DBArtisan User Guide
# Contents

Welcome to DBArtisan ................................................................. 7

- Additional Product Resources ................................................. 8
- Embarcadero Technologies Technical Support ......................... 8
- Technical Requirements .......................................................... 10
- Licensing DBArtisan ............................................................... 12
- Uninstalling DBArtisan ........................................................... 15
- Embarcadero Technologies Problem Report ............................... 15
- Embarcadero Technologies Technical Support ......................... 16

DBArtisan Tutorial ................................................................ 18

- About This DBArtisan Tutorial .............................................. 18
- Session 1: Getting Started with DBArtisan .............................. 19
- Session 2: Object and Schema Management ............................ 21
- Session 3: Database Security Management in DBArtisan ........ 23
- Session 4: DBArtisan Space Management ............................... 24
- Session 5: DBArtisan SQL Management ................................. 25
- Session 6: Job Management .................................................... 27
- Session 7: Data Management .................................................. 27
- Session 8: Performance Management ..................................... 30
- Session 9: Capacity Management .......................................... 31
- Session 10: General Utilities and Tools ................................. 31
- Additional Evaluation Resources ............................................ 31

Application Basics ................................................................. 33

- DBArtisan Product Design ..................................................... 33
- Specifying DBArtisan application and feature options .......... 66

Using .................................................................................. 85

- Datasource Management ..................................................... 86
- Supported Objects ............................................................... 105
- Object Editors ................................................................. 162
- Functionality ................................................................. 276
- SQL Scripting .................................................. 538

MySQL Datasources ............................................................ 577

- Flush Options ............................................................. 577
- Foreign Keys .......................................................... 578
- Functions ............................................................ 580
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexes/Primary Keys/Unique Keys</td>
<td>581</td>
</tr>
<tr>
<td>Users/Privileges</td>
<td>583</td>
</tr>
<tr>
<td>MySQL Databases</td>
<td>586</td>
</tr>
<tr>
<td>Tables</td>
<td>588</td>
</tr>
<tr>
<td>Schema Publication</td>
<td>593</td>
</tr>
<tr>
<td>Server Configuration</td>
<td>595</td>
</tr>
<tr>
<td>Database Backups</td>
<td>595</td>
</tr>
<tr>
<td>Database Restore/Import</td>
<td>600</td>
</tr>
<tr>
<td>Utilities</td>
<td>602</td>
</tr>
<tr>
<td>Schema extraction and migration utilities</td>
<td>603</td>
</tr>
<tr>
<td>Using the Schema Publication wizard</td>
<td>609</td>
</tr>
<tr>
<td>Database Monitor</td>
<td>611</td>
</tr>
<tr>
<td>Instance Configuration</td>
<td>623</td>
</tr>
<tr>
<td>SQL Logfile</td>
<td>624</td>
</tr>
<tr>
<td>Data Export Wizard for IBM DB2 for Linux, Unix, and Windows.</td>
<td>625</td>
</tr>
<tr>
<td>Data Import Wizard for IBM DB2 for Linux, Unix, and Windows.</td>
<td>629</td>
</tr>
<tr>
<td>Load Data Utility Wizard for DB2 OS390</td>
<td>632</td>
</tr>
<tr>
<td>Data Unload Wizard for Microsoft SQL Server and Sybase ASE</td>
<td>639</td>
</tr>
<tr>
<td>Data Load Wizard for Microsoft SQL Server and Sybase ASE</td>
<td>641</td>
</tr>
<tr>
<td>Database Manager</td>
<td>644</td>
</tr>
<tr>
<td>Database Backup Wizard</td>
<td>645</td>
</tr>
<tr>
<td>Database Recovery Wizard</td>
<td>650</td>
</tr>
<tr>
<td>Data Export Wizard for Oracle</td>
<td>655</td>
</tr>
<tr>
<td>Data Import Wizard for Oracle</td>
<td>660</td>
</tr>
<tr>
<td>Archive Activity</td>
<td>664</td>
</tr>
<tr>
<td>Backup Control Files</td>
<td>665</td>
</tr>
<tr>
<td>Chained Row Analysis</td>
<td>666</td>
</tr>
<tr>
<td>Free Space Deficit</td>
<td>667</td>
</tr>
<tr>
<td>Pinned Code</td>
<td>667</td>
</tr>
<tr>
<td>Maximum Extent Limits</td>
<td>668</td>
</tr>
<tr>
<td>Instance Parameter Configuration</td>
<td>669</td>
</tr>
<tr>
<td>Configuring Resource Cost</td>
<td>670</td>
</tr>
<tr>
<td>Session Information</td>
<td>671</td>
</tr>
<tr>
<td>Resource Limits</td>
<td>671</td>
</tr>
<tr>
<td>Top Sessions.</td>
<td>676</td>
</tr>
<tr>
<td>Tools</td>
<td>679</td>
</tr>
<tr>
<td>Find in Files.</td>
<td>680</td>
</tr>
</tbody>
</table>
Database Search ................................................................................................. 680
Script Execution Facility .................................................................................. 681
File Execution Facility ..................................................................................... 682
Scheduling .......................................................................................................... 684
Visual Difference .............................................................................................. 686
Query Builder ................................................................................................... 689
Import Data ........................................................................................................ 721
Data Editor ......................................................................................................... 722
Embarcadero Products ...................................................................................... 729
Code Workbench ............................................................................................... 730
Permissions Management .................................................................................. 735
Add-On Tools ..................................................................................................... 739
Embarcadero SQL Debugger ............................................................................ 740
DBArtisan PL/SQL Profiler ............................................................................... 776
Capacity Analyst .............................................................................................. 788
Getting started with the Capacity Analyst ......................................................... 788
Capacity Analyst tutorial .................................................................................. 798
Using Capacity Analyst .................................................................................... 808
Performance Analyst ......................................................................................... 824
Welcome to Performance Analyst ..................................................................... 825
Performance Analyst Application Basics ......................................................... 833
Using Performance Analyst .............................................................................. 836
Performance Analyst Tools ............................................................................... 836
Performance Analyst for Oracle Tutorial ......................................................... 848
Performance Analyst for Oracle Expert Guide ............................................... 856
Performance Analyst for Microsoft SQL Server Tutorial ............................... 1011
Performance Analyst for Microsoft SQL Server Expert Guide ....................... 1020
Performance Analyst for Sybase Tutorial ....................................................... 1128
Performance Analyst for Sybase ASE Expert Guide ....................................... 1136
Performance Analyst for IBM DB2 for Linux, Unix and Windows Tutorial ....... 1244
Performance Analyst for IBM DB2 for Linux, Unix, and Windows Expert Guide 1252
Space Analyst .................................................................................................... 1386
Welcome to Space Analyst ............................................................................... 1387
Space Analyst Application Basics .................................................................... 1393
Using Space Analyst ........................................................................................ 1395
Space Management Wizard .............................................................................. 1425
Space Analyst for Oracle Tutorial .................................................................... 1432
<table>
<thead>
<tr>
<th>Tutorial/Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Analyst for Microsoft SQL Server Tutorial</td>
<td>1441</td>
</tr>
<tr>
<td>Space Analyst for Sybase ASE Tutorial</td>
<td>1449</td>
</tr>
<tr>
<td>Space Analyst for IBM DB2 for Linux, Unix, and Windows Tutorial</td>
<td>1455</td>
</tr>
<tr>
<td>Space Analyst Expert Guide</td>
<td>1461</td>
</tr>
<tr>
<td>Backup Analyst</td>
<td>1464</td>
</tr>
<tr>
<td>Welcome to Backup Analyst</td>
<td>1465</td>
</tr>
<tr>
<td>Backup Analyst Application Basics</td>
<td>1471</td>
</tr>
<tr>
<td>Using Backup Analyst for Microsoft SQL Server</td>
<td>1476</td>
</tr>
<tr>
<td>Backup Analyst Command Line Reference</td>
<td>1486</td>
</tr>
</tbody>
</table>
Welcome to DBArtisan

Welcome to DBArtisan, the database administration solution for managing IBM DB2 for Linux, Unix, and Windows, IBM DB2 for z/OS and OS/390, Microsoft SQL Server, MySQL, Oracle, and Sybase ASE databases.

The table below describes the major sections of Help.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tutorial</strong></td>
<td>Explores the features and benefits of DBArtisan. You learn how to make schema modifications, add users and grant database privileges. Finally, you explore some of the powerful database utilities.</td>
</tr>
<tr>
<td><strong>Application Basics</strong></td>
<td>Provides conceptual information about DBArtisan’s product design and navigation, as well as datasource management and configuring the application.</td>
</tr>
<tr>
<td><strong>Using DBArtisan</strong></td>
<td>Provides detailed information about using DBArtisan. Using DBArtisan shows how to use each component of the application. Using breaks each process and procedure into simple step-by-step instructions.</td>
</tr>
<tr>
<td><strong>Utilities</strong></td>
<td>Provides detailed information about the extraction, migration and publication of your schema, the loading and unloading of data, database searching, the use of the Script Execution Facility and Query Builder, the scheduling of processes, the backing up and restoration of databases, the use of the Database Monitor, Database Manager, Resource Limits, and Logfile.</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td>Provides detailed information about DBArtisan’s powerful tools to help you create, edit and manage your development environment.</td>
</tr>
<tr>
<td><strong>Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows</strong></td>
<td>Provides detailed information about using the Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows. Includes a step-by-step walk through to familiarize you with the features of the Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows.</td>
</tr>
<tr>
<td><strong>Embarcadero SQL Debugger for Microsoft</strong></td>
<td>Provides detailed information about using the Embarcadero SQL Debugger for Microsoft. Includes a step-by-step walk through to familiarize you with the features of the Embarcadero SQL Debugger for Microsoft.</td>
</tr>
<tr>
<td><strong>Embarcadero SQL Debugger for Oracle</strong></td>
<td>Provides detailed information about using the Embarcadero SQL Debugger for Oracle. Includes a step-by-step walk through to familiarize you with the features of the Embarcadero SQL Debugger for Oracle.</td>
</tr>
<tr>
<td><strong>DBArtisan PL/SQL Profiler</strong></td>
<td>Provides detailed information about using the DBArtisan PL/SQL Profiler. Includes a step-by-step walk through to familiarize you with the features of the PL/SQL Profiler.</td>
</tr>
<tr>
<td><strong>Embarcadero SQL Debugger for Sybase</strong></td>
<td>Provides detailed information about using the Embarcadero SQL Debugger for Sybase. Includes a step-by-step walk through to familiarize you with the features of the Embarcadero SQL Debugger for Sybase.</td>
</tr>
<tr>
<td><strong>Capacity Analyst</strong></td>
<td>Provides detailed information about using Capacity Analyst.</td>
</tr>
<tr>
<td><strong>Performance Analyst</strong></td>
<td>Provides detailed information about using Performance Analyst.</td>
</tr>
<tr>
<td><strong>Space Analyst</strong></td>
<td>Provides detailed information about using Space Analyst.</td>
</tr>
<tr>
<td><strong>Backup Analyst</strong></td>
<td>Provides detailed information about using Backup Analyst.</td>
</tr>
</tbody>
</table>
Additional Product Resources

The Embarcadero Web site is an excellent source for additional product information, including white papers, articles, FAQs, discussion groups, and the Embarcadero Knowledge Base.

Go to www.embarcadero.com/support, or click any of the links below, to find:

- [Documentation](#)
- [Online Demos](#)
- [Technical Papers](#)
- [Discussion Forums](#)
- [Knowledge Base](#)

Embarcadero Technologies Technical Support

If you have a valid maintenance contract with Embarcadero Technologies, the Embarcadero Technical Support team is available to assist you with any problems you have with our applications. Our maintenance contract also entitles registered users of Embarcadero Technologies' products to download free software upgrades during the active contract period.

To save you time, Embarcadero Technologies maintains a [Knowledge Base](#) of commonly-encountered issues and hosts [Discussion Forums](#) that allow users to discuss their experiences using our products and any quirks they may have discovered.

To speak directly with Embarcadero Technical Support, see [Contacting Embarcadero Technologies Technical Support](#) below.

**NOTE:** Evaluators receive free technical support for the term of their evaluation (14 days).

Contacting Embarcadero Technologies Technical Support

When contacting Embarcadero Technologies Technical Support please provide the following to ensure swift and accurate service:

**Personal Information**

- Name
- Company name and address
- Telephone number
- Fax number
- Email address

**Product and System Information**

- Embarcadero product name and version number. This information is found under Help, About.
- Your client operation system and version number.
- Your database and version number.

**Problem Description**
A succinct but complete description of the problem is required. If you are contacting us by telephone, please have the above information, including any error messages, available so that an Embarcadero Technical Support Engineer can reproduce the error and clearly understand the problem.

There are three ways to contact Embarcadero's Technical Support department:

- Via the Web
- Via Phone
- Via Email

**Via the Web**

Embarcadero Technical Support provides an online form that lets you open a Support case via the Web. To access this form, go to [http://www.embarcadero.com/support/open_case.jsp](http://www.embarcadero.com/support/open_case.jsp).

We normally acknowledge the receipt of every case on the same day, depending on the time of submission.

**Via Phone**

**United States**

Embarcadero Technologies Technical Support phone number is (415) 834-3131 option 2 and then follow the prompts. The hours are Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time.

For licensing issues, including Product Unlock Codes, call (415) 834-3131 option 2 and then follow the prompts. The hours are Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time.

The Embarcadero Technologies Technical Support fax number is (415) 495-4418.

**EMEA**

Embarcadero Technologies Technical Support phone number is +44 (0)1628 684 499. The hours are Monday to Friday, 9 A.M. to 5:30 P.M. U.K. time.

For licensing issues, including Product Unlock Codes, call +44 (0)1628-684 494. The hours are Monday to Friday, 9 A.M. to 5:30 P.M. U.K. time.

The Embarcadero Technologies Technical Support fax number is +44 (0)1628 684 401.

**Via Email**

**United States**

Depending on your needs, send your email to one of the following:

- support@embarcadero.com - Get technical support for users and evaluators
- upgrade@embarcadero.com - Request upgrade information
- key@embarcadero.com - Request a product key
- wish@embarcadero.com - Make a suggestion about one of our products

**EMEA**

Depending on your needs, send your email to one of the following:

- uk.support@embarcadero.com - Get technical support for users and evaluators
- uk.upgrade@embarcadero.com - Request upgrade information
- uk.key@embarcadero.com - Request a product key
- uk.wish@embarcadero.com - Make a suggestion about one of our products
Technical Requirements

DBArtisan is a 32-bit application that runs in a Microsoft Windows environment. Before using DBArtisan, please verify that your environment meets the following requirements:

- Microsoft Internet Explorer 5.5 or later

**Hardware**

**Operating System**

**Database**

**NOTE:** Users need full registry privileges during the installation and access to the keys under HKEY_CURRENT_USER in the registry after installation.

Hardware Requirements

Embarcadero Technologies recommends the following minimum hardware requirements:

- 256 MB of memory
- 256 MB of disk space

Operating System Requirements

DBArtisan supports the following operating systems:

- **Windows XP**
- **Windows 2000**

**NOTE:** DBArtisan 8.0.1 was the last version that supported Windows NT 4.0.
Database

The table below describes the database platforms DBArtisan supports and the server and client requirements:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Server</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM DB2 for OS/390 and z/OS</td>
<td>IBM DB2 for OS/390 and z/OS v7, and v8.</td>
<td>When going through a DB2 Connect gateway,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2 Client for Windows 6.0 or later. When</td>
</tr>
<tr>
<td></td>
<td></td>
<td>going directly to the mainframe from the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>client, DB2 Connect Personal Edition v6 or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>higher on the client machine.</td>
</tr>
<tr>
<td>IBM DB2 for Linux, Unix, and</td>
<td>IBM DB2 Universal Database 8.x and 9.x</td>
<td>IBM DB2 Client for Windows 7.2 or later.</td>
</tr>
<tr>
<td>Windows</td>
<td></td>
<td><strong>NOTE:</strong> When using a v8 client, DBArtisan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>only supports connecting to a v8 Database on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Windows, Linux and Unix.</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>Microsoft SQL 2000 and 2005</td>
<td>Microsoft SQL Server Client Library</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> For SQL Server 2005, only object</td>
<td></td>
</tr>
<tr>
<td></td>
<td>types like users, logins, roles, config</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parameters, etc., are supported. We do NOT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>support Microsoft .NET extensions at this</td>
<td></td>
</tr>
<tr>
<td>MySQL</td>
<td>MySQL 4.x. (DBArtisan is tolerant of 5.x but</td>
<td>MySQL ODBC driver 3.5.1 and above</td>
</tr>
<tr>
<td></td>
<td>does not support Stored Procedures,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Triggers, Views.)</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>Oracle 8i, 9i, and 10g</td>
<td>Oracle SQL*Net Client</td>
</tr>
<tr>
<td>Sybase ASE</td>
<td>Sybase System 12.5, Sybase 15</td>
<td>Sybase Open Client</td>
</tr>
</tbody>
</table>

**NOTE:** DBArtisan will not connect to a Sybase database using the Sybase 15 client. **Workaround:** In the Sybase 15 release, Sybase renamed libct.dll and libcs.dll to libsybct.dll and libsybcs.dll, respectively. Sybase has provided a batch command that will rename the files to pre-15 names. It's located in c:\<sybase_client_install_folder>\ocs-15_0\scripts and the file name is copylibs.bat</ul>

Windows XP and 2000 Support

Windows XP, Windows 2000 Professional, Windows 2000 Server, and Windows 2000 Advanced Server have two user security groups:

- Users
- Power Users

Microsoft intentionally does not grant members of the Users Group the authority to install applications or make global changes to the system. Restricted Users are members of the Users Group. Standard users belong to the Power Users Group. Microsoft grants members of the Power Users Group the authority to install programs. You must be a member of the Power Users Group or the Administrators Group in order to install and use Embarcadero Technologies applications.

Because Restricted Users are members of the Users Group, they cannot install and run Embarcadero Technologies applications.

**CAUTION:** You must be a member of the Power Users Group or the Administrators Group in order to install and use Embarcadero Technologies applications.
To open the Group Membership Tab where you can determine your group and review the Microsoft security guidelines:

2. Open Users and Passwords.
3. On the Users Tab, select a user.
4. Click the Properties button.
5. In the Properties dialog box, click the Group Membership Tab.

**Licensing DBArtisan**

DBArtisan includes a trial period in which you can use the application before you must license it. To use DBArtisan after this trial period, you must obtain a license. To continue using the application without interruption, we recommend that you license it as soon as possible.

Licensing DBArtisan is a two-step process. First you need to choose you license type during the installation and then you must license your application.

The table below describes the two license types:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workstation Installation</td>
<td>Installs DBArtisan and the licensing files on the local machine.</td>
</tr>
<tr>
<td>Client Installation</td>
<td>Installs DBArtisan on the local machine but points the licensing files to the license server.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> Embarcadero License Server must be installed and running on the server machine.</td>
</tr>
</tbody>
</table>

Second, you need to complete the four-panel License Request Wizard that lets you request a license from Embarcadero Technologies by e-mail or fax.

**License Request Wizard**

The License Request Wizard lets you specify:

- License type
- Database platforms
- User Information
- Company Information
- Method for submitting your information to Embarcadero Technologies

**Completing the Registration Wizard**

To complete the Registration Wizard, do the following:

1. Open DBArtisan.
2. On the Help menu, click About, and then click License.

DBArtisan opens the first panel of the License Request Wizard.
License Request Wizard - Panel 1
The table below describes the options and functionality on the first panel of the License Request Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Type</td>
<td>Lets you select request a trial license or request a license with a purchase order number or maintenance agreement number.</td>
</tr>
<tr>
<td>Product Configuration</td>
<td>Lets you specify which products and platforms to license.</td>
</tr>
<tr>
<td>Additional Notes and/or Comments</td>
<td>OPTIONAL: Lets you type notes or comments for your license request.</td>
</tr>
</tbody>
</table>

For more information, see License Request Wizard.

License Request Wizard - Panel 2
The table below describes the options and functionality on the second panel of the License Request Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Information</td>
<td>Lets you type your user information.</td>
</tr>
<tr>
<td>Company Information</td>
<td>Lets you type your company information.</td>
</tr>
</tbody>
</table>

For more information, see License Request Wizard.

License Request Wizard - Panel 3
The table below describes the options and functionality on the third panel of the License Request Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Information</td>
<td>Displays machine’s MAC address and description.</td>
</tr>
<tr>
<td>Email Destination</td>
<td>Select <a href="mailto:key@embarcadero.com">key@embarcadero.com</a> if DBArtisan was purchased in the United States. Select <a href="mailto:key@embarcadero.co.uk">key@embarcadero.co.uk</a> if DBArtisan was purchased in the United Kingdom.</td>
</tr>
</tbody>
</table>

For more information, see License Request Wizard.

License Request Wizard - Last Panel
The table below describes the options and functionality on the last panel of the License Request Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary of Request</td>
<td>Displays a summary of your license request.</td>
</tr>
<tr>
<td>Send E-mail</td>
<td>Sends the information, via e-mail message, to Embarcadero Technologies Licensing.</td>
</tr>
</tbody>
</table>
For more information, see License Request Wizard.

### About Embarcadero DBArtisan

Embarcadero DBArtisan offers easy access to your license information. The About Embarcadero DBArtisan dialog box not only lets you see version and build information, but also allows you to see your license type, licensed modules (if any), and server, system, and registration information. You can also connect to numerous Embarcadero Technologies’ links.

**NOTE:** From a client machine, you can see information about the licenses available to your particular machine. Only the Embarcadero License Server Administrator dialog box, housed on a dedicated server machine, gives an overview of the licenses and servers available across the organization.

The table below describes the information available in the About Embarcadero DBArtisan dialog box:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embarcadero DBArtisan version and copyright</td>
<td>Displays the version and build number of the current installation along with the copyright years.</td>
</tr>
<tr>
<td>License Type</td>
<td>Displays the license type, floating or nodelock. Stand-alone installations are nodelock licenses. Floating licenses require that your local machine be connected to an Embarcadero License Server.</td>
</tr>
<tr>
<td>Licensed Modules</td>
<td>Lets you see a list of the possible application modules and whether they are in use or not.</td>
</tr>
<tr>
<td>Server</td>
<td>If you have a floating license, you can see the license server you are connected to. If for some reason you need to connect to a different license server, for example if your enterprise has multiple server machines and the one you are connected to fails, you can ping another server machine and make an alternate connection. <strong>NOTE:</strong> You can enter and ping the information in the Licensing dialog box, but you must restart DBArtisan for the new connection to take place.</td>
</tr>
<tr>
<td>System</td>
<td>The System Info dialog box lets you see the IP address for your machine.</td>
</tr>
<tr>
<td>Info</td>
<td>When you click an active license, you can see the total licenses, the number of licenses in use, the number of available licenses, and the license host(s) (that is, the server machine where the target license resides).</td>
</tr>
<tr>
<td>License</td>
<td>Opens the License Request Wizard.</td>
</tr>
<tr>
<td>Web Links</td>
<td>Makes connections to Embarcadero Technologies available to you, including contact information, downloads, product information, technical support, company information, upgrade requests, and FAQs.</td>
</tr>
</tbody>
</table>
Uninstalling DBArtisan

To uninstall DBArtisan to install a later version of DBArtisan, do the following:

1. Click Start, Programs, Embarcadero DBArtisan 8.0.0, and then Uninstall.
   
   DBArtisan opens the Uninstall Wizard.

2. Complete the wizard panels, and then click Finish.

   DBArtisan deletes DBArtisan and its components.

   **NOTE:** If you reinstall DBArtisan or upgrade to a later version of DBArtisan, DBArtisan rediscovers your datasource connections. You do not need to manually rediscover or reconfigure your datasources.

Opening DBArtisan Help

Although the most common way to access Help is through context-sensitive Help, DBArtisan also offers three additional ways to access Help:

1. On the Help menu, click Help Topics.

2. On the Windows Start button, point to Programs, Embarcadero DBArtisan, and then click DBArtisan Help.

3. On the Windows desktop, open the Embarcadero DBArtisan folder, and then double-click the Help icon.

   **TIP:** To toggle between Help and the application, on the Windows Start button, open DBArtisan Help, and then press ALT+TAB.

For more information, see Using DBArtisan Help.

Using DBArtisan Help

Help is a customizable window that lets you access DBArtisan Help. The table below describes the tabs in the DBArtisan Help window:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>The Table of Contents for the Help system.</td>
</tr>
<tr>
<td>Index</td>
<td>A comprehensive index of DBArtisan Help.</td>
</tr>
<tr>
<td>Search</td>
<td>A search engine for every word in the DBArtisan Help. DBArtisan Help features an advanced full-text search, letting you search using Boolean, wildcard, and nested expressions.</td>
</tr>
<tr>
<td>Favorites</td>
<td>A personalized list where you can access your favorite DBArtisan Help topics. DBArtisan saves this list so it displays every time you open DBArtisan Help.</td>
</tr>
</tbody>
</table>

Embarcadero Technologies Problem Report

In the event that Embarcadero DBArtisan stops responding, an Embarcadero Technologies Problem Report is automatically generated, which includes information about why the program failed. The program launches a dialog box where you can add comments about what you were doing when the problem occurred. Clicking Send Mail emails the report to Embarcadero Technologies.
Embarcadero Technologies Technical Support

DBArtisan is fully supported by the Embarcadero Technologies technical support team. The technical support team is available to assist you with any problems you have with this application. When contacting Embarcadero Technologies Technical Support please provide the following:

- Product name.
- Product version.
- Operating system.
- Database platform and version in which the error occurs.
- Error message (exact text, the Event log, or a screen shot if possible).
- Detailed description of the problem including steps to reproduce.

Contacting Embarcadero Technologies Technical Support
Embarcadero Technologies offers three ways to obtain technical support:

- Web
- Phone
- E-mail

Via the Web

Embarcadero Technologies Web Site
To get to Embarcadero Technologies Technical Support on the Web using your Internet browser:

1. Open your Internet browser.
2. In the address field, type http://www.embarcadero.com/support/opencase.asp
3. Click ENTER.
   The Internet browser opens the Embarcadero Technologies Technical Support Request Web page.
4. Complete the online form.
5. Click Submit.

Via Phone

United States
Embarcadero Technologies Technical Support phone number is (415) 834-3131 option 2 and then follow the prompts. The hours are Monday through Saturday, 6:00 A.M. to 6:00 P.M. Pacific time.

For licensing issues call (415) 834-3131 option 2 and then follow the prompts. The hours are Monday through Friday, 6:00 A.M. to 6:00 P.M., Saturday and Sunday 8 A.M. to 5 P.M. Pacific time.

The Embarcadero Technologies Technical Support fax number is (415) 495-4418.
United Kingdom
Embarcadero Technologies Technical Support phone number is +44 (0)1628 684 499. The hours are Monday to Friday, 9 A.M. to 5:30 P.M. UK time.
For licensing issues call +44 (0)1628-684 494. The hours are Monday to Friday, 9 A.M. to 5:30 P.M. UK time.
The Embarcadero Technologies Technical Support fax number is +44 (0)1628 684 401.

Via E-mail

United States
Embarcadero Technologies offers two ways to contact Technical Support via e-mail:

1. Complete a technical support form on the Embarcadero Technologies Web site:
   Go to the Embarcadero Technologies Technical Support Web page:
   http://www.embarcadero.com/support/opencase.asp
   Complete the form.

2. Send an e-mail message to Embarcadero Technologies. Depending on your needs, send your E-mail to one of the following:
   • support@embarcadero.com - Get technical support for users and evaluators
   • upgrade@embarcadero.com - Request upgrade information
   • key@embarcadero.com - Request a product key
   • wish@embarcadero.com - Make a suggestion about one of our products

E-mail
Embarcadero Technologies offers two ways to contact Technical Support via e-mail:

1. Complete a technical support form on the Embarcadero Technologies Web site:
   Go to the Embarcadero Technologies Technical Support Web page:
   http://www.embarcadero.co.uk/support/opencase.asp
   Complete the form.

2. Send an e-mail message to Embarcadero Technologies. Depending on your needs, send your E-mail to one of the following:
   • support@embarcadero.co.uk - Get technical support for users and evaluators
   • upgrade@embarcadero.co.uk - Request upgrade information
   • key@embarcadero.co.uk - Request a product key
   • wish@embarcadero.co.uk - Make a suggestion about one of our products.
DBArtisan Tutorial

DBArtisan is an industry-leading database administration solution for managing Oracle, Microsoft SQL Server, Sybase Adaptive Server, IBM DB2 for Linux, Unix, and Windows, and IBM DB2 for z/OS and OS/390 databases. Its cross-platform capability allows users to efficiently manage heterogeneous database platforms easily using a single front-end tool. Using DBArtisan, users boost their productivity by utilizing a single tool for all their databases, regardless of database vendor.

Database Administrators

DBArtisan enables database administrators to accomplish more with the time they have available in their workday. It eliminates the tedious tasks associated with researching schema dependencies when making object changes. Also included are a host of utilities, which condense DBA tasks taking hours or days down to minutes.

Developers

DBArtisan provides additional administration functionality to database developers over standard development platforms. Using the powerful schema extraction, schema migration, and publication wizards, developers can quickly extract and move schema from development to other environments, as well as create objects much quicker than using old-fashioned hand coding techniques.

About This DBArtisan Tutorial

This tutorial is intended to help you get started using Embarcadero's DBArtisan, the industry-leading solution for administering enterprise databases from a single point of control. While DBArtisan supports current versions of Oracle, Microsoft SQL Server, Sybase Adaptive Server, IBM DB2, and IBM DB2 for z/OS and OS/390, the examples in this tutorial are Oracle-centric. Unless otherwise noted, all features and functionality highlighted in this tutorial are applicable to all supported platforms.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of DBArtisan. You'll have learned how to competently manage the major database administration disciplines using DBArtisan's standard cross-platform console. In addition, you will have a solid understanding of DBArtisan's more advanced Space, Performance and Capacity management capabilities.

This tutorial is divided into 10 sessions:

- Session 1: Getting Started with DBArtisan
- Session 2: Schema Management
- Session 3: Security Management
- Session 4: Space Management
- Session 5: SQL Management
- Session 6: Job Management
- Session 7: Data Management
- Session 8: Performance Management
- Session 9: Capacity Management
- Session 10: General Utilities and Tools

You can use this basic tutorial as a road map of product highlights and also to help you find your own path to explore DBArtisan.
Once you have started, you can select Help from the toolbar to find the many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see Additional Evaluation Resources.

**Session 1: Getting Started with DBArtisan**

You can obtain the latest version of the DBArtisan software from the Embarcadero web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp

**Start DBArtisan**

1. On the Start menu, point to Programs, Embarcadero DBArtisan 8.1.0, and then select DBArtisan.
   
   The first time DBArtisan opens, it displays a message asking if you would like DBArtisan to automatically discover and register datasources.

2. If you have installed and used other Embarcadero tools that use registered datasources, click Yes.
   
   DBArtisan finds any datasources that are defined in the datasource catalog on the machine identified in the Datasource Tab of the Options Editor (see Setting Environment Options below).
   
   OR
   
   For the purpose of this tutorial, click No.

3. Click OK to continue.

DBArtisan provides a Discover Datasources feature that automatically searches the DBMS configuration files on your system for datasources that are not currently registered. The Discover Datasource feature is a dialog box that contains a list of all unregistered datasources found on your network or local machine, including the name of the server or instance and the type of DBMS for each datasource. Once discovered, you have the option to register datasources.

**Registering Cross-Platform Datasources**

You are presented with the Datasource Registration dialog box if you selected OK in the previous dialog box. The Datasource Registration dialog box also opens from the main menu when you point to Datasource, and then click Register Datasource.

**Registering a Datasource (Oracle Sample)**

On the first panel of the wizard:

1. For What is the DBMS type?, select Oracle.

2. In the Datasource Name text box, type SAMPLE_DATASOURCE for the purpose of this example.

3. Click Next.

   DBArtisan saves your selections and opens the next panel of the Datasource Registration Wizard.

4. In User Id, type the user id for the database.

5. In Password, type the user’s password.

6. To save and encrypt your password, select Auto-Connect?
7. Click **Next**.

DBArtisan opens the final panel of the wizard.

8. In the Datasource Group tree, place the datasource you are registering.

9. Click **Register**.

DBArtisan offers the same easy-to-use Datasource Registration process for IBM DB2, Microsoft SQL Server, Oracle, MySQL, and Sybase connections. The connection information only needs to be set up one time for each datasource and can be saved locally or in a common datasource catalog for use by other Embarcadero products.

You can configure Embarcadero database applications to use a datasource catalog stored in the system registry of your machine (local).

All Embarcadero database administration products share the datasource catalog, which means that when you set up your datasource catalog using one product such as DBArtisan, the same list of datasources is available in other Embarcadero Technologies products. Any changes you make to the datasource catalog are reflected in all Embarcadero database management products.

### General Usability Features

DBArtisan provides many “user in mind” features that make the product configurable to meet individual needs and preferences. These features are designed to shave time off the tasks that you perform many times on any given working day.

#### Retaining Datasource Explorer View Settings

1. At the top of the **Explorer** tree, click to expand the drop-down menu.

2. Select **Retain View Settings**.

The next time you open DBArtisan, the Explorer appears just as you left it. All connections that were present when you closed DBArtisan will be reestablished.

#### Adding Datasource Explorer Bookmarks

1. In the **Explorer** tree, right-click any node.

2. Select **Add Bookmark**.

DBArtisan opens the Add Friendly Bookmark Name dialog box.

3. Click **OK**.

After Bookmarks are defined you can use them to easily navigate to commonly used datasource resources via the main menu Bookmarks item.

#### Setting Keyboard Shortcuts and Hotkeys

1. In any open space above the **Datasource Explorer**, right-click.

DBArtisan opens a shortcut menu.

2. From the shortcut menu, select **Customize**.

The Customize dialog box opens.

3. In the **Customize** dialog box, open the **Keyboard Tab**.

**TIP:** The Keyboard Tab can be used to set Keyboard shortcut hot keys for all areas of DBArtisan functionality.
Opening Most Recently Used Datasources
1 From the File menu, select Recent Datasources, and then choose a datasource.

DBArtisan opens the datasource in the Datasource Explorer, ready to work with an active connection.

Session 2: Object and Schema Management

Advanced Object Creation and Management
DBArtisan provides unparalleled database object management capabilities. Its database platform- and version-specific graphical object editors and wizards enable you to easily create, drop, or alter any of your accessible database objects. The following example walks you through creating and then altering a standard Oracle table object. This concept carries across all supported object types and across all supported platforms.

Creating a Table
1 On the Datasource Explorer, expand an Oracle datasource.
2 On the Oracle datasource, right-click the Tables node, and then select New.

DBArtisan opens the Table wizard and leads you through the process of creating a table object.
3 Complete the wizard panels.
4 Click Finish.

DBArtisan lets you preview any and all generated scripts before you submit them to the database. This is standard for all object related scripts.

Changing a Table
Changes to database tables, such as modifying column lengths, inserting new columns, or deleting unneeded ones, can require dropping of a table. This requires knowledge of the underlying object dependencies so that these dependent objects are rebuilt after the table has been re-created.

DBArtisan provides the ability to perform "extended" table alterations by constructing a SQL script with the steps necessary to save off the original data, create the new table, and populate it with the original data. Once these steps are complete, all dependent objects are then rebuilt and permissions re-applied. Following is a sample table change:

1 From the Datasource Explorer, expand the Tables node, select the table you created in the previous example.
2 Double-click the table.
   OR
2 From the Command menu, click Open.

DBArtisan opens the Table Editor. The Table Editor displays the list of table columns as well as any constraints, storage parameters, space allocation, partitioning, table dependencies, object privileges, and table DDL.
3 Double-click the column you want to modify. In this example, REQUESTING_DOCTOR is being expanded to accommodate new numerical sizing requirements.

DBArtisan opens the Modify Column dialog box.
4 In the Modify Column dialog box, in the Width or Scale text box, type the new value.
5 Click Modify.

DBArtisan saves the changes you made to the table column.
6 On the Table Editor toolbar, select the Alter button.

DBArtisan lets you preview the SQL script before you submit it to the database.

Working with Object DDL
DBArtisan allows you to easily extract DDL for single or multiple objects using several methods. The most straightforward is described here:

1 On the Datasource Explorer, expand an Oracle datasource.
2 On the Oracle datasource, click the Tables node.
3 In the right pane of the Explorer window, right-click any table or group of tables (SHIFT+CLICK), and then select Extract.

The DDL for all highlighted objects is extracted directly in a DDL Editor where it can be altered, executed, and saved to the database. No intermediary steps are required.

Advanced Schema Management
In addition to standardized object support, DBArtisan provides you with advanced Schema management features. These features include full or object-level schema extraction, migration (with or without data), and publication. This example walks you through a simple cross-platform migration between Oracle and SQL Server datasources. Because DBArtisan automatically resolves differences between the disparate DBMS platforms, you can concentrate on what you want to do, rather than how to actually do it. The Schema Migration Wizard sequences the creation of objects in the proper order to eliminate dependency errors. It also has the intelligence to use the fastest method available for copying table data.

Migrating Schema

NOTE: While this example focuses on schema migration, the same wizard principle applies to schema extract and publication.

1 On the Utilities toolbar, click the Migrate button.

DBArtisan opens the Schema Migration Wizard.

2 On the first panel of the wizard, select New Migration Operation.

DBArtisan opens the second panel of the Schema Migration Wizard.

3 From the Select the Source datasource drop-down list, select an Oracle datasource.

4 From the Select the Target datasource drop-down list, select datasource to which you would like to migrate the object.

The SQL Server ER database is selected for this example.

5 Click Next.

DBArtisan opens the next panel of the Schema Migration Wizard.

6 In the Database Object Types box, select the objects you want to migrate to the target datasource, and then select the owner to whom you want to transfer the objects.

7 In the Migration Options box, specify the migration options to use for this migration job. If you only want to transfer a subset of objects, click Customize Object List.
8. Click **Next**.

   DBArtisan opens the next panel of the Schema Migration Wizard.

9. If you chose **Customize Object List**, for each object type select the specific objects you want to migrate using CTRL+CLICK.

10. In the **Migration Options** box, select the options you want to use when performing table migrations.

11. Click **Next**.

   DBArtisan opens the next panel of the Schema Migration Wizard.

12. To have the objects migrated to the target under the same ownership, click **Retain**.

13. If you would like the objects built and transferred to a new owner, click **Transfer**.

14. Click **Next**.

   DBArtisan lets you preview the migration operation.

15. Click **Execute**.

   DBArtisan executes the migration operation.

---

**Session 3: Database Security Management in DBArtisan**

DBArtisan can help you efficiently establish and maintain database security and related objects. Whether you are managing an existing production database or setting up a new environment, you will find consistent support across all supported platforms.

**Creating a User**

**NOTE:** While this example focuses on creating a new Oracle user, the same wizard-driven principle applies to all security objects (groups, roles, etc.).

1. On the **Datasource Explorer**, expand an Oracle datasource, and then the **Security** node.

2. On the **Security** node, right-click **Users**, and then click **New**.

   DBArtisan opens the User Wizard and leads you through the process of adding a user.

3. Complete the **User Wizard**.

4. Click **Finish**.

   DBArtisan allows you to preview any and all generated scripts before they are submitted to the database. This is standard for all object related scripts.

5. Click **Execute** to create the new user.

   DBArtisan opens the User Editor for the new user. The standard User Editor can be used to manage existing database users as shown below.
Granting and Editing User Privileges

Privileges can be easily granted, revoked, or viewed from within either of two editors within DBArtisan; the User Editor, or the individual object editor (table, procedure, etc.). The User Editor provides a tabbed interface, which can be used to view and modify individual attributes of the user.

1. In the User Editor, open the Object Privileges Tab.
2. Select target table(s).
3. To specify which permissions to grant the object, click Grant.
   - If permissions are granted directly to the user, they are noted with a black check mark.
   - Permissions inherited from a group (Sybase) are displayed via blue checkmarks.
   - Permissions inherited from a Role are identified with green checkmarks.
4. Highlight one or more objects (in this case tables), and then click Grant.
   DBArtisan opens the Grant Privilege(s) To dialog box. This dialog box displays the database privileges eligible to be granted to the objects. The user may also specify that the privileges are grantable.
5. Select the desired privileges to be granted, and then click Preview.
   DBArtisan displays the DDL.
6. Click Execute to implement the changes.

Session 4: DBArtisan Space Management

Managing space is vital to ensuring the availability and performance of your databases. DBArtisan incorporates many built-in space features that enable you to smartly manage and exploit all aspects of your database’s storage. The following example walks you through a review of DBArtisan’s built-in support for reporting Oracle tablespace storage and space data.

Managing Object Space

NOTE: While this example is specific to Oracle tablespaces the same concept applies to all supported platforms.

2. On the Oracle datasource, expand the Storage node, and then right-click Tablespaces.
3. Right-click any tablespace listed in the right pane of the Explorer window, and then click Open.
   Embarcadero DBArtisan opens the Tablespaces Editor.
4. On the Tablespaces Editor, click the Storage Tab.
   The Storage Tab displays and lets you edit the tablespace extent limits.
   NOTE: Changes to this tab enable the Alter function on the Tablespace Editor toolbar.
5. On the Tablespaces Editor, click the Space Tab.
   The Space Tab displays a graphical view of the Free space and Fragmentation Index for the target tablespace.
6 Finally, on the Tablespace Editor, click the Map Tab.

The Map Tab displays a color-coded map of the objects contained on the tablespace.

**NOTE:** The map segments are proportional to the actual size of the objects on the tablespace.

### Advanced Space Management

For advanced space analysis and management, DBArtisan’s optional Space Analyst component contains sophisticated diagnostics to help you pinpoint all space-related problems in your database, as well as an intelligent Reorganization Wizard that can reorganize all or selected parts of your database.

**Embarcadero Space Analyst**

1 On the Analyst toolbar click the **Space Analyst** button.

The Space Analyst launches in the DBArtisan workspace.

Embarcadero’s Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result in poor space management.

Please see the Embarcadero Space Analyst evaluation guide for a detailed walkthrough of all available features and functionality.

### Session 5: DBArtisan SQL Management

DBArtisan provides powerful visual tools for creating and analyzing complex SQL statements and server-side code objects. The following examples walk you through DBArtisan’s Visual Query Builder, feature-rich ISQL facility, and some of the advanced analysis and debugging capabilities provided by the Embarcadero SQL Debugger and SQL Profiler.

#### Building Queries

1 From the **Tools** menu, select **Query Builder**.

   **OR**

   In the right pane, right-click a table, and then select **Build Query**.

   DBArtisan opens the Query Builder.

2 In the **Tables/Views Tab**, select the columns to return in the result.

   Query Builder generates the query text in the lower SQL window.

   **TIP:** You can build advanced queries using the options supplied in the DML Tab.

3 After the query is built, click **Execute**.

   Query Builder displays results in the lower SQL window.

#### Using the ISQL Window

1 On the **File** menu, click **New**, and then **SQL**.

   DBArtisan opens the ISQL Editor window.

2 Add SQL code via your method of choice (free-form typing, retrieve from a file, paste copied code, etc.).
The ISQL Editor window includes the following features and options:

- The ISQL window highlights all platform and general keywords and provides the options for SQL code formatting, syntax checking and analysis.
- Once code is executed you have control over whether your transaction is committed or rolled back from the database.
- For all open ISQL windows, there are also options for connection locking, scheduling, executing your code across multiple datasources, explain plan generation, and SQL Tuning.

3 Press F8 prior to SQL execution.

DBArtisan opens the Query Options dialog box that lets you set platform specific Query Options to immediately determine if your code is optimized.

SQL Debugging, Analysis, and Tuning
To analyze and debug your SQL code, DBArtisan provides cross-platform SQL code debuggers. For your Oracle databases DBArtisan offers a robust PL/SQL code profiler that helps you to pinpoint and eliminate “hot spots” within poorly running server-side code. To ensure code efficiency, the ISQL window provides tight integration with Embarcadero’s SQL Tuner, so you can perform multiple “test then tune” iterations without having to leave an open ISQL window.

Debugging SQL

FILE:

While this example is specific to Oracle PL/SQL Debugging the same interface and functionality applies to all of the supported platforms.

1 On the Datasource Explorer, expand any Oracle datasource node.
2 On the Oracle datasource, expand the Procedures node.
3 In the right pane of the Explorer, right-click any stored procedure, and then select Debug.
4 If applicable, enter any input parameters the Procedure Execution input window.
5 Click Continue.

After the SQL Debugger interface is displayed you can step through code, step into dependencies, set and watch variables, and even perform basic code profiling for each line of executed code.

Profiling SQL

FILE:

The Embarcadero SQL Profiler is available only for Oracle version 8.0 and later.

1 From the Tools menu, select SQL Profiling, and then click Start.
2 When prompted add a Profile comment that will help you identify the session later.
3 On the Datasource Explorer, expand any Oracle datasource.
4 On the Oracle datasource, expand the Procedures node.
5 In the right pane of the Explorer window, right-click any stored procedure and select Execute. If applicable, enter any input parameter and press Execute.
6 After execution is complete, from the Tools menu, select SQL Profiling, then click Stop.
7 When prompted select Stop & Analyze.

DBArtisan displays the Run Detail view of the profiling data for the current session.
8 To locate the most expensive code for individual units within the run, double-click any of the units and the Unit Detail view is presented. The Unit Detail view allows you to pinpoint the code that may need to be tuned for efficiency.

9 To find previously captured Profiling sessions, on the Datasource Explorer, select the Performance node, and then PL/SQL Code Profiling.

SQL Tuning

NOTE: The Embarcadero SQL Tuner is available for Oracle only.

DBArtisan lets you easily tune your SQL code using SQL Tuner.

1 Open an ISQL window, and then open, copy, or type the code you want tune.

2 From the ISQL window toolbar, select the Tune Current SQL button.

Your code is ported to the Embarcadero SQL Tuner where multiple optimization options are available. When optimization is complete, SQL Tuner brings your tuned code back to the original DBArtisan ISQL window.

Session 6: Job Management

DBArtisan freely integrates with the Microsoft Windows Task Scheduler, which allows you to schedule virtually any task to run on your computer whenever and however often you want. While this example is specific to an Oracle table redefinition the same concept applies to any job or script that can be scheduled. For this example, you will schedule a table redefinition.

Scheduling Jobs

To schedule a job:

1 On the Datasource Explorer, expand any Oracle datasource.

2 On the Oracle datasource, expand the Tables node, and then right-click any table.

3 Select Extract.

4 From the ISQL window toolbar, click Schedule.

The Scheduler Action dialog box opens where you can provide a name, set notifications, and specify an output directory for the new job.

5 After you have completed the dialog box, click OK.

This opens the Windows Job Scheduler dialog. For the purposes of this exercise, you can either finish scheduling the task and inspect the results when it completes, or you can click Cancel to proceed to the next session.

Session 7: Data Management

DBArtisan provides comprehensive facilities to help you manage the data in all of your databases. A visual data editor helps you add, change, and delete data from your tables with all referential integrity enforced. You can create insert statements for tables using current data and also extract data as XML documents for certain databases. Rounding out its rich schema management capabilities, DBArtisan also allows you to migrate schema objects and associated table data from one database server to another, across the same or different platforms.
Editing Visual Data
To start the Visual Data Editor, do the following:

1. In the **Datasource Explorer**, right-click any table or tables, and select **Edit Data**. DBArtisan opens the Data Editor Filter.

2. In **Columns**, select the columns to include in the Edit.

3. You can also filter the editable rows by including your own Select statement.

4. Click **OK**.

**CAUTION:** In Live mode all changes are applied to the database when you move off of an updated or inserted row. Deleted rows are immediately removed from the database.

**NOTE:** Batch mode allows you to make changes and then save all when all are complete.

5. After you are done editing your data, on the Data Editor toolbar, click **Execute**. DBArtisan commits your changes. Regardless of mode, all of the generated DML statements are viewable in the lower SQL window.

Creating Insert Statements

1. On the **Datasource Explorer**, select any Oracle datasource.

2. On the Oracle datasource, expand the **Tables** node.

3. In the right pane of the Explorer window, right-click any table, and then select **Create Insert Statements**. DBArtisan opens the Create Insert Statements dialog box.

4. In **Columns**, select the columns you want to include in the Insert statement.

5. You can also filter what rows are included by adding your own Select statement.

6. **OPTIONAL:** Select Owner information and row limits

7. Click **OK**.

The resulting insert statements are created and presented in an active ISQL window. At this point they can be executed immediately, scheduled to later or saved. Note that all extracted insert statements can be run against the same or different databases containing a similar schema.

Extracting Data as XML

**NOTE:** This feature is available for Oracle 9i, SQL Server 8.0, and Sybase 12.5.1 and later (for Sybase only, you must purchase and install a special service from Sybase). The following example is specific to Oracle 9i, but the concept applies to SQL Server 8.0 as well.

1. On the **Datasource Explorer**, expand any Oracle datasource.

2. On the Oracle datasource, expand the **Tables** node.

3. In the right pane of the Explorer window, right-click any table listed, and then select **Extract Data as XML**.

4. Select the columns to include in the Insert statement.

5. You can also filter what rows are included by adding your own Select statement.
6 Click **OK**.

The resulting XML document is created and presented in an active XML Editor. At this point the document can be saved in XML format.

---

**Migrating Schema and Data**

DBArtisan provides advanced data management tools that help you move schema and corresponding table data across the same or different platforms. You can copy a single database object, all objects owned by a specific user, or an entire database, all guided by a wizard-driven process.

### Schema and Data Migration

**NOTE:** While this example is specific to an Oracle to SQL Server schema and data migration the same concept applies to any migration involving any combination of the supported platforms.

To open the Schema Migration Wizard:

1. On the **Utilities** toolbar, click the **Migrate** button.
   
   DBArtisan opens the Schema Migration Wizard.

2. Click **New**.
   
   DBArtisan opens the next panel of the Schema Migration Wizard.

3. From the list, select an Oracle datasource.

4. Select the target datasource. This is the datasource to which you would like to migrate the object. For this example, this is the SQL Server 8.0 ER database.

5. Click **Next**.
   
   DBArtisan opens the next panel of the Schema Migration Wizard.

6. Select the objects to be migrated to the target datasource.

7. Select which owner to transfer objects for.

8. Specify the migration options to use for this migration job. If you only want to transfer a subset of objects, click **Customize Object List**.

9. Click **Next**.
   
   DBArtisan opens the next panel of the Schema Migration Wizard.

10. If you selected **Customize Object List**, for each object type, select the specific objects you would like to migrate by pressing **CTRL+CLICK**.

11. Under **Migration Options**, select the options to be used when performing table migrations. To include table data in the migration, click **Copy Table Data**.

12. Click **Next**.
   
   DBArtisan opens the next panel of the Schema Migration Wizard.

13. If you would like to have the objects migrated to the target under the same ownership, select **Retain**.

14. If you would like the objects built and transferred to a new owner, select **Transfer**.

15. Click **Next**.
   
   DBArtisan lets you preview the migration operation.
16 Click **Execute**.

DBArtisan executes the migration job.

### Session 8: Performance Management

DBArtisan offers a number of options to help you manage the performance of your databases. First, DBArtisan ships with a built-in process monitor that helps you understand who is connected to your database along with each user’s current activity and session-related data. For more robust performance details DBArtisan’s Performance Analyst add-on is a powerful client-side database monitor that runs fully contained in the DBArtisan console.

#### Monitoring Sessions

**NOTE:** While this example is specific to Oracle, the Process Monitor is available for all of the supported platforms.

To start the DBArtisan Process Monitor:

1. On the **Datasource Explorer**, select any Oracle datasource.
2. From the **Utilities** menu, select **Database Monitor**.

The Database Monitor includes the following options and features:

- Highlight any session and any currently running SQL is displayed in the lower pane.
- You can drill-down into a specific session to display session-level statistical details, historical and current wait events along with a working copy of the currently running SQL that can be copied to an ISQL for explain plan generation.
- By using the Monitor drop down options you can display more advanced database-level monitoring data such as locks, blocking locks, hit ratio by user, Top 20 SQL etc.

#### Performance Monitoring

For advanced performance monitoring and management, DBArtisan’s optional Performance Analyst provides intelligent diagnostic information and strong drill-down details to help you get to the heart of any episode of performance degradation. Performance Analyst integrates completely with DBArtisan so you can fix any performance problems with a few clicks of the mouse.

**Embarcadero Performance Analyst**

1. On the **Datasource Explorer**, select any Oracle datasource.
2. From the **Analyst** toolbar, click the **Performance Analyst** button.

 The Performance Analyst opens in the DBArtisan workspace for the target Oracle datasource.

Please see the [Embarcadero Performance Analyst Evaluation Guide](#) for a detailed walkthrough of all available features and functionality.
Session 9: Capacity Management

Planning for the future of your critical databases used to be a difficult task. However, DBArtisan’s optional Capacity Analyst tool makes it easy to understand where your databases are today and where they are headed in the future. Capacity Analyst lets you track key database metadata and performance metrics over time so you can perform trend analysis on key areas like growth, object fragmentation, I/O, and session load. Like all of the Analyst Series products, Capacity Analyst runs fully contained within DBArtisan so you have access to smart, built-in forecasting mechanisms that allow you to predict when your databases will run out of space and give you the ability to proactively manage your storage assets, all from the same console.

Advanced Capacity Planning

2. From the Analyst toolbar, click the Capacity Analyst button.

   The Capacity Analyst opens in the DBArtisan workspace for the target Oracle datasource.

Session 10: General Utilities and Tools

No evaluation of DBArtisan would be complete without a mention of the general Utilities and Tools that are available across all supported platforms.

Utilities Menu

The main toolbar Utilities menu contains the more advanced DBArtisan features. The available menu items are context-sensitive and version-specific for the selected datasource DBMS platform.

Tools Menu

The main toolbar Tools menu contains those features that are common across all DBMS platforms. All DBArtisan utilities and tools provide a common interface that walks you through all input and execution requirements. All results are consistently presented so you can easily move between features without effort or confusion.

Additional Evaluation Resources

Embarcadero Technologies provides a variety of resources to help support your evaluation and selection of a development tool for your organization.

Web Site


Electronic Documentation

Detailed reference documentation is available on the DBArtisan Evaluation CD or online at www.embarcadero.com/support.

Online FAQ

The DBArtisan online FAQ provides answers to commonly asked questions regarding licensing, installation and other helpful topics.
E-mail Support
You can contact DBArtisan support engineers, consultants and engineers directly by sending inquiries to support@embarcadero.com.

Telephone Support
We encourage you to call us anytime you would like help or have questions during your evaluation. Please call 415.834.3131 ext. 2, Monday to Friday, 6:00am - 6:00pm PST, Saturday and Sunday, 8:00am - 5:00 PST.
Application Basics

Application Basics is designed to situate you within the application and to provide information about what DBArtisan offers in the way of design, navigation, and application features. The information presented here is high-level and conceptual.

Application Basics is divided into two sections. The table below describes each section:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DBArtisan Product Design</strong></td>
<td>This section describes key interface elements and concepts, including how to use the Datasource Explorer, parallel query execution, and how to navigate and make the most use of DBArtisan's Multiple Document Interface (MDI) environment.</td>
</tr>
<tr>
<td><strong>Specifying DBArtisan application and feature options</strong></td>
<td>This section describes how you can configure the application to meet your specific needs.</td>
</tr>
</tbody>
</table>

**DBArtisan Product Design**

The Datasource Explorer has its own context-sensitive toolbar. As you switch object types in the Datasource Explorer, this toolbar and shortcut menus adapt to expose functionality pertaining to the selected object type. This section describes key interface elements and concepts, such as how to use the Datasource Explorer, parallel query execution, and how to navigate and make the most use of DBArtisan's Multiple Document Interface (MDI) environment.

**TIP:** DBArtisan lets you use a mouse with a wheel to navigate in active windows.

For more information, see:

- Datasource Explorer
- DBArtisan Toolbars and Menus
- DBArtisan Dialog Boxes
- Customizing the DBArtisan Display

**Datasource Explorer**

The Datasource Explorer provides a visual method for browsing, accessing, and manipulating objects. The Explorer is designed to give you an overall picture of the contents and configuration of your datasources. From the Explorer you can connect, disconnect and browse the objects in all of the supported datasources on your network.

The Datasource Explorer is a separate tabular window, with a tree in the left pane. When you select an object on the Datasource Explorer, you can view corresponding data in the right pane.

**TIP:** You can open any number of Explorer tabs so that you can administer and focus on a particular group of servers or objects in one window and accomplish separate tasks in another tab.

For more information, see Datasource Explorer Nodes.
Datasource Explorer Nodes
The Datasource Explorer tree organizes and nests subjects as branches, or nodes. By expanding or collapsing the tree, you can efficiently browse multiple servers and databases and penetrate to the appropriate level of detail.

The Datasource Explorer tree nodes give you quick access to instance and server configuration, as well as security and performance functionality. The node names depend on the platform. For example, the Instance node in IBM DB2 is very similar to the Server node in Sybase ASE.

When you click the Datasource node of the Explorer, DBArtisan lists all the databases available for that datasource. Similarly, by clicking the Databases Node you can view all the database object types available for that particular datasource. Finally, by drilling down to the object type, DBArtisan displays available objects on in the right pane.

TIP: For IBM DB2 for Linux, Unix, and Windows, DBArtisan offers enhanced navigation of the system, instance, and database nodes of the Datasource Explorer. DBArtisan lets you view, manage, and navigate DB2 for Windows/Unix datasources at the instance node level rather than at the database level.

Datasource Explorer Options
The Datasource Explorer offers options that lets you specify the display of the Datasource Explorer. To set the options, click the Datasource Explorer menu bar, and then select option. The table below describes the functionality of the Datasource Explorer:

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organize by Object Type</td>
<td>Select to display objects by object type for all users in the same list.</td>
</tr>
<tr>
<td>Organize by Owner</td>
<td>Select to display objects by owner. Most efficient if you are working with databases containing a high number of objects.</td>
</tr>
<tr>
<td>Show Only My Objects</td>
<td>Select to display the objects you own in the Datasource Explorer. Available if you are organizing the Datasource Explorer by object type.</td>
</tr>
<tr>
<td>Show System Objects</td>
<td>Select to display system objects.</td>
</tr>
<tr>
<td>Full Refresh</td>
<td>Select to refresh.</td>
</tr>
<tr>
<td>Expand All Groups</td>
<td>Select to expand all groups.</td>
</tr>
<tr>
<td>Collapse</td>
<td>Select to collapse all datasources or collapse all groups.</td>
</tr>
<tr>
<td>Retain View Settings</td>
<td>Select to retain the current state of the Datasource Explorer so that the it opens the same way the next time you start DBArtisan.</td>
</tr>
</tbody>
</table>

TIP: You can also set these option on the Explorer Tab of the Options Editor.

Datasource Explorer Platform Nodes
The Datasource Explorer displays the following nodes:

<table>
<thead>
<tr>
<th>IBM DB/2 for Windows, Unix, and Linux</th>
<th>IBM DB/2 for OS390 and Z/OS</th>
<th>Microsoft SQL Server</th>
<th>MySQL</th>
<th>Oracle</th>
<th>Sybase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Processes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Databases Node</td>
<td>Databases Node</td>
<td>Databases Node</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instance Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Node</td>
<td>Performance Node</td>
<td>Performance Node</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**APPLICATION BASICS > DBARTISAN PRODUCT DESIGN**

**EMBARCADERO TECHNOLOGIES > DBARTISAN USER GUIDE 35**

Databases Node

NOTE: The Databases node is available for IBM DB2 for OS390, Microsoft SQL Server, MySQL, and Sybase ASE.

Databases are a collection of tables, or a collection of index spaces and tablespaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery. Information management centers on the deployment and administration of databases and database applications.

**The Datasource Explorer List**

When you select the Databases node of the Datasource Explorer, DBArtisan lists the databases and displays pertinent information about each of them in the right pane. The tables below describes the information displayed for those DBMS that have a Databases node:

<table>
<thead>
<tr>
<th>IBM DB/2 for Windows, Unix, and Linux</th>
<th>IBM DB/2 for OS390 and Z/OS</th>
<th>Microsoft SQL Server</th>
<th>MySQL</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Node description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Node</td>
<td>Storage Node</td>
<td>Storage Node</td>
<td>Storage Node</td>
<td>Storage Node</td>
<td>Storage Node</td>
<td>Storage Node</td>
</tr>
<tr>
<td>Subsystem</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Subsystem</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IBM DB/2 for OS390 and Z/OS</th>
<th>Microsoft SQL Server</th>
<th>MySQL</th>
<th>Sybase</th>
<th>Node description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlterEdts</td>
<td></td>
<td></td>
<td></td>
<td>Time when the most recent alter edit was applied</td>
</tr>
<tr>
<td>Bufferpool</td>
<td></td>
<td></td>
<td></td>
<td>Name of default bufferpool for the database, blank if it is a system database</td>
</tr>
<tr>
<td>CreateEdts</td>
<td></td>
<td></td>
<td></td>
<td>Time when the create statement was created for the database</td>
</tr>
<tr>
<td>Create Date/Time</td>
<td></td>
<td>Create Date/Time</td>
<td></td>
<td>When the database was created</td>
</tr>
<tr>
<td>Creator</td>
<td></td>
<td></td>
<td></td>
<td>Name of creator</td>
</tr>
<tr>
<td>Database Status</td>
<td>Database Status</td>
<td></td>
<td></td>
<td>Status of the database, such as online, offline, read-only or single-user</td>
</tr>
<tr>
<td>DB ID</td>
<td></td>
<td></td>
<td></td>
<td>Internal identification number for database</td>
</tr>
<tr>
<td>Group Member</td>
<td></td>
<td></td>
<td></td>
<td>The DB2 sharing member name</td>
</tr>
<tr>
<td>Index Bufferpool</td>
<td></td>
<td></td>
<td></td>
<td>Indicates name of the default bufferpool for indexes</td>
</tr>
<tr>
<td>Last Dump Tran</td>
<td></td>
<td></td>
<td></td>
<td>The last time that the transaction log for the database was dumped</td>
</tr>
<tr>
<td>MIXED_CCSID</td>
<td></td>
<td></td>
<td></td>
<td>Indicates graphic characters</td>
</tr>
<tr>
<td>Name</td>
<td>Database Name</td>
<td>Name</td>
<td>Database Name</td>
<td>Name of database</td>
</tr>
</tbody>
</table>
For more information, see Datasource Explorer Nodes.

**NOTE:** MySQL does not support the concept of owner and databases for MySQL are registered manually. When you select the database node for MySQL, you see the name of the database.

### Instance Node

**NOTE:** The Instance node is available for IBM DB2 and Oracle.

DBArtisan places Instance as the first level of information under the Datasource node in the Datasource Explorer. Instance includes:

- DB Manager Configuration
- Datasources
- Version
- Configuration
- Oracle Job Queues

### Accessing an Instance

Connect a datasource to access an instance. After you have connected to your datasource:

1. On the **Datasource Explorer**, expand the target datasource node.
2. On the **Datasource Explorer**, expand the **Instance** node.

### Editing Node Connection

1. On the **Datasource Explorer**, expand the target datasource node.
2. On the **Datasource Explorer**, right-click the **Instance** node.
3. Select **Edit Node Connection**.

   DBArtisan opens the Connect Login for Node dialog box.

For more information, see Datasource Explorer Nodes.

---

<table>
<thead>
<tr>
<th>IBM DB/2 for OS390 and Z/OS</th>
<th>Microsoft SQL Server</th>
<th>MySQL</th>
<th>Sybase</th>
<th>Node description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Online Yes or no.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Size Size of the database in MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Stogroup Name of stogroup for the database, blank if it is a system database</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Suspended Yes or no.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Timestamp Indicates timestamp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Type Indicates type of database; W indicates a work file</td>
</tr>
</tbody>
</table>

For more information, see Datasource Explorer Nodes.
Database Configuration

The Database Configuration node displays under the Instance node of the Datasource Explorer. When you select the Configuration node, DBArtisan displays configuration information about your datasource in the right pane of the Explorer window. You can use the Datasource Explorer or shortcut menu to:

- Open the Instance Parameters dialog box.
- Open the Edit Configuration dialog box.
- Extract SQL.
- Execute an HTML report.

Opening the Instance Parameters dialog box

Connect to a datasource to access an instance.

1. On the Datasource Explorer, expand the target datasource node.
2. On the Datasource Explorer, expand the Instance node.
3. Click Configuration.

DBArtisan displays the configuration information in the right pane of the Explorer window.

4. On the Datasource Explorer toolbar, click Open.
   OR
   In the right pane of the Explorer window, right-click the target configuration parameter, and then click Open.
   OR
   In the right pane of the Explorer window, double-click the target configuration parameter, and then click Open.

DBArtisan opens Instance Parameters dialog box.

Version Information

**NOTE:** Version is available for MySQL, Microsoft SQL Server and Oracle.

DBArtisan places Version under either the Server or Instance node of the Datasource Explorer. When you select Version, DBArtisan shows version information about your datasource in the right pane of the Explorer window. Version information varies according to the database platform’s manufacturer.

Displaying Version Information

Connect to a datasource to access version information. After you have connected to a datasource, do the following:

1. On the Datasource Explorer, expand the Datasource node.
2. On the Datasource Explorer, expand the Server or Instance node.
3. Click Version.

DBArtisan displays version information in the right pane of the Explorer window.

Performance Node

**NOTE:** The Performance node is available for all DBMSs.
The Performance node on the Datasource Explorer lets you quickly access to the following features of the Database Monitor:

<table>
<thead>
<tr>
<th>IBM DB/2 for Windows, Unix, and Linux</th>
<th>IBM DB/2 for OS390 and Z/OS</th>
<th>Microsoft SQL Server</th>
<th>MySQL</th>
<th>Oracle</th>
<th>Sybase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocking Locks</td>
<td>Blocking Locks</td>
<td>Blocking Locks</td>
<td>Blocking Locks</td>
<td>Blocking Locks</td>
<td></td>
</tr>
<tr>
<td>Buffer Pools</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image Copies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Locks</td>
<td>Locks</td>
<td>Locks</td>
<td>Locks</td>
<td>Locks</td>
<td>Locks</td>
</tr>
<tr>
<td>Processes</td>
<td>Processes</td>
<td>Processes</td>
<td>Processes</td>
<td>Processes</td>
<td></td>
</tr>
<tr>
<td>Threads</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Traces</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utilities Node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can select the Performance node and view the corresponding statistics in the right pane of the Explorer window. To utilize all Database Monitor functionality, on the Utilities menu, click Database Monitor.

IBM DB2 for Linux, Unix, and Windows and IBM DB2 OS390
Performance is a result of the DB2 API calls. The DB2 API calls are subject to change depending on the version and DB2 SDK call interface of the DB2 server.

**NOTE:** Performances do not appear in the system catalog tables.

For more information, see [Datasource Explorer Nodes](#).

Image Copies

**NOTE:** Image Copies are available for IBM DB2 for z/OS and OS/390 only.

The Image Copies node shows the image copies on the subsystem. Image Copies are a full image copy of all pages in a tablespace, partition, data set, or index space.

For more information, see [Datasource Explorer Nodes](#).

Processes

**NOTE:** Processes is available for IBM DB2, Microsoft SQL Server, MySQL, Oracle, and Sybase ASE.

The Processes node on the Datasource Explorer displays processes that are currently running on your DBMS.

For more information, see [Datasource Explorer Nodes](#).

Locks

**NOTE:** Locks are available for all DBMSs.

The Locks node on the Datasource Explorer displays the lock objects held on the table and row IDs for your database. Locks are a way of controlling concurrent events and access to data. DBArtisan lets you view all locks associated with a database. To utilize all Database Monitor functionality, on the Utilities menu, click Database Monitor.
For more information, see Datasource Explorer Nodes.

**Refreshing the Locks Tab**
1. On the Datasource Explorer, select the **Locks** node. DBArtisan displays the locked sessions in the right pane of the Explorer window.
2. On the Datasource Explorer toolbar, click **Refresh**.

**Blocking Locks**
The Blocking Locks node on the Datasource Explorer displays only the locks that are blocked. To utilize all Database Monitor functionality, on the Utilities menu, click Database Monitor.

For more information, see Datasource Explorer Nodes.

**Refreshing the Blocking Locks Tab**
1. On the Datasource Explorer select the **Blocking Locks** node. DBArtisan displays the blocked sessions in the right pane of the Explorer window.
2. On Datasource Explorer toolbar, click **Refresh**.

**Threads**

**NOTE:** Threads are available for IBM DB2 for OS390.

Threads are control structures in IBM DB2 for z/OS and OS/390, which:

- Describe application connection.
- Trace application progress.
- Process resource functions.
- Communicate SQL statement status after SQL execution.

There are three types of threads:

1. **Active Threads**
2. **Indoubt Threads**
3. **Inactive Threads**

The table below describes the information DBArtisan displays in the right pane when you select the Active, Indoubt or Inactive Thread nodes on the Datasource Explorer:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token</td>
<td>Token number designation</td>
</tr>
<tr>
<td>Name</td>
<td>Location name of the system</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates status of the conversation or socket</td>
</tr>
<tr>
<td>A</td>
<td>Indicates event control</td>
</tr>
<tr>
<td>Req</td>
<td>Indicates Req number value</td>
</tr>
<tr>
<td>ID</td>
<td>Program currently running</td>
</tr>
</tbody>
</table>
For more information, see Datasource Explorer Nodes.

Active Threads: Active threads are the processes running on your database. When you select the Active Thread node on the Datasource Explorer, DBArtisan displays the active threads that are currently running on the IBM DB2 for z/OS and OS/390 database. To utilize all Database Monitor functionality, on the Utilities menu, click Database Monitor.

Refreshing Active Threads
1 On the Datasource Explorer, select the Active Threads node.
   DBArtisan displays the active sessions in the right pane of the Explorer window.
2 In the right pane of the Explorer window, click the target active thread(s).
3 On the Datasource Explorer toolbar, click the Refresh button.
   OR
   In the right pane of the Explorer window, right-click the active thread(s), and then click Refresh.
   DBArtisan refreshes the active thread(s).

Killing Active Threads
1 On the Datasource Explorer, select the Active Threads node.
   DBArtisan displays the active sessions in the right pane of the Explorer window.
2 In the right pane of the Explorer window, click the target active thread(s).
3 On the Datasource Explorer toolbar, click the Kill button.
   OR
   In the right pane of the Explorer window, right-click the active thread(s), and then click Kill.
   DBArtisan refreshes the active thread(s).

For more information, see Datasource Explorer Nodes.

Indoubt Threads: Indoubt threads are processes waiting for resources to become active. When you select the Indoubt Thread node on the Datasource Explorer, DBArtisan displays the indoubt threads that are currently running on the IBM DB2 for z/OS and OS/390 database. To utilize all Database Monitor functionality, on the Utilities menu, click Database Monitor.

Refreshing Indoubt Threads
1 On the Datasource Explorer, select the Indoubt Threads node.
   DBArtisan displays the indoubt sessions in the right pane of the Explorer window.
2 In the right pane of the Explorer window, click the target indoubt thread(s).
3 On the **Datasource Explorer** toolbar, click the **Refresh** button.

 OR

 In the right pane of the **Explorer** window, right-click the indoubt thread, and then click **Refresh**.

 DBArtisan refreshes the indoubt thread.

**Killing Indoubt Threads**

1 On the **Datasource Explorer**, select the **Indoubt Threads** node.

 DBArtisan displays the indoubt sessions in the right pane of the Explorer window.

2 In the right pane of the **Explorer** window, click the target indoubt thread(s).

3 On the **Datasource Explorer** toolbar, click the **Kill** button.

 OR

 In the right pane of the **Explorer** window, right-click the indoubt thread, and then click **Kill**.

 DBArtisan kills the indoubt thread.

For more information, see [Datasource Explorer Nodes](#).

**Inactive Threads:** Inactive threads are processes that are idle. When you select the Inactive Thread node on the Datasource Explorer, DBArtisan displays the inactive threads that are currently running on the IBM DB2 for z/OS and OS/390 database. To utilize all Database Monitor functionality, on the Utilities menu, click Database Monitor.

**Refreshing Inactive Threads**

1 On the **Datasource Explorer**, select the **Inactive Threads** node.

 DBArtisan displays the inactive threads in the right pane of the Explorer window.

2 In the right pane of the **Explorer** window, click the target inactive thread(s).

3 On the **Datasource Explorer** toolbar, click the **Refresh** button.

 OR

 In the right pane of the **Explorer** window, right-click the inactive thread(s), and then click **Refresh**.

 DBArtisan refreshes the inactive thread.

**Killing Inactive Threads**

1 On the **Datasource Explorer**, select the **Inactive Threads** node.

 DBArtisan displays the inactive threads in the right pane of the Explorer window.

2 In the right pane of the **Explorer** window, click the target inactive thread(s).

3 On the **Datasource Explorer** toolbar, click the **Kill** button.

 OR

 In the right pane of the **Explorer** window, right-click the inactive thread(s), and then click **Kill**.

 DBArtisan kills the inactive thread.

For more information, see [Datasource Explorer Nodes](#).
Traces

**NOTE:** Traces are available for IBM DB2 for z/OS and OS/390 only.

The Traces node displays a list of active traces. Traces let you monitor and collect DB2 for OS/390 monitoring, auditing, performance, accounting, statistics, and serviceability (global) data.

**NOTE:** This information is read-only.

For more information, see [Datasource Explorer Nodes](#).

Buffer Pools

The Buffer Pools node displays a list of buffer pools, the main storage that is reserved to satisfy the buffering requirements for one or more table spaces or indexes.

For more information, see [Datasource Explorer Nodes](#).

Utilities Node

**NOTE:** The Utilities node is available for IBM DB2 for OS/390 only.

The Utilities node displays wizards that are typically run from the command line and do not apply to a particular type of schema object. Some of these wizards are also available via the command menus for objects. The wizards include:

- Check Data
- Check Index
- Copy
- Copy to Copy
- Load
- Merge Copy
- Modify Recovery
- Modify Statistics
- Quiesce Tablespace
- Rebuild Index
- Recover
- Reorg Index
- Reorg Tablespace
- Runstats Index
- Runstats Tablespace
- Unload
- Maintain Template

**TIP:** To run a wizard, select the command in the right pane of the application and click the Execute button on the Datasource Explorer toolbar.
For more information, see Datasource Explorer Nodes.

Schema Node

NOTE: Schema node is available for IBM DB2 for Linux, Unix, and Windows, IBM DB2 for OS390, and Oracle.

DBArtisan offers an environment for managing database objects across different database platforms. DBArtisan stores information about server and object types in object editors. Each editor is a tabbed dialog box that groups related information together. Each editor includes a context-sensitive Commands menu with pertinent functionality for the desired object. Many of the object editors contain a DDL tab.

The Schema node displays the underlying target object's SQL. The object editor toolbar has a refresh button, which lets you refresh object editor contents, clear the object editors, and log SQL if SQL Logging is turned on. This can be useful when you have completed an object and want to see the changes immediately. For objects that have dependent objects, such as tables, triggers, procedures, and views, you can view and access their dependent objects directly from the object tab, instead of having to go back to the correct node of the Datasource Explorer.

For more information, see Datasource Explorer Nodes.

Security Node

NOTE: The Security node is available for all DBMSs.

The Security node gives you quick access to related security functions and lets you administer and control security on your database from a single vantage point. Rather than having to sift through a large number of objects, you can simply create and alter security settings to be applied across the server from this top-level.

Each DBMS offers different security options:

<table>
<thead>
<tr>
<th>IBM DB/2 for Windows, Unix, and Linux</th>
<th>IBM DB/2 for OS390 and Z/OS</th>
<th>Microsoft SQL Server</th>
<th>MySQL</th>
<th>Oracle</th>
<th>Sybase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups</td>
<td>Logins</td>
<td>Logins</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Linked Servers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Users</td>
<td>Users</td>
<td>Users</td>
<td>Users</td>
<td>Users</td>
<td></td>
</tr>
</tbody>
</table>

Application Processes

NOTE: The Application Processes node is available for IBM DB2 for z/OS and OS/390.

DBArtisan places Application Processes as the first level of information under the Datasources node in the Datasource Explorer. The Application Processes node of the Datasource Explorer includes portions of the mainframe DB2 subsystem that relate to application programming. The Application Processes node also includes two components that are industry innovations: a search tool for statements, and a means to expose previously established access paths.

Application Processes includes:

* Plan
Plan
A Plan is an executable application created in the bind process. It can include one or more packages or debris.

The Datasource Explorer List
You can select the Plan node in the Datasource Explorer and view the corresponding statistics. The table below describes the information DBArtisan derives and displays about plans from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan Name</td>
<td>Name of plan</td>
</tr>
<tr>
<td>Owner/Creator</td>
<td>User name of plan owner or creator</td>
</tr>
<tr>
<td>Qualifier</td>
<td>Name of qualifier</td>
</tr>
<tr>
<td>Valid</td>
<td>Indicates if plan is valid or invalid</td>
</tr>
<tr>
<td>Operative</td>
<td>Indicates if plan is operative</td>
</tr>
<tr>
<td>Isolation</td>
<td>Level of isolation</td>
</tr>
<tr>
<td>Reoptvar</td>
<td></td>
</tr>
<tr>
<td>Packlists</td>
<td>Number of packlists</td>
</tr>
<tr>
<td>Environments</td>
<td>Number of environments</td>
</tr>
<tr>
<td>Bind Date/Timestamp</td>
<td>Time stamp of bind</td>
</tr>
</tbody>
</table>

TIP: Double-click a specific plan open the Plan Editor.

For more information, see Application Processes.

DBRM Node

NOTE: DBRM is for IBM DB2 for z/OS and OS/390 only.

A DBRM is an application containing information on SQL statements extracted from the DB2 precompiler. A DBRM refers to a specific database and is stored outside of the DB2 directory.

The Datasource Explorer List
When you select the DBRM node of the Datasource Explorer, DBArtisan lists pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays DBRMs from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBRM Name</td>
<td>Displays the name of the DBRM.</td>
</tr>
<tr>
<td>Version</td>
<td>Displays DBRM version.</td>
</tr>
</tbody>
</table>
Server Node

**NOTE:** The Server node is available for MySQL, Microsoft SQL Server, and Sybase ASE.

DBArtisan places Server as the first level of information under the Datasource node in the Datasource Explorer. Server includes:

**MySQL**
- Version
- Server Configuration
- Scheduler

**Microsoft SQL Server**
- Version
- Configuration
- SQL Server Agent
- Error Logs

**Sybase ASE**
- Version
- Configuration

**Accessing a Server**
Connect to a datasource to access a server. After you have connected to your datasource, do the following:

1. On the **Datasource Explorer**, expand the datasource node.
2. On the **Datasource Explorer**, expand the **Server** node.

**Server Version Information**
DBArtisan places Version under the Server node of the Datasource Explorer. When you select Version, DBArtisan shows version information about your datasource in the right pane of the Explorer window.
Displaying Version Information
Connect to a datasource to view version information. After you have connected to a datasource, do the following:

1. On the **Datasource Explorer**, expand the **Datasource** node.
2. On the **Datasource Explorer**, expand the **Server** node.
3. Click **Version**.

DBArtisan displays version information in the right pane of the Explorer window.

For more information, see [Server Node](#).

Configuration
DBArtisan places **Configuration** under the **Server** node of the **Datasource Explorer**. When you select **Configuration**, DBArtisan shows configuration information about your datasource in the right pane of the Explorer window. You can use the Datasource Explorer or shortcut menu to:

- Open the Server Configuration dialog box.
- Open the Edit Configuration dialog box.
- Extract SQL.
- Execute an HTML report.

Opening the Server Configuration Dialog Box
Connect to a datasource to access a server. After you have connected to your datasource, do the following:

1. On the **Datasource Explorer**, expand the datasource node.
2. On the **Datasource Explorer**, expand the **Server** node.
3. Click **Configuration**.

DBArtisan displays configuration information in the right pane of the Explorer window.

4. Select the target configuration parameter.
5. On the **Datasource Explorer** toolbar click **Open**.

   OR

   In the right pane of the **Explorer** window, right-click the configuration parameter, and then click **Open**.

DBArtisan opens the Server Configuration dialog box.

For more information, see [Server Node](#).

Microsoft SQL Server Agent
DBArtisan places **SQL Server Agent** under the **Server** node of the **Datasource Explorer**. DBArtisan assists you in using the Microsoft SQL Server Agent service to automate administration of a Microsoft SQL server. Microsoft SQL Server Agent manages, and schedules, allowing programmed responses to server events or the execution of scheduled administrative tasks. Automating recurring administrative tasks and responses to server events can save considerable time for other administrative tasks.

DBArtisan lets you define the three main components of automatic administration managed by the SQL Server Agent service. These components are:

- **Alerts**
To use the Microsoft SQL Serve Agent features in DBArtisan, you need to install Microsoft SQL Enterprise Manager, Query Analyzer, or Profiler tool.

**NOTE:** The SQL Server Agent must be installed if you have purchased and plan to use the Backup Analyst.

**NOTE:** Be aware that when you work with the Alerts, Operators, or Jobs features and click Help, you connect directly to SQL Server Books online. Embarcadero Technologies is not responsible for the contents of the SQL Server Books materials.

### Alerts

An alert is a definition that matches one or more Microsoft SQL Server events with a response. Errors and messages, or events, are generated by Microsoft SQL Server and entered into the Microsoft Windows application log. SQL Server Agent reads the application log and compares events to alerts that you defined. When SQL Server Agent finds a match, it fires an alert. DBArtisan lets you define alerts to respond to SQL Server events by notifying operators, forwarding the event to another server, or executing a job. These administrative tasks runs automatically if the SQL Server Agent service has been started.

When you select the Alerts node on the Datasource Explorer, DBArtisan displays all alerts for the server in the right pane of the Explorer window. The column headings in the right pane of the Explorer window provide important information about alerts. These headings are:

- Name
- Enabled
- Error
- Severity
- Last Occurred
- E-mail
- Panel
- Net Send
- Count

You can use the shortcut menu to access functionality associated with alerts. Shortcut menus let you:

- View Properties
- Create a New Alert
- Drop an Alert
- Find Object
- Select All
- Unselect All

**NOTE:** On the shortcut menu, when you click New, DBArtisan opens the Microsoft SQL Server Agent Alert Properties dialog box where you can enter the new alert information.

You can also open the *Microsoft SQL Server Agent Alert Properties* dialog box by double-clicking a particular alert, or by clicking the Properties button on the Datasource Explorer.
Operators
Operators are individuals responsible for the maintenance of one or more computers running Microsoft SQL Server. E-mail, panel, and Net Send can notify these operators.

When you select the Operators node on the Datasource Explorer, DBArtisan displays all operators defined for the server in the right pane. The column headings in the right pane of the Explorer window provide important information about operators. These headings are:

- Name
- Enabled
- Last Email
- Last Panel
- Last Net Send

You can use the shortcut menu to access functionality associated with operators. The shortcut functionality lets you:

- View Properties
- Create a New Operator
- Drop an Operator
- Find Object
- Select All
- Unselect All

**NOTE:** On the shortcut menu, when you click New, DBArtisan opens the Microsoft SQL Server Agent Operator Properties dialog box where you can enter the new operator information.

You can also open the *Microsoft SQL Server Agent* Operator Properties dialog box by double clicking a particular operator, or by clicking the Properties button on the Datasource Explorer.

Jobs
Jobs define administrative tasks that can execute on a local server according to schedules or in response to alerts. A job is comprised of job steps that can be:

- Executable Programs
- Windows Commands
- Transact-SQL Statements
- Microsoft ActiveX Script
- Replication Agents

When you select the Jobs node on the Datasource Explorer, DBArtisan displays all jobs for the target server in the right pane of the Explorer window. The column headings in the right pane of the Explorer window provide important information about jobs. These headings are:

- Name
- Category
- Scheduled
- Status
• Next Run Status (Start Date)
• Next Run Date

You can use the shortcut menu to access functionality associated with jobs. The shortcut menu functionality lets you:

• View Properties
• Start Job
• Stop Job
• Enable Job
• Disable Job
• View Job History
• Create a New Job
• Drop a Job
• Find Object
• Select All
• Unselect All

On the Shortcut menu, when you click New, DBArtisan opens the Microsoft SQL Server Agent Job Properties dialog box where you can enter the new job information.

You can also open the Microsoft SQL Server Agent Job Properties dialog box by double clicking on a particular job, or by clicking the Properties button on the Datasource Explorer.

Error Logs
DBArtisan places Error Logs under the Server node of the Datasource Explorer. Error logs are comprised of information recorded by Microsoft SQL Server. When you expand the Error Logs node, you can see the current and archived error logs. When you select the Error Logs node, DBArtisan displays the current and archived error logs in the right pane of the Explorer window. To note information, DBArtisan uses a blue information icon, and to note errors, DBArtisan uses a red error icon.

The column headings in the right pane of the Explorer window provide important information about error logs. These headings are:

• Name
• Source
• Message

You can use the Datasource Explorer toolbar or the shortcut menu to access functionality associated with error logs. The shortcut menu functionality lets you:

• Find
• Toggle All Messages
• Generate an HTML Report
• Refresh the Window
• Find Object
• Select All
• Unselect All

**TIP:** Double-click any error log to display the message.

For more information, see [Server Node](#).

### Storage Node

**NOTE:** The Storage node is available for all DBMSs.

DBArtisan places all storage mechanisms under the Storage node of the Datasource Explorer. Storage objects include:

<table>
<thead>
<tr>
<th>IBM DB/2 for Windows, Unix, and Linux</th>
<th>IBM DB/2 for OS390 and Z/OS</th>
<th>Microsoft SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bufferpools</td>
<td>Bufferpools</td>
<td>Control Files</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Directories</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data Caches</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Database Devices</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dump and Backup Devices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nodegroups</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Redo Log Groups</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rollback Segments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stogroups</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tablespaces</td>
<td>Tablespaces</td>
<td>Tablespaces</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Subsystem

**NOTE:** The Subsystem node is available for IBM DB2 for z/OS and OS/390 only.

The Subsystem node displays detailed information about the DB2 subsystem. Subsystem objects include:

- Connections
- DDF
- Parameters

### Connections

Connections displays the current connections to the subsystem.

**NOTE:** The information is read-only.

For more information, see [Subsystem](#).
DDF
DDF (distributed data facility) displays the information regarding the status and configuration of the DDF, a set of DB2 for OS/390 components through which DB2 for OS/390 communicates with another RDBMS.

NOTE: The information is read-only.

For more information, see Subsystem.

Parameters
Parameters displays the DB2 subsystem parameters.

NOTE: The information is read-only.

For more information, see Subsystem.

Organizing the Explorer
DBArtisan includes functionality that lets you configure how objects are organized in the Datasource Explorer. This option lets you manage and administer your database servers more effectively by resolving differences in structural concepts introduced by a heterogeneous administration environment. You can also set the defaults for the organization of the explorer from the Explorer Tab on the Options Editor.

The table below describes the methods of organizing the Explorer:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organize by Object Owner</td>
<td>The Datasource Explorer displays objects for each user, categorized by object type. This display mode is most efficient if you are working with databases containing a high number of objects.</td>
</tr>
<tr>
<td>Organize by Object Type</td>
<td>The Datasource Explorer displays all objects, categorized by object type, for all users in the same list. This display mode speeds or slows performance in databases that contain many objects.</td>
</tr>
<tr>
<td>Show System Objects</td>
<td>The Datasource Explorer displays all system objects.</td>
</tr>
<tr>
<td>Show Only my Objects</td>
<td>This option applies if you are organizing the Datasource Explorer by object type. Selecting the check box causes DBArtisan to show only the objects you own in the Datasource Explorer.</td>
</tr>
</tbody>
</table>

Organizing the Explorer by Object Owner
1. On the Datasource Explorer, click the drop-down Explorer header button.
2. Click Organize by Object Owner.

DBArtisan dynamically reorganizes the display of the Explorer sorting database objects by object owner.

Organizing the Explorer by Object Type
1. On the Datasource Explorer, click the drop-down Explorer header button.
2. Click Organize by Object Type.

DBArtisan dynamically reorganizes the display of the explorer sorting database objects by type.
Showing or Hiding System Objects
1. On the Datasource Explorer, click the drop-down Explorer header button.
2. To show system objects, click Show System Objects. DBArtisan shows all system objects.
3. To hide system objects, click Show System Objects. DBArtisan hides all system objects.

Displaying the Explorer
DBArtisan provides flexibility in the configuration of its display. DBArtisan lets you open or close multiple Explorer Tab windows. This feature lets you configure the DBArtisan desktop to suit the task at hand. The default setting for the Datasource Explorer is a single tab docked on the left-hand side of the application.

Opening or Reopening a Datasource Explorer Tab
1. On the Datasource menu, click Datasource.

Hiding or Closing a Datasource Explorer Tab
1. Click the target Datasource Explorer Tab.
2. On the Datasource Explorer toolbar, click Close.

Enterprise Management Console
The Enterprise Management Console interface is an enterprise-wide view of managed databases. It is available when the application is focused on the top-level node of a datasource group. You can use this feature to do multiple administrative operations with one or two clicks.

The Enterprise Management Console gives you quick access to the following functionality:
- Connecting to your datasources
- Modifying datasource registration
- Database Monitor
- Data Load
- Data Unload
- Database Schema Extraction
- Database Schema Migration

Parallel Query Execution/Multithreading
DBArtisan takes full advantage of 32-bit architecture by supporting parallel query execution or multithreading.

DBArtisan accelerates the development process by executing multiple queries simultaneously using separate operating system threads and improving response time because most operations can start immediately and return control to the primary application thread.

DBArtisan lets you run multiple, concurrent queries against the same datasource.
DBArtisan employs a database connection pooling technique that dynamically spawns connections as needed and
disconnects them when the associated query has completed.

DBArtisan increases throughput by automatically assigns query threads to separate processors if your system has
multiple CPUs. When you execute parallel queries against different database servers using a multi-processor
computer, you experience near linear gains in throughput.

Describe Window
The Describe Window is a floating window that lets you view procedures, tables, views, and functions (Oracle and IBM
DB2 for Linux, Unix, and Windows only). In the Describe Window you can view columnar information (for tables and
views) or input parameter information (for procedures and functions).

Opening the Describe Window
DBArtisan offers three ways to open the Describe window:

1. In an editor, right-click an object and then click Describe.
2. On the View menu, select an object and then click Describe.
3. On the right pane of the Explorer or in an editor, select an object and then press CTRL+D.

Using the Describe Window
In the Describe window:

1. Click the Name list and then click a name to view a list of types of objects in the database.
2. Click the Owner list and then click an owner to view a list of all owners of objects in the database.
3. Click the Type list and then click a type to view columnar information (for tables and views) or input parameter
   information (for functions and procedures).

DBArtisan Toolbars and Menus
DBArtisan offers multiple ways to navigate through the application:

- Main Menu
- Toolbars
- Shortcut Menus

For more information, see:

Customizing Toolbars
Customizing Commands
Keyboard Commands
Main Menu

You can access DBArtisan’s features from the Main Menu by clicking the menu name, and then selecting the target command on the sub-menu. The menus are context-sensitive, changing as you perform tasks. DBArtisan menu items include:

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Create, open, close, print and save script files and result sets. Set application options and defaults. <strong>Note:</strong> When you open New, you have easy access to a new ISQL window, the DDL Editor, a Java file, or the Active Script.</td>
</tr>
<tr>
<td>Datasource</td>
<td>Create, modify, select, manage, connect to, and disconnect from datasources. Access database object editors. Display or hide the Datasource Explorer. Access session recording for Tivoli integration.</td>
</tr>
<tr>
<td>Utilities</td>
<td>Access utilities to extract schema, migrate schema, export and import data, execute scripts, search databases, monitor process activity and locks, schedule tasks, and restore, manage, and backup databases.</td>
</tr>
<tr>
<td>Logfile</td>
<td>Activate/deactivate, open, flush, and annotate the DBArtisan application log.</td>
</tr>
<tr>
<td>View</td>
<td>Configure DBArtisan to hide or show toolbars, status bars, and windows. Access full screen mode.</td>
</tr>
<tr>
<td>Analyst</td>
<td>Opens the Space Analyst, Capacity Analyst, Backup Analyst, and Performance Analyst.</td>
</tr>
<tr>
<td>Bookmarks</td>
<td>Access and manage bookmarks.</td>
</tr>
<tr>
<td>Window</td>
<td>Cascade and tile open windows. Arrange icons. Toggle between open windows.</td>
</tr>
<tr>
<td>Help</td>
<td>Lets you access context-sensitive Help, link to Embarcadero on the Web, and see license information when you click About...</td>
</tr>
<tr>
<td>Edit</td>
<td>Edit and manipulate the text in your SQL scripts. Available only when an ISQL Window is selected.</td>
</tr>
<tr>
<td>Query</td>
<td>Execute and set query options for a SQL script. Available only when an ISQL Window is selected.</td>
</tr>
<tr>
<td>Format</td>
<td>Format the contents of result sets. Available only when a Result Window is selected.</td>
</tr>
</tbody>
</table>

For more information, see [Toolbars and Menus](#).

Explorer Bookmarks

DBArtisan supports easy datasource navigation with the Explorer bookmarks. You can right-click any node of the Explorer tree and add a bookmark. After a bookmark is created, you can simply select it from the Bookmark menu, and be instantly positioned and connected to the resource in the Datasource Explorer tree.

**Creating Explorer Bookmarks**

1. On the **Datasource Explorer**, right-click the target node, and then select **Add Bookmark**.
   - DBArtisan opens the Add Friendly Bookmark Name dialog box.
2. Type the Explorer bookmark name.
3. Click **OK**.
   - DBArtisan displays the Explorer bookmark under the Bookmarks menu. Explorer bookmarks are organized by platform.
Editing Explorer Bookmarks

1. **On the Main Menu**, select **Bookmarks**.

2. **Select Bookmark Manager**.
   
   DBArtisan opens Bookmark Manager.

3. **To rename the Explorer bookmark**, select the target explorer bookmark, and then click **Rename**.
   
   DBArtisan opens the Edit Bookmark Name dialog box.

4. **Type the new explorer bookmark name**.

5. **Click OK**.

6. **To delete an Explorer bookmark**, select the target bookmark, and then click **Delete**.

   **TIP:** To add explorer bookmarks without using the Add Friendly Bookmark Name dialog box, select **Do not show ‘Add Friendly Bookmark Name’ dialog** option.

**Toolbars**

DBArtisan toolbars change to reflect the element of the application you are using. These toolbars contain buttons that are the fastest way to access commonly used features of DBArtisan. You can move the toolbars to horizontal or vertical positions anywhere on the screen, and you can toggle any of the floating toolbars off and on by using the shortcut menu when the pointer is positioned over one of DBArtisan's toolbars.

**Registration Toolbar**

**Windows Toolbar**

**Main Toolbar**

**Utilities Toolbar**

**Analyst Toolbar**

**Edit Toolbar**
Datasource Explorer Toolbar

Object Editor Toolbar

Moving a Toolbar
1. Click the pointer at the top edge of any toolbar.
2. Drag the toolbar to the new position.

For more information, see Toolbars and Menus.

Customize Dialog Box
The table below describes the options and functionality on the tabs of the Customize dialog box where you can customize the toolbar and commands:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toolbars</td>
<td>Toolbars</td>
<td>Select the toolbars you want to display in the application.</td>
</tr>
<tr>
<td>Application</td>
<td>Microsoft Office 97</td>
<td>Lets you specify DBArtisan's UI appearance. Select for DBArtisan's classic, Microsoft Office 97 style.</td>
</tr>
<tr>
<td>Visual Style</td>
<td>Microsoft Office XP</td>
<td>Lets you specify DBArtisan's UI appearance. Select for Microsoft Office XP style.</td>
</tr>
<tr>
<td></td>
<td>OneNote-style Tabs</td>
<td>Select for DBArtisan to display tabs in the same style as OneNote in Microsoft Office 2003, the note-taking and management program.</td>
</tr>
<tr>
<td>Menu animations</td>
<td></td>
<td>Lets you specify the menu animation style.</td>
</tr>
<tr>
<td>Menu Shadows</td>
<td></td>
<td>Displays shadowed 3D effects.</td>
</tr>
<tr>
<td>Apply</td>
<td></td>
<td>Click to apply the DBArtisan visual style.</td>
</tr>
<tr>
<td>Tools</td>
<td>Menu contents</td>
<td>Lets you define external applications to run from the Tools menu of DBArtisan.</td>
</tr>
<tr>
<td>Keyboard</td>
<td>Category</td>
<td>Select a general category for a hot key for the command.</td>
</tr>
<tr>
<td></td>
<td>Commands</td>
<td>Select a hot key command, based on the general category.</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Displays the command description.</td>
</tr>
<tr>
<td></td>
<td>Set Accelerator for</td>
<td>Select application area where you want new hot key to be active.</td>
</tr>
<tr>
<td></td>
<td>Current Keys</td>
<td>Displays current hot key.</td>
</tr>
<tr>
<td></td>
<td>Press New Shortcut Key</td>
<td>Press keyboard key or an unassigned F key.</td>
</tr>
<tr>
<td>Options</td>
<td>Show ScreenTips on toolbars</td>
<td>Select to display a ScreenTip when you hover your mouse over a button. For example, when you hover your mouse over the New button, DBArtisan displays the ScreenTip &quot;New.&quot;</td>
</tr>
</tbody>
</table>
Shortcut Menus

DBArtisan incorporates shortcut menus to give you another way to access object functionality. These menus mirror the functionality that you can access from application toolbars or the main menu.

Accessing Shortcut Menus

1. On Desktop, right-click DBArtisan.

DBArtisan opens the shortcut menu.

2. Click the target shortcut menu command.

For more information, see:
Toolbars and Menus

Keyboard Commands

DBArtisan provides a number of Keyboard commands that expedite IISQL Editor development. The table below describes the available keyboard commands:

<table>
<thead>
<tr>
<th>Category</th>
<th>Action</th>
<th>Keyboard shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>General editing</td>
<td>Delete one character to the left</td>
<td>BACKSPACE</td>
</tr>
<tr>
<td></td>
<td>Delete one character to the right</td>
<td>DELETE</td>
</tr>
<tr>
<td></td>
<td>Cut selected text to the Clipboard</td>
<td>CTRL+X</td>
</tr>
<tr>
<td></td>
<td>Undo the last action</td>
<td>CTRL+Z</td>
</tr>
<tr>
<td></td>
<td>Redo the last undo operation</td>
<td>CTRL+Y</td>
</tr>
<tr>
<td></td>
<td>Copy text</td>
<td>CTRL+C</td>
</tr>
<tr>
<td></td>
<td>Paste the Clipboard contents</td>
<td>CTRL+V</td>
</tr>
<tr>
<td>Extending selections</td>
<td>One character to the right</td>
<td>SHIFT+RIGHT ARROW</td>
</tr>
<tr>
<td></td>
<td>One character to the left</td>
<td>SHIFT+LEFT ARROW</td>
</tr>
<tr>
<td></td>
<td>To the end of a word</td>
<td>CTRL+SHIFT+RIGHT ARROW</td>
</tr>
<tr>
<td></td>
<td>To the beginning of a word</td>
<td>CTRL+SHIFT+LEFT ARROW</td>
</tr>
<tr>
<td></td>
<td>To the end of a line</td>
<td>SHIFT+END</td>
</tr>
<tr>
<td></td>
<td>To the beginning of a line</td>
<td>SHIFT+HOME</td>
</tr>
<tr>
<td></td>
<td>One line down</td>
<td>SHIFT+DOWN ARROW</td>
</tr>
<tr>
<td></td>
<td>One screen up</td>
<td>SHIFT+PAGE UP</td>
</tr>
<tr>
<td>Category</td>
<td>Action</td>
<td>Keyboard shortcut</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td></td>
<td>To the beginning of a document</td>
<td>CTRL+SHIFT+HOME</td>
</tr>
<tr>
<td></td>
<td>To the end of a document</td>
<td>CTRL+SHIFT+END</td>
</tr>
<tr>
<td></td>
<td>To include the entire document</td>
<td>CTRL+A</td>
</tr>
<tr>
<td>Moving the insertion point</td>
<td>One character to the left</td>
<td>LEFT ARROW</td>
</tr>
<tr>
<td></td>
<td>One character to the right</td>
<td>RIGHT ARROW</td>
</tr>
<tr>
<td></td>
<td>One word to the left</td>
<td>CTRL+LEFT ARROW</td>
</tr>
<tr>
<td></td>
<td>One word to the right</td>
<td>CTRL+RIGHT ARROW</td>
</tr>
<tr>
<td></td>
<td>Up one line</td>
<td>UP ARROW</td>
</tr>
<tr>
<td></td>
<td>Down one line</td>
<td>DOWN ARROW</td>
</tr>
<tr>
<td></td>
<td>To the end of a line</td>
<td>END</td>
</tr>
<tr>
<td></td>
<td>To the beginning of a line</td>
<td>HOME</td>
</tr>
<tr>
<td></td>
<td>Up one screen (scrolling)</td>
<td>PAGE UP</td>
</tr>
<tr>
<td></td>
<td>Down one screen (scrolling)</td>
<td>PAGE DOWN</td>
</tr>
<tr>
<td></td>
<td>To the end of a document</td>
<td>CTRL+END</td>
</tr>
<tr>
<td></td>
<td>To the beginning of a document</td>
<td>CTRL+HOME</td>
</tr>
<tr>
<td>Bookmarks</td>
<td>Toggle bookmark on/off</td>
<td>CTRL+F2</td>
</tr>
<tr>
<td></td>
<td>Go to next bookmark</td>
<td>F2</td>
</tr>
<tr>
<td></td>
<td>Go to previous bookmark</td>
<td>SHIFT+F2</td>
</tr>
<tr>
<td>Splitter windows</td>
<td>Go to next pane</td>
<td>F6</td>
</tr>
<tr>
<td></td>
<td>Go to previous pane</td>
<td>SHIFT+F6</td>
</tr>
<tr>
<td>Debugger operations</td>
<td>Start Debugging</td>
<td>CTRL+F5</td>
</tr>
<tr>
<td></td>
<td>Stop Debugging</td>
<td>SHIFT+F5</td>
</tr>
<tr>
<td></td>
<td>Step Over</td>
<td>F10</td>
</tr>
<tr>
<td></td>
<td>Step Into</td>
<td>F11</td>
</tr>
<tr>
<td></td>
<td>Run to Cursor</td>
<td>CTRL+F10</td>
</tr>
<tr>
<td></td>
<td>Step Out</td>
<td>SHIFT+F11</td>
</tr>
<tr>
<td></td>
<td>Describe from Cursor</td>
<td>CTRL+D</td>
</tr>
<tr>
<td></td>
<td>Insert or Remove Breakpoint</td>
<td>F9</td>
</tr>
<tr>
<td></td>
<td>Toggle (Enable or Disable) Breakpoint</td>
<td>CTRL+F9</td>
</tr>
<tr>
<td></td>
<td>Edit Breakpoint</td>
<td>ALT+F9</td>
</tr>
<tr>
<td></td>
<td>Go</td>
<td>F5</td>
</tr>
<tr>
<td></td>
<td>Restart</td>
<td>CTRL+SHIFT+F5</td>
</tr>
<tr>
<td>Debugger windows</td>
<td>Open or Close Watch Window</td>
<td>ALT+3</td>
</tr>
<tr>
<td></td>
<td>Open or Close Variables Window</td>
<td>ALT+4</td>
</tr>
<tr>
<td></td>
<td>Open or Close Call Stack Window</td>
<td>ALT+5</td>
</tr>
<tr>
<td></td>
<td>Open or Close Dependency Tree Window</td>
<td>ALT+6</td>
</tr>
<tr>
<td>SQL Window</td>
<td>Go to the Result Tab</td>
<td>CTRL+ALT+R</td>
</tr>
<tr>
<td></td>
<td>Go to the Query Tab</td>
<td>CTRL+ALT+Q</td>
</tr>
</tbody>
</table>
Customizing the Tools Menu

The **Customize Tools Menu** dialog box lets you add up to sixteen commands to the Tools menu. You can associate any program that running on your local machine. You can use this feature to associate frequently used programs and commands, thereby adding functionality directly to the DBArtisan application. In addition, you can specify arguments for any command that you add to the Tools menu. You can use these arguments to decipher how you want to execute commands. DBArtisan includes many pre-defined arguments that let you customize the application.

During the initial installation, DBArtisan automatically adds the following items (if installed locally) to the Tools menu.

- ODBC Administrator
- Oracle Service Name Manager
- Sybase Open Client/Server Config
- Sybase Directory Service Edit (DSEDIT)
- SQL Server Network Library Config
- SQL Server Books Online
- DB2 Client Configuration Assistant

For more information, see **Toolbars and Menus**.

**Customize User Tools Menu Dialog Box:** DBArtisan lets you customize the Tools menu.

The table below describes the options and functionality on the Customize User Tools Menu dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu Contents</td>
<td>In the box, click the target command.</td>
</tr>
<tr>
<td>Menu Text</td>
<td>In the box, type the name of the tool as you want to appear on the Tools menu.</td>
</tr>
<tr>
<td>Command</td>
<td>In the box, type the path and name of the program for the new tool or click Browse to locate the program. For example, C:\WINDOWS\NOTEPAD.EXE</td>
</tr>
<tr>
<td>Arguments</td>
<td>To pass an argument to the command, in the box, type any argument or click the Arrow, and then click the target argument. <strong>NOTE:</strong> To display a menu of pre-defined arguments, click the drop-down arrow next to the Argument box, and then select an argument from the list to insert the syntax into the box.</td>
</tr>
<tr>
<td>Initial Directory</td>
<td>To specify an initial directory, in the box, type the path and name of the file directory or click Browse to locate the file directory.</td>
</tr>
<tr>
<td>Add Button</td>
<td>Click to add a new command to Tools menu.</td>
</tr>
<tr>
<td>Remove Button</td>
<td>In the Menu Contents box, click the command you want to remove, and then click the button.</td>
</tr>
<tr>
<td>Move Up Button</td>
<td>Click to move the command up one position in the menu.</td>
</tr>
<tr>
<td>Move Down Button</td>
<td>Click to move the command down one position in the menu.</td>
</tr>
</tbody>
</table>

**Completing the Customize User Tools Menu Dialog Box**

To complete the Customize User Tools Menu dialog box, do the following:

1. On the **Tools** Menu, select **Customize**.

   DBArtisan opens the Customize User Tools dialog box.
2. Select options.
3. Click OK.

For more information, see Toolbars and Menus.

Specifying Arguments
DBArtisan provides a number of pre-defined arguments that you can pass to programs that you have added to the Tools menu. You can specify an argument to be passed to a program for newly added commands by choosing one of DBArtisan's pre-defined arguments or typing in your own arguments. The table below describes the pre-defined arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$FilePath$$</td>
<td>The complete filename of the current source (defined as drive+path+filename); blank if a non-source window is active.</td>
</tr>
<tr>
<td>$$FileDir$$</td>
<td>The directory of the current source (defined as drive+path); blank if a nonsource window is active.</td>
</tr>
<tr>
<td>$$FileName$$</td>
<td>The filename of the current source (defined as filename); blank if the non-source window is active.</td>
</tr>
<tr>
<td>$$FileExt$$</td>
<td>The filename extension of the current source; blank if a non-source window is active.</td>
</tr>
<tr>
<td>$$CurLine$$</td>
<td>The current pointer line position within the active window.</td>
</tr>
<tr>
<td>$$CurCol$$</td>
<td>The current pointer column position within the active window.</td>
</tr>
<tr>
<td>$$CurText$$</td>
<td>The current text (the word under the current pointer position, or the currently selected text, if there is one.)</td>
</tr>
<tr>
<td>$$CurDir$$</td>
<td>The current working directory (defined as drive+path.)</td>
</tr>
<tr>
<td>$$CurDatasource$$</td>
<td>The name of the current datasource as defined in DBArtisan.</td>
</tr>
<tr>
<td>$$CurUserID$$</td>
<td>The name of the current datasource user.</td>
</tr>
<tr>
<td>$$CurPwd$$</td>
<td>The current datasource password.</td>
</tr>
<tr>
<td>$$CurConString$$</td>
<td>The current connection string or server name.</td>
</tr>
</tbody>
</table>

**NOTE:** Arguments are case-sensitive.

The table below describes examples for using the pre-defined arguments and command-line arguments:

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTEPAD.EXE</td>
<td>$$FilePath$$</td>
<td>Launches Microsoft Notepad displaying the contents of the $$FilePath$$ argument.</td>
</tr>
<tr>
<td>iSQL.EXE</td>
<td>-U$$CurUserID$$ -P$$CurPwd$$ -S$$CurConString$$ -i$$FilePath$$</td>
<td>Launches ISQL, automatically connects to the current datasource using the current user name and password, and executes the contents of $$FilePath$$.</td>
</tr>
<tr>
<td>SQLPLUS.EXE</td>
<td>$$CurUserID$$/$$CurPwd$$@$$CurConString$$ @$$_$FilePath$$</td>
<td>Launches SQL*Plus, connects to the current datasource using the current user name and password, and executes the contents of $$FilePath$$.</td>
</tr>
</tbody>
</table>

**TIP:** DBArtisan lets you specify an argument in the DBArtisan opens the Customize dialog box.

For more information, see Toolbars and Menus.
DBArtisan Dialog Boxes
DBArtisan offers the following common dialog boxes. Additional dialog boxes are discussed in the Using section.

- Preview
- Impact Analysis
- Generate Report

DBArtisan Wizards
DBArtisan offers several wizards that let you create objects without knowing the underlying SQL commands. DBArtisan also offers wizards to perform common tasks.

For more information, see:

- Registration Wizard
- Completing a Dialog Box
- Utilities Wizards

DBArtisan Editors
DBArtisan stores information about server and object types in object editors. An editor is a tabbed dialog box that groups related information together. Each editor contains a context-sensitive Commands menu with pertinent functionality for the object. Many of the object editors contain a DDL Tab. This tab displays the underlying target object's SQL.

**TIP:** The Object Editor tool bar has a refresh button, that lets you refresh Object Editor contents, clear the Object Editors, and log SQL if SQL Logging is turned on.

If an objects has dependent objects, such as tables, triggers, procedures and views, you can view and access their dependent objects in the editor.

Customizing the DBArtisan Display
DBArtisan lets you customize your display to suit your work habits.

For more information, see:

- Full Screen Mode
- Active Windows Dialog Box
- Navigating DBArtisan Windows
- Output Window
- Set Sort Columns Dialog Box
Full Screen Mode

DBArtisan is equipped with Full Screen mode capabilities that let you conceal the application framework and take advantage of the entire monitor area. Full screen mode hides any other applications running on the computer and uses every available pixel for DBarisan. Main menu functionality is accessible through keyboard commands when you use Full Screen Mode.

Activating Full Screen Mode

1. On the View menu, click Full Screen.

DBArtisan expands the application to fit the entire monitor area, and displays the Full Screen Mode icon as a stand-alone floating toolbar.

Dismissing Full Screen Mode

1. Click Full Screen.

DBArtisan reduces the size of the application.

TIP: If you closed the Full Screen Mode toolbar, right-click the top of the DBarisan desktop to display the full screen.

For more information, see Customizing the DBarisan Display.

Active Windows Dialog Box

DBArtisan’s multi-tasking environment lets you open multiple active windows. DBarisan’s Active Windows dialog box lets you see all the open windows in the application workspace. DBarisan displays active windows as tabs in the Datasource Explorer. Each tab represents a separate window that lets you execute, create or view the objects or schema in your database.

TIP: DBarisan lets you use a mouse with a wheel to navigate in active windows.

Opening the Active Windows Dialog Box

1. On the Window menu, click Windows.

DBArtisan opens the Active Windows dialog box and any open windows in the application workspace.

The table below describes the options and functionality on the Active Windows dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate</td>
<td>Sets the focus on the selected window and closes the Active Windows dialog box.</td>
</tr>
<tr>
<td>OK</td>
<td>Closes the Active Windows dialog box and makes any changes you have made to the windows in the workspace.</td>
</tr>
<tr>
<td>Save</td>
<td>Saves the contents of the selected window. If you are saving a file, DBarisan prompts you to provide a name and location for the file.</td>
</tr>
<tr>
<td>Close Window</td>
<td>Closes the selected window. If you have not saved the contents of the window, DBarisan prompts you to do so.</td>
</tr>
<tr>
<td>Help</td>
<td>Initiates and displays this Help topic in the DBarisan Help.</td>
</tr>
</tbody>
</table>

For more information, see Customizing the DBarisan Display.
Navigating DBArtisan Windows
DBArtisan offers functionality through the Windows toolbar that lets you:

- Tile windows vertically or horizontally.
- Cascade open windows.
- Minimize all windows and move to the next active window.

**TIP:** DBArtisan lets you use a mouse with a wheel to navigate in active windows.

The table below describes the commands available on the Windows menu:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window</td>
<td>Activates the Active Windows dialog box.</td>
</tr>
<tr>
<td>Cascade</td>
<td>Layers all active windows on the desktop so that you can view all active windows simultaneously.</td>
</tr>
<tr>
<td>Tile Horizontal</td>
<td>Tiles all active windows on the desktop horizontally.</td>
</tr>
<tr>
<td>Tile Vertical</td>
<td>Tiles all active windows on the desktop vertically.</td>
</tr>
<tr>
<td>Next</td>
<td>Moves the focus of the application to the next active window.</td>
</tr>
<tr>
<td>Previous</td>
<td>Moves the focus of the application to the previous active window.</td>
</tr>
</tbody>
</table>

For more information, see [Customizing the DBArtisan Display](#).

Output Window
DBArtisan has a separate window that captures all messages returned by the server and to log entries about the progress of operations started by the application.

The Output Window includes the following tabs:

- General
- Find In Files 1
- Find In Files 2

The Output Window lets you do the following to server messages:

- View
- Save
- Print
- Copy
- Flush

For more information, see [Customizing the DBArtisan Display](#).

**CONFIGURING THE OUTPUT WINDOW**
DBArtisan lets you:

- Float the [Output Window](#).
- Dock the [Output Window](#).
• Hide the **Output Window**.
• Display the **Output Window**.

**Floating the Output Window (with Smart Docking)**
This is the default view when you first open DBArtisan.

1. Place your pointer on the **Output Window** and drag it to another location.
   Arrows appear and you can move the window to the top, bottom, or sides of the view within DBArtisan or to another location on your desktop.
2. To change the location of the **Output Window**, repeat the first step.
3. To disable the feature, right-click and choose **Docking** or **Hide**.

**Docking the Output Window**
1. Place your pointer on the **Output Window**.
2. Right-click, and then click **Docking View**.
   DBArtisan docks the Output Window to the bottom of the application frame.

**Undocking the Output Window**
1. Place your pointer on the **Output Window**.
2. Right-click, and then click **Docking View**.
   DBArtisan floats the Output Window.

**Hiding the Output Window**
1. On the **View** menu, click **Output**.
   DBArtisan hides the Output Window.

**Displaying the Output Window**
1. On the **View** menu, click **Output**.
   DBArtisan displays the Output Window.

For more information, see **Customizing the DBArtisan Display**.

**SAVING OUTPUT WINDOW MESSAGES**
DBArtisan lets you save messages in the **Output Window** to a file.

1. Right-click a message in the **Output Window**, and then click **Save**.
   DBArtisan opens the Save As dialog box.
2. In the **File Name** box, type the location and name of the file.
   DBArtisan defaults the file extension to .msg.
3. Click **Save**.
   DBArtisan saves the file.

For more information, see **Customizing the DBArtisan Display**.
PRINTING OUTPUT WINDOW MESSAGES
DBArtisan lets you print the messages you see in the Output Window.

1. Right-click a message in the Output Window, and then click Print.
   DBArtisan opens the Print dialog box.
2. Configure your print job.
3. Click Print.
   DBArtisan prints the file.
For more information, see Customizing the DBArtisan Display.

COPYING OUTPUT WINDOW MESSAGES
DBArtisan lets you copy all or part of the contents of the Output Window to the clipboard.

1. Select the target Server Messages.
2. Right-click, and then click Copy.
   DBArtisan copies the selected messages to the Windows clipboard.
3. Paste the text into other Windows applications.
For more information, see Customizing the DBArtisan Display.

CLEARING OUTPUT WINDOW MESSAGES
DBArtisan lets you clear server messages from the Output Window.

1. Right-click a message in the Output Window, and then click Clear.
   DBArtisan clears the server messages.
For more information, see Customizing the DBArtisan Display.

Set Sort Columns Dialog Box
The Set Sort Columns dialog box lets you sort multiple columns, and sort column identification, in the Right Pane of the Datasource Explorer.

Completing the Set Columns Dialog Box
To complete the Set Columns dialog box, do the following:

1. In the right pane of the application, right-click a column heading and select Sort Multiple Columns.
   DBArtisan opens the Set Sort Columns dialog box.
2. In Column Name select the column(s) to use to sort the information in the right pane of the application.
3. Click the right arrow to move the column(s) to the Column Name box.
4. Click the up and down arrows to change the order of the sort columns.
For more information, see Set Sort Column Dialog Box.
## Specifying DBArtisan application and feature options

DBArtisan lets you customize the application configuration for your specific needs. All DBArtisan application settings are available in the Options Editor, which is organized in a tabular format based on feature sets. The table below describes the options and functionality on the Options Editor tabs:

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Auto format options</strong></td>
<td>Specifies the style and spacing of SQL statements in an ISQL window.</td>
</tr>
<tr>
<td><strong>Connection options</strong></td>
<td>Specifies the timeout parameters, packet size for a connection, and ANSI to OEM settings.</td>
</tr>
<tr>
<td><strong>Data Editor options</strong></td>
<td>Specifies settings for Data Editor.</td>
</tr>
<tr>
<td><strong>Data transfer options</strong></td>
<td>Sets default directories when performing data unload and load and data export and import operations.</td>
</tr>
<tr>
<td><strong>Datasource Explorer options</strong></td>
<td>Sets defaults for the organization of objects in the Datasource Explorer.</td>
</tr>
<tr>
<td><strong>Datasource options</strong></td>
<td>Specifies how to store the datasource registry on the local machine.</td>
</tr>
<tr>
<td><strong>DDL extract options</strong></td>
<td>Specifies whether or not DBArtisan should include DROP statements when extracting the schema of different database object types.</td>
</tr>
<tr>
<td><strong>Debug options</strong></td>
<td>Sets the duration of your debug initialization and debug session, enable or disable DBMS_OUTPUT, and enable the refresh option.</td>
</tr>
<tr>
<td><strong>Directory options</strong></td>
<td>Sets the default directories for placing the output of different operations such as HTML reports or schema extractions.</td>
</tr>
<tr>
<td><strong>Editors options</strong></td>
<td>Sets defaults for general ISQL display, enabling syntax highlighting and line numbers, command history parameters, formatting, file tracking, and auto-save intervals for ISQL Windows.</td>
</tr>
<tr>
<td><strong>General options</strong></td>
<td>Sets defaults for automatic login, restoring the last session, and other general application options.</td>
</tr>
<tr>
<td><strong>Grid properties (Results window) options</strong></td>
<td>Dictates the physical appearance of the results window grid.</td>
</tr>
<tr>
<td><strong>ISQL options</strong></td>
<td>Sets defaults for the maximum allowable errors before aborting the execution of a SQL script, executing selected text, and the position of Query and Results tabs.</td>
</tr>
<tr>
<td><strong>Java options</strong></td>
<td>Specifies load Java files and drop Java files.</td>
</tr>
<tr>
<td><strong>JDBC options</strong></td>
<td>Lets you specify Java Virtual Machine options in addition to the JDBC driver location for each platform.</td>
</tr>
<tr>
<td><strong>Logging options</strong></td>
<td>Sets defaults for SQL Logging.</td>
</tr>
<tr>
<td><strong>Licensing options</strong></td>
<td>Displays the location of your log file and allows you to turn licensing logging errors on or off. <strong>NOTE:</strong> Accept the defaults unless you are working with Embarcadero Support and are instructed to make changes.</td>
</tr>
<tr>
<td><strong>Migration options</strong></td>
<td>Sets the default options for schema migration operations.</td>
</tr>
<tr>
<td><strong>MySQL Utilities Options</strong></td>
<td>Specifies paths to MySQL utilities.</td>
</tr>
<tr>
<td><strong>Oracle Utilities options</strong></td>
<td>Specifies the location of the Oracle Utilities.</td>
</tr>
<tr>
<td><strong>Perf Center options</strong></td>
<td>Specifies Performance Center’s integration with DBArtisan.</td>
</tr>
<tr>
<td><strong>Query Builder options</strong></td>
<td>Specifies global settings for Query Builder.</td>
</tr>
<tr>
<td><strong>Results (ISQL) options</strong></td>
<td>Specifies auto format result sets, sets the display and format of Result Windows, and the mail file type and default fonts.</td>
</tr>
<tr>
<td><strong>SMTP mail options</strong></td>
<td>Lets you specify defaults for your outgoing mail notifications.</td>
</tr>
<tr>
<td><strong>Space Analyst options</strong></td>
<td>Sets Space Analyst options.</td>
</tr>
<tr>
<td><strong>Warnings options</strong></td>
<td>Activates specific warnings when undesirable actions are attempted against a database.</td>
</tr>
</tbody>
</table>
To specify options for the DBArtisan application or a particular feature:

1. On the **File** menu, click **Options**.
   
   DBArtisan opens the Options Editor.

2. Click the tab corresponding to the feature you want to customize.
   
   Refer to the table above for a description of the available tabs. Consult the referenced topic for detailed descriptions of the feature options available.

3. Set feature options on the tab and then click **OK**.

   **NOTE:** If there is an open document, DBArtisan opens the **Update Document Statement Properties** dialog box. The Update Document Statement Properties dialog box lets you override changes you made to a current document or documents with new setting you made in the Options Editor.

### Auto format options

**NOTE:** This option is for Oracle users only.

After opening the Options editor (see [Specifying DBArtisan application and feature options](#)), you can make changes to the **Auto Format** tab.

The Auto Format Tab of the Options Editor lets you specify the style and spacing of SQL statements in an ISQL windows when you choose to auto format Oracle objects. The options you set on the Auto Format Tab of the Options Editor apply to all SQL statement operations. Before you can set Auto Format options, you must open the Options Editor. The table below describes the options and functionality on the **Auto Format** Tab:

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Style Tab:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Character Case</td>
<td>Lets you specify the character case of Oracle <strong>Keywords</strong> (such as BEGIN, LOOP, and INSERT), non-Oracle keywords such as <strong>Variables</strong> and object names, Oracle <strong>Built-ins</strong>, and <strong>Built-in Packages</strong> (such as DBMS_SQL or DBMS_OUTPUT).</td>
<td>Upper case.</td>
</tr>
<tr>
<td>Spacing Tab:</td>
<td>Tabspace Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Keep Tabs</td>
<td>Keeps the original tabs from the original PL/SQL script.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Extra tabs</td>
<td>Adds tabs to the original PL/SQL script. The Extra tabs option and the Insert Spaces option are mutually exclusive.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Insert Spaces</td>
<td>Replaces tabs from the original PL/SQL script with spaces.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Extra tabs option and the Insert Spaces option are mutually exclusive.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Whitespace Retention</td>
<td>Lets you retain white space.</td>
<td>Not enabled</td>
</tr>
<tr>
<td></td>
<td>General</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Indent Size</td>
<td>Specifies the number of spaces a line indents after a line wrap.</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Right Margin</td>
<td>Specifies the maximum number of characters per line.</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>Defaults</td>
<td>Resets the options on page to the original settings.</td>
<td>Available</td>
</tr>
<tr>
<td>Stacking Tab:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parameter Stacking</td>
<td>Displays the embedded script parameters in a staggered list.</td>
<td>Not Selected</td>
</tr>
</tbody>
</table>
Connection options
After opening the Options editor (see Specifying DBArtisan application and feature options), you can configure a number of server parameters for interacting with your datasources on the Connection tab.

**NOTE:** Setting Connection Options are available for Microsoft SQL Server and Sybase ASE.

The table below describes the options and functionality on the Connection Tab:

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compact (Stacked)</td>
<td>Displays the embedded script parameters by line.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Column Stacking</td>
<td>Standard (Wrapped)</td>
<td>Not Selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displays the embedded script columns in a staggered list.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compact (Stacked)</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Defaults</td>
<td>Resets parameter and column stacking options to default,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>compact (stacked.)</td>
<td></td>
</tr>
</tbody>
</table>

**Data Editor options**
After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Data Editor tab.

**NOTE:** Option parameters set in the Options Editor override options set within Query Builder and Data Editor.
The table below describes the options and functionality on the Data Editor Tab:

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Execution Mode</td>
<td>Live Mode</td>
<td>lets you execute changes one row at a time. Batch Mode lets you make unlimited changes prior to execution. Batch Mode offers three sub-options: Ignore errors-continue processing, Prompt on Error, and Stop after error(s) (number of errors allowed before stopping execution)</td>
<td>Live Mode</td>
</tr>
<tr>
<td>Data Editor File</td>
<td>Mail File Type</td>
<td>Sets the default mail output style as Results, Tab Delimited, Comma Separated, or HTML.</td>
<td>Results</td>
</tr>
<tr>
<td></td>
<td>Include column titles when saving</td>
<td>Includes column titles when saving.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Grid Font</td>
<td>Customizes font style, and size for the Data Editor and the Results Grid.</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Printer Font</td>
<td>Sets font style, and size for printing output.</td>
<td>Available</td>
<td></td>
</tr>
<tr>
<td>Auto Format (Best Fit)</td>
<td>Fits formatting to match your desktop.</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td>Begin and End Transaction Statements</td>
<td>Adds a beginning and ending transaction on each statement.</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td>Default Date/Time Format</td>
<td>Default Date/Time format</td>
<td>Displays the current date/time format and lets you customize the date/time display.</td>
<td>Results</td>
</tr>
<tr>
<td></td>
<td>Use Calendar Control as default</td>
<td>If selected, DBArtisan uses the Calendar Control window.</td>
<td>Not selected</td>
</tr>
<tr>
<td></td>
<td>2 digit year system setting warning</td>
<td>If selected, DBArtisan sends a warning when you use a two digit year system setting.</td>
<td>Selected</td>
</tr>
<tr>
<td>Confirmation Dialog Options</td>
<td>Enabling Show Delete Confirmation Dialog, Show Update Confirmation Dialog, or Show Update LOB Confirmation Dialog lets you display a confirmation dialog box when you use a delete command, update a row, or update a LOB, respectively..</td>
<td>Selected</td>
<td></td>
</tr>
</tbody>
</table>

### Data transfer options

After opening the Options editor (see [Specifying DBArtisan application and feature options](#)), you can make changes to the Data Transfer tab.

You can configure DBArtisan to use default directories when performing data unload and load and data export and import operations. Setting a default directory saves time because it describes a single reference point for loading or unloading, exporting and importing table or view data files.

The table below describes the options and functionality on the Data Transfer Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Unload</td>
<td>Specifies the name and location of the default directory in the Data Unload box.</td>
</tr>
<tr>
<td>Data Load</td>
<td>Specifies the name and location of the default directory in the Data Load box.</td>
</tr>
<tr>
<td>Oracle Export (for Oracle)</td>
<td>Specifies the name and location of the default directory in the Oracle Export box.</td>
</tr>
<tr>
<td>Oracle Import (for Oracle)</td>
<td>Specifies the name and location of the default directory in the Oracle Import box.</td>
</tr>
</tbody>
</table>
Datasource options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Datasource tab. The table below describes the options and functionality on the Datasource Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recent Datasource List Contains</td>
<td>Lets you specify number of datasources to display in the datasource list.</td>
<td>6</td>
</tr>
<tr>
<td>Check Server Component When Connect to DB2/OS390 Database</td>
<td>IBM DB2 for OS/390 ONLY: Lets you select option to check server component when you connect to a database.</td>
<td>Selected</td>
</tr>
<tr>
<td>Default to Alias Usage When Defining New Datasources</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Re-sort the Datasource Combo based on Connection state</td>
<td>When this box is checked, list of datasources on the Datasource toolbar identifies the currently connected datasources above all others. When this box is unchecked, datasources are listed in strict alphabetical order.</td>
<td>Selected</td>
</tr>
</tbody>
</table>

Datasource Explorer options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Explorer tab. The Explorer Tab of the Options Editor lets you configure how objects are organized in the Datasource Explorer.

The table below describes the options and functionality on the Explorer Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organize by Object Owner</td>
<td>Groups objects, by object type, for each user. This display mode is most efficient if you are working with databases containing a high number of objects.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Organize by Object Type</td>
<td>Groups objects by object type for all users in the same list.</td>
<td>Selected</td>
</tr>
<tr>
<td>Show Only My Objects</td>
<td>Shows only the objects you own in the Datasource Explorer. Available if you are organizing the Datasource Explorer by object type.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Show System Objects</td>
<td>Shows all system objects in the Datasource Explorer.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Refresh after Object Commands</td>
<td>Refreshes the Explorer automatically after an object has been modified or created.</td>
<td>Selected</td>
</tr>
<tr>
<td>Enable Landing Page View</td>
<td>Displays informative metadata that gives you a quick overview of your server/database.</td>
<td>Selected</td>
</tr>
<tr>
<td>Retain View Setting on Close</td>
<td>Select to retain the current state of the Datasource Explorer so that it opens the same way the next time you start DBArtisan.</td>
<td>Selected</td>
</tr>
</tbody>
</table>

DDL extract options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the DDL Extract tab.

The DDL Extraction Tab of the Options Editor lets you specify the object types for which DBArtisan should include DROP statements when extracting their schema. Selecting the check boxes instructs DBArtisan to include DROP statements.
You can choose to include DROP statements before you perform ad hoc DDL extractions. You can use this feature to modify and to re-compile database objects. To recompile a database object, drop it before recreating it. This option drops any existing objects of the same name before recreating the object. The data in the existing table is not saved when you specify a DROP statement for extracted DDL.

The table below describes the options and functionality on the **DDL Extract** Tab of the Options Editor:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select RDBMS</td>
<td>Lets you select the appropriate platform to display available object types and the object types whose DROP statements you want to include when extracting their schema.</td>
</tr>
<tr>
<td>Extract Sybase Login with User</td>
<td>Lets you extract the Sybase ASE login when you extract a user.</td>
</tr>
<tr>
<td>Extract to multiple windows</td>
<td>Select to extract the schema for each object into separate DDL windows.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> This option only works when you extract DDL for multiple objects.</td>
</tr>
<tr>
<td>Extract in dependency order</td>
<td>This is the default. When you select a number of objects as part of an extraction job, this option ensures objects will be extracted in the proper dependency order. If the option is not selected, errors may result when you execute the script. It’s also true, however, that loading the dependencies can add significant overhead when you are extracting numerous objects.</td>
</tr>
<tr>
<td>Script Use Statement</td>
<td>This option is for Sybase. Optimizes extraction through generating Use statements in the generated script.</td>
</tr>
<tr>
<td>Auto Format Oracle Objects</td>
<td>This option is for Oracle. Select if you want DDL Extract display format default to Auto Format Parameters.</td>
</tr>
</tbody>
</table>

**CAUTION:** Because dropping an object is a destructive action, you should carefully consider including drop statements before activating this option.

The table below shows the object types you can include a DROP statement with when performing an ad hoc DDL extraction:

<table>
<thead>
<tr>
<th>Option</th>
<th>DB/2 for OS/390 and z/OS</th>
<th>DB/2 for Windows, Unix, and Linux</th>
<th>Oracle</th>
<th>SQL Server</th>
<th>Sybase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Constraints</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Database</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Defaults</td>
<td></td>
<td>Yes (default)</td>
<td>Yes</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
</tr>
<tr>
<td>Functions</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indexes</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
</tr>
<tr>
<td>Packages</td>
<td></td>
<td>Yes (default)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedures</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
<td>Yes (default)</td>
</tr>
<tr>
<td>Rules</td>
<td></td>
<td></td>
<td>Yes (default)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequences</td>
<td></td>
<td>Yes (default)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structure Types</td>
<td></td>
<td>Yes (default)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synonyms</td>
<td></td>
<td>Yes (default)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tables</td>
<td>Yes (default)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Debug options

After opening the Options editor (see [Specifying DBArtisan application and feature options](#)), you can make changes to the **Debug** tab. The Debug Tab of the Options Editor lets you set the duration of your debug initialization and debug session, enable or disable DBMS_OUTPUT, and enable the refresh option.

The table below describes the options and functionality on the **Debug Tab**:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Dependency Tree Option</td>
<td>SYBASE DEBUGGER ONLY: Pre-Fetch Options.</td>
<td>Pre-Fetch All Dependencies</td>
</tr>
<tr>
<td>Profiler</td>
<td>Profiler Time Unit</td>
<td>Lets you select the milliseconds, seconds or minutes.</td>
<td>Milliseconds</td>
</tr>
<tr>
<td></td>
<td>Save Profiler Reports</td>
<td>Lets you save profiler reports and type or browse for the report path.</td>
<td>Not Selected</td>
</tr>
<tr>
<td>Oracle</td>
<td>Initialization Timeout</td>
<td>Specifies the point at which the application stops trying to initialize the debugger. If it cannot initialize the debugger in the specified time, it displays message in the Debug Output window.</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Debug Session Timeout</td>
<td>Specifies the point at which the application terminates your debug session due to idle time.</td>
<td>7200</td>
</tr>
<tr>
<td></td>
<td>Enable DBMS Output</td>
<td>Enables the Oracle built-in package, DBMS_OUTPUT, letting you send messages from stored procedures, packages, and triggers.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Refresh Dependencies for each run</td>
<td>Refreshes the dependencies each time you run the debugger.</td>
<td>Not selected</td>
</tr>
<tr>
<td></td>
<td>Compile with Debug Option</td>
<td>Select options to compile dependent objects while debugging.</td>
<td>Compile dependent options</td>
</tr>
<tr>
<td>DB2</td>
<td>Debug Session Timeout</td>
<td>Specifies the point at which the application terminates your debug session due to idle time.</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>Compile with Debug Option before Debug Session</td>
<td>Lets you specify options.</td>
<td>Prompt Always</td>
</tr>
</tbody>
</table>

### Directory options

After opening the Options editor (see [Specifying DBArtisan application and feature options](#)), you can make changes to the **Directory** tab. DBArtisan lets you configure Directory Options to use default directories when performing certain operations. You can set the default directories for:

- Wizard operations
- Report generation
• Schema extraction
• HTML templates for customizing reports

The table below describes the options and functionality on the **Directories** Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wizard Definitions</td>
<td>Specifies the name and location of the default directory for saving wizard operations. This option applies to wizard operations in which you have the option to save the definition file, such as data unload/load or schema migration operations.</td>
<td>C:\Documents and Settings\Administrator\Application Data\Embarcadero\DBArtisan\DefFiles</td>
</tr>
<tr>
<td>HTML Reports</td>
<td>Specifies the name and location of the default directory for the output from generated HTML reports.</td>
<td>C:\Documents and Settings\Administrator\Application Data\Embarcadero\DBArtisan\Report</td>
</tr>
<tr>
<td>Schema Extraction</td>
<td>Specifies the name and location of the default directory for placing the output from schema extraction operations.</td>
<td>C:\Documents and Settings\Administrator\Application Data\Embarcadero\DBArtisan\Extract</td>
</tr>
<tr>
<td>HTML Template</td>
<td>Specifies the name and location of the default directory where DBArtisan can find the HTML template on which to base HTML reports. This feature lets you customize your HTML reports.</td>
<td>C:\Program Files\Embarcadero\DBA730\HtmlTpl</td>
</tr>
<tr>
<td>User SQL Scripts</td>
<td>Specifies the name and location of the default directory for SQL Scripts.</td>
<td>C:\Documents and Settings\Administrator\Application Data\Embarcadero\DBArtisan\UserSQL Scripts</td>
</tr>
</tbody>
</table>

**Editors options**

After opening the Options editor (see **Specifying DBArtisan application and feature options**), you can make changes to the **Editor** tab. DBArtisan lets you set these ISQL Window options:

• Appearance of the editor
• Syntax coloring and font for SQL scripts
• File tracking
• Line numbers
• Auto-save
• Parameters for the command history feature

The table below describes the options and functionality on the **Editor** Tab:

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window</td>
<td>Show Toolbar</td>
<td>Indicates whether or not DBArtisan should display the ISQL Window toolbar.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Show Status Bar</td>
<td>Indicates whether or not DBArtisan should display the ISQL Window status bar.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Auto-Reload File</td>
<td>If File Tracking is enabled, indicates that the application should automatically reload a file that has been externally modified without prompting you. If you turn this option off, DBArtisan prompts you before reloading your file if external changes have been saved.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Interface Element</td>
<td>Option</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td>File Tracking</td>
<td>Indicates that the ISQL Editor should use the File Tracking Facility to monitor the status of a file. If a file has been modified and saved outside the application, the application loads the most current version of the file into the ISQL Editor based on the options set for Auto-Reload File (see above.)</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Maximize on new or open</td>
<td>Indicates that DBArtisan should maximize the ISQL Editor. If you already have an active MDI Window that is maximized, the default behavior is to maximize a new child window. To deactivate this option, ensure you do not have any active MDI Windows, such as the Explorer.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Auto-Save File</td>
<td>Indicates that files in the ISQL Editor should automatically be saved at the indicated time interval. Specify in minutes the time interval in which files should be saved.</td>
<td>Selected, Every 5 minutes</td>
</tr>
<tr>
<td>Formatting</td>
<td>Auto Indent</td>
<td>Sets automatic indentation for each carriage return, new line in your SQL script.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Expand Tabs</td>
<td>Sets tabs as spaces in result sets.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Tab Size</td>
<td>Indicates the number of spaces you want to substitute for tabs in result sets.</td>
<td>4</td>
</tr>
<tr>
<td>Appearance</td>
<td>Enable Syntax Highlighting</td>
<td>Sets syntax highlighting on so that all keywords and comments are colored for easier reading and debugging.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Show Line Numbers</td>
<td>Places line numbers in the left column of an ISQL Window.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Enable Outlining</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Allow Test Wrapping</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Editor Font</td>
<td>Sets the font face, style, and size for SQL scripts.</td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td>Background Color</td>
<td>Sets the background color of the ISQL window.</td>
<td>White</td>
</tr>
<tr>
<td></td>
<td>Printer Font</td>
<td>Sets printer font face, style, and size for printing scripts from the ISQL Window. If there are no fonts available, install printer fonts in the Fonts in the Windows Control Panel.</td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td>Syntax Coloring</td>
<td>Sets syntax coloring for keywords, comments, quotes, and default text for various file types and scripts from the Syntax Coloring dialog box.</td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td>Emulation</td>
<td>Lets you customize your editing environment to an emulation mode.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command History</td>
<td>Specifies the number of commands you want to save in the Command History list in the top of the ISQL Window toolbar. The maximum value is 99.</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Save Most Recent</td>
<td>Specifies the action you want the application to take when selecting a command from the Command History box. You have the option to be reminded to save a file before overwriting (Ask First), to automatically save a file before overwriting (Always), or to automatically overwrite the file with the command (Never).</td>
<td>Ask First</td>
</tr>
</tbody>
</table>
General options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the General tab. The General tab of the Options Editor lets you specify general application options. The table below describes the options and functionality on the General Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confirm on Exit</td>
<td>Instructs DBArtisan to issue a message confirming the operation before exiting the application.</td>
<td>Selected</td>
</tr>
<tr>
<td>Max Editors in a Single Open Operation</td>
<td>Specifies the maximum number of editors allowable from a single Open operation.</td>
<td>5</td>
</tr>
<tr>
<td>Max Entries in Output Window</td>
<td>Specifies the maximum number of messages that can appear in the Output Window before the contents are flushed. This option conserves memory resources. You can clear the Output Window or raise the maximum number allowed at any time.</td>
<td>1500</td>
</tr>
<tr>
<td>DBA Views (Oracle)</td>
<td>Allows users with DBA role/privileges to view Data Dictionary Usage.</td>
<td>Selected</td>
</tr>
<tr>
<td>ALLViews (Oracle)</td>
<td>Allows all users regardless of assigned privileges to view Data Dictionary Usage.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Preserve Case in Object Identifiers (Oracle)</td>
<td>Preserves case of the database object.</td>
<td>Not selected</td>
</tr>
</tbody>
</table>

Grid properties (Results window) options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Grid Properties tab.

The Grid Properties tab of the Option editor lets you set preferences for the layout and appearance of the grid in an Results window. The table below describes the options and functionality on the Grid Properties tab:

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Titles and Gridlines</td>
<td>3D-Buttons</td>
<td>Enables or disables a 3-d appearance on row and column headings in the grid.</td>
<td>Set</td>
</tr>
<tr>
<td></td>
<td>Horizontal Gridlines</td>
<td>Enables or disables ruling between rows of the grid.</td>
<td>Set</td>
</tr>
<tr>
<td></td>
<td>Vertical Gridlines</td>
<td>Enables or disables ruling between columns of the grid.</td>
<td>Set</td>
</tr>
<tr>
<td></td>
<td>Mark Headings</td>
<td>Enables or disables highlighted row and column headings.</td>
<td>Set</td>
</tr>
<tr>
<td></td>
<td>Preview</td>
<td>Displays a preview of the settings currently selected in the Titles and Gridlines group.</td>
<td></td>
</tr>
</tbody>
</table>

ISQL options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the ISQL tab. The table below describes the options and functionality on the ISQL Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Errors Before Aborting Execution</td>
<td>Sets the maximum number of errors allowed before aborting the execution of a script. A zero indicates that the option is inactive and that you recognize no limit to the number of errors allowed. DBArtisan refers to this value when step executing SQL scripts.</td>
<td>0</td>
</tr>
</tbody>
</table>
Java options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Java tab. The table below describes the options and functionality on the Java Tab:

**NOTE:** Options differ by platform.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute Selected Text</td>
<td>Executes a portion of a selected SQL script.</td>
<td>Selected</td>
</tr>
<tr>
<td>Check Syntax When Executing</td>
<td>For DB2, required to execute DB2 call statements in the ISQL Window.</td>
<td>Not Select</td>
</tr>
<tr>
<td>Automatically lock connection</td>
<td>When disabled, a prompt to commit or rollback the transaction is displayed when you close the ISQL editor window. Enabling this option disables the prompts and locks the connection automatically.</td>
<td></td>
</tr>
<tr>
<td>Prompt to lock database connection</td>
<td>Locks the database connection on execution.</td>
<td>Selected</td>
</tr>
<tr>
<td>Tabs</td>
<td>Sets the appearance of the ISQL Window tabs to either the top or bottom of the ISQL Window.</td>
<td>Top</td>
</tr>
<tr>
<td>JDBC options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Association</td>
<td>Specifies whether the application should open an unknown file type automatically into an ISQL Window or prompt you with a message that DBArtisan does not recognize the file type.</td>
<td></td>
</tr>
<tr>
<td>Enable DBMS Output</td>
<td>ORACLE ONLY: Lets you specify buffer size. 0 is the default.</td>
<td>Selected</td>
</tr>
<tr>
<td>Auto-Commit changes</td>
<td>ORACLE ONLY: Applies auto commit status changes to all open windows.</td>
<td>Not Selected</td>
</tr>
<tr>
<td>Set Isolation Level</td>
<td>DB/2 ONLY: Sets the default for the value of the Isolation Level option on the Query Options dialog. For details, see Query Options.</td>
<td>UNCOMMITTED READ</td>
</tr>
<tr>
<td>Enable Set Query Options</td>
<td>SQL SERVER tab: Sets the default for the value of the Send Set Options setting on the Query Options dialog. For details, see Query Options.</td>
<td>Selected</td>
</tr>
</tbody>
</table>

JDBC options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the JDBC tab.
DBArtisan 8.1.0 requires a JDBC connection. You have the ability, with this panel of the Options Editor, to set or change options that apply to platform-specific JDBC drivers and to the Java virtual machine (JVM) that is running on the client. The table below describes the options and functionality of the JDBC tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Heap Size</td>
<td>Set the size, in MB, for the repository where live and dead objects come into free memory. If memory runs out, JVM executions stop so the garbage can be collected and expunged. Each platform responds differently, so trial and error can help you maximize performance.</td>
<td>64 MB</td>
</tr>
<tr>
<td>Maximum Heap Size</td>
<td>Set the upper limit for your heap size.</td>
<td>64 MB</td>
</tr>
<tr>
<td>Additional Options</td>
<td>Add options here ONLY in consultation with Embarcadero Technical Support</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBMS Type</td>
<td>Select the DBMS appropriate to the driver you want to review/adjust from the drop-down list.</td>
<td></td>
</tr>
<tr>
<td>Search</td>
<td>Click ... to open the JDBC Driver Editor where you can see what drivers for the given platform are available to you. You can assign the JDBC Driver you selected as a default for all platform-specific datasources.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename:</td>
<td>Displays the name of the license log file. Browse to identify where it is located.</td>
<td>guardian.log</td>
</tr>
<tr>
<td>Max Log Size (bytes)</td>
<td>Indicates the maximum size in bytes to which the file can grow.</td>
<td>1048576 bytes</td>
</tr>
<tr>
<td>Log Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Errors</td>
<td>Lets you log license error data.</td>
<td>Selected</td>
</tr>
<tr>
<td>Info</td>
<td>Lets you log license information data</td>
<td>Not selected</td>
</tr>
<tr>
<td>Debugging</td>
<td>Lets you log license debugging data.</td>
<td>Not selected</td>
</tr>
</tbody>
</table>

**Licensing options**

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Licensing tab.

The Licensing Tab of the Options Editor lets you turn on and off different logging options for the licensing data.

**NOTE:** License log file options should only be modified when you are working with Embarcadero Support to identify licensing errors.

The table below describes the options and functionality on the Licensing Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename:</td>
<td>Displays the name of the license log file. Browse to identify where it is located.</td>
<td>guardian.log</td>
</tr>
<tr>
<td>Max Log Size (bytes)</td>
<td>Indicates the maximum size in bytes to which the file can grow.</td>
<td>1048576 bytes</td>
</tr>
<tr>
<td>Log Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Errors</td>
<td>Lets you log license error data.</td>
<td>Selected</td>
</tr>
<tr>
<td>Info</td>
<td>Lets you log license information data</td>
<td>Not selected</td>
</tr>
<tr>
<td>Debugging</td>
<td>Lets you log license debugging data.</td>
<td>Not selected</td>
</tr>
</tbody>
</table>
Logging options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Logging tab. The Logging tab lets you set defaults that specify the behavior and placement of SQL Logging and Output Logging.

The table below describes the options and functionality on the Logging Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log all SQL Statements to a File</td>
<td>Indicates that the application should log all of the SQL that it executes to a file. SQL logging provides an audit trail for DBArtisan. You can examine this log to determine the SQL executed to complete a task.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Logfile Path</td>
<td>If you choose to log the SQL generated by the application, specify the drive, directory, and file name.</td>
<td>None</td>
</tr>
<tr>
<td>Max File Size</td>
<td>Specifies the maximum size for the logfile. When the logfile reaches this threshold, it automatically starts deleting lines in the logfile (starting with the oldest statements) to remain within the specified size limit.</td>
<td>1024 KB</td>
</tr>
<tr>
<td>Truncate</td>
<td>Empties the entire contents of the logfile.</td>
<td>Not available</td>
</tr>
<tr>
<td>Log all Output Messages to a File</td>
<td>Indicates that the application should log all server messages sent to the Output Window. This type of logging lets you monitor only messages issued by the server versus all SQL logged by the application. You can examine this log to determine all messages the server issued.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Logfile Path</td>
<td>If you choose to log the server messages generated in the Output Window, specify the drive, directory, and file name.</td>
<td>None</td>
</tr>
<tr>
<td>Max File Size</td>
<td>Specifies the maximum size for the output logfile. When the output logfile reaches this threshold, it automatically starts deleting lines in the file (starting with the oldest statements) to remain within the specified size limit.</td>
<td>1024 KB</td>
</tr>
<tr>
<td>Truncate</td>
<td>Empties the entire contents of the output logfile.</td>
<td>Not available</td>
</tr>
</tbody>
</table>

Migration options

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Migration tab. The Migration tab sets the default values effective when you open the Migration Wizard, a utility that lets you migrate schema and data. For information on this utility, see Using the schema migration wizard.

The Migration Wizard option defaults you can set on the Migration tab are:

- **Script Dependencies**
- **Group Dependencies**
- **Script... options**
- **Data Movement**
- **Batch Size**
- **Overwrite Existing**
- **Generate Script**
MySQL Utilities Options
After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the MySQL Utilities tab. DBArtisan integrates with MySQL utilities. For DBArtisan to access these utilities, you need to specify their location in the MySQL Utilities Tab of the Options Editor. You can use the MySQL Dump and Import Utilities. The table below describes the options and functionality on the MySQL Utilities Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysqladmin</td>
<td>Specifies a path for the MySQL dump utility, mysqladmin.exe. By default, MySQL installs this utility in the C:\mysql\bin directory.</td>
</tr>
<tr>
<td>mysqlimport</td>
<td>Specifies a path for the MySQL import utility, mysqlimport.exe. By default, MySQL installs this utility in the C:\mysql\bin directory.</td>
</tr>
</tbody>
</table>

Oracle Utilities options
After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Oracle Utilities tab.

DBArtisan integrates with multiple Oracle utilities. For DBArtisan to access these utilities, you need to specify their location in the Oracle Utilities Tab of the Options Editor. You can use the Oracle Export, Import Utilities, and SQL * Loader. The table below describes the options and functionality on the Oracle Utilities Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export</td>
<td>Specifies a path for the Oracle Export utility. By default, Oracle installs to C:\Orant\Bin directory.</td>
</tr>
<tr>
<td>Import</td>
<td>Specifies a path for the Oracle Import utility. By default, Oracle installs to C:\Orant\Bin directory.</td>
</tr>
<tr>
<td>SQL*Loader</td>
<td>Specifies a path for the SQL * Loader utility. By default, Oracle installs to C:\Orant\Bin directory.</td>
</tr>
<tr>
<td>Default Bind Size</td>
<td>Specifies the bind size. Default is set to 70KB.</td>
</tr>
</tbody>
</table>

Perf Center options
After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Performance Center tab. DBArtisan lets you customize Performance Center’s integration with DBArtisan. The table below describes the options and functionality on the Perf Center Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Name or IP Address</td>
<td>Specify where the Performance Center Server is installed.</td>
<td>localhost</td>
</tr>
<tr>
<td>Port Number</td>
<td>Specify the port for the Apache server or IIS web server.</td>
<td>80</td>
</tr>
<tr>
<td>Test</td>
<td>Verifies the settings.</td>
<td></td>
</tr>
<tr>
<td>Connect to the server using</td>
<td>Specify if you want DBArtisan to open the Web Client or the Performance Center .exe file within DBArtisan.</td>
<td>Web Client</td>
</tr>
</tbody>
</table>

Query Builder options
After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Query Builder tab. Option parameters set on the Options Editor elicit a prompt if there are different options set on an open individual session. Global options override properties set within individual Query Builder sessions.
The table below describes the options and functionality on the Query Builder Tab:

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Generation</td>
<td>Generate Use Database statement</td>
<td>Adds a line of SQL code indicating which database or instance is used in the statement.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Generate owner names</td>
<td>Adds a line of SQL code showing the table owner name as part of the query.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Include Row Count limits</td>
<td>Includes the output row limit set in the Execution settings.</td>
<td>Not selected</td>
</tr>
<tr>
<td></td>
<td>Generate SQL/92 if supported by DBMS</td>
<td>SQL/92 is a standard for relational database management systems.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Execution</td>
<td>To set the number of rows in your result set, type the number in the dialog box. This lessens congestion of server processes when queries execute by setting row count limits.</td>
<td>1000 rows</td>
<td></td>
</tr>
<tr>
<td>General</td>
<td>Show Column Data types in Query Diagram</td>
<td>Reveals the data type in each column for tables in the SQL Diagram pane.</td>
<td>Not selected</td>
</tr>
<tr>
<td></td>
<td>Confirm on Item delete</td>
<td>Opens a Confirm Delete dialog box when an item is deleted.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Auto Populate Views</td>
<td>Checks syntax every time an execute statement, refresh or copy statement begins.</td>
<td>Not selected</td>
</tr>
<tr>
<td></td>
<td>Auto Format</td>
<td>Automatically sets style and spacing of display.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Auto Join</td>
<td>Automatically detects names and data types, and create joins for multiple tables.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Require Indexes</td>
<td>Joins only indexed columns. Requires an indexed column for joins.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Require same data type</td>
<td>Automatically joins columns with the same data type.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Syntax Checker</td>
<td>Automatically checks SELECT and CREATE VIEW statements for errors.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Warn on non index join</td>
<td>Returns a warning when it detects a join against a non-indexed column, or a column not participating in a primary key</td>
<td>Not selected</td>
</tr>
<tr>
<td>Display</td>
<td>Lets you sets the style, size, and color of Column Font and Title Font. Also lets you set the background Table Color for the SQL Diagram Pane.</td>
<td>Available</td>
<td></td>
</tr>
</tbody>
</table>

**Results (ISQL) options**

After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the Results tab. The table below describes the options and functionality on the Results Tab:

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results Option pane</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Result Window</td>
<td>Single Window</td>
<td>Displays all results in one tabbed result window. Multiple result sets together in the window. Single Window and Multiple Windows options are mutually exclusive.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Multiple Windows</td>
<td>Displays multiple result sets one result set per window. Single Window and Multiple Windows options are mutually exclusive.</td>
<td>Not selected</td>
</tr>
<tr>
<td>Interface Element</td>
<td>Option</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>Attached to Editor</td>
<td>Sets results as tabbed windows attached to the ISQL window. Used in conjunction with Single Window option or Multiple Window option. Attached and Unattached options are mutually exclusive.</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td>Unattached</td>
<td>Sets results appear separate from the ISQL Window. Used in conjunction with Single Window option or Multiple Windows option. Attached to Editor and Unattached options are mutually exclusive.</td>
<td>Not Selected</td>
<td></td>
</tr>
<tr>
<td>Reuse Window</td>
<td>Sets new result sets to overwrite any current result sets in an open Result Window. Only valid for Single and Attached to Editor combination.</td>
<td>Selected</td>
<td></td>
</tr>
<tr>
<td>Results File</td>
<td>Mail File Type</td>
<td>Selects the file type to use when sending result sets via a MAPI-compliant mail package. Valid formats include the proprietary Results type, Tab delimited, Comma separated, and HTML.</td>
<td>Results</td>
</tr>
<tr>
<td>Schedule File Type</td>
<td>Selects the schedule file type. Valid formats include Tab delimited, Comma separated, and HTML.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Include column titles when saving</td>
<td>Includes column titles when saving a result set. If this option is turned off, column titles DBArtisan does not save result sets.</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>Result Set Options</td>
<td>Default Rowcount</td>
<td>Lets you limit the number of rows returned to the result window of the ISQL window</td>
<td>0</td>
</tr>
<tr>
<td>Sybase and SQL Server: Text Size (bytes)</td>
<td>Lets you specify the text size.</td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td>Oracle:LONG Size (bytes)</td>
<td>Lets you specify the LONG size.</td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td>LOB Preview: Text Size (bytes)</td>
<td>Specifies the length of the preview of LOB column data.</td>
<td>4096</td>
<td></td>
</tr>
<tr>
<td>Results Format pane</td>
<td>Column Formatting</td>
<td>Sets column widths automatically to accommodate the longest piece of data in a column. Large queries depend on the longest row formatting, so activating this option can affect performance.</td>
<td>Selected</td>
</tr>
<tr>
<td>Use pre-defined column</td>
<td>Lets you select column type and character length</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>Enable Date/Time Format</td>
<td>Lets you select the date/time format.</td>
<td>Not selected.</td>
<td></td>
</tr>
<tr>
<td>Format</td>
<td>Standard Grid</td>
<td>Displays all result sets in a standard grid format. DBArtisan displays result sets in grid format in IISQL Editors that are opened after you have selected this option. DBArtisan does not display IISQL Editors that are already open.</td>
<td>Selected</td>
</tr>
<tr>
<td>HTML</td>
<td>Displays all result sets as HTML tables. DBArtisan displays result sets in HTML format in ISQL Editors that are opened after you have selected this option. DBArtisan does not display ISQL Editors that are already open.</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>ASCII Text</td>
<td>Displays all result sets as ASCII Text. DBArtisan displays result sets in ASCII Text format in ISQL Editors that are opened after you have selected this option. DBArtisan does not display ISQL Editors that are already open.</td>
<td>Not selected</td>
<td></td>
</tr>
<tr>
<td>Grid Font Button</td>
<td>Click to open the Font dialog box. Select the font, style, and size for the result sets grid.</td>
<td>Available</td>
<td></td>
</tr>
</tbody>
</table>
SMTP mail options

After opening the Options editor (see *Specifying DBArtisan application and feature options*), you can make changes to the SMTP Mail tab. The SMTP Mail Tab of the Options Editor lets you specify outgoing notification e-mail message options.

The table below describes the options and functionality on the SMTP Mail tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name that appear as the e-mail sender.</td>
<td>Name you specified during installation.</td>
</tr>
<tr>
<td>E-mail Address</td>
<td>Address to send e-mails from DBArtisan.</td>
<td>E-mail address you specified during installation.</td>
</tr>
<tr>
<td>Authentication</td>
<td>Lets you specify authentication options.</td>
<td>None</td>
</tr>
<tr>
<td>User Name</td>
<td>User name for authentication.</td>
<td>Not available</td>
</tr>
<tr>
<td>Password</td>
<td>Password for authentication.</td>
<td>Not available</td>
</tr>
<tr>
<td>Host Name</td>
<td>SMTP server for outgoing messages. For Microsoft Outlook, select Tools, and then Accounts. On the Mail Tab, select target account, and then click Properties. On Servers Tab, copy the Outgoing Mail(SMPT) and paste.</td>
<td>Host Name you specified during installation.</td>
</tr>
<tr>
<td>Port Number</td>
<td>Port number you connect to on your outgoing SMTP server.</td>
<td>25</td>
</tr>
<tr>
<td>Test</td>
<td>Opens an SMTP Configuration Test e-mail addressed to your e-mail address. Click Send Mail to send the e-mail.</td>
<td>Available</td>
</tr>
<tr>
<td>Bind to</td>
<td>Your IP address the message is bound to.</td>
<td>ANY IP ADDRESS</td>
</tr>
<tr>
<td>Encoding</td>
<td>E-mail encoding.</td>
<td>Western Europe (ISO)</td>
</tr>
<tr>
<td>Send messages Mime encoded</td>
<td>Messages encoded using Multipurpose Internet Mail Extensions (MIME) support enriched content and attachments.</td>
<td>Selected</td>
</tr>
<tr>
<td>Send all messages as HTML</td>
<td>Messages include text formatting.</td>
<td>Selected</td>
</tr>
<tr>
<td>Auto Connect to the Internet</td>
<td>DBArtisan connects to internet at launch.</td>
<td>Selected</td>
</tr>
</tbody>
</table>
Space Analyst options
After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the **Space Analyst** tab. The table below describes the options and functionality on the **Space Analyst** Tab:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>Exclude objects owned by SYS in Reorganization Wizard.</td>
<td>The Space Analyst Reorganization Wizard excludes SYS objects.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Auto populate Table Reorganization Details Grid</td>
<td>Space Analyst auto-populates table reorganization details.</td>
<td>Not Selected</td>
</tr>
<tr>
<td></td>
<td>Auto populate Index Reorganization Details Grid</td>
<td>Space Analyst auto-populates index reorganization details.</td>
<td>Not Selected</td>
</tr>
<tr>
<td></td>
<td>Populate Index Reorganization Details Grid with Deleted Leaf Statistics</td>
<td>Space Analyst gathers deleted leaf row statistics for an index and displays details.</td>
<td>Not Selected</td>
</tr>
<tr>
<td>SQL Server, Sybase, and IBM DB/2 for Windows, Unix, and Linux</td>
<td>Offers the option to update the Table Details Grid and Index Details Grid.</td>
<td></td>
<td>Not Selected</td>
</tr>
</tbody>
</table>

Warnings options
After opening the Options editor (see Specifying DBArtisan application and feature options), you can make changes to the **Warnings** tab. DBArtisan lets you issue warning messages to users whenever improper operations are attempted in a database. Warning messages differ by platform.

The tables below describes the options of the **Warnings** Tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an index on the same tablespace as the associated table</td>
<td>Issues a warning message whenever a user does not create an index on a different tablespace than the table. This makes it possible for the server to read the index and table data in parallel.</td>
<td>Not Selected</td>
<td>DB2</td>
</tr>
<tr>
<td>Create an index on the same tablespace as the associated table</td>
<td>Issues a warning message whenever a user does not create an index on a different tablespace than the table. This makes it possible for the server to read the index and table data in parallel.</td>
<td>Not Selected</td>
<td>Oracle</td>
</tr>
<tr>
<td>Create an object in the SYSTEM tablespace</td>
<td>Issues a warning message whenever a user tries to create or place an object on the SYSTEM tablespace.</td>
<td>Not Selected</td>
<td>Oracle</td>
</tr>
<tr>
<td>Create a user with default or temp tablespace as the SYSTEM tablespace</td>
<td>Issues a warning message when a user is created with a default or temp tablespace on the SYSTEM tablespace.</td>
<td>Not Selected</td>
<td>Microsoft SQL Server and Sybase ASE</td>
</tr>
<tr>
<td>Create an object in the master database</td>
<td>Issues a warning message when an object is created in the master database.</td>
<td>Selected</td>
<td>Microsoft SQL Server and Sybase ASE</td>
</tr>
<tr>
<td>Create a table or index on the default segment</td>
<td>Issues a warning message when a table or index is created on the default segment.</td>
<td>Not Selected</td>
<td>Microsoft SQL Server and Sybase ASE</td>
</tr>
<tr>
<td>Create a non-clustered index on same segment as the associated table</td>
<td>Issues a warning message when a non-clustered index is created on the same segment as the associated table.</td>
<td>Not Selected</td>
<td>Microsoft SQL Server and Sybase ASE</td>
</tr>
</tbody>
</table>
Using

Using shows how to use each component of the application. Using breaks each process and procedure into simple step-by-step instructions.

Using is divided into sections. The table below describes each section:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasource Management</td>
<td>This section describes the purpose of datasources, the process of establishing datasource connections, and managing your datasources and datasource groups.</td>
</tr>
<tr>
<td>Supported Objects</td>
<td>This section describes the different platform objects the application supports.</td>
</tr>
<tr>
<td>Object Editors</td>
<td>This section describes Object editors. Editors let you modify existing objects.</td>
</tr>
<tr>
<td>Functionality</td>
<td>This section describes the different platform object functionalities.</td>
</tr>
<tr>
<td>SQL Scripting</td>
<td>This section describes the SQL scripting environment, that lets you write, debug, test and deploy solid SQL code for your database applications.</td>
</tr>
<tr>
<td>Permissions Management</td>
<td>This section describes the Permissions Management features.</td>
</tr>
</tbody>
</table>
Datasource Management

After installing DBArtisan, you must set up datasources to establish reusable connections to your database servers. A datasource is a database connection profile that is similar to the connection information that you have stored in your SQL.INI or WIN.INI files. DBArtisan stores information about the datasource specification in the system registry and provides a visual interface for maintaining it.

This section describes the function of datasources, the process of establishing datasource connections and managing your datasources and datasource groups.

Registering or Editing a Datasource

A datasource is the connection profile that lets you have access to a database, and, therefore, to your data. So, when you register a datasource, you're telling DBArtisan how to get at the database(s) you want to operate on. In fact, all database activities are performed through datasources, so you need a datasource profile for each database instance (or database server) in your enterprise. Each connection profile (i.e., datasource) includes a unique name, the connection string, the default user ID, and optionally, a password.

DBArtisan requires a JDBC connection. If you did not already have a JDBC driver, one was included as a part of the installation process (to see what's available, go to C:\ProgramFiles\Embarcadero\DBA851\Java\drivers). The DBArtisan installation provides a JDBC Type-4 thin driver for each of the platforms except IBM DB2 (see note below) if one was not previously installed.

**NOTE:** JDBC Type-4 Drivers require TCP/IP. If you are upgrading from a prior version of DBArtisan and want to connect to a datasource that is not using TCP/IP, you will have to edit your current datasource registration information.

**NOTE:** IBM DB2 does not use a Type-4 “thin” JDBC driver because it won’t connect to DB2 version 7 or the OS/390. Instead, IBM DB2 packages a Type-2 driver by default.

Datasource Registration

Each database instance must be registered. Whether you are registering a new datasource or editing preexisting connection information, you use the Datasource Registration dialog box.

**To register a new datasource:**

On the toolbar, click **Datasource**, and then click **Register Datasource** from the drop-down list.

**OR**

On the **Explorer**, right-click **Managed Datasources**, and then click **Register Datasource**.

**To edit registration information:**

On the toolbar, click **Datasource**, and then click **Edit Registration** from the drop-down menu.

**OR**

On the **Explorer**, right-click on the appropriate datasource, and select **Edit Datasource Registration** from the drop-down list.

The table below describe the fields you will encounter when registering a datasource or editing the connectivity information:

<table>
<thead>
<tr>
<th>Panel/Tab</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Type</td>
<td>Select the radio box corresponding to the DBMS type for the datasource</td>
<td></td>
</tr>
</tbody>
</table>
**DATASOURCE MANAGEMENT**

<table>
<thead>
<tr>
<th>Panel/Tab</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection Information</strong></td>
<td></td>
<td>For detailed descriptions of the options on this tab, see <a href="#">Datasource connection information on page 88</a>.</td>
</tr>
<tr>
<td>Advanced</td>
<td>JDBC Driver</td>
<td>You should see the JDBC Driver that's appropriate to the platform. If you select ..., the JDBC Driver Editor opens letting you browse to a new or possibly undetected JDBC driver. For details, see <a href="#">JDBC Driver Editor</a>.</td>
</tr>
<tr>
<td></td>
<td>Connection URL</td>
<td>The connection URL cannot be edited—it's for reference only.</td>
</tr>
<tr>
<td></td>
<td>URL Check</td>
<td>Indicates whether the connection URL is valid or not.</td>
</tr>
<tr>
<td><strong>Security Parameters</strong></td>
<td>User ID</td>
<td>The User ID that DBArtisan will use to connect to the datasource.</td>
</tr>
<tr>
<td></td>
<td>Password</td>
<td>The password associated with the User ID.</td>
</tr>
<tr>
<td></td>
<td>Connect As: (Oracle)</td>
<td>When relevant, choose the appropriate user/administrator level.</td>
</tr>
<tr>
<td></td>
<td>Domain:</td>
<td>For MS SQL only: Identify the domain if the user has restricted access.</td>
</tr>
<tr>
<td></td>
<td>Auto connect?</td>
<td>Spares the user from reentering the password every time you connect.</td>
</tr>
<tr>
<td></td>
<td>Connect using Windows</td>
<td>Login to the server is verified using Windows authentication</td>
</tr>
<tr>
<td></td>
<td>Authentication (IBM DB/2 for</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Windows, Unix, and Linux, MySQL, Oracle, SQL Server)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Connect using Kerberos</td>
<td>If this option is selected, login to the server is verified using Kerberos authentication.</td>
</tr>
<tr>
<td></td>
<td>Authentication (Sybase)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test Connection</td>
<td>When you test the connection, you're testing both the native and JDBC connectivity. If the connection fails, read the error message and backtrack as necessary.</td>
</tr>
<tr>
<td><strong>Database Group</strong></td>
<td></td>
<td>Select the datasource folder under which you want this datasource to appear.</td>
</tr>
</tbody>
</table>

If you have completed the datasource registration and still encounter problems, please refer to the [Unable to Connect?](#) discussion.

**NOTE:** After you successfully connect to a datasource, you can see the name of the host and the connection string information displayed at the bottom of the DBArtisan window. For Oracle and Sybase ASE, should a question arise, you can see if perhaps you set up the datasource NOT to use the alias file. You can also use these cues to remind yourself that if you changed the alias file, that change will not be reflected here unless you very specifically made a change to the defaults in the Options Editor-JDBC tab.
Datasource connection information

The following table lists the Connection Information tab options, DBMS-by-DBMS, available when registering or editing a datasource.

<table>
<thead>
<tr>
<th>DBMS</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM DB/2 for Windows, Unix, and Linux</td>
<td>Server</td>
<td>The name of the server.</td>
</tr>
<tr>
<td></td>
<td>Schema ID</td>
<td>This is an optional field. If you want to include the current schema as part of the datasource connection properties, you can set the implicit schema for unqualified object references to a schema different from the user’s login id.</td>
</tr>
<tr>
<td></td>
<td>Function Path</td>
<td>This is an optional field. If you choose, you can enter the appropriate function path name so the search scope of functions will not be limited to the IBM DB2 library.</td>
</tr>
<tr>
<td></td>
<td>Datasource Name:</td>
<td>This field is automatically populated with the server name, but you can rename it to whatever you want.</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Use Network Library Configuration</td>
<td>When you select this, you can connect to a Net-Library that is listening on the server that is configured to either the Named Pipes or TCP/IP protocols.</td>
</tr>
<tr>
<td></td>
<td>Alias</td>
<td>This box is enabled only when you opt to use the Network Library Configuration</td>
</tr>
<tr>
<td></td>
<td>Protocol: TCP/IP or Named Pipes</td>
<td>You need to select one or the other to register a SQL Server database.</td>
</tr>
<tr>
<td></td>
<td>Host</td>
<td>Enabled only when you have not selected an Alias from the Network Library.</td>
</tr>
<tr>
<td></td>
<td>Port (optional)/Pipe Name</td>
<td>Depending on your section, you can optionally indicate the port for TCP/IP or the pipe name depending on your means of connection.</td>
</tr>
<tr>
<td></td>
<td>Default DB (optional)</td>
<td>Optionally give the default name for the database.</td>
</tr>
<tr>
<td></td>
<td>Datasource Name</td>
<td>The field is automatically populated with the host name, but you can rename to whatever you want.</td>
</tr>
<tr>
<td>MySQL</td>
<td>Server</td>
<td>Enter the name of the host, for example doctest01.</td>
</tr>
<tr>
<td></td>
<td>Port (optional)</td>
<td>The default port is 3306. You do not need to enter this information.</td>
</tr>
<tr>
<td></td>
<td>Default Database</td>
<td>You must enter the name of the default database.</td>
</tr>
<tr>
<td></td>
<td>Datasource Name</td>
<td>This field is automatically populated with the server name, but you can rename it to whatever you want.</td>
</tr>
<tr>
<td>Oracle</td>
<td>Use TNS Names Alias/oracle Alias</td>
<td>If this option is checked, look for the Alias you want to register from the drop-down list. The remaining fields will automatically populate. If it is unchecked, you must manually enter the requisite information.</td>
</tr>
<tr>
<td></td>
<td>Host</td>
<td>Manually enter the name of the host machine.</td>
</tr>
<tr>
<td></td>
<td>Port</td>
<td>The default is 1521, but you can change it to wherever the Oracle listener is set up.</td>
</tr>
<tr>
<td></td>
<td>SID/Service Name / Type: SERVICE_NAME or SID</td>
<td>Enter the SID or Service Name to correspond with the option you select.</td>
</tr>
<tr>
<td></td>
<td>Instance Name</td>
<td>The specific name used to identify the Oracle instance (the SGA and the Oracle processes).</td>
</tr>
<tr>
<td></td>
<td>Datasource Name</td>
<td>The field is automatically populated with the host name, but you can change it.</td>
</tr>
</tbody>
</table>
JDBC Driver Editor

The following table lists the controls available on the JDBC Driver Editor dialog. You use this dialog when defining or editing a datasource.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver Type</td>
<td>The database platform.</td>
</tr>
<tr>
<td>Driver List</td>
<td>You see what platform-specific drivers are available to you and which driver (highlighted) is currently associated with the server you are connecting to.</td>
</tr>
<tr>
<td>Driver Name</td>
<td></td>
</tr>
<tr>
<td>Driver Class</td>
<td>The name of the class the driver uses for creating connections to the server.</td>
</tr>
<tr>
<td>Version</td>
<td>The driver’s version.</td>
</tr>
<tr>
<td>Driver supports the use of native aliases</td>
<td>DBArtisan cannot detect whether the driver you are using can support native aliases. You need to know by checking the driver’s documentation.</td>
</tr>
<tr>
<td>Required jars</td>
<td>DBArtisan detects the required jars and lists the path to them in the little window. You can select or add new jar files as required by your driver.</td>
</tr>
</tbody>
</table>

For context information, see Registering or Editing a Datasource.
Unable to Connect?
If you are having trouble connecting to a datasource, as a first step please check the connection information to make sure it has been captured correctly.

1. Verify that the user/password combination is correct.

2. Check that the correct JDBC Driver is being called. You can see all platform-specific drivers when you browse from the Advanced Settings panel of the registration wizard. You can see ALL drivers by going to C:\ProgramFiles\Embarcadero\DBA851\Java\drivers. A list of supported drivers follows.

**Drivers supported by DBArtisan:**

**Oracle:**
- JDBC OCI client-side driver--Type 2, Versions 8, 9, and 10. This is shipped by Oracle.
- Oracle 9.0.2 Thin Driver--Type 4, shipped with DBArtisan.

**Sybase:**
- jConnect v5.5--Type 4, shipped with DBArtisan.

**Microsoft SQL Server:**
- jTDS Driver--Type 4, shipped with DBArtisan.

**NOTE:** This is the only driver for MS SQL Server that we support.

**IBM DB2**--Type 2 driver comes from IBM.

**MySQL:**
- MySQL driver--shipped with DBArtisan.

3. Verify that the connection URL is correct. The URL always begins with JDBC and includes a permutation of the driver.

Datasources
A datasource is a database connection profile. A datasource includes:

- Name
- Connection String
- Default User ID
- Optional Password Specification

All database operations are performed through datasources. You can create a datasource profile for each database instance (or database server) in your enterprise, and you can create multiple datasource profiles with different logins for a single database. The table below describes the data items stored for each datasource:

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique, user-defined name for the datasource.</td>
</tr>
<tr>
<td>Connection String</td>
<td>For Oracle: the SQL*Net connect string, for Sybase ASE: the database server name.</td>
</tr>
<tr>
<td>Default User</td>
<td>Default user ID to use when logging in to the datasource. Can be null.</td>
</tr>
<tr>
<td>Default Password</td>
<td>Default password to use when logging in to the datasource. This is encrypted. Can be null.</td>
</tr>
</tbody>
</table>
Available Functionality for registered datasources

DBArtisan offers the following functionality for the Datasource Explorer:

- Change Group
- Connect
- Disconnect
- Drop
- Edit Datasource Registration
- Extract
- Monitor
- New UDB Datasource
- Register Datasource
- Report
- Unregister Datasource

Related Topics

- Selecting Datasources
- Viewing Datasource Properties

Datasource Catalog

The Datasource Catalog is a collection of defined datasources. It is stored in the system registry of your computer. All Embarcadero database management products share the datasource catalog, which means that when you set up your datasource catalog using one product such as DBArtisan, the same list of datasources is available in other Embarcadero Technologies products. Any changes you make to the datasource catalog are reflected in all Embarcadero database management products.

Sharing a Datasource Catalog

You can configure Embarcadero database applications to use a datasource catalog stored in the system registry of your machine (local).

You can determine the location of your datasource catalog by inspecting the application status bar. The 'Catalog' indicator displays 'Local' if the catalog is stored in your local system registry.

When you configure the location of the datasource catalog, you are affecting all Embarcadero datasource applications as they all read and maintain the same datasource catalog.

**NOTE:** You need the proper permissions to read/write registry entries on another computer. See your network administrator if you have problems.

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-Connect Flag</td>
<td>If Yes, then automatically login using default user and password. If No, open Login dialog box.</td>
</tr>
<tr>
<td>Default Database</td>
<td>SYBASE ASE ONLY: Database to automatically use after logging in.</td>
</tr>
</tbody>
</table>
Automatically Discovering Datasources

The first time you run DBArtisan a dialog box displays, giving you the option to Auto-Discover all configured datasources. If you click Yes, the DBArtisan Auto-Discover feature searches the DBMS configuration files on your computer and automatically discovers all the datasources that you are licensed for. For example, if you have a cross-platform license, Discover Datasources finds all unregistered datasources. If you have an Oracle only license, Discover Datasources finds all unregistered Oracle datasources.

**NOTE:** Microsoft SQL Server datasources are registered through a Windows system call to your network. Provide login information (user name and password) the first time you connect to a datasource.

**NOTE:** IBM DB2 for Linux, Unix, and Windows databases use ODBC/CLI or DB2 (attach) to connect. Therefore, you need the proper ODBC/CLI Connection established in order for the auto-discover feature to find your IBM DB2 for Linux, Unix, and Windows databases, including registering the DB2 datasource to ODBC as a system datasource. Although your datasources are auto-discovered, provide login information (user name and password) the first time you connect to a datasource.

In addition to Auto-Discovering your database servers, the application creates Datasource Groups based on RDBMS type. Each registered datasource is placed in its respective Datasource Group. For example, all Microsoft SQL Server datasources are added to the Microsoft SQL Server Group. Each registered datasource is placed in its respective Datasource Group.

For more information on how to configure your datasources, see Working with Datasources.

Changing Datasource Groups

DBArtisan lets you change datasource groups by:

- Dragging the datasource between groups.
- Invoking the Change Group dialog box.

**Dragging and Dropping Between Groups**

1. On the Datasource Explorer, left-click the datasource group you want to move, drag it over the folder for the new group, and release the pointer.

**Using the Change Group Dialog Box**

Disconnect your datasource before changing groups.

1. On the Datasource Explorer, right-click the datasource you want to move, and then click Change Group. DBArtisan opens the Change Group dialog box.
2. In the Select Group tree, click new group.
3. Click OK. DBArtisan changes groups.

For more information, see Datasource Groups.
Connect
DBArtisan lets you set datasources to automatically connect each time you open the application. The first time you start the application, DBArtisan prompts you to register your datasources. During this process, you can select the Auto Connect check box, which automatically connects all registered datasource each subsequent time you open the application.

If you did not check the Auto Connect box, or if you clicked No when prompted to connect to a database after registering, you must connect manually, each time you want to access that datasource, using the Datasource Login dialog box. If you later want to automatically connect your datasources, you can edit the datasource to make that change.

TIP: To configure your datasource to login automatically, refer to Registering or Editing a Datasource.

The table below describes the options and functionality on the Datasource Login dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login ID</td>
<td>Lets you type the Login ID.</td>
</tr>
<tr>
<td>Password</td>
<td>Lets you type the password.</td>
</tr>
<tr>
<td>Auto Connect</td>
<td>Select to automatically connect to the datasource in the future, select the check box.</td>
</tr>
</tbody>
</table>

Completing the Datasource Login Dialog Box
To complete the Datasource Login dialog box, do the following:

1. On the Datasource menu, click Connect.
   DBArtisan opens the Datasource Login dialog box.
2. In Login ID type the Login ID.
3. In Password type the password.
4. Select Auto Connect to automatically connect to the datasource in the future.
5. Click Connect.
   DBArtisan opens the Datasource Landing Page in the right pane of the application.

For more information, see Working with Datasources.

Datasource Landing Page
The Datasource Landing Page contains informative metadata that gives you a quick overview of your server/database. The Datasource Landing Page opens when you automatically connect to a datasource or complete the Datasource Login Dialog Box.

The Datasource Landing Page consists of two views. The top view contains of text metadata and the bottom contains a graphical display of the memory allocations.

For Oracle 9i, the Datasource Landing Page also indicates if the cache advice mechanism is turned on.

For IBM DB2 for Windows/Unix, the metadata requires an attachment to the instance. Use the database user id and password for the attachment.
Disconnect
When you disconnect from a server, the application immediately breaks the connection between any open ISQL Windows, the servers, and databases. Although your ISQL Windows are still visible, the connections are no longer valid. If you attempt to execute a script, DBArtisan attempts to reconnect to a registered datasource, if available.

Completing the Disconnect Dialog Box
To complete the Datasource Login dialog box, do the following:

1. On the Datasource menu, click Disconnect.
   - DBArtisan opens a dialog box asking if you want to commit all pending transactions for that connection or to rollback all before disconnecting. You cannot disconnect if there is an uncommitted transaction.
2. Click Yes.
   - DBArtisan confirms you want to disconnect and closes the dialog box.

Discover Datasource
DBArtisan discovers datasources residing on your system that are not currently registered datasources through a Windows system call to your network. The Discover Datasource dialog box includes a list, which includes the name of the server or instance and the type of DBMS of all unregistered datasources found on your network or local machine. Once discovered, you have the option to register datasources.

Completing the Discover Datasources Dialog Box
1. On the Datasource menu, click Discover Datasource.
   - DBArtisan opens the Discover Datasources dialog box.
2. Select the check box next to the datasource you want to register.
3. Click Select All to select all the datasources on the list.
4. Click Register.
   - DBArtisan registers the datasource or datasources selected.
5. Click OK.
   - DBArtisan closes the DBArtisan message.

For more information, see:
Datasources
Working with Datasources

Extract Statements
DBArtisan lets you extract the sp_configure statements for setting and displaying your server configuration parameters into an Interactive SQL window.

1. On the Datasource Explorer, select the Configuration node.
   - DBArtisan displays the server configuration parameters in the right pane of the Datasource Explorer.
2. In the right pane of the Datasource Explorer, click the target server configuration parameter(s).
On the **Datasource Explorer** menu, click **Extract**.

DBArtisan opens the DDL Editor.

For more information, see [Servers](#).

### Managing Datasources

The **Manage Datasources dialog box**:

- Lets you manage and datasources throughout your enterprise from a single vantage point.
- Displays in grid format all relevant registered datasource information.
- Unifies complicated and widespread information about your datasources.
- Lets you directly access other datasource features such as adding, modifying, deleting and discovering datasources.

### Completing the Manage Datasources Dialog Box

1. On the **Datasource** menu, click **Manage Datasources**.

DBArtisan opens the Manage Datasources dialog box.

The Manage Datasources grid format lets you access and view datasource specifications. The table below describes the options and functionality on the grid:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasource Name</td>
<td>Uses an explorer-like interface to display all registered datasources and their groups. You can navigate this column in the same manner as the datasource explorer, by clicking on nodes to expand or collapse your view of the datasources.</td>
</tr>
<tr>
<td>Connect String</td>
<td>Displays the full connection string for the datasource.</td>
</tr>
<tr>
<td>Default User Id</td>
<td>Displays the Default User ID for the datasource.</td>
</tr>
<tr>
<td>Auto-Connect?</td>
<td>Indicates whether the Auto Connect feature is turned on or off.</td>
</tr>
<tr>
<td>Host Computer</td>
<td>Displays the name of the Host Computer if one has been configured.</td>
</tr>
<tr>
<td>Default Schema</td>
<td>Displays view default schemas for your DB2 datasources.</td>
</tr>
</tbody>
</table>

For more information, see:

- [Datasources](#)
- [Using the Manage Datasources Dialog Box](#)

### Manage Datasources Dialog Box

The **Manage Datasource dialog box** lets you access most datasource management tasks. The Manage Datasource dialog box lets you:

- Register datasources
- Edit datasources
- Discover datasources
- Delete datasources
Editing a Datasource

1. On the Datasource menu, click Manage Datasources.
   DBArtisan opens the Manage Datasources dialog box.
2. Click a datasource from the Datasource Name column, and then click the Edit button.
   OR
   Double-click the datasource from the grid.
   DBArtisan opens the Edit Registration dialog box.

Discovering Datasources

1. On the Datasource menu, click Manage Datasources.
   DBArtisan opens the Manage Datasources dialog box.
2. Click a datasource from the Datasource Name column, and then click Discover.
   DBArtisan opens the Discover Datasource dialog box.

Deleting a Datasource

1. On the Datasource menu, click Manage Datasources.
   DBArtisan opens the Manage Datasources dialog box.
2. Click one or more datasource(s) from the Deleting a Datasource Name column.
3. Click Delete.
   DBArtisan displays a message.
4. Click Yes.
   DBArtisan confirms you want to unregister the datasource.

For more information, see Managing Datasources.

Managing Datasource Properties

The Datasource Properties dialog box displays the name, type, version, status and mode of the datasource. The Datasource Properties box also lets you view the middleware or connectivity software that is being used to establish a particular datasource connection. You can use this information to troubleshoot connectivity problems, determining vital information such as the server version, connectivity library used, and library version and date.

Completing the Datasource Properties Dialog Box

1. On the Datasource Explorer, click a datasource with an established connection.
2. On the Datasource menu, click Properties.
   DBArtisan opens the Datasource Properties dialog box.

For more information, see:

Datasources
Creating a Report
DBArtisan lets you generate a report containing detailed information about your server configuration parameters. The report derives its contents from the information displayed in the right hand side of the Datasource Explorer. To facilitate the online publication of these reports, DBArtisan generates the report in Hypertext Mark-Up Language (HTML). You can view the report directly in DBArtisan's built-in HTML browser.

NOTE: For more information on generating reports, see Schema Publication.

1 Click the Configuration node on the Datasource Explorer.
DBArtisan lists server configuration information in the right pane.

2 In the right pane, click one or more parameters.

NOTE: You can select a continuous range of objects by holding down the SHIFT key while selecting with the pointer. You can select a non-contiguous set of objects of holding down the CTRL key while selecting with the pointer.

3 On the Datasource Explorer menu, click Report.
DBArtisan opens the Report dialog box.

Report Dialog Box
The table below describes the options and functionality on the Report dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Home Page File Name</td>
<td>Lets you specify the name of the .htm report or click the browse button to open the Save As dialog box.</td>
</tr>
<tr>
<td>Report Title</td>
<td>Lets you specify a report title.</td>
</tr>
</tbody>
</table>

For more information, see Generating a Report on Server Configuration.

Selecting Datasources
The DBArtisan Select Datasource dialog box lets you select a datasource and connect to it.

Completing the Select Datasource Dialog Box
1 On the Datasource menu, click Select to open the Select Datasource dialog box.
2 Click the Datasource list box, and then click the target datasource.
3 Click Connect to connect to the datasource.
4 Click the Database list, and then click the target database.
5 Click OK to close the Select Datasource dialog box.

For more information, see:
Datasources
Working with Datasources1
Unregistering Datasource

DBArtisan lets you unregister datasources when you no longer need them.

**TIP:** Removing a datasource from DBArtisan does not delete the physical database. It simply removes the datasource definition, and connection information, from the DBArtisan catalog.

1. On the **Datasource** menu, click **Unregister**.
   DBArtisan opens a dialog box.
2. Click **Yes**.
   DBArtisan confirms you want to unregister the datasource.

**NOTE:** The datasource manager is shared across Embarcadero's database management products. When you remove a datasource in any of Embarcadero's database management tools the datasource is removed across all relevant products.

For more information, see:

- **Datasources**
- **Working with Datasources**

...displays the results in the right panel of the application.

Datasource Groups

DBArtisan lets you define datasource groups to organize the datasources in your enterprise. Datasource Groups behave as folders in the Windows Explorer, allowing you to group related datasources together. If you manage or monitor many Microsoft SQL Servers, datasource groups are a great mechanism for alleviating desktop clutter.

Anywhere that datasources are presented in a hierarchical tree format, datasource group folders expand to display one or more contained datasources. Upon installation of the first Embarcadero database management product, an initial datasource group is called Managed Datasources. You can rename this group.

**Available Functionality**

DBArtisan offers the following functionality for Datasource Groups:

- **Delete**
- **New Datasource Group**
- **Rename Datasource Group**
Delete

DBArtisan lets you remove database groups that you no longer need to access, or that have become obsolete.

1. On the **Datasource Explorer**, click the datasource.
2. On the **Datasource Explorer** tool bar, click **Delete**.
   - OR
   - Click the **Delete** button.
   - OR
   - Right-click the datasource, and then click **Remove Datasource Group**.

DBArtisan removes the Datasource Groups.

3. Click **Yes**.

DBArtisan confirms you want to remove the Datasource Group.

For more information, see [Datasource Groups](#).

New Datasource Group

DBArtisan lets you define datasource groups to organize the datasources in your enterprise. Datasource Groups behave as folders in the Windows Explorer, allowing you to group related datasources together. If you manage or monitor many Microsoft SQL Servers, datasource groups are a great mechanism for alleviating desktop clutter.

Anywhere that datasources are presented in a hierarchical tree format, datasource group folders expand to display one or more contained datasources. Upon installation of the first Embarcadero database management product, an initial datasource group is called Managed Datasources. You can rename this group.

**TIP:** SQL Server does not provide true clustering because separate instances of a database are actually running on each server in a cluster. To simulate a consolidated view, you can name and group clustered datasources within the Explorer by setting up a datasource group, named, for example, Cluster3. After you name the new group, simply drag-and-drop the datasources that make up the cluster into Cluster3.

**Completing the New Datasource Group Dialog Box**

1. On the **Datasource Explorer**, right-click the datasource group folder, and then click **New Datasource Group**.
   
   DBArtisan opens the New Datasource Group dialog box.

2. In the **Datasource Group Name** box, type the new name of the datasource group.

3. Click **OK**.
   
   DBArtisan closes the New Datasource Group dialog box.

Rename Datasource Group

DBArtisan lets you:

- Rename a datasource group.
- Change the members of a datasource group.
- Change the group to which a datasource belongs.
Completing the Rename Datasource Group
1. On the Datasource Explorer, right-click the datasource group folder, and then click Rename Datasource Group.
   DBArtisan opens the Rename Datasource Group.
2. In the Datasource Group Name box, type the new name of the datasource group.
3. Click OK.
   DBArtisan closes the Rename Datasource Group dialog box.

Servers
DBArtisan lets you manage your servers from the Datasources menu. From the Datasources menu you can:

- Shutdown a server.
- Configure a server.
- Extract server configuration parameters.
- Generate a report on server configuration.

Shut Down a Server
DBArtisan lets you shutdown a server using the Shutdown Server dialog box.

Completing the Shutdown Server Dialog Box
1. In the Explorer, select the target datasource.
2. On the Datasource menu, click Shutdown.
   DBArtisan opens the Shutdown Server dialog box.

The table below describes the options and functionality on the Shutdown Sever dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Shuts down the server in an orderly fashion. DBArtisan disables logins, and waits for currently executing Transact-SQL statements and stored procedures to finish.</td>
</tr>
<tr>
<td>Immediate</td>
<td>Shuts down the server immediately. DBArtisan does not perform checkpoints in every database. The server terminates all user processes and rolls back any active transactions.</td>
</tr>
</tbody>
</table>

For more information, see Servers.
Configuring the Server
To configure and tune your server, DBArtisan offers the Server Configuration dialog box that lets you view and edit the configuration parameters for each server. These configuration parameters are stored in the system catalog and used by DBArtisan when restarting the server. For detailed information on the meaning and purpose of each server configuration parameter, see the System Administration Guide and Performance and Tuning Guide.

NOTE: Many changes to server configuration parameters require you to shutdown and to restart DBArtisan before the changes take effect.

Completing the Server Configuration Dialog Box
NOTE: For Oracle the following dialog box is called Instance Parameters, and for SQL Server it is Server Configuration.

1. On the Datasource menu, click Configure.

The table below describes the options and functionality on the Server Configuration dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Button</td>
<td>Click to modify the target parameter. Opens the Edit Configuration dialog box.</td>
</tr>
<tr>
<td>New Value</td>
<td>Lets you type the value for the parameter.</td>
</tr>
</tbody>
</table>

Oracle Managed Files (OMF)
For Oracle 9i only, DBArtisan supports Oracle Managed Files (OMF.) OMF automatically manages files for tablespace, redo logs, and control files. Oracle automatically creates and removes operating system files as you create and drop storage objects.

You can enter the following query to set the db_create_file_dest parameter at the system level. You can set the parameter value for the session level by using Alter session clause:

```sql
ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/u01/oradata/sample2'
```

TIP: You can also set the parameter at the system level in the Edit Configuration dialog box.

For more information, see Servers.

Edit Configuration Dialog Box
The table below describes the options and functionality on the Edit Configuration dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Value</td>
<td>Lets you type the value for the parameter.</td>
</tr>
</tbody>
</table>

NOTE: To modify a parameter, the parameter must have a Yes in the Session Modifiable or System Modifiable column of the Instance Parameter grid.

For more information, see Servers.

Session Recording
DBArtisan includes a session recording feature that records changes to the server. The Session Recording feature lets you create a script file that you can execute as a batch at another time.
The session recording feature automates repetitive tasks that you routinely perform by issuing the commands in the collected script with a single command. Session recording lets you:

- Streamline time consuming, repetitive tasks.
- Execute multiple actions with from a single file.
- Automate a complex series of tasks by creating a batch file.
- Distribute automated tasks to multiple users with the session recording file.

The Session Recording feature is not available for operations that are destructive to your server. These operations include:

- Schema Migration
- Data Load
- ISQL and DDL Editors

When you activate session recording, these operations are grayed out and not available.

You can also use the session recording feature for Tivoli Integration. DBArtisan records schema change management sessions to instruction files, which are read by Tivoli Management Systems. Tivoli subsequently transports these to multiple managed endpoints for local execution.

For more information, see:

Recording a Session
Executing a Session File

Recording a Session
You can create a session file that includes a series of commands or actions using DBArtisan Session Recording Mode dialog box.

Completing the Session Recording Mode Dialog Box
1 On the Datasource menu, click Session Recording.

The table below describes the options and functionality on the Enter Session Recording Mode dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Name</td>
<td>Lets you type the name of the session.</td>
</tr>
<tr>
<td>Session File</td>
<td>Lets you type the location and session file name or click the browse button. DBArtisan uses *.ses file extension for session files.</td>
</tr>
</tbody>
</table>

**NOTE:** DBArtisan appends each task that you record in the session file. In place of an Execute button, DBArtisan displays a Record button. To record a certain task, click the Record Button to open the Task Recording dialog box.

2 When you want to exit session recording mode, open an ISQL Window that lets you review the session file.

3 On the Datasources menu, click Session Recording.
   DBArtisan opens an SQL window.

For more information, see:
Executing a Session File

You can execute a session file which includes a series of collected commands in a single batch from an ISQL Window.

1. On the File menu, click Open.
   
   DBArtisan opens the Open Files dialog box.

2. Select the session file(s) (*.ses) from the file list, or in the File Name box, type a session file name.

3. Click OK.
   
   DBArtisan opens the session file into a SQL window.

4. On the SQL tool bar, click the Execute button.

5. On the SQL tool bar, click the Step-Execute button to see a graphical display to identify the execution path of the script, and then click the Execute button to execute the session file.

6. On the SQL tool bar, click the Query Plan button to see a graphical display to identify the execution path of the script.

7. Click the Execute button.
   
   DBArtisan executes the script.

For more information, see Session Recording.

Database Object Finder

The Database Object Finder is a search utility that lets you search through all your database objects. You can narrow your search by type of object and object owner, and you can set the search direction. The Database Object Finder also includes many options that let you further focus your search. The Database Object Finder displays your search results in the Database Object Found box at the bottom of the Database Object Finder dialog box. For successful searches, DBArtisan lets you open the appropriate object editor from the Database Object Finder.

Completing the Database Object Finder

1. Connect to a datasource.

2. On the Datasource menu, click Find Object.

The table below describes the options and functionality on the Database Object Finder dialog box.
NOTE: If the search is successful, DBArtisan displays the search string in the Database Object Found box. Click the Open button and DBArtisan opens the object editor for the target object.

Changing an ODBC Datasource to a Native Datasource

CAUTION: The section below refers only to Microsoft SQL Server connectivity.

Microsoft SQL Server uses ODBC to connect to Microsoft SQL Servers. DBArtisan requires native connectivity. To connect through DBArtisan, register your Microsoft SQL Server(s) using native connectivity in the Microsoft SQL Server Client Utility.

To change your ODBC servers to native connectivity, do the following:

1. Open the Microsoft SQL Server Client Network Utility dialog box, CLICONFG.exe.
2. On the General Tab, click Add.
   Microsoft SQL Server opens the Add Network Library Configuration dialog box.
3. Click OK.
   DBArtisan returns to the Client Network Utility dialog box.
4. In the Client Network Utility dialog box, click Apply.
   DBArtisan adds the server.
5. Open DBArtisan and the Datasource Registration Wizard.

For more information, see Datasources.

### Option Table

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object to Find</td>
<td>Lets you type the target string of text. You can also click the list to choose from a list of up to ten previous search strings.</td>
</tr>
<tr>
<td>Type of Object</td>
<td>Lets you select the target database object type.</td>
</tr>
<tr>
<td>Object Owner</td>
<td>Lets you select the target database object owner.</td>
</tr>
<tr>
<td>Search Direction</td>
<td>Lets you select a search direction: From Beginning, Down, or Up.</td>
</tr>
<tr>
<td>Case-Sensitive Search</td>
<td>Select to perform the search with the same capitalization as the search string.</td>
</tr>
<tr>
<td>Find Entire String Only</td>
<td>Select to perform the search using the entire search string, not partial strings.</td>
</tr>
<tr>
<td>Search My Objects Only</td>
<td>Select to perform the search only on your database objects.</td>
</tr>
</tbody>
</table>

### Server Alias Table

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server alias</td>
<td>In the box, type the unique name of the server.</td>
</tr>
<tr>
<td>Network libraries</td>
<td>In the box, click the appropriate option button to specify the network library that connects to the server.</td>
</tr>
<tr>
<td>Computer name</td>
<td>In the box, type the name of the target computer.</td>
</tr>
<tr>
<td>Port number</td>
<td>In the box, type the port number of the target computer.</td>
</tr>
</tbody>
</table>
Supported Objects

DBArtisan lets you manage database objects across different database platforms. The table below indicates the objects that DBarisan supports by platform:

<table>
<thead>
<tr>
<th>Objects</th>
<th>IBM DB2 for Windows/Unix</th>
<th>IBM DB2 for OS/390 and z/OS</th>
<th>Microsoft SQL Server</th>
<th>MySQL</th>
<th>Oracle</th>
<th>Sybase ASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Backup Devices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bufferpools</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check Constraints</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clusters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Control Files</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Database Devices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Database Links</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Databases</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Caches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DBRM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Defaults</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Directories</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dump Devices</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Extended Procedures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Foreign Keys</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Functions</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Groups</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indexes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Java Classes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Java Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Java Sources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Job Queues</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Libraries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linked Servers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Logs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Materialized Query Tables</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Objects</td>
<td>IBM DB2 for Windows/Unix</td>
<td>IBM DB2 for OS/390 and z/OS</td>
<td>Microsoft SQL Server</td>
<td>MySQL</td>
<td>Oracle</td>
<td>Sybase ASE</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------</td>
<td>-----------------------------</td>
<td>----------------------</td>
<td>-------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>Materialized Views</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Materialized View Logs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nodegroups</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outlines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Package Bodies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packages</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plans</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary Keys</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Procedures</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Profiles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redo Log Groups</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote Servers</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Roles</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Rollback Segments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rules</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Segments</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequences</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snapshots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snapshot Logs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Stogroups</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structured Types</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Summary Tables</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synonyms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tables</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Tablespaces</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triggers</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Type Bodies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique Keys</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>User Datatypes</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>User Messages</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Aliases

NOTE: This object is supported by IBM DB2 for Linux, Unix, and Windows, IBM DB2 for OS/390 and z/OS, Microsoft SQL Server, and Sybase ASE.

Aliases let you assume the permissions of another database user without creating a separate user identity. You can use an alias when a user requires only temporary access to a database. You can also use an alias to mask a user's identity.

Datasource Explorer List

When you select the Aliases node of the Datasource Explorer, DBArtisan lists the aliases defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about aliases from the system catalog:

NOTE: Columns differ by platform.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias Owner</td>
<td>The identity of the alias' owner.</td>
</tr>
<tr>
<td>Alias Name</td>
<td>Name of the alias</td>
</tr>
<tr>
<td>Aliased Object Owner</td>
<td>User who owns the object that has been aliased</td>
</tr>
<tr>
<td>Aliased Object Name</td>
<td>Name of the object that has been aliased</td>
</tr>
<tr>
<td>Create Date</td>
<td>Date and time the alias was created</td>
</tr>
</tbody>
</table>

Available Functionality

DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Report</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bufferpools

NOTE: This object is supported by IBM DB2 for OS/390 and z/OS and IBM DB2 for Linux, Unix, and Windows only.

Bufferpools are areas of system memory to cache table and index data pages. DBArtisan retrieves table and index data pages from their physical storage location and places them in the bufferpool so that when you modify or access data, you can retrieve the data from memory rather than a physical device or file. Bufferpools improve database system performance because DBArtisan can retrieve the data quickly from memory.
Datasource Explorer List
When you select the Bufferpools node of the Datasource Explorer, DBArtisan lists the tablespaces defined in a database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about tablespaces from the system catalog:

### IBM DB2 for Windows/Unix

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bufferpool Name</td>
<td>Name of the bufferpool</td>
</tr>
<tr>
<td>Node Group or Storage Group</td>
<td>Name of the nodegroups or storage groups that are assigned to the bufferpool</td>
</tr>
<tr>
<td># Pages</td>
<td>The number of pages assigned to the bufferpool</td>
</tr>
<tr>
<td>Page Size</td>
<td>The size of the pages in the bufferpool</td>
</tr>
<tr>
<td>Extended Storage</td>
<td>Whether or not the bufferpool uses extended storage</td>
</tr>
</tbody>
</table>

### IBM DB2 for OS/390 and z/OS

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer Pool Name</td>
<td>Buffer pool external name (BP0, BP1, ..., BP49, or BP32K, BP32K1, ...,BP32K9).</td>
</tr>
<tr>
<td>ID</td>
<td>Buffer pool internal identifier (0-49, 80-89, 100-109, 120-129)</td>
</tr>
<tr>
<td># Use Count</td>
<td>Number of open table spaces or index spaces that reference this bufferpool. (Inactive pools have a zero use count.)</td>
</tr>
<tr>
<td>Size</td>
<td>User-specified virtual buffer pool size.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of virtual buffer pool to be allocated. PRIMARY - Used to maintain the coherency of cached data. This structure is used for page registration and cross-invalidation. DATASPACE - A range of up to 2 gigabytes of contiguous virtual storage addresses that a program can directly manipulate. Unlike an address space, a data space can hold only data; it does not contain common areas, system data, or programs.</td>
</tr>
<tr>
<td>Hiper Pool Size</td>
<td>User-specified hiper pool size.</td>
</tr>
<tr>
<td># VPSEQT</td>
<td>Sequential steal threshold for the virtual buffer pool.</td>
</tr>
<tr>
<td># VPPSEQT</td>
<td>Parallel sequential threshold for the virtual buffer pool.</td>
</tr>
<tr>
<td># VPXSEQT</td>
<td>Portion of the virtual buffer pool that is used for processing queries that originate on other members of the data sharing group.</td>
</tr>
<tr>
<td># HPSEQT</td>
<td>Sequential steal threshold for the hiperpool.</td>
</tr>
<tr>
<td># DWQT</td>
<td>Deferred write threshold for the virtual buffer pool. It is expressed as a percentage of the total virtual buffer pool size, and valid values range from 0 to 90. This threshold determines when deferred writes begin, based on the number of unavailable (non-stealable) buffers.</td>
</tr>
<tr>
<td># VDWQT1</td>
<td>Vertical deferred write threshold for the virtual buffer pool expressed as a percentage of the total virtual buffer pool size, and valid values range from 0 to 90. This attribute determines when deferred writes begin, based on the number of updated pages for a given data set.</td>
</tr>
<tr>
<td># VDWQT2</td>
<td>Vertical deferred write threshold for the virtual buffer pool expressed as an absolute number of buffers. VDWQT2 only applies when VDWQT1 is 0; VDWQT2 can range from0 to 9999.</td>
</tr>
<tr>
<td>CASTOUT</td>
<td>CASTOUT option for all hiperpools. The possible values are YES and NO.</td>
</tr>
<tr>
<td>PGSTEAL</td>
<td>Page stealing algorithm DB2 uses for the virtual buffer pool.LRU or FIFO are the possible options.</td>
</tr>
</tbody>
</table>
Available Functionality
DBArtisan offers the following functionality for this object:

NOTE: Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
<td>Status</td>
<td></td>
</tr>
</tbody>
</table>

Check Constraints

NOTE: This object is supported by all platforms.

Check constraints are data values that are acceptable in a column. They are logical expressions that verify column values meet defined acceptance criteria.

Datasource Explorer List

When you select the Check Constraints node of the Datasource Explorer tree, DBArtisan lists the check constraints defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about check constraints from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>Name of the table owner of the table with the check constraint</td>
</tr>
<tr>
<td>Table Name</td>
<td>Table with the check constraint</td>
</tr>
<tr>
<td>Constraint Name</td>
<td>Name of the check constraint</td>
</tr>
</tbody>
</table>

Available Functionality

DBArtisan offers the following functionality for this object:

NOTE: Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
<td>Status</td>
<td></td>
</tr>
</tbody>
</table>

Clusters

NOTE: This object is supported by Oracle only.

Clusters provide an optional method of storing table data. A cluster comprises of a group of tables that share the same data blocks, and which are grouped together because they share common columns and are often used together. The related columns of tables stored in a cluster are known as the cluster key.

There are two types of clusters:

- Index
- Hash

Index clusters store the cluster data together and index the cluster key, which should make them faster at retrieving a range of data rows.
Hash clusters apply hashing functions to the cluster key to determine the physical location of a data row, which should make them faster at retrieving specific data rows.

**NOTE:** To place a table on a cluster, include the ON CLUSTER syntax within the CREATE TABLE statement. Placing a table on a cluster precludes you from placing it on a tablespace or defining the associated storage parameters.

Datasource Explorer List
When you select the Clusters node of the Datasource Explorer, DBArtisan lists the clusters defined in a database, and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about clusters from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Owner of the cluster.</td>
</tr>
<tr>
<td>Cluster Name</td>
<td>Name of the cluster</td>
</tr>
<tr>
<td>Tablespace</td>
<td>Tablespace on which the cluster is placed</td>
</tr>
<tr>
<td>Type</td>
<td>Whether the cluster is an index or a hash cluster</td>
</tr>
<tr>
<td>%Free</td>
<td>The percentage of space reserved on each data block for update operations</td>
</tr>
<tr>
<td>%Used</td>
<td>The minimum percentage of space that must be filled on each data block</td>
</tr>
<tr>
<td>Key Size</td>
<td>Cluster key size in bytes</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Allocate Extent</th>
<th>Analyze</th>
<th>Deallocate Unused Space</th>
<th>Drop</th>
<th>Extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate</td>
<td>New</td>
<td>Open</td>
<td>Report</td>
<td>Truncate</td>
</tr>
</tbody>
</table>

Control Files
**NOTE:** This object is supported by Oracle only.

Control files map to the structures that make up the database including the location of datafiles. Control files are read-only because they are only constructed during database creation.

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Backup Control File</th>
<th>Open</th>
<th>Report</th>
</tr>
</thead>
</table>

Database Devices
**NOTE:** This object is supported by Microsoft SQL Server or later and Sybase only.
Database Devices are the fundamental storage mechanism for Microsoft SQL Server. They bridge the physical world of your hard disks and the logical internal structure of Microsoft SQL Server. When you create a device, Microsoft SQL Server writes a file on your hard disk and pre-initializes the panels for quick allocation to databases. Microsoft SQL Server stores the resulting panel range in the system catalog and refers to it when determining how to allocate space on a device to a particular database.

Datasource Explorer List

When you select the Database Devices node located under the Storage node of the Datasource Explorer tree, DBArtisan lists the database devices and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about database devices from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Device Name</td>
<td>Name of the database device.</td>
</tr>
<tr>
<td>Size (MB)</td>
<td>Defined size of the database device.</td>
</tr>
<tr>
<td>Used (MB)</td>
<td>Space on the database device that has already been allocated to databases.</td>
</tr>
<tr>
<td>Free (MB)</td>
<td>Remaining space that has not been allocated on the database device.</td>
</tr>
<tr>
<td>Physical Name</td>
<td>Name and location of the *.dat file that stores the database device.</td>
</tr>
</tbody>
</table>

Available Functionality

DBArtisan offers the following functionality for this object:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Extract</td>
<td>New</td>
<td>Open</td>
<td>Reinitialize</td>
</tr>
<tr>
<td>Resize Disk</td>
<td>Refit</td>
<td>Report</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Database Links

**NOTE:** This object is supported by Oracle and Sybase only.

Database links are named schema objects that describe a path from one database to another. Database links are implicitly used when a reference is made to a global object name in a distributed database. To use a database link, either it is public or you own it.

**NOTE:** Oracle syntax does not let you alter an existing database link. To change its definition, drop and re-create it.

Datasource Explorer List

When you select the Database Links node of the Datasource Explorer, DBArtisan lists the database links defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about database links from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Owner of the database link</td>
</tr>
<tr>
<td>Database Link Name</td>
<td>Name of the database link</td>
</tr>
<tr>
<td>User</td>
<td>The user account for connecting to the remote database</td>
</tr>
<tr>
<td>Host</td>
<td>Instructions for connecting to the remote database</td>
</tr>
<tr>
<td>Created</td>
<td>The date and time that the database link was created</td>
</tr>
</tbody>
</table>
Available Functionality

DBArtisan offers the following functionality for this object:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extract</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Migrate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Databases

NOTE: Databases are available for Microsoft SQL Server and Sybase ASE. For Oracle and IBM DB2 for Linux, Unix, and Windows, IBM DB2 for OS/390 and z/OS databases are called Instances.

Databases are a collection of tables, or a collection of index spaces and table spaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

For more information, see Available Functionality.

IBM DB2 for Linux, Unix, and Windows Instances

Databases are a collection of tables, or a collection of index spaces and table spaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

Datasource Explorer List

When you select the Instances node of the Datasource Explorer, DBArtisan lists the databases and displays pertinent information about each of them in the right pane. The tables below describes how this information is displayed:

<table>
<thead>
<tr>
<th>IBM DB2 for Windows/Unix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Column Name</strong></td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Creator</td>
</tr>
<tr>
<td>Stogroup</td>
</tr>
<tr>
<td>Bufferpool</td>
</tr>
<tr>
<td>DB ID</td>
</tr>
<tr>
<td>Timestamp</td>
</tr>
<tr>
<td>Type</td>
</tr>
<tr>
<td>Group Member</td>
</tr>
<tr>
<td>CreateEdts</td>
</tr>
<tr>
<td>AlterEdts</td>
</tr>
<tr>
<td>MIXED_CCSID</td>
</tr>
<tr>
<td>Index Bufferpool</td>
</tr>
</tbody>
</table>
Microsoft SQL Server Databases

Databases are a collection of tables, or a collection of index spaces and tablespaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

NOTE: Microsoft SQL Server recommends that you do not create any user objects, such as tables, views, stored procedures, or triggers, in the master database. The master database includes the system tables that store the system information used by SQL Server, such as configuration option settings.

Datasource Explorer List

When you select the Databases node of the Datasource Explorer, DBArtisan lists the databases and displays pertinent information about each of them in the right pane. The tables below describes how this information is displayed:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>Name of the database</td>
</tr>
<tr>
<td>Database Status</td>
<td>Status of the database, such as online, offline, read-only or single-user</td>
</tr>
<tr>
<td>Create Date/Time</td>
<td>When the database was created</td>
</tr>
</tbody>
</table>

IBM DB2 for OS/390 and z/OS Instances

Databases are a collection of tables, or a collection of index spaces and tablespaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

Datasource Explorer List

When you select the Instances node of the Datasource Explorer, DBArtisan lists the databases and displays pertinent information about each of them in the right pane. The tables below describes how this information is displayed:
IBMB2 for OS/390 and z/OS

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of database</td>
</tr>
<tr>
<td>Creator</td>
<td>Name of creator</td>
</tr>
<tr>
<td>Stogroup</td>
<td>Name of stogroup for the database, blank if it is a system database</td>
</tr>
<tr>
<td>Bufferpool</td>
<td>Name of default bufferpool for the database, blank if it is a system database</td>
</tr>
<tr>
<td>DB ID</td>
<td>Internal identification number for database</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Indicates timestamp</td>
</tr>
<tr>
<td>Type</td>
<td>Indicates type of database; W indicates a work file</td>
</tr>
<tr>
<td>Group Member</td>
<td>The DB2 sharing member name</td>
</tr>
<tr>
<td>CreateEdts</td>
<td>Time when the create statement was created for the database</td>
</tr>
<tr>
<td>AlterEdts</td>
<td>Time when the most recent alter edit was applied</td>
</tr>
<tr>
<td>MIXED_CCSID</td>
<td>Indicates graphic characters</td>
</tr>
<tr>
<td>Index Bufferpool</td>
<td>Indicates name of the default bufferpool for indexes</td>
</tr>
</tbody>
</table>

Sybase ASE Databases

Databases are a collection of tables, or a collection of index spaces and tablespaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

Datasource Explorer List

When you select the Databases node of the Datasource Explorer, DBArtisan lists the databases and displays pertinent information about each of them in the right pane. The tables below describes how this information is displayed:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>Name of the database</td>
</tr>
<tr>
<td>Database Status</td>
<td>Status of the database, such as online or offline</td>
</tr>
<tr>
<td>Size</td>
<td>Size of the database in MB</td>
</tr>
<tr>
<td>Last Dump Tran</td>
<td>The last time that the transaction log for the database was dumped</td>
</tr>
<tr>
<td>Create Date/Time</td>
<td>When the database was created</td>
</tr>
</tbody>
</table>

For more information, see [Available Functionality](#).
Available Functionality for Databases

DBArtisan offers the following functionality for databases:

**NOTE:** Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Backup</th>
<th>Checkpoint</th>
<th>Coalesce</th>
<th>DBCC</th>
<th>Detach/Attach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Filter</td>
<td>Extract</td>
<td>Migrate</td>
<td>Move Log</td>
</tr>
<tr>
<td>New</td>
<td>Open</td>
<td>Quiesce</td>
<td>Rename</td>
<td>Report</td>
</tr>
<tr>
<td>Set Online/Offline</td>
<td>Shrink</td>
<td>Start Database</td>
<td>Stop Database</td>
<td>Unquiesce</td>
</tr>
<tr>
<td>Update Statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Data Caches**

**NOTE:** This object is supported by Sybase only.

Data Caches are devices stored in memory. If configured properly, they can substantially boost performance by allowing users to read or write data in memory instead of to or from physical disk. With System 11, you can now “borrow” memory from the default data cache and allocate it to customized data caches.

Named data caches let you achieve substantially greater concurrency. Data caches also increase the likelihood that Sybase ASE reads data from memory and not disk. DBArtisan lets you bind particularly active databases (for example, tempdb), tables (for example, syslogs and sysindexes) or indexes to dedicated data caches. In addition, you can boost performance by creating memory pools, each configured for different I/O throughput (2 K to 16 K), within every named data cache.

In spite of their potential, named data caches are not a panacea for performance problems. You should still carefully optimize your queries and indexing strategy. Poor configuration of named data caches can lead to worse performance because precious memory resources are misallocated. To determine data cache performance, you should examine the cache hit ratio.

**The Datasource Explorer List**

When you select the Data Caches node located under Storage in the Datasource Explorer tree, DBArtisan lists the data caches and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about data caches from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Cache Name</td>
<td>Name of the data cache</td>
</tr>
<tr>
<td>Type</td>
<td>Type of the data cache, whether it is for mixed use or log only</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the data cache</td>
</tr>
<tr>
<td>Configured</td>
<td>Configured size of the data cache in bytes</td>
</tr>
<tr>
<td>Run</td>
<td>Run size of the data cache in bytes</td>
</tr>
<tr>
<td>Overhead</td>
<td>Overhead required to manage the data cache in bytes</td>
</tr>
</tbody>
</table>

**Available Functionality**

DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>New</th>
<th>Open</th>
<th>Report</th>
</tr>
</thead>
</table>
**DBRM**

**NOTE:** This object is supported by IBM DB2 for OS/390 and z/OS only.

A Database Request Module (DBRM) is an application containing information on SQL statements extracted from the DB2 precompiler. A DBRM refers to a specific database and is stored outside of the DB2 directory.

**The Datasource Explorer List**

When you select the DBRM node of the Datasource Explorer, DBArtisan lists pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays DBRMs from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBRM Name</td>
<td>Displays the name of the DBRM.</td>
</tr>
<tr>
<td>Version</td>
<td>Displays DBRM version.</td>
</tr>
<tr>
<td>Contoken</td>
<td>The contoken (consistency token) in the object compares to the contoken in the plan/package. If the contokens match, then the program will execute. If they do not match and autorebind = y, and the rebind is successful, then the program will execute. If they do not match and autorebind = n or fails, then an -818 is returned to the program - contoken mismatch, and the job ends. DB2 displays the mismatched contokens. The analysis associated with a -818 requires finding the DBRM or package contoken. This pane lets you easily expose the contokens and find the version mismatch.</td>
</tr>
<tr>
<td>Language</td>
<td>Displays the Host language.</td>
</tr>
<tr>
<td>Precompile Timestamp</td>
<td>Precompilation time of the DBRM.</td>
</tr>
</tbody>
</table>

**Available Functionality**

DBArtisan offers the following functionality for this object:

- **Drop**
- **Extract**
- **New**
- **Open**
- **Report**

**Defaults**

**NOTE:** This object is supported by Microsoft SQL Server and Sybase only.

Defaults promote data integrity by supplying a default value to a table column if the user does not explicitly provide one. They are reusable objects that you can bind to table columns or user datatypes.

**Datasource Explorer List**

When you select the Defaults node of the Datasource Explorer tree, DBArtisan lists the defaults defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about defaults from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Owner</td>
<td>User who owns the default</td>
</tr>
<tr>
<td>Default Name</td>
<td>Name of the default</td>
</tr>
<tr>
<td>Create/Date Time</td>
<td>When the default was created</td>
</tr>
</tbody>
</table>
Available Functionality

DBArtisan offers the following functionality for this object:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Extract</td>
<td>Migrate</td>
<td>New</td>
<td>Open</td>
</tr>
<tr>
<td>Rename</td>
<td>Report</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Directories

**NOTE:** This object is supported by Oracle only.

Directories create an alias to an external operating system directory to your database files, which can be used for storing large binary object files. When you create a directory, provide the full path name to the outside operating system where the BFILE's are stored. This object lets you store large files, such as video, outside of the database. The directory object lets you provide a simple alias to the full path name of an outside server's file system, which you can then use to point to the files when creating procedural logic objects. This saves the developer from having to type the full path name when coding.

To create a Directory object, you need CREATE ANY DIRECTORY system privileges. You also create or have in place a corresponding operating system directory to store the file. This directory must have the correct read permissions for Oracle processes.

Datasource Explorer List

When you select the Directories node of the Datasource Explorer, DBArtisan lists the directories defined in a database, and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about directories from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the directory</td>
</tr>
<tr>
<td>File Name</td>
<td>Physical path and name of the operating system file that corresponds to the directory</td>
</tr>
</tbody>
</table>

Dump and Backup Devices

**NOTE:** This object is supported by Microsoft SQL Server and Sybase only.

Dump devices store backups of databases and transaction logs. In Microsoft SQL Server, dump devices have been renamed to backup devices. Before you can dump a database or its transaction log, first create a dump device that stores the resulting backup file. Microsoft SQL Server and Sybase ASE lets you create tape, disk and diskette dump devices. Disk dump devices are a fast and convenient mechanism for backups. Tape dump devices facilitate off site storage for enhanced disaster recovery. Similarly, diskette dump devices facilitate off site storage, but only for very small databases.
Datasource Explorer List
When you select the Dump Devices or Backup Devices node located under the Storage node of the Datasource Explorer tree, DBArtisan lists the devices and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about these devices from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dump/Backup Device Name</td>
<td>Name of the dump or backup device.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of the dump device, either disk, tape or diskette.</td>
</tr>
<tr>
<td>Physical Name</td>
<td>Name of the dump device recognized by the operating system.</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>New</th>
<th>Open</th>
<th>Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Contents</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Event Monitors

**NOTE:** This object is supported by IBM DB2 for Windows/Unix only.

Event Monitors lets you monitor specified events that occur on the database including deadlocks, connections, and statements. Because monitoring is configurable by type, you can record activity against all trackable objects or only those you consider most important.

**NOTE:** Event Monitors cannot be altered using the ALTER command. An Event Monitor has to be dropped and re-created.

Datasource Explorer List
When you select the Event Monitors node of the Datasource Explorer tree, DBArtisan lists the event monitors defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about event monitors from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor Owner</td>
<td>User who owns the monitor</td>
</tr>
<tr>
<td>Monitor Name</td>
<td>Name of the monitor</td>
</tr>
<tr>
<td>Monitor Type</td>
<td>Type of the monitor</td>
</tr>
<tr>
<td>Autostarts</td>
<td>Indicates if event monitor autostarts</td>
</tr>
<tr>
<td>On Node</td>
<td>Indicates if event monitor is on node</td>
</tr>
<tr>
<td>Scope</td>
<td>Scope of the monitor</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename</td>
<td>Report</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Contents</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Extended Procedures

**NOTE:** This object is supported by Sybase ASE only.

Extended Procedures are dynamic link libraries that can be used to load and execute application routines written in other programming languages, such as C or Visual Basic. Extended Procedures function and appear in the same manner as normal stored procedures in that you can pass parameters to them and obtain results.

**NOTE:** Extended Procedures can only be accessed on the Master database.

Datasource Explorer List

When you select the Extended Stored Procedures node of the Datasource Explorer tree, DBArtisan lists the extended stored procedures defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about extended stored procedures from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>User who owns the stored procedure</td>
</tr>
<tr>
<td>Procedure Name</td>
<td>Name of the Stored Procedure</td>
</tr>
<tr>
<td>DLL</td>
<td>The routine called by the procedure</td>
</tr>
<tr>
<td>Created</td>
<td>When the stored procedure was created</td>
</tr>
</tbody>
</table>

Available Functionality

DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Execute</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Foreign Keys

**NOTE:** This object is supported by all platforms.

Foreign keys enforce referential integrity between tables by verifying the existence of foreign key values in the parent table before letting you insert or update foreign key values in the child table.

Available Functionality

DBArtisan offers the following functionality for this object:

**NOTE:** Functionality differs by platform.

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Functions

**NOTE:** This object is supported by all platforms.
Functions are subroutines that you define. Functions are useful for reusable application logic. You can use functions to determine the best methods for controlling access and manipulation of the underlying data contained in an object.

The table below describes the types of user-defined functions that DBArtisan lets you create:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column or External Table Function</td>
<td>You can write in a host programming language, such as C. This function can act on a table and returns a table value rather than a scalar value.</td>
</tr>
<tr>
<td>External Scalar Function</td>
<td>You can write in a language other than SQL, such as C++ or Java and returns a scalar value to the program. This type of function is referenced by the CREATE FUNCTION statement and can be used to perform computations on data contained in the database but cannot directly reference the data.</td>
</tr>
<tr>
<td>OLEDB Function</td>
<td>Accesses OLE DB data in user-defined OLE DB external tables.</td>
</tr>
<tr>
<td>Sourced Function</td>
<td>Inherits the semantics of another function and can be an operator.</td>
</tr>
<tr>
<td>Template Function</td>
<td>Partial functions that do not contain any executable code. Mainly used in a federated database to map the template function to a data source function -Oracle, SQL Server, Sybase, etc. A function mapping needs to be created in conjunction with the template function.</td>
</tr>
</tbody>
</table>

Datasource Explorer List

When you select the Functions node of the Datasource Explorer, DBArtisan lists the functions defined in a database, and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about functions from the system catalog:

**NOTE:** Columns differ by platform.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Owner</td>
<td>Owner of the function</td>
</tr>
<tr>
<td>Function Name</td>
<td>Name of the function</td>
</tr>
<tr>
<td>Status</td>
<td>Whether the function is valid or invalid</td>
</tr>
<tr>
<td>Specific Name</td>
<td>Unique identity of the function</td>
</tr>
<tr>
<td>Origin</td>
<td>Indicates if the function is system-generated or user-defined</td>
</tr>
<tr>
<td>Language</td>
<td>The programming language which was used to create the function.</td>
</tr>
<tr>
<td>Implementation</td>
<td>The programming language which was used to create the function.</td>
</tr>
<tr>
<td>Created</td>
<td>The date and time that the function was created</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Date that the function was last modified</td>
</tr>
</tbody>
</table>

Available Functionality

DBArtisan offers the following functionality for this object:

**NOTE:** Functionalities differ by platform.
Groups

NOTE: This object is supported by IBM DB2 for Windows/Unix, Oracle, and Sybase only.

Groups are a defined collection of database users. The primary use of groups is to consolidate the management of permissions. By batching together similar users into groups, you can greatly reduce the number of commands required to set permissions.

Every user automatically belongs to the public group. To assign a user to another group, add the user to that group. Then the user belongs to that group and public.

NOTE: A user can only belong to one group at a time other than public.

Datasource Explorer List

When you select the Groups node of the Datasource Explorer, DBArtisan lists the groups defined in a database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about the group:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Name</td>
<td>Name of the group</td>
</tr>
</tbody>
</table>

NOTE: In Oracle, there is only one group PUBLIC.

Available Functionality

DBArtisan offers the following functionality for this object:

NOTE: Functionalities differs by platform.

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename</td>
<td>Report</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Image Copies

NOTE: This object is supported by IBM DB2 for OS/390 and zOS.

Image Copies are a full image copy of all pages in a tablespace, partition, data set, or index space. Image Copies can also be an incremental image copy of only pages that have been modified since the last use of the Image Copy Wizard.

Available Functionality

DBArtisan offers the following functionality for this object:

• Recover
• Modify Recovery
  • Copy to Copy
• MergeCopy
Indexes

NOTE: This object is supported by all platforms.

Indexes are optional structures associated with tables. You can create indexes specifically to speed SQL statement execution on a table. When properly used, indexes are the primary means of reducing disk I/O. Indexes are logically and physically independent of the data in the associated table. Unique Indexes guarantee that no two rows of a table have duplicate values in the columns that define the index.

For more information, see:

IBM DB2 for Linux, Unix, and Windows Indexes
IBM DB2 for OS/390 and z/OS Indexes
Microsoft SQL Server Indexes
Oracle Indexes
Sybase ASE
Available Functionality

IBM DB2 for Linux, Unix, and Windows Indexes
IBM DB2 for Linux, Unix, and Windows offers two types of indexes:

- Unique
- Non-Unique

Unique Indexes guarantee that no two rows of a table have duplicate values in the columns that define the index.

When you select the indexes node of the Datasource Explorer, DBArtisan lists the indexes defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about indexes from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>User who owns the indexed table</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the indexed table</td>
</tr>
<tr>
<td>Index Owner</td>
<td>Name of the index owner</td>
</tr>
<tr>
<td>Index Name</td>
<td>Name of the index</td>
</tr>
<tr>
<td>Type</td>
<td>Indicates if the index is primary, unique, or non-unique</td>
</tr>
<tr>
<td>#Levels</td>
<td>The number of levels of the index's B-Tree. If this value is greater than 3, then you should consider rebuilding the index.</td>
</tr>
<tr>
<td>#Columns</td>
<td>The number of table columns that are indexed.</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.

Microsoft SQL Server Indexes
Microsoft SQL Server offers two types of indexes: clustered and non-clustered. Clustered indexes physically sort table data to match their logical order. Non-clustered indexes only order the table data logically. In a database, an index lets you speed queries by setting pointers that allow you to retrieve table data without scanning the entire table. An index can be unique or non-unique.
Microsoft SQL Server creates indexes as B-Trees, which are a series of pointers mapping index pages to their underlying data pages. As tables and, therefore, indexes grow, the number of levels in the B-Tree increases. The B-Tree of a clustered index is shorter than that of a non-clustered index because the leaf level of a clustered index is the data page.

A sound indexing strategy is critical to overall system performance. One pitfall to avoid is placing many indexes on a table without regard for their cumulative cost. Remember that indexes improve read but slow write performance because Microsoft SQL Server must update more information in the system catalog. Consequently, extra indexes can actually slow overall performance if data modification occurs frequently on the table. To determine the efficacy of indexes, you should tune your queries using SHOWPLAN and IO STATISTICS and analyze the selectivity of indexes using DBCC SHOW_STATISTICS.

DBArtisan separates system indexes from user-defined indexes in the Datasource Explorer to ensure that system indexes are not accidentally altered or dropped.

**Datasource Explorer List**

When you select the indexes node of the Datasource Explorer, DBArtisan lists the indexes defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about indexes from the system catalog:

<table>
<thead>
<tr>
<th>Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Column Name</strong></td>
</tr>
<tr>
<td>Table Owner</td>
</tr>
<tr>
<td>Table Name</td>
</tr>
<tr>
<td>Index Name</td>
</tr>
<tr>
<td>Clustered</td>
</tr>
<tr>
<td>Unique</td>
</tr>
<tr>
<td>File Group</td>
</tr>
</tbody>
</table>

For more information, see [Available Functionality](#).

**Oracle Indexes**

Oracle offers two types of indexes. The table below describes these indexes:

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>A table index is defined on an individual table.</td>
</tr>
<tr>
<td>Cluster</td>
<td>A cluster index is defined on a set of tables physically stored together in a cluster. In an Oracle database, both table and cluster indexes use a B-tree structure.</td>
</tr>
</tbody>
</table>

The indexing strategy, particularly with large, active tables, is critical to overall system performance. The optimal definition and number of indexes for a given table is determined by the mix of access paths to that table performing insert, update, delete and select operations. For example, adding or changing an index can speed up your selects but slow your inserts, updates and deletes. Careful tuning and testing helps you achieve the best overall performance.

**TIP:** Indexes generally improve read operations in a database, but you should not place too many indexes on some tables. Since Oracle must maintain each index along with its referenced table, placing too many indexes on a table that is the object of much insert, update, and delete activity, can actually degrade performance.
Even when an index exists on a table, the way a SQL statement is coded can actually disallow the use of the index. To prevent this from happening, follow these rules of thumb:

- Try not to use SQL statements that include the NOT IN, NOT LIKE, <>, IS NULL operators because they typically suppress the use of indexes.
- When referencing concatenated indexes with queries, be sure the leading column in the index is used. If it isn't, the index won't be used at all.
- Avoid using functions in WHERE predicates.

If you must use functions, and you are using Oracle8i, investigate the use of function-based indexes.

**Datasource Explorer List**

When you select the indexes node of the Datasource Explorer, DBArtisan lists the indexes defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derivs and displays about indexes from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>User who owns the indexed table</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the indexed table</td>
</tr>
<tr>
<td>Index Name</td>
<td>Name of the index</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates if the index is valid or invalid</td>
</tr>
<tr>
<td>Unique</td>
<td>Indicates if the index is unique or non-unique</td>
</tr>
<tr>
<td>Levels</td>
<td>The number of levels of the index's B-Tree. If this value is greater than 3, then you should consider rebuilding the index.</td>
</tr>
<tr>
<td>Dist. Keys</td>
<td>The number of distinct values found in the table based on the index. A low value indicates that the index is not very specific and, therefore, could be unnecessary.</td>
</tr>
</tbody>
</table>

For more information, see [Available Functionality](#).

**Index Partitions**

Index partitions are similar to table partitions. There are three types of partitioned indexes that Oracle supports:

1. **Local prefixed**
2. **Local nonprefixed**
3. **Global prefixed**

**NOTE:** An index cannot be partitioned if it is a cluster index or if the index is defined on a clustered table.

**Local prefixed and nonprefixed indexes**

A local partitioned index has keys that refer to rows in a single table partition. A local partitioned index is automatically partitioned to mirror the underlying table. The number of partitions or subpartitions and the partition bounds for the partitioned index correspond with the partitions on the table. Oracle maintains this correspondence. If the table partitions are altered, the index partitions are altered accordingly.

A local partitioned index is prefixed if it is partitioned on the same column as the underlying table. The local partitioned index is nonprefixed if it is partitioned on a different column.

Global prefixed indexes
A global partitioned index can refer to rows in more than one table partition or subpartition. Global partitioned indexes are more difficult to manage than local partitioned indexes because any change in the underlying table partition affects all partitions in a global index. As a result, there is increased partition maintenance.

**NOTE:** A global index can only be range partitioned but it can be defined on any kind of partitioned table.

### IBM DB2 for OS/390 and z/OS Indexes

IBM DB2 for OS/390 and z/OS offers two types of indexes:
- Unique
- Non-Unique

Unique Indexes guarantee that no two rows of a table have duplicate values in the columns that define the index.

Non-Unique indexes let table rows have duplicate values in the columns that define the indexes.

### Datasource Explorer List

When you select the indexes node of the Datasource Explorer, DBArtisan lists the indexes defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about indexes from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Name</td>
<td>Name of the index</td>
</tr>
<tr>
<td>Index Owner</td>
<td>Name of the index owner</td>
</tr>
<tr>
<td>Organization</td>
<td>Indicates the object reference</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the indexed table</td>
</tr>
<tr>
<td>Table Owner</td>
<td>User who owns the indexed table</td>
</tr>
<tr>
<td>Database</td>
<td>Name of the database</td>
</tr>
<tr>
<td>Bufferpool</td>
<td>Name of the Bufferpool used for the index</td>
</tr>
<tr>
<td>Last RUNSTATS</td>
<td>Indicates the time of the last RUNSTAT</td>
</tr>
<tr>
<td>Levels</td>
<td>The number of levels of the index's B-Tree. If this value is greater than 3, then you should consider rebuilding the index.</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.

### Sybase ASE Indexes

Sybase ASE offers two types of indexes: clustered and non-clustered. Clustered indexes physically sort table data to match their logical order. Non-clustered indexes only order the table data logically. In a database, an index lets you speed queries by setting pointers that let you retrieve table data without scanning the entire table. An index can be unique or non-unique.

Sybase ASE creates indexes as B-Trees, which are a series of pointers mapping index pages to their underlying data pages. As tables and, therefore, indexes grow, the number of levels in the B-Tree increases. The B-Tree of a clustered index is shorter than that of a non-clustered index because the leaf level of a clustered index is the data page.
A sound indexing strategy is critical to overall system performance. One pitfall to avoid is placing many indexes on a table without regard for their cumulative cost. Remember that indexes improve read but slow write performance because Sybase ASE must update more information in the system catalog. Consequently, extra indexes can actually slow overall performance if data modification occurs frequently on the table. To determine the efficacy of indexes, you should tune your queries using SHOWPLAN and IO STATISTICS and analyze the selectivity of indexes using DBCC SHOW_STATISTICS.

DBArtisan separates system indexes from user-defined indexes in the Datasource Explorer to ensure that system indexes are not accidentally altered or dropped.

Datasource Explorer List
When you select the indexes node of the Datasource Explorer tree, DBArtisan lists the indexes defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about indexes from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>User who owns the indexed table</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the indexed table</td>
</tr>
<tr>
<td>Index Name</td>
<td>Name of the index</td>
</tr>
<tr>
<td>Clustered</td>
<td>Whether the index is clustered or non-clustered</td>
</tr>
<tr>
<td>Unique</td>
<td>Whether or not the index is unique</td>
</tr>
<tr>
<td>Segment</td>
<td>The segment on which the index is placed</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.

Available Functionality for Indexes
DBArtisan offers the following functionality for indexes:

NOTE: Functionality differs by platform.

<table>
<thead>
<tr>
<th>Analyze</th>
<th>Allocate Extent</th>
<th>Check Index</th>
<th>Convert to Partitioned</th>
<th>Copy to Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deallocate Unused Space</td>
<td>Drop</td>
<td>DBCC</td>
<td>Estimate Size</td>
<td>Extract</td>
</tr>
<tr>
<td>Filter</td>
<td>Image Copy</td>
<td>Migrate</td>
<td>Modify Recovery</td>
<td>Modify Statistics</td>
</tr>
<tr>
<td>New</td>
<td>Open</td>
<td>Place</td>
<td>Rebuild (IBM DB2 for OS/390 and z/OS)</td>
<td>Rebuild (Oracle)</td>
</tr>
<tr>
<td>Rename</td>
<td>Reorganize</td>
<td>Report</td>
<td>Runstats Index</td>
<td>Update Statistics</td>
</tr>
</tbody>
</table>

Instance

NOTE: This object is support for IBM DB2 and Oracle.

DBArtisan places Instance as the first level of information under the Datasource node in the Datasource Explorer. Instance includes:

• DB Manager Configuration

• Datasources
Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Edit Node Connection</th>
<th>New UDB Database</th>
<th>Quiesce</th>
<th>Unguesce</th>
</tr>
</thead>
</table>

**Java Classes**

**NOTE:** This object is supported by Oracle 8i or later only.

The Java Classes contain compiled Java code. Java Classes are made up of a group of data items, with associated functions that perform operations. The data items are called fields or variables; the functions are referred to as methods.

**TIP:** Oracle is shipped with a JVM (Java Virtual Machine). The JVM provided by Oracle sits atop the Oracle RDBMS and interacts directly with the RDBMS instead of the operating system.

Available Functionality

- **Drop**

**Java Resources**

**NOTE:** This object is supported by Oracle only.

The Java Resources node of the Explorer tab offers support for browsing Java resources.

Available Functionality

- **Drop**

**Java Sources**

**NOTE:** This object is supported by Oracle 8i or later only.

Java Sources contain the uncompiled Java source code.

**TIP:** Oracle is shipped with a JVM (Java Virtual Machine). The JVM provided by Oracle sits atop the Oracle RDBMS and interacts directly with the RDBMS instead of the operating system.

Available Functionality

<table>
<thead>
<tr>
<th>Compile</th>
<th>Create</th>
<th>Edit</th>
<th>Load Java</th>
<th>Drop</th>
</tr>
</thead>
</table>

**Job Queues**

**NOTE:** This object is supported by Oracle only.

Job Queues are built-in mechanisms that let you schedule a variety of SQL-based or command-line driven tasks.
Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Run</th>
<th>New</th>
<th>Open</th>
<th>Enable</th>
<th>Disable</th>
</tr>
</thead>
</table>

Libraries

**NOTE:** This object is supported by Oracle only.

Libraries are an object type introduced in Oracle8 that represent a call to an operating system shared library. After the call is made, libraries can be used by SQL or PL/SQL to link to external procedures or functions. Libraries are only to be used on operating systems that support shared libraries and dynamic linking. Libraries serve as pointers or aliases to physical operating system shared library files and do not have existence as a physical object on their own, rather they rely on the physical existence of the files in the external operating system library to which they refer. To access the function or procedures stored in the library, you need execute privileges at the operating system level where the shared library resides.

Datasource Explorer List

When you select the libraries node of the Datasource Explorer, DBArtisan lists the libraries defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about indexes from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>User who owns the library</td>
</tr>
<tr>
<td>Library Name</td>
<td>Name of the library</td>
</tr>
<tr>
<td>File Name</td>
<td>Name and location of the shared operating system library file</td>
</tr>
<tr>
<td>Dynamic</td>
<td>Whether the referenced operating system file supports dynamic linking</td>
</tr>
<tr>
<td>Status</td>
<td>Whether or not the library is valid. Libraries are not validated upon creation, but rather when they are called in a procedure or function.</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename</td>
<td>Report</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Linked Servers

**NOTE:** This object is supported by Microsoft SQL Server or later.

Linked servers let you create links to OLE DB data sources. With a linked server you can execute commands against OLE DB data sources on different servers. Using linked servers you can:

- Access remote servers
- Issue enterprise-wide distributed queries, updates, commands, and transactions
Distributed Queries
Before you run a distributed query, write the following as the first statement of your SQL script:

```sql
SET ANSI_NULLS ON
GO
SET ANSI_WARNINGS ON
GO
```

Datasource Explorer List
When you select the Linked Servers node located under the Storage node of the Datasource Explorer, DBArtisan lists the linked servers and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about linked servers from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linked Server Name</td>
<td>Name of the server.</td>
</tr>
<tr>
<td>Net Name</td>
<td>Reserved (name of the server).</td>
</tr>
<tr>
<td>Product Name</td>
<td>Product name for the linked server.</td>
</tr>
<tr>
<td>Ole DB Provider Name</td>
<td>Name of the OLE DB provider accessing the server.</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
<th>Report</th>
</tr>
</thead>
</table>

Logins

**NOTE:** This object is supported by Microsoft SQL Server and Sybase only.

Logins let you access your account. Your login account controls access to the server and all of the databases within it. Only the System Administrator or System Security Officer can create logins. Once you can log into a server, you need additional privileges to access user databases. Specifically, each database owner adds the login as a user or alias to the database.

**NOTE:** Logins are available for [Microsoft SQL Server](https://www.microsoft.com/en-us/sql-server) and [Sybase ASE](https://www.sybase.com/en_us/).
Datasource Explorer List
When you select the Logins node located under the Storage node of the Datasource Explorer tree, DBArtisan lists the logins and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about logins from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>Name of the login.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of login Standard, NT Group, or NT User.</td>
</tr>
<tr>
<td>Access</td>
<td>Indicates if the login has been granted access to the server or not.</td>
</tr>
<tr>
<td>Default Database</td>
<td>Database that is assigned as the default for the login.</td>
</tr>
<tr>
<td>Server Role</td>
<td>Shows whether or not a server role is assigned to the login.</td>
</tr>
<tr>
<td>Date/Time Created</td>
<td>When the login was created.</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.

Sybase ASE Logins
Logins let you access your account. Your login account controls access to the server and all of the databases within it. Only the System Administrator or System Security Officer can create logins. Once you can log into a server, you need additional privileges to access user databases. Specifically, each database owner adds the login as a user or alias to the database.

The Datasource Explorer List
When you select the Logins node located under the Storage node of the Datasource Explorer tree, DBArtisan lists the logins and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about logins from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>Name of the login</td>
</tr>
<tr>
<td>Default Database</td>
<td>Database that is assigned as the default for the login</td>
</tr>
<tr>
<td>CPU</td>
<td>Cumulative CPU usage by the login since statistics were last flushed</td>
</tr>
<tr>
<td>I/O</td>
<td>Cumulative I/O usage by the login. Since statistics were last flushed</td>
</tr>
<tr>
<td>Create Date/Time</td>
<td>When the login was created</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Change Password</th>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Report</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Materialized Query Tables

NOTE: This object is supported by IBM DB2 for Linux, Unix, and Windows version 8.

A materialized query table is a table whose definition is based on the result of a query. The materialized query table typically contains pre-computed results based on the data existing in the table or tables that its definition is based on. If the SQL compiler determines that a query will run more efficiently against a materialized query table than the base table or tables, the query quickly executes against the materialized query table.

Datasource Explorer List
When you select the Materialized Query Tables node of the Datasource Explorer, DBArtisan lists the materialized query tables defined in a database and displays pertinent information about each of them in the right pane.

The table below describes the information DBArtisan derives and displays about materialized query tables from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>User who owns the table</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the table</td>
</tr>
<tr>
<td># Rows</td>
<td>Number of rows of data the table includes</td>
</tr>
<tr>
<td>% Free</td>
<td>The amount of remaining space allocated to the table</td>
</tr>
<tr>
<td>Tablespace</td>
<td>The name of the tablespace on which the table resides</td>
</tr>
<tr>
<td>Index TS</td>
<td>The name of the tablespace on which the table's indexes reside</td>
</tr>
<tr>
<td>Long TS</td>
<td>The name of the tablespace on which the long objects (LOB,CLOB) stored on the table reside</td>
</tr>
<tr>
<td>Refresh Mode</td>
<td>Either Deferred or Immediate</td>
</tr>
<tr>
<td>Refresh Time</td>
<td>Last time table was refreshed</td>
</tr>
<tr>
<td>Create Time</td>
<td>When the table was created</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Create Insert Statements</th>
<th>Create Synonym</th>
<th>Drop</th>
<th>Edit Data</th>
<th>Extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate</td>
<td>New</td>
<td>Open</td>
<td>Refresh Table</td>
<td>Rename</td>
</tr>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Materialized Views

NOTE: This object is supported by Oracle only.

Materialized views are used to dynamically copy data between distributed databases. There are two types of materialized views:

- Complex
- Simple
Complex materialized views copy part of a master table or data from more than one master table. Simple materialized views directly copy a single table. You cannot directly update the underlying data contained in materialized views.

**NOTE:** Materialized Views are only in Oracle 8.1 or later. If you are using Oracle 8.0 or earlier, see [Snapshots](#).

### Datasource Explorer List
When you select the Materialized Views node of the Datasource Explorer, DBArtisan lists the materialized views defined in a database and displays pertinent information about each of them in the right pane.

The table below describes the information DBArtisan derives and displays about materialized views from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Materialized View Owner</td>
<td>User who owns the materialized view</td>
</tr>
<tr>
<td>Materialized View Name</td>
<td>Name of the materialized view</td>
</tr>
<tr>
<td>Last Refresh</td>
<td>Date the materialized view was last refreshed.</td>
</tr>
<tr>
<td>Refresh Type</td>
<td>The method by which the materialized view refreshes itself (fast, force or complete).</td>
</tr>
</tbody>
</table>

### Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Compile</th>
<th>Create Synonym</th>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
<td></td>
</tr>
</tbody>
</table>

### Materialized View Logs

**NOTE:** This object is supported by Oracle only.

Materialized View logs are tables that maintain a history of modifications to the master table, and they are used to refresh simple materialized views. When you create a materialized view log, Oracle automatically creates a log table to track data changes in the master table and a log trigger to maintain the data in the log table.

**NOTE:** Materialized View Logs are in Oracle 8i or later. If you are using Oracle 8.0 or earlier, see [Snapshot Logs](#).

### Datasource Explorer List
When you select the Materialized View Logs node of the Datasource Explorer, DBArtisan lists the materialized views logs defined in a database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about materialized view logs from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Materialized View Log Owner</td>
<td>User who owns the materialized view log</td>
</tr>
<tr>
<td>Materialized View Log Name</td>
<td>Name of the materialized view log</td>
</tr>
<tr>
<td>Log Table</td>
<td>A system-generated table that tracks changes to data in the master table</td>
</tr>
<tr>
<td>Log Trigger</td>
<td>A system-generated trigger that maintains data in the Log Table</td>
</tr>
</tbody>
</table>
Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Nodegroups

**NOTE:** This object is supported by IBM DB2 for Windows/Unix only.

Nodegroups are a named subset of one or more partitions in a database. DBArtisan places Nodegroup under the Storage node in the Datasource Explorer.

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rebind</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Oracle Job Queues

**NOTE:** This object is supported by Oracle only.

Oracle job queues are built-in mechanisms in the Oracle database that let you schedule a variety of SQL-based or command-line driven tasks. Basically, they are database-driven schedulers.

Datasource Explorer List

When you select the Oracle Job Queues node of the Datasource Explorer, DBArtisan lists the job queues defined in the database and displays pertinent information about each of them in the right pane. DBArtisan displays the following columns:

- Logon User
- Job ID
- Job
- Running
- Start Date/Time
- Next Date/Time
- Failures
- Broken

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outlines

**NOTE:** This object is supported by Oracle 8.1 or later only.

Outlines are a set of results for the execution plan generation of a particular SQL statement. When you create an outline, plan stability examines the optimization results using the same data used to generate the execution plan. That is, Oracle uses the input to the execution plan to generate an outline, and not the execution plan itself.

**Datasource Explorer List**

When you select the Outlines node of the Datasource Explorer, DBArtisan lists the outlines defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about stored outlines from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>User who owns the stored outline.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the stored outline.</td>
</tr>
<tr>
<td>Category</td>
<td>Indicates the category of the stored outline.</td>
</tr>
<tr>
<td>Used</td>
<td>Indicates if the outline is used or unused.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Date and time the outline was created.</td>
</tr>
<tr>
<td>Version</td>
<td>Indicates the version of Oracle for the stored outline.</td>
</tr>
</tbody>
</table>

**Available Functionality**

DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Change Category</th>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Reassign by Category</td>
<td>Rebuild</td>
<td>Rename</td>
<td>Report</td>
</tr>
</tbody>
</table>

**Packages**

**NOTE:** This object is supported by IBM DB2 for Linux, Unix, and Windows, IBM DB2 for OS/390 and z/OS, and Oracle only.

Packages contain all the information needed to process SQL statements from a single source file. You can use packages to process and call batches of SQL. Depending on the platform, packages can include:

- Procedures
- Functions
- Types
- Variables
- Constants
- Exceptions
• Cursors
• Subprograms

Packages offer a number of important advantages over using standalone procedures and functions, including the ability to:

• Modify package objects without recompiling dependent database objects.
• Declare global variables and cursors that can be shared within the package.
• Grant privileges more efficiently.
• Load multiple package objects into memory at once.

Packages usually have two parts: a header or specification and a body, although sometimes the body is unnecessary. The package header declares the members of the package while the body details the logic underlying each of the package components.

**NOTE:** DBArtisan splits out package headers and package bodies in the Datasource Explorer; however, you create both a package header and body from the packages node.

**Datasource Explorer List**

When you select the Packages node of the Datasource Explorer, DBArtisan lists the packages defined in a database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about packages from the system catalog:

**NOTE:** Columns differ by platform.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>User who owns the package</td>
</tr>
<tr>
<td>Package Name</td>
<td>Name of the package</td>
</tr>
<tr>
<td>Collection</td>
<td>Server generated identification</td>
</tr>
<tr>
<td>Collection ID</td>
<td>Server generated identification</td>
</tr>
<tr>
<td>Contoken</td>
<td>The consistency token in the object is compared to the contoken in the package. This pane lets you easily expose the contokens and find the version mismatch.</td>
</tr>
<tr>
<td>Creator</td>
<td>The user who generated the package</td>
</tr>
<tr>
<td>Qualifier</td>
<td>Name of package creator</td>
</tr>
<tr>
<td>Status</td>
<td>Whether the package is valid or invalid</td>
</tr>
<tr>
<td>Explain Snapshot</td>
<td>Whether or not the snapshot information should be gathered for each static SQL statement in the calling application</td>
</tr>
<tr>
<td>Binder</td>
<td>The user who generated the package</td>
</tr>
<tr>
<td>Sections</td>
<td>The number of sections that comprise the package</td>
</tr>
<tr>
<td>Last Bind Date</td>
<td>The date that the package was last bound to an external program</td>
</tr>
<tr>
<td>Valid</td>
<td>Whether the package is valid or invalid</td>
</tr>
<tr>
<td>Operative</td>
<td>Whether the package is operative or not</td>
</tr>
<tr>
<td>Created</td>
<td>Date and time the package was created</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Date and time the package was last modified</td>
</tr>
</tbody>
</table>
Available Functionality
DBArtisan offers the following functionality for this object:

**NOTE:** Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Bind</th>
<th>Compile</th>
<th>Create Synonym</th>
<th>Drop</th>
<th>Explain Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract</td>
<td>Filter</td>
<td>Flush Cache</td>
<td>Migrate</td>
<td>New</td>
</tr>
<tr>
<td>Open</td>
<td>Rebind</td>
<td>Report</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Package Bodies

**NOTE:** This object is supported by Oracle only.

Package Bodies implement the package specification in that the package body includes the definition of every cursor and subprogram declared in the package specification. While Package Bodies are listed as a separate object in the Datasource Explorer, they are created on the Packages Editor in conjunction with Packages.

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Compile</th>
<th>Create Synonym</th>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Open</td>
<td>Report</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Plans

**NOTE:** This object is supported by IBM DB2 for OS/390 and z/OS only.

A Plan is an executable application created in the bind process. It can include one or more packages or debris.

The Datasource Explorer List
You can select the Plan node in the Datasource Explorer and view the corresponding statistics. The table below describes the information DBArtisan derives and displays about plans from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan Name</td>
<td>Name of plan</td>
</tr>
<tr>
<td>Owner/Creator</td>
<td>User name of plan owner or creator</td>
</tr>
<tr>
<td>Qualifier</td>
<td>Name of qualifier</td>
</tr>
<tr>
<td>Valid</td>
<td>Indicates if plan is valid or invalid</td>
</tr>
<tr>
<td>Operative</td>
<td>Indicates if plan is operative</td>
</tr>
<tr>
<td>Isolation</td>
<td>Level of isolation</td>
</tr>
<tr>
<td>Reoptvar</td>
<td></td>
</tr>
<tr>
<td>Packlists</td>
<td>Number of packlists</td>
</tr>
<tr>
<td>Environments</td>
<td>Number of environments</td>
</tr>
<tr>
<td>Bind Date/Timestamp</td>
<td>Time stamp of bind</td>
</tr>
</tbody>
</table>
Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Bind</th>
<th>Copy Object Names</th>
<th>Drop</th>
<th>Filter</th>
<th>Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Rebind</td>
<td>Report</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Primary Keys**

**NOTE:** This object is supported by all platforms.

Primary Keys are a set of table columns that can uniquely identify every row of a table.

Available Functionality
DBArtisan offers the following functionality for this object:

**NOTE:** Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
<td>Status</td>
<td></td>
</tr>
</tbody>
</table>

**Procedures**

**NOTE:** This object is supported by all platforms.

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

**NOTE:** Only IBM DB2 for OS/390 and z/OS SQL stored procedures created by Rapid SQL, DBArtisan, or IBM's Stored Procedure Builder can be retrieved by Rapid SQL or DBArtisan.

**Sybase ASE**

Procedures perform procedural logic in your Sybase ASE applications. They are batches of SQL statements that are compiled and stored in the system catalog. Procedures execute faster than embedded SQL statements because they are pre-compiled and have execution plans for use by the optimizer. When you create a procedure, Sybase ASE builds a query tree, which it stores in a system table. When you execute a procedure for the first time, Sybase ASE loads it from the system table, compiles, and optimizes it. Sybase ASE places the resulting query plan in the procedure cache where it remains on a most recently used basis. In addition to better performance, procedures yield other benefits, including easier code maintenance, additional security and reduced network traffic.

In addition, DBArtisan now supports extended procedures. Extended procedures are dynamic link libraries that can be used to load and execute application routines written in other programming languages, such as C or Visual Basic. Extended procedures function and appear in the same manner as normal procedures in that you can pass parameters to them and obtain results.
Datasource Explorer List

When you select the Procedures node of the Datasource Explorer, DBArtisan lists procedures defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about procedures from the system catalog:

**NOTE:** Columns differ by platform.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>User who owns the procedure</td>
</tr>
<tr>
<td>Procedure Name</td>
<td>Name of the procedure</td>
</tr>
<tr>
<td>Procedure Owner</td>
<td>User who owns the procedure</td>
</tr>
<tr>
<td>Schema</td>
<td>Indicates the schema of the procedure</td>
</tr>
<tr>
<td>Owner</td>
<td>User who owns the procedure</td>
</tr>
<tr>
<td>External Name</td>
<td>The load module name for the procedure</td>
</tr>
<tr>
<td>Language</td>
<td>The programming language in which the procedure body was written</td>
</tr>
<tr>
<td>WLM Environment</td>
<td>Indicates the WLM application environment</td>
</tr>
<tr>
<td>Create Time</td>
<td>Indicates the time the procedure was created</td>
</tr>
<tr>
<td>Specific Name</td>
<td>The name that uniquely identifies the procedure</td>
</tr>
<tr>
<td>Tran Mode</td>
<td>Whether the procedure is chained or unchained</td>
</tr>
<tr>
<td>Origin</td>
<td>Indicates if the procedure is was created externally or internally</td>
</tr>
<tr>
<td>Language</td>
<td>The programming language in which the procedure body was written</td>
</tr>
<tr>
<td>Implementation</td>
<td>How the procedure is implemented or called</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates if the procedure is valid or invalid</td>
</tr>
<tr>
<td>Created</td>
<td>Date and time the procedure was created</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Date and time the procedure was last modified</td>
</tr>
</tbody>
</table>

Available Functionality

DBArtisan offers the following functionality for this object:

**NOTE:** Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Compile</th>
<th>Create Synonym</th>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
<td></td>
</tr>
</tbody>
</table>

Profiles

**NOTE:** This object is supported by Oracle only.

Profiles are a mechanism for allocating system and database resources to users. Profiles let you specify limits on:

- Number of sessions
- CPU time
• Connect time
• Idle time
• Logical reads and space in the SGA's shared pool

You can assign profiles to one or more users. The database's default profile and all of its resource limits are assigned to users without a specific profile assignment.

**Datasource Explorer List**
When you select the Profiles node of the Datasource Explorer, DBArtisan lists the profiles defined in a database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about profiles from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Name</td>
<td>Name of the profile</td>
</tr>
</tbody>
</table>

**Available Functionality**
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Redo Log Groups**

*NOTE:* This object is supported by Oracle only.

Redo Logs save information on the state of the database after an update. If the database fails before the update is written to the datafiles, the update can be re-done, bringing the changes back after a crash, so it appears the update did happen.

*TIP:* Have sufficient redo logs available so that other Oracle threads do not have to wait until one is free before performing an update.

Redo logs are arranged in groups, and when a particular group is being written to, each member receives the same data.

*TIP:* To avoid crashes, have two logfiles in a group, each one being on a different (physical) disk.

**Available Functionality**
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Remote Servers**

*NOTE:* This object is supported by Microsoft SQL Server and Sybase ASE only.
Remote Servers are servers on a network that you can access through your local server. Instead of logging directly into a remote server, you can log into your local server and execute a remote procedure call to it.

The local server routes all commands to the remote server. To communicate with a remote server, you log into a local server and submit your remote procedure calls for remote servers. The remote server receives remote procedure calls from the local server and processes them accordingly.

Datasource Explorer List
When you select the Remote Servers node located under the Storage node of the Datasource Explorer tree, DBArtisan lists the remote servers and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about remote servers from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Server Name</td>
<td>Name of the server</td>
</tr>
<tr>
<td>Net Name</td>
<td>Reserved (name of the server)</td>
</tr>
<tr>
<td>Product Name</td>
<td>Product name for the remote server(e.g. Microsoft SQL Server)</td>
</tr>
<tr>
<td>Ole Provider Name</td>
<td>Name of the Ole provider accessing the server</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Server Name</td>
<td>Name of the server</td>
</tr>
<tr>
<td>Network Name</td>
<td>Name of the network on which the Remote Server resides</td>
</tr>
<tr>
<td>Timeouts</td>
<td>Shows whether or not the Timeout option has been enabled. If it is set to 1 then the site handler for the remote login will timeout if there is no remote user activity for one minute.</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Roles

**NOTE:** This object is supported by **Microsoft SQL Server**, **Oracle** and **Sybase only**.

Roles are sets of user privileges you associate with access to objects within a database. Roles streamline the process of granting permissions. You can use roles to grant sets of permissions and privileges to users and groups.

For more information, see [Available Functionality](#).
Microsoft SQL Server Roles
Roles let you manage permissions for users to perform a set of activities based on job functions. Roles are a mechanism for streamlining the assignment of object and system privileges to multiple users. Instead of granting sets of privileges to individual users, you can create a role, grant system and object privileges to that role, and then grant that role to all the users who should share the same privileges. You can grant one or more roles to the same user. Therefore, if you need to change the permissions for a certain role you can do so, and not have to grant or revoke the permissions for each individual user.

Datasource Explorer List
When you select the Roles node of the Datasource Explorer tree, DBArtisan lists the roles defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about roles from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Name</td>
<td>Name of the role</td>
</tr>
<tr>
<td>Role Type</td>
<td>Whether the role is a standard Microsoft SQL Server role or an application role</td>
</tr>
<tr>
<td>Created</td>
<td>When the role was created</td>
</tr>
<tr>
<td>Updated</td>
<td>When the role was last updated</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.

Oracle Roles
Roles are a mechanism for streamlining the assignment of object and system privileges to multiple users. Instead of granting sets of privileges to individual users, you can create a role, grant system and object privileges to that role, then simply grant that role to all the users who should share the same privileges. You can grant one or more roles to the same user.

The Datasource Explorer List
When you select the Roles node of the Datasource Explorer, DBArtisan lists the roles defined in the database, displaying pertinent information about each of them in the right pane. DBArtisan derives and displays the following information about roles from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Name</td>
<td>Name of the role</td>
</tr>
<tr>
<td>Password Required</td>
<td>Password authentication for the role. The valid values are Yes (specific password required), No (no password required), External (password is authenticated through the operating system) or Global (password is authenticated through the security domain central authority).</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.

Sybase ASE Roles
Roles are a mechanism for streamlining the assignment of object and system privileges to multiple users. Instead of granting sets of privileges to individual users, you can create a role, grant system and object privileges to that role, then simply grant that role to all the users who should share the same privileges. You can grant one or more roles to the same user.
The Datasource Explorer List

When you select the Roles node of the Datasource Explorer, DBArtisan lists the roles defined in the database, displaying pertinent information about each of them in the right pane. DBArtisan derives and displays the following information about roles from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Name</td>
<td>Name of the role</td>
</tr>
</tbody>
</table>

For more information, see [Available Functionality](#).

Available Functionality for Roles

DBArtisan offers the following functionality for roles:

<table>
<thead>
<tr>
<th>Action</th>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
<th>Report</th>
</tr>
</thead>
</table>

Rollback Segments

**NOTE:** This object is supported by Oracle only.

Rollback segments manage all transactions in your Oracle databases. They maintain read consistency among concurrent users in a database and can rollback transactions. They are transaction logs that maintain discrete records of changes to data. By maintaining a history of data changes, rollback segments can rollback uncommitted transactions so that data is rolled back to their prior state.

**TIP:** Databases with large user bases and intensive data manipulation characteristics can benefit from having multiple rollback tablespaces that are placed on different server drives or file systems. This strategy aids in the reduction of I/O and rollback segment contention.

Because all changes are logged to a rollback segment, they can become a performance bottleneck as the number of concurrent users and transaction activity increases. You can tune rollback segment performance by creating additional ones to handle increased traffic. To determine whether rollback segment performance is satisfactory, you should monitor the ratio of gets to waits. If you see that the number of waits is climbing, create another rollback segment to further spread concurrency.

**NOTE:** The analysis for determining when to add a rollback segment can be quite complicated. As a rule-of-thumb, however, many DBAs simply plan on allocating a rollback segment for every four concurrent users.

The proper sizing of rollback segments is critical to their overall performance. Performance degrades whenever a rollback segment must extend, wrap or shrink in response to transaction loads. Ideally, you want to make their extents as small as possible while ensuring that each transaction fits into a single extent, which in practice is very difficult to achieve.

**NOTE:** You should carefully monitor each rollback segments for the occurrence of extends, wraps and shrinks. If you see a high level of extends and wraps, then the rollback segment is probably too small. If you see a high level of shrinks, then you should increase the rollback segment's optimal size.
Datasource Explorer List
When you select the Rollback Segments node of the Datasource Explorer, DBArtisan lists the rollback segments defined in a database, displaying pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about rollback segments from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rollback Segment Name</td>
<td>Name of the Rollback Segment</td>
</tr>
<tr>
<td>Owner</td>
<td></td>
</tr>
<tr>
<td>Tablespaces</td>
<td>Name of the tablespace where the rollback segment is placed</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates if the rollback segment is on-line or off-line</td>
</tr>
<tr>
<td>Gets</td>
<td>Number of rollback segment header requests. *It is an indicator of activity levels on the rollback segment.</td>
</tr>
<tr>
<td>Waits</td>
<td>Number of rollback segment header requests that resulted in waits. It is an indicator of concurrency. *If you see a significant number of waits, then you should consider adding rollback segments to spread concurrency.</td>
</tr>
<tr>
<td>Extents</td>
<td>A logical unit of database storage space allocation that comprises a number of contiguous data blocks. Oracle allocates a new extent for the segment when the existing is completely used.</td>
</tr>
<tr>
<td>Size</td>
<td>The space in bytes allocated for the rollback segment</td>
</tr>
<tr>
<td>Bytes Written</td>
<td>Current datafile size in bytes</td>
</tr>
<tr>
<td>Current Xacts</td>
<td></td>
</tr>
<tr>
<td>Shrinks</td>
<td>The cumulative number of times Oracle has truncated extents from the rollback segment</td>
</tr>
<tr>
<td>Optimal Size</td>
<td>The OPTIMAL size in bytes of the rollback segment</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>Shrink</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Rules
NOTE: This object is supported by Microsoft SQL Server and Sybase only.

Rules promote data integrity by allowing you to validate the values supplied to a table column. They are reusable objects that you can bind to table columns or user datatypes. For example, you can create a rule, bind it to a column in a table and have it specify acceptable values that can be inserted into that column.

Datasource Explorer List
When you select the Rules node of the Datasource Explorer tree, DBArtisan lists the rules defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about rules from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>User who owns the rule.</td>
</tr>
<tr>
<td>Rule Name</td>
<td>Name of the rule.</td>
</tr>
</tbody>
</table>
Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename</td>
<td>Report</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Segments

**NOTE:** This object is supported by Sybase only.

Segments are a mechanism for placing tables and indexes on specific logical partitions. You create segments on one or more fragments of a database. You can map segments to specific database fragments, which in turn reside on specific hard disks; and, mapping segments lets you increase i/o throughput by placing intensively used tables and indexes on different physical devices. You can allocate tables and indexes to segments by including placement statements at the end of CREATE TABLE or CREATE INDEX statements.

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sequences

Sequences are programmable database objects that generate a definable sequence of values. Once defined, a sequence can be made available to many users. A sequence can be accessed and incremented by multiple users with no waiting. A sequence can be used to automatically generate primary key values for tables. When you create a sequence, you can define its initial value, increment interval and maximum value.

Datasource Explorer List
When you select the Sequences node of the Datasource Explorer, DBArtisan lists the sequences defined in a database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about sequences from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Owner</td>
<td>User who owns the sequence</td>
</tr>
<tr>
<td>Sequence Name</td>
<td>Name of the Sequence</td>
</tr>
<tr>
<td>Last Number</td>
<td>Last value generated by the sequence</td>
</tr>
<tr>
<td>Max Value</td>
<td>Maximum allowable value of the sequence. Unless the sequence is allowed to cycle, it stops generating values once it reaches the maximum value.</td>
</tr>
<tr>
<td>Created</td>
<td>Date and time the sequence was created</td>
</tr>
</tbody>
</table>
Available Functionality

DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Create Alias</th>
<th>Create Synonym</th>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate</td>
<td>New</td>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
</tr>
<tr>
<td>Restart</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Snapshots

NOTE: This object is supported by Oracle only.

Snapshots are used to dynamically copy data between distributed databases. There are two types of snapshots:

- Complex
- Simple

NOTE: Snapshots are available in Oracle 8.0 and earlier. For Oracle 8i or later, see Materialized Views.

Complex snapshots copy part of a master table or data from more than one master table. Simple snapshots directly copy a single table. You cannot directly update the underlying data contained in snapshots.

NOTE: To use snapshots, use the Oracle 7 Distributed Option.

Available Functionality

DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Create Synonym</th>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rename</td>
<td>Report</td>
<td></td>
</tr>
</tbody>
</table>

Snapshot Logs

NOTE: This object is supported by Oracle only.

Snapshot logs are tables that maintain a history of modifications to the master and they are used to refresh simple snapshots. When you create a snapshot log, Oracle automatically creates a log table to track data changes in the master table and a log trigger to maintain the data in the log table.

NOTE: Snapshot Logs are available in Oracle 8.0 or earlier. For Oracle 8i or later, see Materialized View Logs.

Available Functionality

DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Stogroups

NOTE: This object is supported by IBM DB2 for OS/390 and z/OS only.

Stogroups are storage groups, named sets of volumes on which DB2 for OS/390 data is stored.

When you select the Stogroups node of the Datasource Explorer, DBArtisan lists pertinent information about each of them in the right pane.

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Structured Types

NOTE: This object is supported by IBM DB2 for Linux, Unix, and Windows and IBM DB2 for OS/390 and z/OS only.

Structured types define an abstract data type or object composed of a collection of similar types of data. For example, create an structured type that defines a full address rather than the pieces of an address, such as city, state and postal code. An structured type stores the pieces of an address in a single type, storing them in the same location and allowing the full address to be accessed and manipulated as single unit rather than multiple units.

Structured types are useful for ensuring uniformity and consistency as they are defined as single encapsulated entity that can be reused in other structured types and objects. They also offer flexibility by allowing for the creation of objects that represent real-world situations which is limited in relational objects.

Datasource Explorer List
When you select the Types node of the Datasource Explorer, DBArtisan lists the types defined in the database, displaying pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about types from the system catalog.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type Owner</td>
<td>User who owns the structured type</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the structured type</td>
</tr>
<tr>
<td>Status</td>
<td>Whether the structured type is valid or invalid</td>
</tr>
<tr>
<td>Created</td>
<td>Date and time the structured type was created</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Date and time the structured type was last modified</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>New</th>
<th>Report</th>
</tr>
</thead>
</table>
Summary Tables

NOTE: This object is supported by IBM DB2 for Linux, Unix, and Windows version 7.

A summary table is a table whose definition is based on the result of a query. The summary table typically contains pre-computed results based on the data existing in the table or tables that its definition is based on. If the SQL compiler determines that a query will run more efficiently against a summary table than the base table or tables, the query quickly executes against the materialized query table.

Datasource Explorer List
When you select the SummaryTables node of the Datasource Explorer, DBArtisan lists the summary tables defined in a database and displays pertinent information about each of them in the right pane.

The table below describes the information DBArtisan derives and displays about summary tables from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>User who owns the table</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the table</td>
</tr>
<tr>
<td># Rows</td>
<td>Number of rows of data the table includes</td>
</tr>
<tr>
<td>% Free</td>
<td>The amount of remaining space allocated to the table</td>
</tr>
<tr>
<td>Tablespace</td>
<td>The name of the tablespace on which the table resides</td>
</tr>
<tr>
<td>Index TS</td>
<td>The name of the tablespace on which the table's indexes reside</td>
</tr>
<tr>
<td>Long TS</td>
<td>The name of the tablespace on which the long objects (LOB,CLOB) stored on the table reside</td>
</tr>
<tr>
<td>Refresh Mode</td>
<td>Either Deferred or Immediate</td>
</tr>
<tr>
<td>Refresh Time</td>
<td>Last time table was refreshed</td>
</tr>
<tr>
<td>Create Time</td>
<td>When the table was created</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Compile</th>
<th>Create Synonym</th>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
<td></td>
</tr>
</tbody>
</table>

Synonyms

NOTE: This object is supported by IBM DB2 for OS/390 and z/OS and Oracle only.

Synonyms are an alternate name for an object in the database. Depending on the platform, you can define synonyms on tables, views, sequences, procedures, functions, packages and materialized views. There are two types of synonyms:

- Public
- Private

Public synonyms can be used by everyone in the database. Private synonyms belong to the synonym owner.
Synonyms simplify object names and let you:

- Reference an object without needing to specify its owner.
- Reference a remote object without needing to specify its database.
- Alias an object so its purpose becomes more understandable.

**Datasource Explorer List**

When you select the Synonyms node of the Datasource Explorer, DBArtisan lists the synonyms defined in a database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about synonyms from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonym Owner</td>
<td>User who owns the synonym</td>
</tr>
<tr>
<td>Synonym Name</td>
<td>Name of the synonym</td>
</tr>
<tr>
<td>Base Object</td>
<td>Base object referenced by the synonym</td>
</tr>
<tr>
<td>Created</td>
<td>Date and time the synonym was created</td>
</tr>
<tr>
<td>Creator</td>
<td>Name of the owner of the synonym</td>
</tr>
<tr>
<td>Created By</td>
<td>Name of the user who created the synonym</td>
</tr>
<tr>
<td>Object Name</td>
<td>Indicates the name of the object</td>
</tr>
<tr>
<td>Object Creator</td>
<td>Name of the owner of the object</td>
</tr>
</tbody>
</table>

**Available Functionality**

DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Tables**

*NOTE:* This object is supported by all platforms.

Tables are the basic unit of data storage. Tables store all the data accessible to users in rows and columns. Each column has a name, datatype and other associated properties. After you define a table, users can insert valid data into the table, which you can later query, update and delete.

*NOTE:* DBArtisan separates system tables from user-defined tables in the Datasource Explorer to ensure that system tables are not accidentally altered or dropped.

**Datasource Explorer List**

When you select the Tables node of the Datasource Explorer, DBArtisan lists the tables defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about tables from the system catalog:
### IBM DB2 for Windows/Unix

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>User who owns the table</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the table</td>
</tr>
<tr>
<td># Rows</td>
<td>Number of rows of data the table includes</td>
</tr>
<tr>
<td>% Free</td>
<td>The amount of remaining space allocated to the table</td>
</tr>
<tr>
<td>Tablespace</td>
<td>The name of the tablespace on which the table resides</td>
</tr>
<tr>
<td>Index TS</td>
<td>The name of the tablespace on which the table’s indexes reside</td>
</tr>
<tr>
<td>Long TS</td>
<td>The name of the tablespace on which the long objects (LOB,CLOB) stored on the table reside</td>
</tr>
<tr>
<td>Type</td>
<td>Object type</td>
</tr>
<tr>
<td>Created</td>
<td>When the table was created</td>
</tr>
<tr>
<td>Last Runstats</td>
<td>When the script was last run</td>
</tr>
</tbody>
</table>

### Microsoft SQL Server

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>User who owns the table</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the table</td>
</tr>
<tr>
<td># Rows</td>
<td>Number of rows of data the table includes</td>
</tr>
<tr>
<td>File Group</td>
<td>The file group on which the table is placed</td>
</tr>
<tr>
<td>Created</td>
<td>When the table was created</td>
</tr>
</tbody>
</table>

### Oracle

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>User who owns the table</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the table</td>
</tr>
<tr>
<td># Rows</td>
<td>Number of rows of data the table includes</td>
</tr>
<tr>
<td>% Free</td>
<td>The percentage of space reserved on each data block for update operations</td>
</tr>
<tr>
<td>% Used</td>
<td>The minimum percentage of space that must be filled on each data block before another data block is allocated to the table</td>
</tr>
<tr>
<td>Tablespace</td>
<td>Name of the tablespace on which the table resides</td>
</tr>
<tr>
<td>Created</td>
<td>When the table was created</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Oracle 7 or later When the table was last changed</td>
</tr>
<tr>
<td>Last Analyzed</td>
<td>Oracle 8 or later When the table was last analyzed</td>
</tr>
</tbody>
</table>
Available Functionality

DBArtisan offers the following functionality for this object:

**NOTE:** Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Activate Logging</th>
<th>Allocate Extent</th>
<th>Analyze</th>
<th>Build Query</th>
<th>Clone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convert to Partitioned</td>
<td>Copy Name</td>
<td>Create Alias</td>
<td>Create Insert Statements</td>
<td>Create Like</td>
</tr>
<tr>
<td>Create Synonym</td>
<td>Create View</td>
<td>DBCC</td>
<td>Deallocate Unused Space</td>
<td>Disable Triggers</td>
</tr>
<tr>
<td>Enable Triggers</td>
<td>Drop</td>
<td>Edit Data</td>
<td>Estimate Size</td>
<td>Extract</td>
</tr>
<tr>
<td>Indexes</td>
<td>Lock</td>
<td>Migrate</td>
<td>New</td>
<td>Open</td>
</tr>
<tr>
<td>Place</td>
<td>Quiesce Tablespaces</td>
<td>Recompile</td>
<td>Rename</td>
<td>Reorganize</td>
</tr>
<tr>
<td>Report</td>
<td>Schema</td>
<td>Select * From</td>
<td>Set Integrity</td>
<td>Triggers</td>
</tr>
<tr>
<td>Truncate</td>
<td>Update Statistics</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tablespaces

**NOTE:** This object is supported by IBM DB2 for OS/390 and z/OS, IBM DB2 for Linux, Unix, and Windows, and Oracle only.

Tablespaces are storage structures that act as partitions for the database. You can create a tablespace to store table data and other objects related to table performance such as indexes or large object data. Tablespaces are used to manage large complex databases. Once you have created a tablespace, you can place objects on it.

**TIP:** Create separate tablespaces for your tables and indexes and put each tablespace on a different drive or file system. Segmenting tables and their corresponding indexes in this manner helps eliminate I/O contention at the server level.

**NOTE:** IBM DB2 for Linux, Unix, and Windows lets you assign a location for table or index data directly to physical storage devices. Each tablespace can also be broken down into a collection of containers which are the actual physical storage files or devices. You can then spread the data or database objects across multiple file systems, thereby giving you the necessary space for objects that require it.

Once you have created a tablespace, you can place individual tables and indexes on it. Because tablespaces map to physical drives, you can improve I/O performance by placing tables and their indexes on physically separated table spaces.

**Datasource Explorer List**

When you select the Tablespaces node on the Datasource Explorer, DBArtisan lists the tablespaces defined in the database, displaying pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about triggers from the system catalog:

**NOTE:** Columns differ by platform.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace Name</td>
<td>Name of the tablespace</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the tablespace</td>
</tr>
<tr>
<td>Size</td>
<td>Total size (MB) and data blocks allocated to a tablespace.</td>
</tr>
<tr>
<td>Used</td>
<td>Total space used and used data blocks in a tablespace</td>
</tr>
<tr>
<td>Free</td>
<td>Total unused space and unused data blocks in a tablespace</td>
</tr>
<tr>
<td>Percent Free</td>
<td>Minimum percentage of free space in a block</td>
</tr>
<tr>
<td>Data</td>
<td>The type of data stored on the table space</td>
</tr>
<tr>
<td>Type</td>
<td>Whether the table space is system manage or database managed</td>
</tr>
<tr>
<td>Extent</td>
<td>The number of extents assigned to the tablespace. Extents are issued in 4K pages.</td>
</tr>
<tr>
<td>Prefetch</td>
<td>The number of pages to be fetched prior to being referenced</td>
</tr>
<tr>
<td>Overhead</td>
<td>The estimated time of the waiting period to initialize a new I/O operation</td>
</tr>
<tr>
<td>Transfer</td>
<td>The estimated time required to read one extent</td>
</tr>
<tr>
<td>Database</td>
<td>Name of the database selected for the table space</td>
</tr>
<tr>
<td>Creator</td>
<td>Indicates who created the tablespace</td>
</tr>
<tr>
<td># Partitions</td>
<td>Indicates the number of partitioned table spaces</td>
</tr>
<tr>
<td>Layout</td>
<td>Type of table space layout</td>
</tr>
</tbody>
</table>
Available Functionality

DBArtisan offers the following functionality for this object:

NOTE: Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Change Status</th>
<th>Check Data</th>
<th>Check Index</th>
<th>Coalesce</th>
<th>Copy to Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Extract</td>
<td>Filter</td>
<td>Image Copy</td>
<td>Migrate</td>
</tr>
<tr>
<td>Modify Statistics</td>
<td>New</td>
<td>Open</td>
<td>Quiesce (IBM DB2 for Linux, Unix, and Windows)</td>
<td>Rebuild</td>
</tr>
<tr>
<td>Rename</td>
<td>Reorganize</td>
<td>Report</td>
<td>Runstats Index</td>
<td>Runstats Tablespace</td>
</tr>
<tr>
<td>Start Database</td>
<td>Stop Database</td>
<td>Switch Online</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Triggers

NOTE: This object is supported by all platforms.

Triggers are a special type of procedure that automatically fire when defined data modification operations (insert, update or delete) occur on a target table. Triggers fire after an insert, update or delete, but belong to the same transaction as the data modification operation.

Because triggers fire automatically and are part of the same transaction as the initiating SQL statement, they are often used to enforce referential integrity in a database. For example, an insert trigger can validate that a foreign key value exists in the primary key of the parent table. Similarly, you can write a trigger to cascade the deletion or update of data in a parent table to all corresponding data rows in child tables.

While triggers are a powerful data validation mechanism, they do complicate the task of debugging and tuning SQL. Because triggers fire inside transactions, they hold open transactions, contributing to concurrency problems. Accordingly, you should move all non-critical data validation outside of triggers to shorten transaction lengths to their minimum.

Datasource Explorer List

When you select the Triggers node on the Datasource Explorer, DBArtisan lists the triggers defined in the database, displaying pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about triggers from the system catalog:

NOTE: Columns differ by platform.
Available Functionality

DBArtisan offers the following functionality for this object:

**NOTE:** Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Compile</th>
<th>Disable/Enable</th>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate</td>
<td>New</td>
<td>Open</td>
<td>Rename</td>
<td>Report</td>
</tr>
</tbody>
</table>

Types

**NOTE:** This object is supported by Oracle only.

Types define an abstract data type or object composed of a collection of similar types of data. For example, create an object type that defines a full address rather than the pieces of an address, such as city, state and postal code. An object type stores the pieces of an address in a single type, storing them in the same location and allowing the full address to be accessed and manipulated as a single unit rather than multiple units.

Object types are useful for ensuring uniformity and consistency as they are defined as single encapsulated entity that can be reused in other object types and objects. They also offer flexibility by allowing for the creation of objects that represent real-world situations which is limited in relational objects.
You can choose to create a type that is incomplete, complete, a VARRAY, or a nested table or any combination of the above. An incomplete type specifies no attributes and can be used for circular references such as person - female. It lets the type be referenced before it is complete. The VARRAY type can be used to store small sets of related data. For example, if you have ten offices (each one with a different description) at a particular division in your company, you could create a VARRAY of 10 to hold the details of these offices. The values for a VARRAY type must be fixed and known and small values as they are stored in RAW format. A nested table type can be used when data is repeated for the same entity an unknown number of times and storage is a concern.

Datasource Explorer List
When you select the Types node of the Datasource Explorer, DBArtisan lists the types defined in the database, displaying pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about types from the system catalog.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type Owner</td>
<td>User who owns the object type</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the object type</td>
</tr>
<tr>
<td>Status</td>
<td>Whether the object type is valid or invalid</td>
</tr>
<tr>
<td>Created</td>
<td>Date and time the object type was created</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Date and time the object type was last modified</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Compile</th>
<th>Drop</th>
<th>Extract</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Type Bodies

**NOTE:** This object is supported by Oracle only.

Type Bodies implement object type specification by containing the definition of every cursor and subprogram declared in the object type specification. While Type Bodies are listed as a separate object in the Datasource Explorer, they are created on the Types editor in conjunction with Types.

Available Functionality
DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Compile</th>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Report</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unique Keys

**NOTE:** This object is supported by all platforms.

Unique keys can enforce logical keys that are not chosen as the primary key. They enforce uniqueness for specified columns in a table.
Datasource Explorer List
When you select the Unique Keys node of the Datasource Explorer, DBArtisan lists the unique keys defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about unique keys from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique Index Name</td>
<td>Name of the unique index</td>
</tr>
<tr>
<td>Index Owner</td>
<td>Name of the index owner</td>
</tr>
<tr>
<td>Table Name</td>
<td>Name of the indexed table</td>
</tr>
<tr>
<td>Table Owner</td>
<td>User who owns the indexed table</td>
</tr>
<tr>
<td>Database</td>
<td>Name of the database</td>
</tr>
<tr>
<td>Bufferpool</td>
<td>Name of the Bufferpool used for the index</td>
</tr>
<tr>
<td>Last RUNSTATS</td>
<td>Indicates the time of the last RUNSTATS</td>
</tr>
<tr>
<td>Levels</td>
<td>The number of levels of the index's B-Tree. If this value is greater than 3, then you should consider rebuilding the index.</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

**NOTE:** Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
<th>Rename</th>
<th>Report</th>
<th>Status</th>
</tr>
</thead>
</table>

User Datatypes

**NOTE:** This object is supported by IBM DB2 for Linux, Unix, and Windows, IBM DB2 for OS/390 and z/OS, Microsoft SQL Server, and Sybase only.

User-defined datatypes promote domain consistency by streamlining the definition of commonly used table columns in a database. You can build a customized datatype from system datatypes and bind defaults and rules to it to enhance integrity. When you reference the user datatype in a column, the column assumes all of the properties of the user datatype.

Datasource Explorer List
When you select the User Datatypes node of the Datasource Explorer, DBArtisan lists the user datatypes defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about user datatypes from the system catalog:

**NOTE:** Columns differ by platform.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datatype Owner</td>
<td>User who owns the datatype</td>
</tr>
<tr>
<td>Datatype Name</td>
<td>Name of the user datatype</td>
</tr>
</tbody>
</table>
Available Functionality

DBArtisan offers the following functionality for this object:

NOTE: Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Type</td>
<td>Base datatype parameters for the user datatype</td>
</tr>
<tr>
<td>Bound Rule</td>
<td>The rule, if any, bound to the user datatype</td>
</tr>
<tr>
<td>Bound Default</td>
<td>The default, if any, bound to the user datatype</td>
</tr>
<tr>
<td>Length</td>
<td>The length of the datatype</td>
</tr>
<tr>
<td>Remark</td>
<td>Any comments that pertain to the datatype</td>
</tr>
<tr>
<td>Schema</td>
<td>Schema of the datatype</td>
</tr>
<tr>
<td>Created By</td>
<td>Who created the user datatype</td>
</tr>
<tr>
<td>Source Schema</td>
<td>Schema of the source data type</td>
</tr>
<tr>
<td>Source Type</td>
<td>Source datatype parameters for the user datatype</td>
</tr>
<tr>
<td>Datatype Owner</td>
<td>User who owns the datatype</td>
</tr>
<tr>
<td>Length</td>
<td>The length of the datatype</td>
</tr>
<tr>
<td>Remarks</td>
<td>Any comments that pertain to the datatype</td>
</tr>
</tbody>
</table>

User Messages

NOTE: This object is supported by IBM DB2 for Windows/Unix, Microsoft SQL Server, and Sybase only.

User Messages lets you catalog error messages that your database applications can re-use. Microsoft SQL Server stores your error messages in a system table, sysmessages. To return error messages from stored procedures and triggers, you need to call a system stored procedure and pass an error number as a parameter.

NOTE: User Messages can only be accessed from the master database for a specific server.

Datasource Explorer List

When you select the User Messages node located under the master database of the Datasource Explorer tree, DBArtisan lists the user messages defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about user messages from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Msg#</td>
<td>Number assigned to the message</td>
</tr>
<tr>
<td>Language</td>
<td>Language in which the user message is written</td>
</tr>
<tr>
<td>Severity</td>
<td>Level of severity assigned to the user message</td>
</tr>
<tr>
<td>Description</td>
<td>Error text description associated with the user message</td>
</tr>
</tbody>
</table>
Available Functionality

DBArtisan offers the following functionality for this object:

<table>
<thead>
<tr>
<th>Drop</th>
<th>Extract</th>
<th>Migrate</th>
<th>New</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Users

**NOTE:** This object is supported by all platforms.

A user is an individual with access to the DBMS.

For more information, see:

- IBM DB2 for Linux, Unix, and Windows Users
- IBM DB2 for OS/390 and z/OS Users
- Microsoft SQL Server Users
- MySQL Users
- Oracle Users
- Sybase ASE Users
- Available Functionality

IBM DB2 for Linux, Unix, and Windows Users

IBM DB2 for Linux, Unix, and Windows relies on your machine’s operating system to define users for the server. You create and maintain users through your operating system.

**NOTE:** DBArtisan lets you set a default SQLID user for a DB2 datasource. This lets the user set the implicit schema for the unqualified object references to a schema different from the user’s login id.

Datasource Explorer List

When you select the Users node of the Datasource Explorer, DBArtisan lists the users defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about users from the system catalog:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>Name of the user</td>
</tr>
<tr>
<td>Connect</td>
<td>Indicates if the user has connect privileges</td>
</tr>
<tr>
<td>Create Table</td>
<td>Indicates if the user has create table privileges</td>
</tr>
<tr>
<td>Create package</td>
<td>Indicates if the user has create package privileges</td>
</tr>
<tr>
<td>Register Function</td>
<td>Indicates if the user can register functions</td>
</tr>
<tr>
<td>Admin</td>
<td>Indicates if the user has admin privileges</td>
</tr>
<tr>
<td>Create Schemas Implicitly</td>
<td>Indicates if the user can create objects on which schemas are created implicitly</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.
IBM DB2 for OS/390 and z/OS Users

IBM DB2 relies on the operating system you are running to define users for the server. This means that users are created and maintained through your operating system.

**NOTE:** DBArtisan lets you set a default SCHEMA user for a DB2 datasource. This lets the user set the implicit schema for the unqualified object references to a schema different from the user's login id.

Datasource Explorer List

When you select the Users node of the Datasource Explorer, DBArtisan lists the users defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about users from the system catalog:

<table>
<thead>
<tr>
<th>IBM DB2 for OS/390 and z/OS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Name</td>
<td>Description</td>
</tr>
<tr>
<td>User Name</td>
<td>Name of the user</td>
</tr>
<tr>
<td>System</td>
<td>Aggregate number of system privileges</td>
</tr>
<tr>
<td>Database</td>
<td>Indicates the aggregate number of privileges for the database</td>
</tr>
<tr>
<td>Plan</td>
<td>Indicates the aggregate number of privileges for the plan</td>
</tr>
<tr>
<td>Packages</td>
<td>Indicates the aggregate number of privileges for the packages</td>
</tr>
<tr>
<td>Table</td>
<td>Indicates the aggregate number of table privileges</td>
</tr>
<tr>
<td>Column</td>
<td>Indicates the aggregate number or column privileges</td>
</tr>
<tr>
<td>Schema</td>
<td>Indicates the aggregate number of privileges for the schema</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.

Microsoft SQL Server Users

Microsoft SQL Server controls access at the database level by requiring the System Administrator or Database Owner to add a login as a database user or alias. After you create a database user, you can implement further security by the granting or revoking the privileges for that user on specific database objects. To consolidate the process of granting or revoking permissions to many users, the database owner can assign users to groups.

Datasource Explorer List

When you select the Users node of the Datasource Explorer tree, DBArtisan lists the users defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about users from the system catalog:

<table>
<thead>
<tr>
<th>Microsoft SQL Server</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Name</td>
<td>Description</td>
</tr>
<tr>
<td>User Name</td>
<td>Name of the User</td>
</tr>
<tr>
<td>Login Name</td>
<td>Name of the login that maps to the user</td>
</tr>
<tr>
<td>Created</td>
<td>When the user was created</td>
</tr>
<tr>
<td>Updated</td>
<td>When the User was last updated</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.
Oracle Users
To access an Oracle database, you need a user account authenticated with a password. A user account is what Oracle uses to permit access by the user. You can assign the following optional properties to the user:

- Default tablespace
- Temporary tablespace
- Quotas for allocating space in tablespaces
- Profile containing resource limits

Datasource Explorer List
When you select the Users node of the Datasource Explorer, DBArtisan lists the users defined in the database, displaying pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about users from the system catalog.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>Name of the User</td>
</tr>
<tr>
<td>Default Tablespace</td>
<td>Name of the default tablespace assigned to the user</td>
</tr>
<tr>
<td>Temp Tablespace</td>
<td>Name of the temporary tablespace the user is placed on</td>
</tr>
<tr>
<td>Profile</td>
<td>Profile assigned to the user to govern resource utilization</td>
</tr>
<tr>
<td>Create Date</td>
<td>Date and time the user was created</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.

Sybase ASE Users
Sybase ASE controls access at the database level by requiring the System Administrator or Database Owner to add a login as a database user or alias. After you create a database user, you can implement further security by granting or revoking the privileges for that user on specific database objects. To consolidate the process of granting or revoking permissions to many users, the database owner can assign users to groups.

The Datasource Explorer List
When you select the Users node of the Datasource Explorer tree, DBArtisan lists the users defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about users from the system catalog.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>Name of the user</td>
</tr>
<tr>
<td>Login Name</td>
<td>Name of the login that maps to the user</td>
</tr>
<tr>
<td>Group Name</td>
<td>Name of the group to which the user belongs</td>
</tr>
</tbody>
</table>

For more information, see Available Functionality.
Available Functionality for Users
DBArtisan offers the following functionality for users:

**NOTE:** Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Change Password</th>
<th>Drop</th>
<th>Extract</th>
<th>Filter</th>
<th>Migrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Open</td>
<td>Report</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Views

**NOTE:** This object is supported by all platforms.

Views are SQL queries stored in the system catalog that customize the display of data contained in one or more tables. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables. Views let you:

- View a customized selection of data from one or more tables. As a result, you can display data more cogently to different sets of users, even though the underlying data is the same.
- Restricting access to a defined set of rows and columns.

Datasource Explorer List

When you select the Views node of the Datasource Explorer, DBArtisan lists the views defined in the database and displays pertinent information about each of them in the right pane. The table below describes the information DBArtisan derives and displays about views from the system catalog:

**NOTE:** Columns differ by platform.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Owner</td>
<td>User who owns the view</td>
</tr>
<tr>
<td>View Name</td>
<td>Name of the view</td>
</tr>
<tr>
<td>Created</td>
<td>Date and time the view was created</td>
</tr>
<tr>
<td>Owner</td>
<td>User who owns the view</td>
</tr>
<tr>
<td>Created By</td>
<td>Who created the view</td>
</tr>
<tr>
<td>Status</td>
<td>Whether the view is valid or invalid</td>
</tr>
<tr>
<td>Created</td>
<td>Date and time the view was created</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Date and time the view was last modified</td>
</tr>
</tbody>
</table>

Available Functionality
DBArtisan offers the following functionality for this object:

**NOTE:** Functionalities differ by platform.

<table>
<thead>
<tr>
<th>Build Query</th>
<th>Compile</th>
<th>Copy Name</th>
<th>Create Alias</th>
<th>Create Synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Filter</td>
<td>Extract</td>
<td>New</td>
<td>Open</td>
</tr>
<tr>
<td>Rename</td>
<td>Report</td>
<td>Select * From</td>
<td>Updating statistics for views (IBM DB2 for Windows, Unix, and Linux)</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------</td>
<td></td>
</tr>
</tbody>
</table>
Object Editors

An Object editor is a tabbed dialog box that stores information about existing server and database types and objects, and lets you modify those items. For an introduction to object editors, see Overview and common usage of Object editors.

**NOTE:** Availability of editors for specific object type varies from DBMS to DBMS. Similarly, functionality offered for an editor for an object type common to two or more DBMS, will differ from DBMS to DBMS.

The following tables lists the type of objects for which an object editor is supported, refers you to detailed instructions on that editor, and lists the DBMS for which that editor is available.

<table>
<thead>
<tr>
<th>Object Type</th>
<th>For more information:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>Using the Aliases Editor (IBM DB2, Sybase)</td>
</tr>
<tr>
<td>Backup Devices</td>
<td>Using the Backup Devices/Dump Devices editor (Microsoft SQL Server)</td>
</tr>
<tr>
<td>Bufferpools</td>
<td>Using the Bufferpools Editor (IBM DB2)</td>
</tr>
<tr>
<td>Check Constraints</td>
<td>Using the Check Constraints editor</td>
</tr>
<tr>
<td>Clusters</td>
<td>Using the Clusters Editor (Oracle)</td>
</tr>
<tr>
<td>Data Caches</td>
<td>Using the Data Caches Editor (Sybase)</td>
</tr>
<tr>
<td>Database Devices</td>
<td>Using the Database Devices Editor (Sybase)</td>
</tr>
<tr>
<td>Database Links</td>
<td>Using the Database Links editor (Oracle)</td>
</tr>
<tr>
<td>Databases</td>
<td>Using the Databases editor (IBM DB2, Microsoft SQL Server, Sybase ASE)</td>
</tr>
<tr>
<td>DBRMs</td>
<td>Using the DBRM Editor (IBM DB2 for OS/390 and z/OS)</td>
</tr>
<tr>
<td>Defaults</td>
<td>Using the Defaults Editor (Microsoft SQL Server, Sybase)</td>
</tr>
<tr>
<td>Directories</td>
<td>Using the Directories Editor (Oracle)</td>
</tr>
<tr>
<td>Event Monitors</td>
<td>Using the Event Monitors editor (IBM DB2 for Windows, Unix and Linux)</td>
</tr>
<tr>
<td>Extended Procedures</td>
<td>Using the Extended Procedures Editor (Microsoft SQL Server, Sybase)</td>
</tr>
<tr>
<td>Foreign Keys</td>
<td>Using the Foreign Keys editor</td>
</tr>
<tr>
<td>Functions</td>
<td>Using the Functions editor (IBM DB2, Microsoft SQL Server, Oracle)</td>
</tr>
<tr>
<td>Groups</td>
<td>Using the Groups editor (IBM DB2 for Windows, Unix, and Linux, Oracle, Sybase)</td>
</tr>
<tr>
<td>Indexes</td>
<td>Using the Indexes editor</td>
</tr>
<tr>
<td>Job Queue</td>
<td>Using the Job Queue Editor (Oracle)</td>
</tr>
<tr>
<td>Libraries</td>
<td>Using the Libraries Editor (Oracle)</td>
</tr>
<tr>
<td>Linked Servers</td>
<td>Using the Linked Servers Editor (Microsoft SQL Server)</td>
</tr>
<tr>
<td>Logins</td>
<td>Using the Logins Editor (Microsoft SQL Server, Sybase)</td>
</tr>
<tr>
<td>Materialized Query Tables</td>
<td>Using the Materialized Query Tables Editor (IBM DB2 for Linux, Unix, and Windows)</td>
</tr>
<tr>
<td>Materialized Views</td>
<td>Using the Materialized Views (Snapshot) Editor (Oracle)</td>
</tr>
<tr>
<td>Materialized View Logs</td>
<td>Using the Materialized View (Snapshot Logs) Logs editor (Oracle)</td>
</tr>
<tr>
<td>Nodegroups</td>
<td>Using the Nodegroups editor (IBM DB2)</td>
</tr>
<tr>
<td>Outlines</td>
<td>Outlines Editor for Oracle</td>
</tr>
<tr>
<td>Package Bodies</td>
<td>Using the Package Bodies Editor (Oracle)</td>
</tr>
<tr>
<td>Packages</td>
<td>Using the Packages editor (IBM DB2, Oracle)</td>
</tr>
</tbody>
</table>
Overview and common usage of Object editors

An Object editor lets you view and modify settings and properties of existing object types and servers on a datasource. It also lets you add new resources and provides access to related, datasource management facilities. For example, the Tables editor lets you add or insert, edit, or drop columns, work with permissions to work with that table, access information on physical storage and the distribution of data and indexes across table spaces, and so on.

Each tab on the Object editor lets you perform a logical task or collection of logical tasks for that object type. The Object editor toolbar has a set of commands common to all object types and also includes a **Command** menu with commands specific to the object type you are currently working with. For example:

<table>
<thead>
<tr>
<th>Object Type</th>
<th>For more information:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plans</td>
<td>Using the Plans Editor (IBM DB2 for OS/390 and z/OS)</td>
</tr>
<tr>
<td>Primary Keys</td>
<td>Using the Primary Keys editor</td>
</tr>
<tr>
<td>Procedures</td>
<td>Using the Procedures editor</td>
</tr>
<tr>
<td>Profiles</td>
<td>Using the Profiles Editor (Oracle)</td>
</tr>
<tr>
<td>Redo Log Groups</td>
<td>Redo Log Groups Editor (Oracle)</td>
</tr>
<tr>
<td>Remote Servers</td>
<td>Using the Remote Servers Editor (Microsoft SQL Server, Sybase)</td>
</tr>
<tr>
<td>Roles</td>
<td>Using the Roles editor (Microsoft SQL Server, Oracle, Sybase)</td>
</tr>
<tr>
<td>Rollback Segments</td>
<td>Using the Rollback Segments Editor (Oracle)</td>
</tr>
<tr>
<td>Rules</td>
<td>Using the Rules editor (Microsoft SQL Server, Sybase)</td>
</tr>
<tr>
<td>Schemas</td>
<td>Using the Schemas editor (Microsoft SQL Server 2005)</td>
</tr>
<tr>
<td>Segments</td>
<td>Using the Segments Editor (Sybase)</td>
</tr>
<tr>
<td>Sequences</td>
<td>Using the Sequences editor (IBM DB2 for Linux, Unix, and Windows, Oracle)</td>
</tr>
<tr>
<td>Stogroups</td>
<td>Using the Stogroups Editor (IBM DB2 for OS/390 and z/OS)</td>
</tr>
<tr>
<td>Structured Types</td>
<td>Using the Structured Types Editor (IBM DB2 for Linux, Unix, and Windows)</td>
</tr>
<tr>
<td>Synonyms</td>
<td>Using the Synonyms editor (IBM DB2 for OS/390 and z/OS, Oracle)</td>
</tr>
<tr>
<td>Tables</td>
<td>Using the Tables editor</td>
</tr>
<tr>
<td>Tablespaces</td>
<td>Using the Tablespaces editor (DB2, Oracle)</td>
</tr>
<tr>
<td>Triggers</td>
<td>Using the Triggers editor</td>
</tr>
<tr>
<td>Unique Keys</td>
<td>Using the Unique Keys editor</td>
</tr>
<tr>
<td>User Datatypes</td>
<td>Using the User Datatypes editor (IBM DB2, Microsoft SQL Server, Sybase)</td>
</tr>
<tr>
<td>User Messages</td>
<td>Using the User Messages Editor (Microsoft SQL Server, Sybase)</td>
</tr>
<tr>
<td>Users</td>
<td>Using the Users editor</td>
</tr>
<tr>
<td>View</td>
<td>Using the Views editor</td>
</tr>
</tbody>
</table>
In order to work with object editors, you must know how to:

- **Open an object editor** on a server or database object.
- **View and modify database object properties** using functionality common to all object editors as well as object-specific functionality available to specific object editors.
- **Preview and submit changes** to effect your changes to the datasource.

**Open an object editor**

Object editors are accessed from the Datasource Explorer pane of the DBArtisan main window.

**To open an Object editor on a specific database object**

1. Connect to the datasource that contains the object you want to modify. For more information, see [Datasource Management](#).

2. On the Datasource Explorer, expand the target datasource.

3. Continue to expand folders under the datasource until the type of object you want to view or modify is visible.
4 On the **Datasource Explorer**, click on the specific type of object that you want to view or modify.

All objects of that type on the datasource are displayed in the right pane of the explorer window.

5 In the right pane of the Explorer window, right-click the specific object you want to view or modify, and then select **Open**.

DBArtisan opens the Object editor for that specific object type.

With the Object editor open, you can **View and modify database object properties**.

### View and modify database object properties

There are two categories of tasks you can perform using an Object editor:

- **Work with tasks and settings of specific object types**
- **Work with tasks and settings common to Object editors**

#### Work with tasks and settings of specific object types

Many of the tasks you perform using an Object editor are specific to the object type you are working with. For example, the Triggers editor lets you work directly with SQL code for the trigger while the Tables editor lets you work with columns of the table. Object-specific tasks are documented in the topics for specific Object editors provided later in this chapter.

#### Work with tasks and settings common to Object editors

The following tasks can be performed against many of the support Object editors:
• **Use the Object editor toolbar**
• **Open another object in the same object editor explorer**
• **Add a comment to a database object**
• **View SQL/DDL for a database object**
• **Work with privileges and permissions**

**Use the Object editor toolbar**
The Object editor toolbar appears above the tabs of the Object editor.

```
Create    Alter    Drop...   Extract   Migrate   Report   Command   Refresh   Close
```

It provides the following functions:

- **Create**
  - Launches a DBArtisan object creation wizard. For more information, see [Creating an Object](#).

- **Alter**
  - Enabled when a property of the current database has been modified. Opens a dialog that lets you preview the SQL code that will effect the modification and then submit the change. For more information, see [Preview](#).

- **Drop**
  - Lets you drop one or more database objects and remove their definition from the system catalog. For more information, see [Drop](#).

- **Extract**
  - DBArtisan lets you extract data from one database to another database and extract the statements required to create objects into an Interactive SQL window. For more information, see [Extract](#).

- **Migrate**
  - Lets you move object schema between DBMS or servers by copying the DDL between the source and target databases. For details, see [Migrate](#).

- **Report**
  - Lets you generate detailed or summary reports on database objects. For details, see [Report](#).

- **Command Menu**
  - Provides menu commands specific to the object being viewed or modified. For a listing of commands available, see the topic for the specific Object editor, later in this chapter.

- **Refresh**
  - Refreshes or clears Object editor contents.

- **Close**
  - Closes the Object editor and if appropriate, prompts you to verify any changes.

Commands on the Object editor toolbar are disabled if they do not apply to the object type being viewed or modified.

**Open another object in the same object editor explorer**
When an object editor is open, the area between the object editor toolbar and the tabs has one or more dropdown lists. For example:

<table>
<thead>
<tr>
<th>Owner</th>
<th>Table</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN8910</td>
<td>EMP</td>
<td>PERSON</td>
</tr>
</tbody>
</table>

These lists allow you to qualify any object of the same type as the object currently displayed in the Object editor, and display information for that object in the current Object editor. The number of, and specific dropdown lists, differ according to the type of Object editor but there are always sufficient controls to uniquely identify another object of the same type.
To display the information for another object of the same type:

- Use the dropdown lists to specify a different object.

Add a comment to a database object
The object editors for certain object types feature a **Comment** tab that lets you add an explanatory note to specific object definitions. Comments are stored in the REMARKS column of the object’s system-catalog.

**To add a comment to an object:**
1. Open an object editor on an object type that permits comments. See the topics for specific Object editors later in this chapter for information on whether that object type supports comments.
2. Click the **Comment** tab.

   ![Comment tab](image)

   - **Comment:**

   ![Comment area](image)

   3. In the **Comment** area, type an explanatory note of up to 254 characters long.

   **NOTE:** Right-clicking in the Comment area opens a context menu that lets you perform edit text operations such as search, selection, copying and pasting, as well as coding-specific operations such as enabling/disabling line numbers and indenting lines.

   4. Preview and submit changes.

View SQL/DDL for a database object
The object editors for most object types feature a **DDL** tab that lets you view an object’s underlying SQL code.

**To view the underlying DDL/SQL for a database object:**
1. Open an object editor on an object type for which DDL code can be displayed. See the topics for specific Object editors later in this chapter for information on whether that object type supports display of underlying SQL code.
2. Click the **DDL** tab.

   **NOTE:** Right-clicking in the Comment area opens a context menu that lets you perform edit text operations such as search, selection, copying and pasting, as well as coding-specific operations such as enabling/disabling line numbers and indenting lines.

Work with privileges and permissions
When you open an Object editor on an object with associated privileges, the **Permissions** (or **Object Permissions** or **System Permissions**) tab for that editor displays the relevant privileges and lets you make changes accordingly:

For database objects such as tables or procedures, to which permissions or privileges are granted, a **Permissions** tab lets you manage permissions or privileges to that object from all grantees, such as users or groups, on that datasource. The tab has a navigation area, a command area, and a display area.
The Navigation area lets you change the content of the Display area to view more specific privilege details. For example, when viewing privileges for a database object such as a table, the dropdown lists in the navigation area, if present, let you drill down to populate the display area with privileges for a lower level component, such as a particular column. Similarly, when viewing privileges for a grantee such as a user, the Object Permissions tab’s Navigation area lets you populate the display area with that user’s permissions on specific object types, such as tables or procedures.

The Display area shows privilege details for an object or grantee. When viewing privileges for a grantee such as a user or role, the display area shows privileges for that recipient on objects currently qualified by the current selection in the Navigation area. When viewing privileges for an object such as a table, the display area shows the privileges for all grantees. Each cell in the Display area corresponds to a specific permissions and a cell representing a granted permission shows a distinctive icon.

The Command area lets you initiate granting or revocation of permissions.

To view or modify privileges using an Object editor:
1. Open an object editor on a database object with associated privileges or permissions.
2. Click the Privileges (or Object Privileges or System Privileges) tab.
3. Use the dropdown lists in the navigation area, if present, to populate the Display area with details for more specific or different permissions.
4. In the Display area, select a cell corresponding to a specific permission.
5 Use the Command area controls to **Grant Privilege(s) To** or **Revoke Privilege(s) From**, database objects/grantees.

6 **Preview and submit changes.**

## Work with object dependencies

For objects such as views and procedures, whose definition references other objects, such as tables, DBArtisan lets you view all dependencies. The Object editors for referencing or referenced objects have a **Dependencies** tab that shows all dependencies for an object.

### Dependency Name

<table>
<thead>
<tr>
<th>Referencing Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions</td>
</tr>
<tr>
<td>Packages</td>
</tr>
<tr>
<td>Procedures</td>
</tr>
<tr>
<td>Synonyms</td>
</tr>
<tr>
<td>Triggers</td>
</tr>
<tr>
<td>Views</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Referenced Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
</tr>
<tr>
<td>Tables</td>
</tr>
<tr>
<td>CE.INVENTORIES</td>
</tr>
<tr>
<td>CE.WAREHOUSES</td>
</tr>
<tr>
<td>Views</td>
</tr>
<tr>
<td>CE.PRODUCTS</td>
</tr>
</tbody>
</table>

In addition to letting you view dependencies, Object editors let you drop a referencing or referenced object, or open an Object editor on that object.

### To manage dependences for an object:

1 **Open an object editor** on a database object with referencing or referenced objects. See the object editor descriptions in subsequent topics to verify that an Object editor supports display of dependencies.

2 Click the **Dependencies** tab.

   The **Dependency Name** area lists all referenced or referencing objects. Objects are grouped according to their object type.

3 Optionally, select a referenced or referencing object in the right pane and either:
   - Use the **Edit** button to open an Object editor on that object
   - Use the **Drop** button to initiate dropping that object.

### Preview and submit changes

After you use an Object editor to modify the settings or properties of a database object, you can preview the SQL that will be executed to effect those changes on the datasource. You can then submit the SQL for execution on the server.
To preview and submit object editor changes to a database object:

1. Click the **Alter** button on the Object editor toolbar. For details, see Use the Object editor toolbar.

   A Preview: Alter dialog opens.

   ![Preview: Alter dialog]

2. Use one of the following options to submit and execute your changes:
   - **Execute** button - executes the changes immediately.
   - **Schedule** button - opens the Job Scheduler to let you schedule the change. For more information, see Scheduling.
   - **Batch** button - (IBM DB2 for OS/390 and Z/OS only) lets you specify the job cards used to execute the changes as a batch job. For information on how to check status, see Database Monitor.

Using the Aliases Editor (IBM DB2, Sybase)

The Aliases editor lets you modify settings and options of existing aliases.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- View the basic definition details of an Alias (IBM DB2, Sybase)

You also use this editor to perform the following common Object editor tasks:

- Add a comment to a database object (IBM DB2 for Windows, Unix, and Linux only)
- View SQL/DDL for a database object

View the basic definition details of an Alias (IBM DB2, Sybase)

The **Definition** tab of the Aliases Editor displays basic definition details for an alias:
• For IBM DB2, this includes the object type, owner, and name of the target of the alias
• For Sybase, this includes the name of the alias and the associated user

To view the definition for an alias:
1. Open an object editor on the alias.
2. Click the Definition tab.

Using the Backup Devices/Dump Devices editor (Microsoft SQL Server)

The Backup Devices/Dump Devices Editor lets you view and modify dump device definitions.

CAUTION: Before dumping a database or its transaction log, create a dump device on which to store the backup file. Microsoft SQL Server lets you create tape or disk dump devices.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

• View the definition for a Backup Device/Dump Device (Microsoft SQL Server)

You also use this editor to perform the following common Object editor tasks:

• View SQL/DDL for a database object

Lastly, the Command menu on the Object editor toolbar provides the following command:

• View Contents

View the definition for a Backup Device/Dump Device (Microsoft SQL Server)

The Definition tab lets you view general information for every backup or dump device on the current datasource.

To view basic definition details for a backup device/dump device:
1. Open an object editor on the backup/dump device.
2. Click the Definition tab.

The Device Type and Physical Name for the device are displayed, as well as the skip/noskip setting for the ANSI Labels option.

For more information, see Backup Devices/Dump Devices Editor for Microsoft SQL Server.
Using the Bufferpools Editor (IBM DB2)
The Bufferpools editor lets you modify settings and options of existing bufferpools.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage basic definition settings for a buffer pool (IBM DB2 for Windows, Unix and Linux)
- Manage nodegroups assigned to a buffer pool (IBM DB2 for Windows, Unix, and Linux)
- View buffer pool utilization details (IBM DB2 for OS/390 and z/OS)

You also use this editor to perform the following common Object editor tasks:

- Work with privileges and permissions
- View SQL/DDL for a database object

Manage basic definition settings for a buffer pool (IBM DB2 for Windows, Unix and Linux)
The Definition tab of the Buffer Pools Editor lets you pool storage and other details for a buffer pool.

1  Open an object editor on the buffer pool to be modified.
2  Click the Definition tab.
3  Use the following table as a guide to understanding and modifying the settings on this page:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage (IBM DB2 for Windows, Unix, and Linux)</td>
<td>For IBM DB2 for Windows, Unix, and Linux, lets you view the current Page Size, enable/disable Extended Storage, and specify the Number of Pages properties. For IBM DB2 for OS/390 and Z/OS, lets you set the Size (VPSIZE), Type (VPTYPE), Hiperpool Size (HPSIZE), Castout, Page Stealing Algorithm (PGSTEAL) properties.</td>
</tr>
<tr>
<td>Partitions (IBM DB2 for Windows, Unix, and Linux)</td>
<td>Lets you select a node and click Edit to provide a new partition page size.</td>
</tr>
<tr>
<td>Threshold (IBM DB2 for OS/390 and Z/OS)</td>
<td>Lets you set the Sequential Steal (VPSEQT), Parallel Sequential (VPPSEQT), Assisting Parallel Sequential (VPXSEQT), Hiperpool Sequential Steal Threshold (HPSEQT), Deferred Write Threshold (DWQT), and Vertical Deferred Write Threshold (DDWQT) properties.</td>
</tr>
</tbody>
</table>

4  Preview and submit changes.

Manage nodegroups assigned to a buffer pool (IBM DB2 for Windows, Unix, and Linux)
The Nodegroups tab of the Buffer Pools Editor lets you manage the nodegroups that are assigned to the buffer pool.

1  Open an object editor on the buffer pool to be modified.
2  Click the Nodegroups tab.
3 Use the **Join Buffer Pool** and **Leave Buffer Pool buttons** to move nodegroups between the **Nodegroups Not in Buffer Pool** and **Nodegroups in Buffer Pool** lists.

4 **Preview and submit changes.**

**View buffer pool utilization details (IBM DB2 for OS/390 and z/OS)**

The **Usage Details** tab of the Buffer Pools Editor lets you monitor buffer pool utilization.

1 **Open an object editor** on the buffer pool.

2 Click the **Usage Details** tab.

**Using the Check Constraints editor**

The Check Constraints editor lets you modify settings and options of existing check constraints.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to **Open an object editor**, **View and modify database object properties**, and how to **Preview and submit changes**. For detailed information, see **Overview and common usage of Object editors**.

You use this editor to perform the following tasks specific to this object type:

- **Modify a Check Constraint condition definition (IBM DB2, Microsoft SQL Server, Oracle)**

You also use this editor to perform the following common Object editor tasks:

- **Add a comment to a database object**

- **View SQL/DDL for a database object**

Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename</td>
<td>YES</td>
<td>Yes</td>
<td>YES</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td></td>
<td>Yes</td>
<td>YES</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Modify a Check Constraint condition definition (IBM DB2, Microsoft SQL Server, Oracle)**

The **Definition** tab of the Check Constraints editor lets you modify the condition defining a check constraint. Depending on the DBMS that DBArtisan is working with, you can also use the tab to enable or disable the check constraint or change the replication setting.

**To modify a check constraint definition:**

1 **Open an object editor** on the check constraint.

2 Click the **Definition** tab.
3 Edit the check constraint condition definition in the **Check Condition** text area.

**TIP:** As an aid in editing the check constraint, use the **Table Columns** button to open a dialog that lets you paste the name of a column on the current datasource at the cursor location.

4 If you are using DBArtisan against a Microsoft SQL Server or IBM DB2 for Windows, Unix, and Linux DBMS, use the **Enabled** check box to enable or disable the check constraint.

5 If you are using DBArtisan against a Microsoft SQL Server DBMS, use the **Not For Replication** check box to enable or disable the check constraint for replication.

6 **Preview and submit changes.**

### Using the Clusters Editor (Oracle)

The Clusters editor lets you modify settings and options of existing clusters.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to [Open an object editor](#), [View and modify database object properties](#), and how to [Preview and submit changes](#). For detailed information, see [Overview and common usage of Object editors](#).

You use this editor to perform the following tasks specific to this object type:

- **Manage general properties and hash specifications for a Cluster (Oracle)**
- **Manage storage for a Cluster (Oracle)**
- **Manage performance for a Cluster (Oracle)**
- **View usage and space distribution for a Cluster (Oracle)**
- **View the table details for a Cluster (Oracle)**

You also use this editor to perform the following common Object editor tasks:

- **View SQL/DDL for a database object**

Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocate Extent</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deallocate Unused Space</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Truncate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Manage general properties and hash specifications for a Cluster (Oracle)

The **Columns** tab of the Clusters Editor lets you view and manage general properties and hash specifications for every cluster on the current datasource.
To view or manage general properties and hash specifications for a cluster:

1. **Open an object editor** on the cluster to be modified.
2. Click the **Columns** tab.
3. Use the controls in the **General Properties** group to specify a hash or index **Cluster Type** and to specify a **Key Size** for the cluster.
4. If you specified a hash cluster type, use the controls in the **Hash Specifications** group to specify the number of hash keys and to specify use of either the default hash function or a specified expression used as the hash function.
5. **Preview and submit changes**.

### Manage storage for a Cluster (Oracle)

The **Storage** tab of the Clusters Editor lets you manage storage for every cluster on the current datasource. The table below describes the options and functionality on the **Storage** tab of the Clusters Editor:

To manage data block storage and extents settings for a cluster:

1. **Open an object editor** on the cluster to be modified.
2. Click the **Storage** tab.
3. Use the table below as a guide to understanding cluster storage settings:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Block Storage</strong></td>
<td>Each transaction that updates a data block requires a transaction entry. Percent Free - minimum percentage of free space in a block&lt;br&gt;Percent Used - minimum percentage of used space in a block&lt;br&gt;Initial transactions - The initial parameter ensures that a minimum number of concurrent transactions can update a data block, avoiding the overhead of allocating a transaction entry dynamically.&lt;br&gt;Maximum transactions - The maximum parameter limits concurrency on a data block.</td>
</tr>
<tr>
<td><strong>Extents</strong></td>
<td>The unit of space allocated to an object whenever the object needs more space. Initial Extent - The initial space extent (in bytes) allocated to the object.&lt;br&gt;Next Extent - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.&lt;br&gt;Minimum Extents - The appropriate minimum extents value for the object.&lt;br&gt;Maximum Extents - The appropriate maximum extents value for the object.&lt;br&gt;Percent Increase - Magnifies how an object grows and, can materially affect available free space in a tablespace. Select a value in the corresponding box.</td>
</tr>
</tbody>
</table>

4. Optionally, modify the **Percent Free**, **Percent Used**, or **Max Transactions** values.
5. **Preview and submit changes**.

### Manage performance for a Cluster (Oracle)

The **Performance** tab of the Clusters Editor lets you manage performance for every cluster on the current datasource. The table below describes the options and functionality on the **Performance** tab of the Clusters Editor:
To manage performance for a cluster:
1. Open an object editor on the cluster to be modified.
2. Click the Performance tab.
3. Use the table below as a guide to understanding or planning changes to cluster performance settings:

<table>
<thead>
<tr>
<th>Option Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Query Option group</td>
<td>Lets you specify Degrees and Instances settings for processing queries using many query server processes running against multiple CPUs, which provides substantial performance gains such as reduction of the query completion time.</td>
</tr>
<tr>
<td>Cache group</td>
<td>The Cache setting keeps the blocks in memory by placing it at the most recently used end. This option is useful for small lookup tables.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

View usage and space distribution for a Cluster (Oracle)
The Space tab of the Clusters Editor lets you view the usage and the distribution of space for clusters on the current datasource.

To view usage and space distribution for a cluster:
1. Open an object editor on the cluster.
2. Click the Space tab.
3. Use the table below as a guide to understanding the cluster usage and space distribution settings:

<table>
<thead>
<tr>
<th>Option Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Utilization group</td>
<td>Displays the percent of space reserved for future updates.</td>
</tr>
<tr>
<td>FreeLists group</td>
<td>Displays the allocation of data blocks when concurrent processes are issued against the cluster. Identifying multiple freelists can reduce contention for freelists when concurrent inserts take place and potentially improve the performance of the cluster.</td>
</tr>
<tr>
<td>Extents group</td>
<td>The unit of space allocated to an object whenever the object needs more space.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

View the table details for a Cluster (Oracle)
The Tables tab of the Clusters Editor lets you details for each cluster column. Details for each column include the key column in the cluster, the clustered table name, and the key column in the table.

To view tables that share columns in a cluster:
1. Open an object editor on the cluster to be modified.
2. Click the Tables tab.
Using the Data Caches Editor (Sybase)
The Data Caches editor lets you modify settings and options of existing data caches.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- View identifying information for a Data Cache (Sybase)
- Manage object binding for a Data Cache (Sybase)
- Manage buffer pools for a Data Cache (Sybase)

You also use this editor to perform the following common Object editor tasks:

- View SQL/DDL for a database object

View identifying information for a Data Cache (Sybase)
The Information tab of the Data Caches Editor lets you manage basic settings for data caches on the current datasource. After creating or deleting a cache, Sybase ASE requires that you restart the target server to activate the new cache. Until you restart the server the status of the new cache remains Pending Active/Pending Delete.

To manage basic data cache settings:

1. Open an object editor on the data cache to be modified.
2. Click the Information tab.
3. Use the table below as a guide to changing settings on this tab:

<table>
<thead>
<tr>
<th>Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Type</td>
<td>Whether the cache can store data and log pages (mixed) or log pages only (logonly). Because you cannot change the Cache Type for the default cache (it is always default), it does not display in the Cache Type list.</td>
</tr>
<tr>
<td>Cache Policy</td>
<td>The replacement policy for the cache: strict (the default value) and relaxed.</td>
</tr>
<tr>
<td>Status</td>
<td>The current status of the cache: Active (the cache is currently active), Pending Active (the cache is new and becomes active once the server is restarted), and Pending Delete (the cache is still active, but is deleted once the server is restarted). You can only change a mixed data cache if it does not have any non-log objects bound to it.</td>
</tr>
<tr>
<td>Configured Size</td>
<td>The size of the cache after the server is restarted.</td>
</tr>
<tr>
<td>Run Size</td>
<td>The size that the cache is currently using. For the default cache, this is the amount of all cache space not explicitly configured for another cache.</td>
</tr>
<tr>
<td>Overhead</td>
<td>The amount of memory required to manage the cache, in MB. This memory comes from the default cache.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.
Manage object binding for a Data Cache (Sybase)
The **Objects** tab of the Data Caches Editor lets you manage database object binding for caches on the current datasource. You can bind the following items to a mixed cache:

- Entire Database
- Index(es)
- Table(s)
- Text and Image Columns from Tables

The data cache must be active before you can bind an object to it. To activate a new data cache, Sybase ASE requires that you restart the server. The bindings take effect immediately after the server restart.

**NOTE:** You can bind the syslogs table to a logonly cache.

To manage object bindings for a data cache:
1. **Open an object editor** on the data cache to be modified.
2. Click the **Objects** tab.
   
   Currently bound objects are stored under the folders displayed on the tab.
3. Take one of the following actions:
   - Initiate binding another object to the cache by clicking the **Bind** button. For detailed information, see **Bind Data Cache**.
   - Select a bound object and unbind the object by clicking the **Unbind** button and using the **Unbind Data Cache** dialog to verify that you want to unbind that object.
   - Create a caching strategy for a selected, bound index or table by clicking the **Strategy** button. For detailed information, see **Create a cache strategy for a bound index or table**.

Create a cache strategy for a bound index or table
The Cache Strategy dialog box lets you regulate the cache for bound objects.

To regulate the cache for a bound index or table:
1. **Open an object editor** on the data cache to be modified.
2. Click the **Objects** tab.
3. Click a bound index or table, and then click **Strategy**.
   
   The **Cache Strategy** dialog opens.
4. If you selected a bound table, use the **Apply To** radio set to specify the entire table or text only.
5. Enable or disable the **Prefetch** feature.
6. Select the **MRU** or **LRU Replacement** strategy.
7. Click **Execute** or use one of the other execution options.

Manage buffer pools for a Data Cache (Sybase)
The **Pools** tab of the Data Caches Editor lets you manage buffer pools for every cache on the current datasource. Sybase ASE requires that you restart the server to activate any changes that you make to the cache configuration.
To manage buffer pools for a data cache:

1. Open an object editor on the data cache to be modified.

2. Click the Pools tab.

   Information for each bufferpool on the tab includes the name and type of the Pool, Configured Size of the cache after server restart, current Run Size of the pool, and current Wash Size.

   **NOTE:** Wash size must be a minimum of 10 pages and a maximum of 80 percent of the number of buffers in the pool. The buffers used by the sort cannot be used by another other task on the server until the sort completes. Therefore, to avoid seriously affecting query processing if you are creating indexes while other transactions are active, you are limited to setting the number of sort buffers to 80 percent of the pool size.

3. Optionally, use the Add button to add a new buffer pool or select a buffer pool and use the Edit button to modify the settings for that buffer pool.

4. Preview and submit changes.

**Using the Database Devices Editor (Sybase)**

The Database Devices editor lets you modify settings and options of existing database devices.

   **NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- View identifying information for a Database Device (Sybase)
- View size and database fragments for a Database Device (Sybase)
- Manage Database Device mirroring (Sybase)

You also use this editor to perform the following common Object editor tasks:

- View SQL/DDL for a database object

**View identifying information for a Database Device (Sybase)**

The Information tab of the Database Devices Editor lets you view identifying information for database devices on the current datasource and lets you set a database device as the default device.
To view basic information for a database device:

1. **Open an object editor** on the database device to be modified.
2. Click the **Information** tab.

   The **General** and **Page Range** groups on the tab display the following information:

<table>
<thead>
<tr>
<th>Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Number</td>
<td>The unique number for the database device.</td>
</tr>
<tr>
<td>Default Device</td>
<td>Whether the database device is the default device. CREATE and ALTER DATABASE commands that do not specify a location use the default database device.</td>
</tr>
<tr>
<td>Physical Name</td>
<td>The name of the Windows file for the target database device.</td>
</tr>
<tr>
<td>Description</td>
<td><strong>OPTIONAL:</strong> Any user-defined comments for the target database device.</td>
</tr>
<tr>
<td>First Virtual Page</td>
<td>The first page number for the target database device.</td>
</tr>
<tr>
<td>Last Virtual Page</td>
<td>The last page number for the target database device.</td>
</tr>
</tbody>
</table>

   **TIP:** During Sybase ASE installation, the master database is designated as the default database device. To prevent the master database from getting cluttered and to ease database recovery, Sybase recommends that you change the default device from the master database to another database.

3. Optionally, set the **Default Device** check box to designate this device as the default device.
4. **Preview and submit changes**.

### View size and database fragments for a Database Device (Sybase)

The **Space** tab of the Database Devices Editor lets you view the size and database fragments for database devices on the current datasource.

To view space details for a database device:

1. **Open an object editor** on the database device.
2. Click the **Space** tab.

   Details include the **Device Size**, **Space used**, **Free Space**, and the size for all **Database Fragments**.

### Manage Database Device mirroring (Sybase)

The **Mirror** tab of the Database Devices Editor lets you view and manage database device mirroring for every database device on the current datasource.

To view or manage mirroring for a database device:

1. **Open an object editor** on the database device to be modified.
2. Click the **Mirror** tab.

   The tab displays the **Mirror Name** and **Mirror Status**.

3. Optionally, you can **Mirror a Database Device (Sybase)** that currently has mirroring disabled, or **Unmirror a Database Device (Sybase)** that currently has mirroring enabled.
Mirror a Database Device (Sybase)
Mirroring protects against the loss of data due to disk failure by maintaining a fully redundant copy of data on a separate disk.

To mirror a database device:
1. Open an object editor on the database device to be mirrored.
2. Click the Mirror tab.
   The Mirror Status field shows the current mirroring status for the device.
3. Click the Mirror button.
   The Mirror Device dialog opens
4. Use the table below as a guide to choosing mirroring options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mirror Name</td>
<td>Lets you specify the mirror name.</td>
</tr>
<tr>
<td>Serial Writes</td>
<td>Forces serial writes to the mirror device.</td>
</tr>
<tr>
<td>Contiguous</td>
<td>Specifies a contiguous mirror device used when the initial medium becomes full, allowing continuation of the backup operation.</td>
</tr>
</tbody>
</table>
5. Click the Mirror button.

Unmirror a Database Device (Sybase)

To unmirror a database device:
1. Open an object editor on the database device to be unmirrored.
2. Click the Mirror tab.
   The Mirror Status field shows the current mirroring status for the device.
3. Click the Unmirror button.
   The Unmirror Device dialog opens
Use the table below as a guide to understanding the options on this dialog:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Name</td>
<td>Displays the database device name.</td>
</tr>
<tr>
<td>Mirror Name</td>
<td>Displays the mirror name.</td>
</tr>
<tr>
<td>Mirror Side</td>
<td>Lets you specify the primary or secondary device.</td>
</tr>
<tr>
<td></td>
<td>Primary - Selects the database device mirror side.</td>
</tr>
<tr>
<td></td>
<td>Secondary - Selects the mirror device mirror side.</td>
</tr>
<tr>
<td>Unmirroring Mode</td>
<td>Lets you specify the unmirror as temporary or permanent.</td>
</tr>
<tr>
<td></td>
<td>Retain - Sets the unmirror as temporary.</td>
</tr>
<tr>
<td></td>
<td>Remove - Sets the unmirror as permanent.</td>
</tr>
</tbody>
</table>
Using the Database Links editor (Oracle)
The Database Links editor lets you modify settings and options of existing database links.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- View connection string information for a Database Link (Oracle)

You also use this editor to perform the following common Object editor tasks:

- View SQL/DDL for a database object

Lastly, the **Command** menu on the Object editor toolbar provides the following command:

- Rename

View connection string information for a Database Link (Oracle)
The **Definition** tab of the Database Links Editor displays connection string information for database links on the current datasource.

**To view connection information for a database link:**

1. Open an object editor on the database link to be modified.
2. Click the **Definition** tab.
   
   Details include the **Public** setting, the **User** name and **Password** associated with the database link, and a **Connect String**.

Using the Databases editor (IBM DB2, Microsoft SQL Server, Sybase ASE)
The Databases editor lets you modify settings and options of existing databases.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage tablespaces, file details, or database fragments for a database (IBM DB2 for Windows, Unix, and Linux, Microsoft SQL Server, Sybase)
- View and modifying database configuration/options (IBM DB2 for Windows, Unix, and Linux)
- View and modifying database configuration/options (Microsoft SQL Server)
- View and modify database options (Sybase)
- View database space usage (Microsoft SQL Server)
- View database space usage (Sybase)
• Manage basic properties of a database (IBM DB2 for OS/390 and Z/OS)
• Manage database transaction logs (Microsoft SQL Server)

You also use this editor to perform the following common Object editor tasks:

• Work with privileges and permissions
• View SQL/DDL for a database object
• Work with object dependencies

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach</td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td>For more information, see Detach/Attach.</td>
</tr>
<tr>
<td>Backup</td>
<td></td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>For more information, see Database Recovery Wizard.</td>
</tr>
<tr>
<td>Change Comment</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For more information, see Change Database Comment.</td>
</tr>
<tr>
<td>Checkpoint</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBCC</td>
<td>YES</td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detach</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For more information, see Detach/Attach.</td>
</tr>
<tr>
<td>Move Log</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td>YES</td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restore</td>
<td>YES</td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td>For more information, see Database Recovery Wizard.</td>
</tr>
<tr>
<td>Set Online/Offline</td>
<td>YES</td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shrink</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update Statistics</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** For IBM DB2 (version 9), DBArtisan supports quick schema copying. For details, see Copy Schema.

Manage tablespaces, file details, or database fragments for a database (IBM DB2 for Windows, Unix, and Linux, Microsoft SQL Server, Sybase)

Depending on the DBMS that DBArtisan is working with, the Placement tab of the Databases Editor lets you manage the following:

• IBM DB2 - tablespace details
• Microsoft SQL Server - file details
• Sybase - database fragment details (data and log devices)
To work with tablespace, file details, or database fragments for a database:
1. Open an object editor on the database to be modified.
2. Click the Placement tab.
3. Take one of the following actions:
   - If working against an IBM DB2 for Windows/Unix/Linux DBMS, you can select a tablespace and click the Edit button to open an object editor on that tablespace. For more information, see Using the Tablespaces editor (DB2, Oracle).
   - If working against a Microsoft SQL Server DBMS, you can view the File Group Name, Default File Group, Read-only File Group, Device File Name, Physical File Name, and Max Size properties. You can set the Size, File Growth Rate, Default File Group, and Unlimited Max Size properties.
   - If working against a Sybase DBMS, you can select a fragment to view the Device Name, Size, and Device Type for the fragment. You can also click the New button to open a dialog that lets you provide the data device and log device details for a new database fragment.
   
   **NOTE:** We strongly recommend that you place the transaction log on a separate device from all other database fragments.
4. Preview and submit changes.

View and modifying database configuration/options (IBM DB2 for Windows, Unix, and Linux)
The Options tab of the Databases Editor lets you view or modify configuration settings or options for databases on the current datasource.

To view or modify database configuration settings and options:
1. Open an object editor on the database to be modified.
2. Click the Options tab.

The following table shows the options and configuration settings you can view on this tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>adsm_mgmtclass</td>
<td>adsm_nadename</td>
</tr>
<tr>
<td>appl_ctl_heap_sz</td>
<td>autorestart</td>
</tr>
<tr>
<td>catalogcache_sz</td>
<td>chngpgs_thresh</td>
</tr>
<tr>
<td>country</td>
<td>database_consistent</td>
</tr>
<tr>
<td>dft_extent_sz</td>
<td>dft_loadadres</td>
</tr>
<tr>
<td>dft_sqlmathwarm</td>
<td>dft_prefetch_sz</td>
</tr>
<tr>
<td>d1_num_copies</td>
<td>d1_time_drop</td>
</tr>
<tr>
<td>estore_seg_sz</td>
<td>indexsort</td>
</tr>
<tr>
<td>logbufsz</td>
<td>logfilsize</td>
</tr>
<tr>
<td>logretain</td>
<td>logsecond</td>
</tr>
<tr>
<td>maxlocks</td>
<td>mincommit</td>
</tr>
<tr>
<td>numsegfs</td>
<td>num_db_backups</td>
</tr>
<tr>
<td>num_ioservers</td>
<td>num_quantiles</td>
</tr>
<tr>
<td>restore_pending</td>
<td>segdetect</td>
</tr>
<tr>
<td>softmax</td>
<td>softmax</td>
</tr>
<tr>
<td>stat_heap_sz</td>
<td>stat_heap_sz</td>
</tr>
</tbody>
</table>

The table shows the options and configuration settings you can view on this tab.
If the Modifiable field for a setting indicates that the setting is modifiable, select that setting and click the Edit button to open a dialog that will let you edit setting value and execute your changes.

**View and modifying database configuration/options (Microsoft SQL Server)**

The *Options* tab of the Databases Editor lets you view and modify the database options for databases on the current datasource.

**To modify database options:**

1. [Open an object editor](#) on the database to be modified.
2. Click the *Options* tab.
3. In the *Creation Properties* group, you can choose values for the *Compatible Level* and *Owner* settings.
4. Use the following table as a guide to the *Properties* you can set on this tab:

<table>
<thead>
<tr>
<th>ANSI null default</th>
<th>ANSI nulls</th>
<th>ANSI padding</th>
<th>ANSI warnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>arithabort</td>
<td>auto create statistics</td>
<td>auto update statistics</td>
<td>auto close</td>
</tr>
<tr>
<td>auto shrink</td>
<td>concat null yields null</td>
<td>cursor close on commit</td>
<td>db chaining</td>
</tr>
<tr>
<td>dbo use only</td>
<td>default to local cursor</td>
<td>merge publish</td>
<td>numeric roundabout</td>
</tr>
<tr>
<td>offline</td>
<td>published</td>
<td>quoted identifier</td>
<td>read only</td>
</tr>
<tr>
<td>recursive triggers</td>
<td>select into/bulkcopy/pllsort</td>
<td>single user</td>
<td>subscribed</td>
</tr>
<tr>
<td>torn page detection</td>
<td>trunc log on chkpt</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. [Preview and submit changes](#).

**TIP:** The *Compatible level* option sets certain database behaviors to be compatible with the specified earlier version of Microsoft® SQL Server. The compatibility level affects the behaviors in the specified database, not the entire server. To set database options for all future databases, set the database options on the model database.

**View and modify database options (Sybase)**

The *Properties* tab of the Databases Editor lets you view and modify configuration settings and options for databases on the current datasource.

**To modify database configuration settings and options:**

1. [Open an object editor](#) on the database to be modified.
2. Click the *Properties* tab.
3. In the *Creation Properties* group, you can choose an *Owner* for the database.
4 Use the following table as a guide to enabling/disabling the **Options** on this tab:

<table>
<thead>
<tr>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>abort tran on log full</td>
</tr>
<tr>
<td>allow nulls by default</td>
</tr>
<tr>
<td>auto identity</td>
</tr>
<tr>
<td>dbo use only</td>
</tr>
<tr>
<td>ddl in tran</td>
</tr>
<tr>
<td>identity in nonunique index</td>
</tr>
<tr>
<td>no chkpt on recovery</td>
</tr>
<tr>
<td>no free space acctg</td>
</tr>
<tr>
<td>read only</td>
</tr>
<tr>
<td>select into/bulkcopy/plsort</td>
</tr>
<tr>
<td>single user</td>
</tr>
<tr>
<td>trunc log on chkpt</td>
</tr>
</tbody>
</table>

**NOTE:** For Sybase 15, under Creation Properties, a Type property of **TEMP** indicates that this table was created as a temporary table. Temporary tables have a distinctive icon in the datasource tree.

5 **Preview and submit changes.**

**TIP:** To set database options for all future databases, set the database options on the model database.

**View database space usage (Microsoft SQL Server)**

The **Space** tab of the Databases Editor lets you view pie charts showing the data space usage and the transaction log (if available) space usage for every database on the current datasource.

To view database space usage and transaction log usage:

1. Open an object editor on the database to be modified.
2. Click the **Space** tab.

**View database space usage (Sybase)**

The **Space** tab of the Databases Editor lets you view pie charts showing the data space usage and the transaction log (if available) space usage for every database on the current datasource.

To view database space usage and transaction log usage:

1. Open an object editor on the database to be modified.
2. Click the **Space** tab.

**TIP:** Double-click a slice in the pie chart for detailed statistics.

**Manage basic properties of a database (IBM DB2 for OS/390 and Z/OS)**

The **Properties** tab of the Databases Editor lets you manage the parameters for the database and display and alter the database coded character set identifier (CCSID).

To view basic properties of a database:

1. Open an object editor on the database to be modified.
2. Click the **Properties** tab.
3. Use the following table as a guide to understanding and modifying the properties on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Properties</th>
</tr>
</thead>
</table>
Manage database transaction logs (Microsoft SQL Server)

The Transaction Log tab of the Databases Editor lets you manage transaction logs for every database on the current datasource.

To manage transaction log details for a database:
1. Open an object editor on the database to be modified.
2. Click the Transaction Log tab.
3. Select a transaction log from the list on the left and in the Log Device Properties group, modify the Size, File Growth Rate, Max Size, and Unlimited Max Size settings.
4. Optionally, take one of the following actions:
   • Click the New button to provide details for a new transaction log
   • Select a transaction log and click the Delete button to delete the log
5. Preview and submit changes.

Using the DBRM Editor (IBM DB2 for OS/390 and z/OS)

The DBRM editor lets you modify settings and options of existing DBRMs.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- **View DBRM header contents (IBM DB2 for OS/390 and z/OS)**
- **View the plan containing a DBRM (IBM DB2 for OS/390 and z/OS)**
- **View collections, plans, and packages for a DBRM (IBM DB2 for OS/390 and z/OS)**
- **View information on SQL statements in a DBRM (IBM DB2 for OS/390 and Z/OS)**

**View DBRM header contents (IBM DB2 for OS/390 and z/OS)**

The Description tab of the DBRM Editor displays the main contents of the DBRM header information, including the location of the source, and the characteristics of the DBRM.
To view DBRM header contents:
1. Open an object editor on the DBRM.
2. Click the Description tab.
3. Use the following table as a guide to understanding the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Lets you view the Precompile Timestamp, Language, Consistency Token, String Delimiter, Katakana charset, Mixed Character Set, Decimal Point, Dec31, and Version properties.</td>
</tr>
</tbody>
</table>

View the plan containing a DBRM (IBM DB2 for OS/390 and z/OS)
The Plans tab of the DBRM Editor displays the plans that contain the DBRM, and the plan creator. The DBRM name, version, and plan name are from the Datasource Explorer.

To work with plans of a DBRM:
1. Open an object editor on the DBRM to be modified.
2. Click the Plans tab.
3. Optionally, take one of the following actions:
   • Select a plan and click Edit to open object editor for that plan.
   • Select a plan and click Rebind to open the Rebind Plans dialog.
   • Select a plan and click Free to open the Free Plans dialog.
4. Preview and submit changes.

View collections, plans, and packages for a DBRM (IBM DB2 for OS/390 and Z/OS)
The Packages tab of the DBRM Editor displays a cross-reference to the collections, packages, and plans where the DBRM was bound.

To work with packages for a DBRM:
1. Open an object editor on the DBRM.
2. Click the Packages tab.
3. Optionally, take one of the following actions:
   • Select a package and click Edit to open object editor for that package.
   • Select a package and click Rebind to open the Rebind Packages dialog.
   • Select a package and click Free to open the Free Packages dialog.
4. Preview and submit changes.

View information on SQL statements in a DBRM (IBM DB2 for OS/390 and Z/OS)
The Statements tab of the DBRM Editor displays the information about the SQL statements in the DBRM. The information is for programmers reference only.
To open an ISQL Editor window on a DBRM statement:
1. Open an object editor on the DBRM to be modified.
2. Click the Statements tab.
3. Select a statement and click Explain to copy the statement to an ISQL Editor window.

**NOTE:** For more information, see ISQL Editor.

### Using the Defaults Editor (Microsoft SQL Server, Sybase)

The Defaults editor lets you modify settings and options of existing defaults.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- **Modify the value of a Default (Microsoft SQL Server, Sybase)**

You also use this editor to perform the following common object editor tasks:

- **Work with object dependencies**

The Command menu on the Object editor toolbar provides the following command:

- **Rename**

### Modify the value of a Default (Microsoft SQL Server, Sybase)

The Properties tab of the Defaults Editor lets you modify the value of a default.

**To modify the value of a default:**
1. Open an object editor on the default to be modified.
2. Click the Properties tab.
3. Modify the Value.
4. Preview and submit changes.

### Using the Directories Editor (Oracle)

The Directories editor lets you modify settings and options of existing directories.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- **View SQL for a Directory (Oracle)**
You also use this editor to perform the following common Object editor tasks:

- Work with privileges and permissions

View SQL for a Directory (Oracle)
The Definition tab of the Directories Editor lets you view the SQL code for every directory on the current datasource.

To view the path of a directory:
1. Open an object editor on the directory to be modified.
2. Click the Definition tab.

Using the Event Monitors editor (IBM DB2 for Windows, Unix and Linux)
The Event Monitors editor lets you view and modify parameters, WHERE clause conditions, and partitions for event monitors on a datasource.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Specify event types and partitions for an Event Monitor (IDM DB2 for Windows, Unix, and Linux)
- Specify target, buffersize, and file attributes for an Event Monitor (IDM DB2 for Windows, Unix, and Linux)
- View the event type and condition for an Event Monitor (IDM DB2 for Windows, Unix, and Linux)

You also use this editor to perform the following common Object editor tasks:

- View SQL/DDL for a database object

Specify event types and partitions for an Event Monitor (IDM DB2 for Windows, Unix, and Linux)
The Definition tab of the Event Monitors Editor lets you specify event types and database partitions for event monitors on the current datasource.

To specify event types and partitions for an event monitor:
1. Open an object editor on the event monitor to be modified.
2. Click the Definition tab.
3. Use the table below as a guide to modifying the event monitor:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Types</td>
<td>Choose an event type or types for this monitor (bufferpools, connections, database, deadlocks, statements, tables, tablespaces, transactions)</td>
</tr>
<tr>
<td>Database Partition</td>
<td>Displays the partition associated with the event monitor.</td>
</tr>
</tbody>
</table>
Specify target, buffersetze, and file attributes for an Event Monitor (IDM DB2 for Windows, Unix, and Linux)

The Target tab of the Event Monitors Editor lets you specify target, buffersetze, and file attributes for every event monitor on the datasource.

To manage target, buffersetze and file attributes for an event monitor:
1. Open an object editor on the event monitor to be modified.
2. Click the Target tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autostart</td>
<td>Specify if the event monitor is to start automatically when the database starts.</td>
</tr>
<tr>
<td>Deadlock Monitoring</td>
<td>If you chose a Deadlock event type, specify the scope (Local or Global) and the With Details option.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

View the event type and condition for an Event Monitor (IDM DB2 for Windows, Unix, and Linux)

The Conditions tab of the Event Monitors Editor lets you view event type and condition for event monitors on the datasource.

To view the event type and condition for an event monitor:
1. Open an object editor on the event monitor to be modified.
2. Click the Conditions tab.

Using the Extended Procedures Editor (Microsoft SQL Server, Sybase)

The Extended Procedures editor lets you modify settings and options of existing extended procedures.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.
You use this editor to perform the following tasks specific to this object type:

- Manage basic definition details for an Extended Procedure (Microsoft SQL Server, Sybase)

You also use this editor to perform the following common Object editor tasks:

- Work with object dependencies
- Work with privileges and permissions
- View SQL/DDL for a database object

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

Manage basic definition details for an Extended Procedure (Microsoft SQL Server, Sybase)

The Definition tab of the Extended Procedures Editor lets you view basic definition details and modify the library name associated with a procedure.

To work with basic definition details for an extended procedure:

1. Open an object editor on the extended procedure to be modified.
2. Click the Definition tab.
3. Optionally, change the Library Name for the procedure.
4. Preview and submit changes.

Using the Foreign Keys editor

The Foreign Keys Editor lets you manage columns of foreign keys on a datasource.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage Foreign Key columns (All DBMS)

You also use this editor to perform the following common Object editor tasks:

- View SQL/DDL for a database object
Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rename</strong></td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td></td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Manage Foreign Key columns (All DBMS)**

The **Column Mapping** tab of the Foreign Keys Editor lets you manage columns for every foreign key on the current datasource.

**To manage foreign key columns:**

1. Open an object editor on the foreign key to be modified.
2. Click the **Column Mapping** tab.
   - The existing column mapping for the foreign key is represented by selected columns in the **Main Table** and **Referenced Table** lists. Additional candidates in the **Referenced Table** list are indicated by enabled column check boxes.
3. If necessary, use the **Table** dropdown in the **Referenced Table** group to choose a new table for this foreign key.
4. Select or deselect columns in the **Main Table** list and **Referenced Table** list to form the referential constraint between the two tables.
5. Preview and submit changes.

**Manage Foreign Key properties (All DBMS)**

The **Properties** tab of the Foreign Keys Editor lets you manage basic options for every foreign key on the current datasource.

**To manage foreign key options:**

1. Open an object editor on the foreign key to be modified.
2. Click the **Properties** tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Lets you specify the following DBMS-specific settings:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Delete Rule</strong> and <strong>Update Rule</strong> (IBM DB2 for Windows, Unix, and Linux)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Delete Rule</strong> (IBM DB2 for OS/390 and Z/OS)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Enabled, Not For Replication, Delete Rule</strong>, and <strong>Update Rule</strong> (Microsoft SQL Server)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Enabled</strong> and <strong>Delete Cascade</strong> (Oracle)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Match Full</strong> (Sybase)</td>
</tr>
</tbody>
</table>
Using the Functions editor (IBM DB2, Microsoft SQL Server, Oracle)

The Functions Editor lets you view and modify function definitions, status, and size details, and manage function dependencies.

**NOTE:** Functions are available in Microsoft SQL Server 8.0 or later.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage function parameters and return parameters (IBM DB2)
- Manage the body of a function (IBM DB2 for Windows, Unix, and Linux)
- Modify function definition DDL (Microsoft SQL Server, Oracle)
- Manage function properties (IBM DB2 for Windows, Unix, and Linux)
- View function properties (Oracle)

You also use this editor to perform the following common Object editor tasks:

- Add a comment to a database object
- View SQL/DDL for a database object
- Work with privileges and permissions
- Work with object dependencies

Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Synonym</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>For more information, see Add-On Tools</td>
</tr>
<tr>
<td>Debug</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Describe</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Manage function parameters and return parameters (IBM DB2)

The Functions Editor has **Parameters**, **Return Scalar**, and **Return Columns** tabs that let you manage inputs and outputs of a function:
To manage function inputs and outputs:

1. Open an object editor on the function to be modified.
2. Click the Parameters, Return Scalar, or Return Columns tab.

   The tab has a list of parameters, and for the currently selected parameter, details such as the type and size of the parameter are shown in the Attributes group.

3. Optionally, take one of the following actions:
   - In the attribute area, provide new values, as permissible. Attributes differ by the tab you chose but typically include items such as Type, Precision, Scale, As Locator, and Size.
   - Click the New button to provide datatype and size details for a new parameter.
   - Select a parameter and click Drop to delete that parameter.
4. Preview and submit changes.

Manage the body of a function (IBM DB2 for Windows, Unix, and Linux)

The Body tab of the Functions editor lets you manage the body of a function.

To modify the body of a function:

1. Open an object editor on the function to be modified.
2. Click the Body tab.
3. Optionally, modify the code in the text area of the tab.
4. Preview and submit changes.

Modify function definition DDL (Microsoft SQL Server, Oracle)

The Definition tab of the Functions Editor lets you modify the DDL defining a function.

To view the DDL for a function:

1. Open an object editor on the function to be modified.
2. Click the Definition tab.
3. Optionally, modify the DDL in the Function Text area as required.
4. Preview and submit changes.

Manage function properties (IBM DB2 for OS/390 and Z/OS)

The Properties tab of the Functions Editor lets you work with single value properties of a function.

   NOTE: Options differ by type of function.

To manage definition settings for a function:

1. Open an object editor on the function to be modified.
2. Click the Definition tab.
3 Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>Lets you view the Create Timestamp, Schema, Routine ID, and Origin properties. Lets you set the External Name and Collection ID properties.</td>
</tr>
<tr>
<td>Run Time</td>
<td>Lets you view the Result Sets property. Lets you set the WLM Environment, WLM For Nested, ASU Time, and Run Options properties.</td>
</tr>
<tr>
<td>Structure</td>
<td>Lets you view the Language, Parameter Style, Number of LOB Columns, Number of Parameters, and Allow Parallel properties. Lets you set the Program Type, Security Type, SQL Access Level, Inherit Special Registers, Fenced, DBINFO, Deterministic, Called On Null Input, External Action, Stay Resident, Final Call, Scratchpad, and Scratchpad Length properties.</td>
</tr>
<tr>
<td>Run Estimates</td>
<td>Lets you view the Initial I/Os, I/Os Per Invocation, Initial Instructions, and Instructions Per Invocation properties.</td>
</tr>
<tr>
<td>Java Structure</td>
<td>Lets you view the Java Class, Jar ID, Package ID, Method ID, Jar Schema, and Java Signature properties.</td>
</tr>
</tbody>
</table>

4 Preview and submit changes.

Manage function properties (IBM DB2 for Windows, Unix, and Linux)
The Properties tab of the Functions Editor lets you work with single value properties of a function.

**NOTE:** Options differ by type of function.

To manage definition settings for a function:
1 Open an object editor on the function to be modified.
2 Click the Definition tab.
3 Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>If the function body is written in C or OLE, select DB2SQL. If the function body is written in JAVA, select DB2GENERAL. <strong>NOTE:</strong> If the function body is written in OLEDB, this parameter cannot be changed.</td>
</tr>
<tr>
<td>Return Type</td>
<td>Specifies the type of value returned by the function.</td>
</tr>
<tr>
<td>External Name</td>
<td>To change the external name of the function, type the new external name.</td>
</tr>
<tr>
<td>SQL Access Level</td>
<td>Specifies the SQL Access Level for the function.</td>
</tr>
<tr>
<td>Origin</td>
<td>Specifies the function’s origin setting.</td>
</tr>
<tr>
<td>Threadsafe</td>
<td>Specifies whether the function is declared as Threadsafe.</td>
</tr>
<tr>
<td>Fenced</td>
<td>Select to make the function run in a memory space separate from the database.</td>
</tr>
<tr>
<td>Scratchpad</td>
<td>Specifies whether the function can keep track of its state from invocation to invocation.</td>
</tr>
<tr>
<td>Scratchpad Length</td>
<td>IBM DB2 for Linux, Unix, and Windows 8 ONLY: Lets you type the length of the scratchpad. <strong>NOTE:</strong> External scalar and external table functions default is 100.</td>
</tr>
<tr>
<td>Allow Parallel</td>
<td>Enable to allow parallel processing</td>
</tr>
</tbody>
</table>
View function properties (Oracle)
The Properties tab of the Functions Editor displays the status and size information for every function on the current datasource.

To view size and status details for a function:
1. Open an object editor on the function to be modified.
2. Click the Properties tab.
3. Use the following table as a guide to understanding the settings on this tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Call</td>
<td>Specifies whether the function is defined with the Final Call option.</td>
</tr>
<tr>
<td>Parameter Style</td>
<td>To change the parameter style of the function, select the new parameter style. If the function body is written in OLEDB, this parameter cannot be changed.</td>
</tr>
<tr>
<td>Inherit Special Registers</td>
<td>Specifies whether the function has the Inherit Special Registers clause enabled.</td>
</tr>
<tr>
<td>DBInfo</td>
<td>Enable to allow use of the DBINFO parameter with this function.</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Select to make the function return different results from calls with the same parameters.</td>
</tr>
<tr>
<td>External Action</td>
<td>Select to allow the function to take action on an object not managed by DB2.</td>
</tr>
<tr>
<td>Called on Null Input</td>
<td>Specifies whether the function is to execute if no parameters are passed.</td>
</tr>
</tbody>
</table>

4. Optionally, use the Retrieve Size Info button to update the values in the Size group.
5. Preview and submit changes.

Using the Groups editor (IBM DB2 for Windows, Unix, and Linux, Oracle, Sybase)
The Groups editor lets you modify settings and options of existing groups.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage group users (Sybase)

You also use this editor to perform the following common Object editor tasks:

- Work with privileges and permissions
• View SQL/DDL for a database object

Manage group users (Sybase)
The Users tab of the Groups Editor lets you add users to, or remove users from, every group on the current datasource.

To manage the members for a group:
1. Open an object editor on the group to be modified.
2. Click the Users tab.
3. To move a user from one of the Users Not In Group or Users In Group lists to the other list, select the user in the list and click the Join Group or Leave Group button.
4. Preview and submit changes.

Using the Indexes editor
The Indexes editor lets you modify settings and options of existing indexes.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

• Manage index columns (All DBMS)
• Set index properties (IBM DB2 for Windows, Unix, and Linux, Microsoft SQL Server, Oracle, Sybase)
• View space details for indexes (IBM DB2, Oracle)
• Manage storage for indexes (IBM DB2 for OS/390 and Z/OS, Oracle, Sybase)
• Display page and row statistics for an index (Microsoft SQL Server, Sybase)
• Manage index partitions (Oracle)

You also use this editor to perform the following common Object editor tasks:

• Add a comment to a database object
• View SQL/DDL for a database object
• Work with privileges and permissions

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Command</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocate Extent</td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze</td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convert to Partitioned</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>
Manage index columns (All DBMS)

The Columns tab of the Indexes Editor lets you manage the columns for indexes on the current datasource.

   NOTE: For IBM DB2 for Windows, Unix, and Linux, you can segregate Include columns; columns that are to be part of the index but not part of the unique key. For details, see Manage include columns for an index (IBM DB2 for Windows, Unix, and Linux).

To manage the columns for an index:
1. Open an object editor on the index to be modified.
2. Click the Columns tab.
   - The tab shows a listing of the columns currently defined as part of this index.
3. Use the Sort column check box to specify a sort option.
4. Optionally, take one of the following actions:
   - Click the New button to add a new column to the index.
   - Select an existing column and click the Delete button to delete that column from the index.
5. Preview and submit changes.

Manage include columns for an index (IBM DB2 for Windows, Unix, and Linux)

For IBM DB2 for Windows, Unix, and Linux, you can separate index columns into those that form the unique key and Include columns that are to be part of the index but not part of the unique key. The Include Columns tab of the Index editor lets you manage Include columns.

To manage the include columns for an index:
1. Open an object editor on the index to be modified.
2. Click the Include Columns tab.
   - The tab shows a listing of the Include columns currently defined as part of this index.

<table>
<thead>
<tr>
<th></th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCC</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deallocate Unused Space</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimate Size</td>
<td>YES</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Place</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rebuild Unusable Local Indexes</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td>YES</td>
<td>YES</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Reorganize</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Shrink</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Update Statistics</td>
<td>YES</td>
<td>YES</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
3 Optionally, take one of the following actions:
   • Click the New button to add a new column to the index.
   • Select an existing column and click the Delete button to delete that column from the index.

4 Preview and submit changes.

Set index properties (IBM DB2 for Windows, Unix, and Linux, Microsoft SQL Server, Oracle, Sybase)

The Properties tab of the Indexes Editor lets you manage basic properties for indexes on the current datasource.

To modify the properties of an index:
1 Open an object editor on the index to be modified.
2 Click the Properties tab.
3 Use the following table as a guide to setting options on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>For IBM DB2 for Windows, Unix, and Linux, lets you view the Defined By property. Lets you set the</td>
</tr>
<tr>
<td></td>
<td>Index Type, Clustered, and Allow Reverse Scans properties.</td>
</tr>
<tr>
<td></td>
<td>For IBM DB2 for OS/390 and Z/OS, lets you set the Buffer Pool, Piece Size, Close, and Copy properties.</td>
</tr>
<tr>
<td></td>
<td>For Microsoft SQL Server, lets you set Index Type, Clustered, Ignore Duplicate Key, and Statistics Recompute properties.</td>
</tr>
<tr>
<td></td>
<td>For Oracle, lets you view IsValid and Function-Based properties. Lets you set Index Type, No Sort, Logging, and Reverse properties.</td>
</tr>
<tr>
<td></td>
<td>For Sybase, lets you view the Ignore Duplicate Rows property. Lets you set Index Type, Clustered, Ignore Duplicate Key, and Maximum Rows Per Page properties.</td>
</tr>
<tr>
<td>Storage</td>
<td>For IBM DB2 for Windows, Unix, and Linux, lets you set Percent Free and Minimum Percent Used settings.</td>
</tr>
<tr>
<td></td>
<td>For Microsoft SQL Server, lets you set Fill Factor, File Group, Pad Index, and Sort in Tempdb settings.</td>
</tr>
<tr>
<td>Parallel Query Option (Oracle only)</td>
<td>Lets you view the No Parallel Execution property. Lets you set the Parallel Degree and Parallel Instances values.</td>
</tr>
</tbody>
</table>

TIP: For Oracle, the Parallel server query option lets you process queries, using many query server processes, running against multiple CPUs. This option provides substantial performance gains such as reduction of the query completion time. You can also opt for NOPARALLEL execution in the event that the cost of synchronization from the parallel processing is too high and throughput declines.

4 Preview and submit changes.

View space details for indexes (IBM DB2, Oracle)

The Space tab of the Indexes Editor lets you view storage attributes, statistics, and card details for indexes on the current datasource.
To manage space settings for an index:
1. Open an object editor on the index to be modified.
2. Click the Space tab.
3. Use the following table as a guide to understanding the settings displayed:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes (IBM DB2)</td>
<td>Lets you view Make Unique, System Required, Total Keys, Page Fetch Pairs, and Distinct Keys settings.</td>
</tr>
<tr>
<td>Space Utilization (Oracle)</td>
<td>Size and number of Blocks</td>
</tr>
<tr>
<td>Statistics</td>
<td>For IBM DB2, lets you view Index Level, Cluster Ratio, Cluster Factor, Leaf Pages, Sequential Pages, and Density settings. For Oracle, lets you view Index Level, Distinct Keys, Cluster Factor, Leaf Blocks, Leaf Blks/Key and Data Blks/Key properties.</td>
</tr>
<tr>
<td>Cards (IBM DB2 for Windows, Unix, and Linux)</td>
<td>For IBM DB2 for Windows, Unix, and Linux, lets you view First Key, First 2 Keys, First 3 Keys, First 4 Keys and Full Keys settings. For IBM DB2 for OS/390 and Z/OS lets you First Key and Full Keys settings.</td>
</tr>
<tr>
<td>Extents (Oracle)</td>
<td>No. of extents, Max Extents and for each extent, the Extent ID, File ID, Block ID, and number of Blocks.</td>
</tr>
</tbody>
</table>

Manage storage for indexes (IBM DB2 for OS/390 and Z/OS, Oracle, Sybase)

The Storage tab of the Indexes Editor lets you manage storage for every index on the current datasource.

To manage storage settings for an index:
1. Open an object editor on the index to be modified.
2. Click the Storage tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes (IBM DB2 for OS/290 and Z/OS)</td>
<td>Lets you view the Storage Group and VCAT Catalog properties. Lets you specify the Primary Space Allocation, Secondary Space Allocation, Erase, Free Page, Percent Free, and GBP Cache properties.</td>
</tr>
<tr>
<td>Storage (Microsoft SQL Server, Sybase)</td>
<td>For Microsoft SQL Server, lets you specify a Filegroup, specify a Fill Factor value, enable/disable the Pad Index feature, and specify a Sort in Tempdb choice. For Sybase, lets you specify a Segment and Reserve Page Gap value.</td>
</tr>
<tr>
<td>Data Block Storage (Oracle)</td>
<td>Lets you choose the Tablespace, and specify Percent Free, Initial Transactions, and Max Transactions values.</td>
</tr>
<tr>
<td>Extents (Oracle)</td>
<td>Lets you specify Initial Extent, Next Extent, Minimum Extents, Maximum Extents, and Percent Increase values.</td>
</tr>
<tr>
<td>Freelists (Oracle)</td>
<td>Lets you specify Freelists and Freelist Groups values.</td>
</tr>
<tr>
<td>Cache Strategy (Sybase)</td>
<td>Lets you specify a Replacement Strategy and enable/disable Prefetch.</td>
</tr>
<tr>
<td>Buffer Pool (Oracle)</td>
<td>Lets you specify a buffer pool.</td>
</tr>
</tbody>
</table>
Display page and row statistics for an index (Microsoft SQL Server, Sybase)
The Statistics tab of the Indexes Editor lets you view the page and row statistics for every index on the current datasource.

To view page and row statistics for an index:
1. Open an object editor on the index to be modified.
2. Click the Statistics tab.
3. Use the following table as a guide to understanding the settings displayed:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Statistics</td>
<td>Data Pages, Reserved Pages, Used Pages, OAM Page Ratio (Sybase), Index Page Ratio (Sybase), and Total Pages Modified (Microsoft SQL Server)</td>
</tr>
<tr>
<td>Row Statistics</td>
<td>Maximum Row Size, Minimum Row Size, Max Size of Non-Leaf Index Row, Maximum Rows Per Page (Sybase), and Total Rows Modified (Microsoft SQL Server)</td>
</tr>
</tbody>
</table>

Manage index partitions (IBM DB2 for OS/390 and Z/OS)
The Partitions tab of the Indexes Editor lets you partition indexes on the current datasource.

To manage partitions for an index:
1. Open an object editor on the index to be modified.
2. Click the Partitions tab.
3. Optionally, select a partition and click Edit to modify the settings for that partition.
4. Preview and submit changes.

Manage index partitions (Oracle)
The Partitions tab of the Indexes Editor lets you partition indexes on the current datasource.

NOTE: Options vary for partitioned and non-partitioned tables.

To manage partitions for an index:
1. Open an object editor on the index to be modified.
2. Click the Partitions tab.
3. Use the following table as a guide to understanding the settings on this page:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Lets you view the Locality (Global/Local), Alignment (Prefix/Non-Prefix), Partition Type (including Range-Hash Composite or Range-List Composite), and Subpartition type properties.</td>
</tr>
<tr>
<td>Columns</td>
<td>Displays partitioning columns.</td>
</tr>
<tr>
<td>Partition Definitions</td>
<td>Displays details for each partition.</td>
</tr>
</tbody>
</table>
Using the Job Queue Editor (Oracle)

The Job Queue editor lets you modify settings and options of existing jobs.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Modify the definition of a Job (Oracle)
- Manage the schedule and status of a Job (Oracle)

You also use this editor to perform the following common Object editor tasks:

- View SQL/DDL for a database object

Modify the definition of a Job (Oracle)

The Definition tab of the Job Queue Editor lets you modify the WHAT parameter value for a job.

To modify the code executed for a job:

1. Open an object editor on the job queue to be modified.
2. Click the Definition tab.
3. Modify the statement, function call, or procedure call in the Job Definition area.
4. Preview and submit changes.

Manage the schedule and status of a Job (Oracle)

The Properties tab of the Job Queue Editor lets you change a job's schedule and status.

To modify job interval or enable/disable a job:

1. Open an object editor on the job queue to be modified.
2. Click the Properties tab.
3. Use the settings in the Schedule group to specify that the job is to run once only or is to run according to a schedule. If it is to run on a schedule, specify the interval in days or using a custom date expression.
4. Use the settings in the Status group to enable or disable the job.
5. Preview and submit changes.
Using the Libraries Editor (Oracle)
The Libraries editor lets you modify settings and options of existing libraries.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

• Display definition information for a Library (Oracle)

You also use this editor to perform the following common Object editor tasks:

• View SQL/DDL for a database object
• Work with privileges and permissions
• Work with object dependencies

Lastly, the Command menu on the Object editor toolbar provides the following command:

• Rename

Display definition information for a Library (Oracle)
The Definition tab of the Libraries Editor lets you modify the file name and path for a library and view the current status and whether the library is dynamic.

To work with basic definition information for a library:
1 Open an object editor on the library.
2 Click the Definition tab.
3 Optionally, change the file and path specification in the File Name field.
4 Preview and submit changes.

Using the Linked Servers Editor (Microsoft SQL Server)
The Linked Servers editor lets you modify settings and options of existing linked servers.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

• Manage Linked Server options and functions (Microsoft SQL Server)
• Manage logins for Linked Servers (Microsoft SQL Server)

You also use this editor to perform the following common Object editor tasks:

• Work with privileges and permissions
Manage Linked Server options and functions (Microsoft SQL Server)

The **Information** tab of the Linked Servers Editor lets you view and modify options and functions available for linked servers.

**To manage linked server options and functions:**

1. **Open an object editor** on the linked server to be modified.
2. **Click the Information tab.**
3. **Use the table below to make modifications.**
4. **Preview and submit changes.**

The table below describes the options and functionality on the **Information** tab of the Linked Servers Editor:

<table>
<thead>
<tr>
<th>Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server</td>
<td>Indicates a linked server defined through Microsoft SQL Server. The name must be the server’s network name. Any tables retrieved from the server are from the default database defined for the login on the linked server.</td>
</tr>
<tr>
<td>OLE DB Product Name</td>
<td>If the server is not an instance Microsoft SQL Server, click the provider list, and then click the name of the OLE DB provider managing access to the specified linked server. An OLE DB provider is a software component that exposes OLE DB interfaces. Each OLE DB provider exposes data from a particular type of data source (for example SQL Server databases, Access databases, or Excel spreadsheets).</td>
</tr>
<tr>
<td>Datasource</td>
<td>Lets you type the name of the OLE DB provider managing the access to the linked server. An OLE DB provider is a software component that exposes OLE DB interfaces. Each OLE DB provider exposes data from a particular type of data source (for example SQL Server databases, Access databases, or Excel spreadsheets).</td>
</tr>
<tr>
<td>OLE DB Provider Name</td>
<td>If the server is not an instance Microsoft SQL Server, click the provider list, and then click the name of the OLE DB provider managing access to the specified linked server. An OLE DB provider is a software component that exposes OLE DB interfaces. Each OLE DB provider exposes data from a particular type of data source (for example SQL Server databases, Access databases, or Excel spreadsheets).</td>
</tr>
<tr>
<td>Provider Location</td>
<td>Lets you type the OLE DB location property corresponding to the linked server. An OLE DB provider is a software component that exposes OLE DB interfaces. Each OLE DB provider exposes data from a particular type of data source (for example SQL Server databases, Access databases, or Excel spreadsheets).</td>
</tr>
<tr>
<td>Provider Connection String</td>
<td>Lets you type the OLE DB provider string property corresponding to the linked server. An OLE DB provider is a software component that exposes OLE DB interfaces. Each OLE DB provider exposes data from a particular type of data source (for example SQL Server databases, Access databases, or Excel spreadsheets).</td>
</tr>
<tr>
<td>Catalog</td>
<td>Lets you type the OLE DB catalog property corresponding to the linked server. An OLE DB provider is a software component that exposes OLE DB interfaces. Each OLE DB provider exposes data from a particular type of data source (for example SQL Server databases, Access databases, or Excel spreadsheets).</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the **Information** tab of the Linked Servers Editor:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Access Server</td>
<td>Indicates whether the target linked server is enabled for distributed query access.</td>
</tr>
<tr>
<td>RPC Out Server</td>
<td>Lets you select the server with RPC encryption.</td>
</tr>
<tr>
<td>RPC</td>
<td>Enables RPC encryption from the server.</td>
</tr>
</tbody>
</table>
Manage logins for Linked Servers (Microsoft SQL Server)

The Logins tab of the Linked Servers Editor lets you manage every login for every linked server on the current server. This tab lets you:

- View linked logins.
- Add linked logins.
- Modify linked logins.
- Drop linked logins.

To manage logins for a linked server:

1. Open an object editor on the linked server to be modified.
2. Click the Logins tab.
3. Optionally, create a new login by clicking Add, use the following table as a guide to completing the settings on the Create Linked Login dialog, and click an execution option when complete:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linked Server Name</td>
<td>Displays the linked server name. Linked servers let you create links to OLE DB data sources.</td>
</tr>
<tr>
<td>Optional Settings</td>
<td>Lets you set optional settings: Use SQLServer authenticated login credentials to login to linked server. No remote login or password used to login to linked server. All users will use the same login to login to the linked server.</td>
</tr>
<tr>
<td>Local Login Name</td>
<td>Lets you select a local login name.</td>
</tr>
<tr>
<td>Linked Login Name</td>
<td>Lets you type in a linked login name.</td>
</tr>
<tr>
<td>Linked Password</td>
<td>Lets you enter a password.</td>
</tr>
<tr>
<td>Confirm Password</td>
<td>Confirms the password.</td>
</tr>
</tbody>
</table>

4. Optionally, select a login, click the Modify button, and change the login names and password.
5. Optionally, select a login and click the Drop button to delete that login.
Using the Logins Editor (Microsoft SQL Server, Sybase)

The Logins editor lets you modify settings and options of existing logins.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to [Open an object editor](#), [View and modify database object properties](#), and how to [Preview and submit changes](#). For detailed information, see [Overview and common usage of Object editors](#).

You use this editor to perform the following tasks specific to this object type:

- Manage Login information (Microsoft SQL Server)
- Manage user accounts for Logins (Microsoft SQL Server, Sybase)
- Manage roles for Logins (Microsoft SQL Server, Sybase)
- Manage accounting statistics for Logins (Sybase)

You also use this editor to perform the following common Object editor tasks:

- [View SQL/DDL for a database object](#)

Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Password</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Like</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Manage Login information (Microsoft SQL Server)

The **Definition** tab of the Logins Editor lets you view and modify information for every login on the current server.

To manage basic definition for a login:

1. [Open an object editor](#) on the login to be modified.
2. Click the **Definition** tab.
3. Use the table below as a guide to modifying values on this tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Database</td>
<td>The database that Microsoft SQL Server places the target user into when they log in to the target server.</td>
</tr>
<tr>
<td>Default Language</td>
<td>The language that Microsoft SQL Server uses to displays the target login's system messages.</td>
</tr>
<tr>
<td>Currently Logged In</td>
<td>The login status of the target login.</td>
</tr>
</tbody>
</table>

4. [Preview and submit changes](#).
Manage Login information (Sybase)

The **Definition** tab of the Logins Editor lets you view and modify information for every login on the current server.

**To manage basic definition for a login:**

1. Open an object editor on the login to be modified.
2. Click the **Definition** tab.
3. Use the table below as a guide to modifying values on this tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Name</td>
<td><strong>OPTIONAL:</strong> A user-defined descriptive name for the target login.</td>
</tr>
<tr>
<td>Default Database</td>
<td>The database that Sybase ASE places the target user into when they log in to the target server.</td>
</tr>
<tr>
<td>Default Language</td>
<td>The language that Sybase ASE uses to displays the target login’s system messages.</td>
</tr>
<tr>
<td>Password Status</td>
<td>The status, current, expired, of the password.</td>
</tr>
<tr>
<td>Last Password Change</td>
<td>The data and time of the last time that the target login’s password changed.</td>
</tr>
<tr>
<td>Password Expiration</td>
<td>Duration, in days, before passwords expire.</td>
</tr>
<tr>
<td>Minimum Password Length</td>
<td>Minimum number of characters in a password.</td>
</tr>
<tr>
<td>Maximum Login Attempts</td>
<td>Maximum number of login attempts before the login (account) is suspended.</td>
</tr>
<tr>
<td>Currently Logged In</td>
<td>The login status of the target login.</td>
</tr>
<tr>
<td>Account Status</td>
<td>Whether the login is locked or unlocked.</td>
</tr>
</tbody>
</table>

**TIP:** To identify the login, in the Full Name box, type the actual name of the user who uses the target login.

4. Preview and submit changes.

Manage user accounts for Logins (Microsoft SQL Server, Sybase)

The **Users** tab of the Logins Editor lets you manage each login’s user account(s) for every database on the current server. Before a login name can access a user database, it must be added as a user or alias to that database by the system administrator or database owner.

**NOTE:** User ids and passwords are not shared across Windows servers unless the Security Account Delegation option is turned on in the AD Security groups.

**To manage database accounts for logins:**

1. Open an object editor on the login to be modified.
2. Click the **Users** tab.
3. Use the **Add User** button to add a selected database from the **Databases where the login does NOT have a user account** to the **Databases where the login HAS a user account** list. Conversely, remove a login from a database by using the **Drop User** button to move a selected database from the **Databases where the login HAS a user account** list.
4. Preview and submit changes.
Manage roles for Logins (Microsoft SQL Server, Sybase)

The (Server) Roles tab of the Logins Editor lets you manage every login’s role(s). The tab includes a table listing every role of which the target login is a member.

The Roles tab lets you:

- Grant Roles
- Assign Default Roles (Sybase only)
- Revoke Roles

To manage roles for a login:

1. Open an object editor on the login to be modified.
2. Click the Roles (Sybase) or Server Roles (Microsoft SQL Server) tab.
   Any existing roles assigned to this login are displayed on the tab.
3. Grant Roles or Revoke Role as needed.
4. Optionally, if you are working with Sybase, use the Def. Roles button to open a dialog that lets you add or delete roles from the set of default roles assigned to this login.

Manage accounting statistics for Logins (Sybase)

The Accounting tab of the Logins Editor lets you manage chargeback accounting statistics for every login on the current server. Chargeback accounting statistics are CPU and I/O usage statistics that Sybase ASE accumulates for every login. To start a new accounting period, the system administrator must clear all previous login statistics.

To view or clear accounting statistics for a login:

1. Open an object editor on the login to be modified.
2. Click the Accounting tab.
3. Click the Clear Statistics button to start a new accounting interval.
4. Preview and submit changes

Using the Materialized Query Tables Editor (IBM DB2 for Linux, Unix, and Windows)

The Materialized Query Tables editor lets you modify settings and options of existing materialized query tables.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage columns of a Materialized Query Table (IBM DB2 for Linux, Unix, and Windows)
- Manage query and query options for a Materialized Query Table (IBM DB2 for Linux, Unix, and Windows)
- Manage performance for a Materialized Query Table (IBM DB2 for Linux, Unix, and Windows)
• Display space details for a Materialized Query Table (IBM DB2 for Linux, Unix, and Windows)

You also use this editor to perform the following common Object editor tasks:

• Add a comment to a database object
• Work with privileges and permissions
• View SQL/DDL for a database object
• Work with object dependencies

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Query</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Alias</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Like</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create View</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td>For details, see Creating an Object.</td>
</tr>
<tr>
<td>Describe</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drop Materialized Query</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td>For more information, see Drop</td>
</tr>
<tr>
<td>Indexes</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lock</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quiesce Tablespaces</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refresh Table</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td>For more information, see Refresh Table.</td>
</tr>
<tr>
<td>Reorganize</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td>For details, see Schema Wizard for IBM DB2 for Linux, Unix, and Windows.</td>
</tr>
<tr>
<td>Select * From</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set Integrity</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update Statistics</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Manage columns of a Materialized Query Table (IBM DB2 for Linux, Unix, and Windows)

The Columns tab of the Materialized Query Tables Editor lets you manage columns for every table on the current datasource. The Columns tab lets you:

• Add columns
• Insert columns
• Edit columns
• Drop columns
To manage the columns of a materialized query table:
1. Open an object editor on the materialized query table to be modified.
2. Click the Columns tab.
3. Add, Insert, or Edit columns for the materialized query table.
4. Remove columns using the Drop button.
5. Preview and submit changes.

Manage query and query options for a Materialized Query Table (IBM DB2 for Linux, Unix, and Windows)
The Definition tab of the Materialized Query Tables Editor lets you manage the SQL for every table on the current datasource.

To manage basic definition details for a materialized query table:
1. Open an object editor on the materialized query table to be modified.
2. Click the Definition tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Query</td>
<td>Lets you view the base query.</td>
</tr>
<tr>
<td>Materialized Query Table</td>
<td>Lets you specify an immediate or deferred Refresh, and explicitly set a</td>
</tr>
<tr>
<td>Options</td>
<td>Materialized Query Table as Enabled or Disabled or go with the Default.</td>
</tr>
<tr>
<td></td>
<td>This tab also displays whether the Materialized Query Table is maintained</td>
</tr>
<tr>
<td></td>
<td>by the system or by a user.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

Manage performance for a Materialized Query Table (IBM DB2 for Linux, Unix, and Windows)
The Performance tab of the Materialized Query Tables Editor lets you manage performance for every table on the current datasource.

To manage performance settings for a materialized query table:
1. Open an object editor on the materialized query table to be modified.
2. Click the Performance tab.
3. Use the table below as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace Placement</td>
<td>Displays Table Tablespace, Index Tablespace, and Long Tablespace settings for the materialized query table.</td>
</tr>
</tbody>
</table>
Display space details for a Materialized Query Table (IBM DB2 for Linux, Unix, and Windows)

The Space tab of the Materialized Query Tables Editor lets you view page and row space usage details for materialized query tables on the current datasource.

To view space details for a materialized query table:

1. Open an object editor on the materialized query table to be modified.
2. Click the Storage tab.
3. Use the table below as a guide to understanding the settings on this tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Information</td>
<td>Lets you view Row Count and Num. of Overflow Rows values.</td>
</tr>
<tr>
<td>Row Information</td>
<td>Lets you view Num. of Pages with Rows, Num. of Pages, Percent Free values.</td>
</tr>
</tbody>
</table>

Using the Materialized Views (Snapshot) Editor (Oracle)

The Materialized Views editor lets you modify settings and options of existing materialized views.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- View Materialized View (Snapshot) properties (Oracle)
- Partition Materialized Views (Oracle)
- Manage storage for a Materialized View (Snapshot) (Oracle)
- Display queries for a Materialized View (Oracle)

You also use this editor to perform the following common Object editor tasks:

- Work with privileges and permissions
• View SQL/DDL for a database object
• Work with object dependencies

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Synonym</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

View Materialized View (Snapshot) properties (Oracle)
The Information tab of the Materialized Views Editor displays the status and size information for every materialized view on the current datasource.

To manage properties for a materialized view:
1. Open an object editor on the materialized view to be modified.
2. Click the Information tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>Lets you work with the Table Name and Master View properties.</td>
</tr>
<tr>
<td>Last Refresh</td>
<td>Lets you work with the Last Date and Errors properties.</td>
</tr>
<tr>
<td>Refresh Configuration</td>
<td>Lets you work with the Type, Refresh Method, and Mode properties.</td>
</tr>
<tr>
<td>Rollback Usage</td>
<td>Lets you work with the Local and Master properties.</td>
</tr>
<tr>
<td>Options</td>
<td>Lets you work with the Updatable and Enable Query Rewrite properties.</td>
</tr>
</tbody>
</table>
4. Preview and submit changes.

Partition Materialized Views (Oracle)
The Partitions tab of the Materialized Views Editor lets you view partitioning columns and manage partitions on a materialized view.

To manage materialized view partitions:
1. Open an object editor on the materialized view to be modified.
2. Click the Partitions tab.
3. Optionally take one of the following actions:
   • Click Add to create a new partition.
   • Select a partition and click Edit to modify the partition details.
   • Select a partition and click Drop to delete that partition.
4 Optionally, use one of the following **Command** menu items:

- Allocate Extent
- Deallocate Unused Space
- Mark Local Indexes Unusable
- Rebuild Unusable Local Indexes
- Split Partition
- *Preview and submit changes.*

### Manage storage for a Materialized View (Snapshot) (Oracle)

The **Storage** tab of the Materialized Views Editor lets you manage storage for every Materialized View on the current datasource.

**To manage storage settings for a materialized view:**

1. [Open an object editor](#) on the materialized view to be modified.
2. Click the **Storage** tab.
3. Use the following table to understand and modify the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Placement</td>
<td>Lets you work with <strong>Tablespace</strong> and <strong>Cluster</strong> values.</td>
</tr>
<tr>
<td>Data Block Storage</td>
<td>Lets you work with <strong>Percent Free</strong>, <strong>Percent Used</strong>, <strong>Initial transactions</strong>, and <strong>Maximum Used</strong> values.</td>
</tr>
<tr>
<td>Extents</td>
<td>Lets you work with <strong>Initial Extent</strong>, <strong>Next Extent</strong>, <strong>Minimum Extents</strong>, <strong>Maximum Extents</strong>, and <strong>Percent Increase</strong> values.</td>
</tr>
</tbody>
</table>

4. [Preview and submit changes](#).

### Display queries for a Materialized View (Oracle)

The **Query** tab of the Materialized Views Editor displays SQL for materialized view queries on the current datasource.

**To view a materialized view query:**

1. [Open an object editor](#) on the materialized view to be modified.
2. Click the **Query** tab.

   The SQL query is displayed in the **Query** area.

### Using the Materialized View (Snapshot Logs) Logs editor (Oracle)

The Materialized View Logs editor lets you modify settings and options of existing materialized view logs.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to [Open an object editor](#), [View and modify database object properties](#), and how to [Preview and submit changes](#). For detailed information, see [Overview and common usage of Object editors](#).
You use this editor to perform the following tasks specific to this object type:

- View log table information for a Materialized View Log (Snapshot Logs) (Oracle)
- Manage storage for a Materialized View Log (Snapshot Logs) (Oracle)
- Manage performance for a Materialized View Log (Oracle)

You also use this editor to perform the following common Object editor tasks:

- View SQL/DDL for a database object
- Work with object dependencies

View log table information for a Materialized View Log (Snapshot Logs) (Oracle)

The Information tab of the Materialized View Logs Editor displays information on the log table used for replication.

To view replication log table details for a materialized view log:
1. Open an object editor on the materialized view log to be modified.
2. Click the Information tab.

Manage storage for a Materialized View Log (Snapshot Logs) (Oracle)

The Storage tab of the Materialized View Logs Editor lets you manage storage for materialized view logs on the current datasource.

To manage storage settings for a materialized view log:
1. Open an object editor on the materialized view log to be modified.
2. Click the Storage tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Block Storage</td>
<td>Lets you view the associated Tablespace and Initial Transactions value, and modify the Percent Free, Percent Used, and Max Transactions values.</td>
</tr>
<tr>
<td>Extents</td>
<td>Lets you view Initial Extent, Next Extent, Minimum Extents, Maximum Extents, and Percent Increase values.</td>
</tr>
<tr>
<td>Column Filtering</td>
<td>Lets you select the filter columns to be recorded in the materialized view log. You can specify only one primary key, one ROWID and one filter column list per materialized view log. The ROWID is a globally unique identifier for a row in a database. It is created at the time the row is inserted into a table, and destroyed when it is removed from a table.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes

Manage performance for a Materialized View Log (Oracle)

The Performance tab of the Materialized View Logs Editor lets you manage performance for every materialized view log on the current datasource.
To manage performance for a materialized view log:

1. **Open an object editor** on the materialized view log to be modified.

2. **Click the Performance tab.**

3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
</table>
| **Parallel Query Option** | The Parallel server query option lets you process queries using many query server processes running against multiple CPUs. This option provides substantial performance gains such as reduction of the query completion time.  
  Degrees - Lets you type a value indicating the number of query server processes that should be used in the operation.  
  Instances - Lets you type a value indicating how you want the parallel query partitioned between the Parallel Servers. |
| **Logging**            | Select **Logging** to create a log for all Materialized View updates.                                                                     |
| **Cache**              | Select **Cache** if you want Oracle to put data you access frequently at the most recently used end of the list in the buffer cache when a full table scan is performed. This option is useful for small lookup tables. |

4. **Preview and submit changes.**

**Using the Nodegroups editor (IBM DB2)**

The Nodegroups editor lets you modify settings and options of existing nodegroups.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to **Open an object editor**, **View and modify database object properties**, and how to **Preview and submit changes**. For detailed information, see **Overview and common usage of Object editors**.

You use this editor to perform the following tasks specific to this object type:

- **Manage nodegroup partitions (IBM DB2)**

You also use this editor to perform the following common Object editor tasks:

- **View SQL/DDL for a database object**
- **Add a comment to a database object**

Lastly, the **Command** menu on the Object editor toolbar provides the following command:

- **Redistribute**

**Manage nodegroup partitions (IBM DB2)**

The **Partitions** tab of the Nodegroups Editor lets you add partitions to, and remove partitions from any defined Nodegroup.

**To manage partitions for a nodegroup:**

1. **Open an object editor** on the nodegroups to be modified.

2. **Click the Partitions tab.**
3 Use the **Join Nodegroup** and **Leave Nodegroups** buttons to move nodegroups between the **Partitions Not in Nodegroup** and **Partitions in Nodegroup** lists.

4 **Preview and submit changes.**

### Outlines Editor for Oracle

The Outlines Editor lets you:

- View outline properties.
- View and modify outline definition.
- View outline DDL.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to **Open an object editor**, **View and modify database object properties**, and how to **Preview and submit changes**. For detailed information, see **Overview and common usage of Object editors**.

**NOTE:** The only SQL statements possible with stored outlines are SELECT, DELETE, UPDATE, INSERT...SELECT, and CREATE TABLE...AS SELECT.

### Using the Package Bodies Editor (Oracle)

The Package Bodies editor lets you modify settings and options of existing packages.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to **Open an object editor**, **View and modify database object properties**, and how to **Preview and submit changes**. For detailed information, see **Overview and common usage of Object editors**.

You use this editor to perform the following common Object editor tasks:

- **View SQL/DDL for a database object**

Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compile</strong></td>
<td></td>
<td></td>
<td><strong>YES</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Create Synonym</strong></td>
<td></td>
<td></td>
<td><strong>YES</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Using the Packages editor (IBM DB2, Oracle)

The Packages editor lets you modify settings and options of existing packages.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to **Open an object editor**, **View and modify database object properties**, and how to **Preview and submit changes**. For detailed information, see **Overview and common usage of Object editors**.
You use this editor to perform the following tasks specific to this object type:

- **View packages definitions (IBM DB2 for Windows, Linux, and Unix)**
- **Manage package properties (IBM DB2 for OS/390 and Z/OS)**
- **Display bind parameters of a package (IBM DB2 for OS/390 and Z/OS)**
- **Display plans contained in a package (IBM DB2 for OS/390 and Z/OS)**
- **Display the contents of a package statement (IBM DB2 for OS/390 and Z/OS)**
- **Manage run-time environment information for a package (IBM DB2 for OS/390 and Z/OS)**
- **Display the command originally used to build a package (IBM DB2 for OS/390 and Z/OS)**
- **Modify the package header specifications (Oracle)**
- **Modify the package body specifications (Oracle)**
- **Display status and size details for a package (Oracle)**

You also use this editor to perform the following common Object editor tasks:

- **Add a comment to a database object**
- **View SQL/DDL for a database object**
- **Work with object dependencies**

Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Synonym</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flush Cache</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>Rebind Packages</td>
</tr>
<tr>
<td>Rebind</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**View packages definitions (IBM DB2 for Windows, Linux, and Unix)**

The **Definition** tab of the Packages Editor displays basic definition settings for a package.

**To display definition settings for a package:**

1. **Open an object editor** on the package to be modified.
2. **Click the Definition tab.**

The following table lists the settings available on this tab:

<table>
<thead>
<tr>
<th>Binder</th>
<th>Definer</th>
<th>Default Schema</th>
<th>Degree</th>
<th>Function Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language Level</td>
<td>SQL Math Warn</td>
<td>SQL Warn</td>
<td>Buffered Inset</td>
<td>Status</td>
</tr>
<tr>
<td>Code Page</td>
<td>Total Sections</td>
<td>Multi-node Bound</td>
<td>Intra-partition</td>
<td>Query Optimization</td>
</tr>
<tr>
<td>Cursor Blocking</td>
<td>Isolation Level</td>
<td>Date Time Format</td>
<td>Last Bind Time</td>
<td>Explicit Bind Time</td>
</tr>
<tr>
<td>Explain Level</td>
<td>Explain Snapshot</td>
<td>Explain Mode</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Manage package properties (IBM DB2 for OS/390 and Z/OS)
The Properties tab of the Packages Editor displays properties of any package on the datasource.

To view properties for a package:
1. Open an object editor on the package to be modified.
2. Click the Properties tab.

The following table lists the settings available on this tab:

<table>
<thead>
<tr>
<th>Average Size</th>
<th>Bind Time</th>
<th>Consistency Token</th>
<th>Dec31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal Point</td>
<td>Group Member</td>
<td>Katakana Charset</td>
<td>Language</td>
</tr>
<tr>
<td>Mixed Character Set</td>
<td>Operative</td>
<td>Package Size</td>
<td>Package Source</td>
</tr>
<tr>
<td>PDS Name</td>
<td>Precompile Timestamp</td>
<td>Release Bound</td>
<td>String Delimiter</td>
</tr>
<tr>
<td>SysEntries</td>
<td>Timestamp</td>
<td>Valid</td>
<td></td>
</tr>
</tbody>
</table>

Display bind parameters of a package (IBM DB2 for OS/390 and Z/OS)
The Bind Parameters tab of the Packages Editor displays bind parameters of any package on the datasource.

To work with bind parameters of a package:
1. Open an object editor on the package to be modified.
2. Click the Bind Parameters tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Properties</th>
</tr>
</thead>
</table>

4. Preview and submit changes

Display plans contained in a package (IBM DB2 for OS/390 and Z/OS)
The Plan/Packlists tab of the Packages Editor displays the plans contained in the package if the package was bound individually, or as part of a complete collection id.

To work with the plans of a package:
1. Open an object editor on the package to be modified.
2. Click the Plan/Packlists tab.
3. Optionally, take one of the following actions:
   - Select a plan and click Edit to open object editor for that plan.
   - Select a plan and click Rebind to open the Rebind Plans dialog.
   - Select a plan and click Free to open the Free Plans dialog.
4  **Preview and submit changes.**

Display the contents of a package statement (IBM DB2 for OS/390 and Z/OS)
The **Statements** tab of the Packages Editor shows the contents of any package statement on the datasource.

To copy package statement SQL to an ISQL Editor window:
1  **Open an object editor** on the package to be modified.
2  Click the **Statements** tab.
3  Select a statement and click **Explain** to copy the statement to an ISQL Editor window.

**NOTE:** For more information, see ISQL Editor.

Manage run-time environment information for a package (IBM DB2 for OS/390 and Z/OS)
The **Environments** tab of the Packages Editor displays run-time environments information for every package on the datasource.

To manage environments for a package:
1  **Open an object editor** on the package to be modified.
2  Click the **Environments** tab.
3  Optionally, take one of the following actions:
   • Use the arrow buttons to move environments between the **Enabled Environments** and **Disabled Environments** lists.
   • Select an environment and click **Edit Connections** to open the Connection Editor.
4  **Preview and submit changes.**

Display the command originally used to build a package (IBM DB2 for OS/390 and Z/OS)
The **Commands** tab of the Packages Editor displays the command that originally built the package, for every package on the datasource.

To display the command for a package:
1  **Open an object editor** on the package to be modified.
2  Click the **Commands** tab.

Modify the package header specifications (Oracle)
The **Header** tab of the Packages Editor lets you modify the package header specifications.
To modify a package header:
1. Open an object editor on the package to be modified.
2. Click the Header tab.
3. Modify the text of the header specification in the Package Header Text area.
4. Preview and submit changes.

Modify the package body specifications (Oracle)
The Body tab of the Packages Editor lets you modify the package body specifications.

To modify the body of a package:
1. Open an object editor on the package to be modified.
2. Click the Body tab.
3. Modify the text of the package body in the Package Body area.
4. Preview and submit changes.

Display status and size details for a package (Oracle)
The Information tab of the Packages Editor displays the status and size information for package headers and bodies on the current datasource.

To view status and size information for package headers and bodies:
1. Open an object editor on the package to be modified.
2. Click the Information tab.
3. Use the following table as a guide to understanding the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header and Body</td>
<td>Lets you view Status, Created, and Last Modified Source Size, Parsed Size, Code Size, and Error Size details for the header and body of the package.</td>
</tr>
</tbody>
</table>

4. Optionally, use the Retrieve Size Info button to update the Size values.
5. Preview and submit changes.

Using the Plans Editor (IBM DB2 for OS/390 and z/OS)
The Plans editor lets you modify settings and options of existing plans.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage Plan properties (IBM DB2 for OS/390 and z/OS)
• View DBRM information for a Plan (IBM DB2 for OS/390 and z/OS)
• Manage packages in a Plan (IBM DB2 for OS/390 and z/OS)
• Manage contents of a Plan (IBM DB2 for OS/390 and z/OS)
• Manage run-time environment information for a Plan (IBM DB2 for OS/390 and z/OS)
• View the originating command for a Plan (IBM DB2 for OS/390 and z/OS)

You also use this editor to perform the following common Object editor tasks:
• Work with privileges and permissions
• Work with object dependencies

Manage Plan properties (IBM DB2 for OS/390 and z/OS)
The Definition tab of the Plans Editor displays properties of any plan on the datasource.

To manage plan properties:
1. Open an object editor on the plan to be modified.
2. Click the Definition tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Lets you view the Timestamp, Group Member, Plan Size, Average Size, Valid, Operative, Pkg. List Entries, SysEntries, and Release Bound properties. Lets you set the Owner, Qualifier, and Current Server properties.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

View DBRM information for a Plan (IBM DB2 for OS/390 and z/OS)
The DBRMs tab of the Plans Editor lists details for each DBRM associated with a plan.

To view DBRM information for a plan:
1. Open an object editor on the plan to be modified.
2. Click the DBRMs tab.
3. Optionally, select a DBRM from the list and click Edit to open the DBRM editor on that DBRM. For more information, see Using the DBRM Editor (IBM DB2 for OS/390 and z/OS).

Manage packages in a Plan (IBM DB2 for OS/390 and z/OS)
The Packages tab of the Plans Editor displays details for each packages associated with a plan.
To manage packages associated with a plan:
1. Open an object editor on the plan.
2. Click the Packages tab.
3. Optionally, take one of the following actions:
   - Select a package and click Edit to open the Package editor on that package. For details, see Using the Packages editor (IBM DB2, Oracle).
   - Select a package and click Rebind to open the Rebind Packages dialog.
   - Select a package and click Free to open the Free Packages dialog.

Manage contents of a Plan (IBM DB2 for OS/390 and z/OS)
The DBRM/Packages tab of the Plans Editor displays the entire contents of the plan, DBRMs and packages, in a single display. The first column contains either a 'D' for DBRM or 'P' for packages.

To work with plan contents:
1. Open an object editor on the plan.
2. Click the DBRM/Packages tab.
3. Optionally, take one of the following actions:
   - Select a package list entry and click Edit to open the Package editor on that package. For details, see Using the Packages editor (IBM DB2, Oracle).
   - Select a package list entry and click Rebind to open the Rebind Packages dialog.
   - Select a package list entry and click Free to open the Free Packages dialog.
4. Preview and submit changes.

Manage run-time environment information for a Plan (IBM DB2 for OS/390 and z/OS)
The Environments tab of the Plans Editor displays run-time environments information for every plan on the datasource.

To manage environments for a plan:
1. Open an object editor on the plan to be modified.
2. Click the Environments tab.
3. Optionally, take one of the following actions:
   - Use the arrow buttons to move environments between the Enabled Environments and Disabled Environments lists.
   - Select an environment and click Edit Connections to open the Connection Editor.
4. Preview and submit changes.
View the originating command for a Plan (IBM DB2 for OS/390 and z/OS)

The Commands tab of the Plans Editor displays the command that originally built the plan, for every plan on the datasource.

To view the command for a plan:
1. Open an object editor on the plan.
2. Click the Commands tab.

Using the Primary Keys editor

The Primary Keys editor lets you modify settings and options of existing primary keys.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Specify Primary Key columns (IBM DB2, Microsoft SQL Server, Oracle, Sybase)
- Manage space allocations for a primary key (IBM DB2, Oracle)
- Manage Primary Key properties (IBM DB2, Microsoft SQL Server, Sybase)
- Manage storage for Primary Keys (Oracle)
- Display page and row statistics for a primary key (Microsoft SQL Server, Sybase)
- Manage Primary Key partitions (Oracle)

You also use this editor to perform the following common Object editor tasks:

- View SQL/DDL for a database object
- Add a comment to a database object

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type.

The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocate Extent</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deallocate Unused Space</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rebuild</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td>YES</td>
<td>YES</td>
<td>Yes</td>
<td>YES</td>
<td>Yes</td>
</tr>
<tr>
<td>Status</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>
Specify Primary Key columns IBM DB2, Microsoft SQL Server, Oracle, Sybase)

The **Columns** tab of the Primary Keys Editor lets you specify the columns in the index for any primary key on the datasource.

**To manage the columns for a primary key:**

1. Open an object editor on the primary key to be modified.
2. Click the **Columns** tab.
   - Details for the columns making up the primary key are listed on the tab.
3. Optionally, take one of the following actions:
   - Select a column and click the New button to add a new column to the primary key.
   - Select a column and click the Delete button to remove the column from the primary key.
4. **Preview and submit changes.**

Manage space allocations for a primary key (IBM DB2, Oracle)

The **Space** tab of the Primary Keys Editor lets you view storage attributes, statistics, and card details for primary keys on the current datasource.

**To manage space for a primary key:**

1. Open an object editor on the primary key to be modified.
2. Click the **Space** tab.
3. Use the following table as a guide to understanding the settings displayed:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes (IBM DB2)</td>
<td>Lets you view <strong>Make Unique</strong>, <strong>System Required</strong>, <strong>Total Keys</strong>, <strong>Page Fetch Pairs</strong>, and <strong>Distinct Keys</strong> settings.</td>
</tr>
<tr>
<td>Statistics</td>
<td>For IBM DB2, lets you view <strong>Index Level</strong>, <strong>Cluster Ratio</strong>, <strong>Cluster Factor</strong>, <strong>Leaf Pages</strong>, <strong>Sequential Pages</strong>, and <strong>Density</strong> properties. For Oracle, lets you view <strong>Index Level</strong>, <strong>Distinct Keys</strong>, <strong>Cluster Factor</strong>, <strong>Leaf Blocks</strong>, <strong>Leaf Blks/Key</strong>, and <strong>Data Blks/Key</strong> properties.</td>
</tr>
<tr>
<td>Cards (IBM DB2 for Windows, Unix, and Linux)</td>
<td>For IBM DB2 for Windows, Unix, and Linux, lets you view <strong>First Key</strong>, <strong>First 2 Keys</strong>, <strong>First 3 Keys</strong>, <strong>First 4 Keys</strong> and <strong>Full Keys</strong> settings. For IBM DB2 for OS/390 and Z/OS, lets you view <strong>First Key</strong> and <strong>Full Keys</strong> properties.</td>
</tr>
<tr>
<td>Extents (Oracle)</td>
<td>Displays the <strong>No. of Extents</strong> and <strong>Max Extents</strong>, and for each extent, the <strong>Extent ID</strong>, <strong>File ID</strong>, <strong>Block ID</strong>, and number of <strong>Blocks</strong>.</td>
</tr>
</tbody>
</table>

Manage Primary Key properties (IBM DB2, Microsoft SQL Server, Sybase)

The **Properties** tab of the Primary Keys Editor lets you manage properties for every primary key on the current datasource.
To modify the properties of a primary key:
1. Open an object editor on the primary key to be modified.
2. Click the Properties tab.
3. Use the following table as a guide to setting options in the General group:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>For IBM DB2 for Windows, Unix, and Linux, lets you view the Defined by setting. For IBM DB2 for OS/390 and Z/OS, lets you set Buffer Pool, Close, Copy and Piece Size properties. For Microsoft SQL Server, lets you set the Clustered properties. For Oracle, lets you view Deferred, Cascade, and Update Date properties. Lets you set No Sort, Logging, Reverse, Deferrable, Validate, Enabled, and Rely properties. For Sybase, lets you set the Clustered and Maximum Rows Per Page properties.</td>
</tr>
<tr>
<td>Storage</td>
<td>For IBM DB2 for Windows, Unix, and Linux, lets you view the Percent Free and Minimum Percent Used settings. For Microsoft SQL Server, lets you set the File Group and File Factor option. For Sybase, lets you set Reserve Page Gap, Segment Name, and Fill Factor properties.</td>
</tr>
<tr>
<td>Enforcing Index (Oracle)</td>
<td>Lets you view User Defined, Index Owner, and Index Name properties.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

Manage storage for primary keys (IBM DB2 for OS/390 and Z/OS)
The Storage tab of the Primary Keys Editor lets you manage storage for every primary key on the current datasource.

To manage storage settings for a primary key:
1. Open an object editor on the primary key to be modified.
2. Click the Storage tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
</table>

4. Optionally, use the Edit button to initiate managing storage group and VCAT catalogue options.
5. Preview and submit changes.

Manage storage for Primary Keys (Oracle)
The Storage tab of the Primary Keys Editor lets you manage storage for every primary key on the current datasource.
To manage storage settings for a primary key:
1. **Open an object editor** on the primary key to be modified.
2. Click the **Storage** tab.
3. Use the table below as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Block Storage</td>
<td>Lets you specify <strong>Tablespace, Percent Free, Initial Transactions, and Max Transactions</strong> values.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> You should never place primary keys on the SYSTEM tablespace.</td>
</tr>
<tr>
<td>Extents</td>
<td>Displays <strong>Initial Extent, Next Extent, Percent Increase, Minimum Extents</strong> and <strong>Maximum Extents</strong> values.</td>
</tr>
<tr>
<td>Freestack</td>
<td>Lets you specify <strong>Freestack</strong> and <strong>Freestack Groups</strong> values.</td>
</tr>
<tr>
<td>Buffer Pool</td>
<td>Lets you specify a buffer pool.</td>
</tr>
</tbody>
</table>

4. **Preview and submit changes.**

Display page and row statistics for a primary key (Microsoft SQL Server, Sybase)
The **Statistics** tab of the Primary Keys Editor lets you view the page and row statistics for every primary key on the current datasource.

To view page and row statistics for a primary key:
1. **Open an object editor** on the index to be modified.
2. Click the **Statistics** tab.
3. Use the following table as a guide to understanding the settings displayed:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Statistics</td>
<td>Data Pages, Reserved Pages, Used Pages, OAM Page Ratio (Sybase only), Index Page Ratio (Sybase only), and <strong>Total Pages Modified</strong> (Microsoft SQL Server only)</td>
</tr>
<tr>
<td>Row Statistics</td>
<td><strong>Maximum Rows Per Page</strong> (Sybase only), <strong>Minimum Row Size</strong>, <strong>Maximum Row Size</strong>, <strong>Max Size of Non-Leaf Index Row</strong>, and <strong>Total Rows Modified Since Last</strong> (Microsoft SQL Server only)</td>
</tr>
</tbody>
</table>

Manage Primary Key partitions (Oracle)
The **Partitions** tab of the Primary Keys Editor lets you partition primary keys on the current datasource.

**NOTE:** If the primary key is not currently partitioned, this tab shows a **Partition** button you can use to partition it. For more information, see **Convert to Partitioned**.

To manage partitions for a primary key:
1. **Open an object editor** on the primary key to be modified.
2. Click the **Partitions** tab.
3 Use the following table as a guide to understanding the settings on this page:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td>Lets you view the <strong>Locality</strong> (Global/Local), <strong>Alignment</strong> (Prefixed/Non-Prefixed), <strong>Partition Type</strong> (including Range-Hash Composite or Range-List Composite), and <strong>Subpartition type</strong> properties.</td>
</tr>
<tr>
<td><strong>Columns</strong></td>
<td>Displays partitioning columns.</td>
</tr>
<tr>
<td><strong>Partition Definitions</strong></td>
<td>Displays details for each partition.</td>
</tr>
</tbody>
</table>

**Using the Procedures editor**

The Procedures editor lets you modify settings and options of existing procedures.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to [Open an object editor](#), [View and modify database object properties](#), and how to [Preview and submit changes](#). For detailed information, see [Overview and common usage of Object editors](#).

You use this editor to perform the following tasks specific to this object type:

- [Modify procedure properties (IBM DB2 for OS/390 and Z/OS)](#)
- [Modify procedure properties (IBM DB2 for Windows, Unix, and Linux)](#)
- [Modify a procedure definition (Microsoft SQL Server, Oracle, Sybase)](#)
- [Modify the body of a procedure (IBM DB2 for Windows, Unix, and Linux)](#)
- [Manage procedure parameters (IBM DB2)](#)
- [View status and size details for a procedure (Oracle)](#)

You also use this editor to perform the following common Object editor tasks:

- [Work with privileges and permissions](#)
- [Add a comment to a database object](#)
- [View SQL/DDL for a database object](#)
- [Work with object dependencies](#)
Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Synonym</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Debug</td>
<td>YES</td>
<td>YES</td>
<td>Yes</td>
<td>YES</td>
<td>Yes</td>
<td>For more information, see Add-On Tools</td>
</tr>
<tr>
<td>Describe</td>
<td>YES</td>
<td>YES</td>
<td>Yes</td>
<td>YES</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td>YES</td>
<td>YES</td>
<td>Yes</td>
<td>YES</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** For SQL stored procedures, you can change the build options by opening the Stored Procedure Build Option dialog box. Select **Build** from the **Command** menu drop-down list.

**Modify procedure properties (IBM DB2 for OS/390 and Z/OS)**

The **Properties** tab of the Procedures Editor lets you modify a procedure to change its definition. To modify a procedure, DBArtisan must drop then re-create the procedure.

To work with basic properties for a procedure:

1. **Open an object editor** on the procedure.
2. Click the **Properties** tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>Lets you view the <strong>Schema</strong> and <strong>Routine ID</strong> properties. Lets you set the <strong>External Name</strong> and <strong>Collection ID</strong> properties.</td>
</tr>
<tr>
<td>Run Time</td>
<td>Lets you set the <strong>Result Sets</strong>, <strong>WLM Environment</strong>, <strong>WLM For Nested</strong>, <strong>ASU Time</strong>, and <strong>Run Options</strong> properties.</td>
</tr>
<tr>
<td>Structure</td>
<td>Lets you view the <strong>Language</strong>, <strong>Number of LOB Columns</strong>, <strong>Number of Parameters</strong>, and <strong>DBINFO</strong> properties. Lets you set the <strong>Program Type</strong>, <strong>Security Type</strong>, <strong>SQL Access Level</strong>, <strong>Parameter Style</strong>, <strong>Inherit Special Registers</strong>, <strong>Fenced</strong>, <strong>Commit on Return</strong>, <strong>Deterministic</strong>, and <strong>Stay Resident</strong> properties.</td>
</tr>
<tr>
<td>Run Estimates</td>
<td>Lets you view the <strong>Initial I/Os</strong>, <strong>I/Os Per Invocation</strong>, <strong>Initial Instructions</strong>, and <strong>Instructions Per Invocation</strong> properties.</td>
</tr>
<tr>
<td>Java Structure</td>
<td>Lets you view the <strong>Java Class</strong>, <strong>Jar ID</strong>, <strong>Package ID</strong>, <strong>Method ID</strong>, <strong>Jar Schema</strong>, and <strong>Java Signature</strong> properties.</td>
</tr>
</tbody>
</table>

4. **Preview and submit changes.**

**Modify procedure properties (IBM DB2 for Windows, Unix, and Linux)**

The **Properties** tab of the Procedures Editor lets you modify a procedure to change its definition. To modify a procedure, DBArtisan must drop then re-create the procedure.
To work with basic definition settings for a procedure:

1. Open an object editor on the procedure.
2. Click the Procedures tab.
3. Use the table below as a guide to understanding the settings on this page:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>If the function body is written in C or OLE, select DB2SQL. If the function body is written in JAVA, select DB2GENERAL. <strong>NOTE:</strong> If the function body is written in OLEDB, this parameter cannot be changed.</td>
</tr>
<tr>
<td>External Name</td>
<td>To change the external name of the function, type the new external name.</td>
</tr>
<tr>
<td>SQL Access Level</td>
<td>Specifies the SQL Access Level for the function.</td>
</tr>
<tr>
<td>Result Sets</td>
<td>Number of result sets</td>
</tr>
<tr>
<td>External Action</td>
<td>Select to allow the function to take action on an object not managed by DB2.</td>
</tr>
<tr>
<td>New Save Point</td>
<td>Specifies whether the function can use a new save-point.</td>
</tr>
<tr>
<td>Threadsafe</td>
<td>Specifies whether the function is declared as Threadsafe.</td>
</tr>
<tr>
<td>Fenced</td>
<td>Select to make the function run in a memory space separate from the database.</td>
</tr>
<tr>
<td>Parameter Style</td>
<td>To change the parameter style of the function, select the new parameter style. <strong>NOTE:</strong> If the function body is written in OLEDB, this parameter cannot be changed.</td>
</tr>
<tr>
<td>Program Type</td>
<td>Displays the Program Type option.</td>
</tr>
<tr>
<td>Inherit Special Registers</td>
<td>Specifies whether the function has the Inherit Special Registers clause enabled.</td>
</tr>
<tr>
<td>DBInfo</td>
<td>Enable to allow use of the DBINFO parameter with this function.</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Select to make the function return different results from calls with the same parameters.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

Modify a procedure definition (Microsoft SQL Server, Oracle, Sybase)

The **Definition** tab of the Procedures Editor lets you modify the SQL code for procedures on the current datasource.

To modify the body for a procedure:

1. Open an object editor on the procedure to be modified.
2. Click the **Definition** tab.
3. Modify the SQL code in the text area.
4. Preview and submit changes.

Modify the body of a procedure (IBM DB2 for Windows, Unix, and Linux)

The **Body** tab of the Procedures Editor lets you modify the SQL code for procedures on the current datasource.
To modify the body for a procedure:
1. Open an object editor on the procedure to be modified.
2. Click the Body tab.
3. Modify the SQL code in the text area.
4. Preview and submit changes.

Manage procedure parameters (IBM DB2)
The Parameters tab of the Procedures Editor lets you modify the input and output parameters of a procedure:

To modify the input and output parameters for a procedure:
1. Open an object editor on the procedure to be modified.
2. Click the Parameters tab.
   For each parameter, the tab lists details such as the datatype of the parameter and whether the parameter is an input or output parameter.
3. Optionally, take one of the following actions:
   • In the Datatype attributes, provide a new Type or Parameter Mode for a selected parameter. Depending on the DBMS that DBArtisan is working with, other parameters such as Size, Precision, or Scale may be available for viewing or modification.
   • Click the New button to provide datatype and size details for a new parameter.
   • Select a parameter and click the Delete button to drop that parameter.
4. Preview and submit changes.
   NOTE: You cannot use host variables in the CALL statement for the name of the procedure.

View status and size details for a procedure (Oracle)
The Properties tab of the Procedures Editor displays the status and size information for procedures on the current datasource.

To view size information on a procedure:
1. Open an object editor on the procedure to be modified.
2. Click the Properties tab.
3. Use the following table as a guide to understanding the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Lets you view Status, and Last Modified properties.</td>
</tr>
<tr>
<td>Size Information</td>
<td>Lets you view Source Size, Parsed Size, Code Size, and Error Size</td>
</tr>
</tbody>
</table>
Using the Profiles Editor (Oracle)
The Profiles editor lets you modify settings and options of existing profiles.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

• Manage Profile limits (Oracle)
• Manage users for Profiles (Oracle)

You also use this editor to perform the following common Object editor tasks:

• Work with privileges and permissions

Manage Profile limits (Oracle)
The Resources tab of the Profiles Editor lets you manage profile limits.

To modify limits associated with a profile:
1. Open an object editor on the profile to be modified.
2. Click the Resources tab.
3. Use the following table as a guide to modifying profile limits:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Limits</td>
<td>Lets you specify Composite Limit and Private SGA settings.</td>
</tr>
<tr>
<td>Session Limits</td>
<td>Lets you specify the limit on the amount of private space a session can allocate in the shared pool of the SGA. Specific settings are Sessions Per User, CPU Per Session, Logical Reads.</td>
</tr>
<tr>
<td>Time Limits</td>
<td>Lets you specify the limit on total connection time per session. Specific settings are Connect Time and Idle Time.</td>
</tr>
<tr>
<td>Call Limits</td>
<td>Lets you specify the CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds. Specific settings are CPU Per Call and Logical Reads.</td>
</tr>
<tr>
<td>Login Limits</td>
<td>Lets you specify the number of Failed Login Attempts on the user account before the account is locked and the Account Lock Time.</td>
</tr>
<tr>
<td>Password Limits</td>
<td>Lets you specify the Lifetime, Reuse Time, Reuse Max, Grace Period and a Verify Function for passwords.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

Manage users for Profiles (Oracle)
The Users tab of the Profiles Editor lets you manage users to be associated with profiles on the current datasource.

To assign users to or unassign users from a profile:
1. Open an object editor on the profile to be modified.
2. Click the Users tab.
3 Use the **Assign** button to open a dialog that lets you assign a user to this profile or select a user from the list, and click the **Unassign** button to open a dialog prompting you to confirm that the user is to be unassigned.

4 **Preview and submit changes.**

**Redo Log Groups Editor (Oracle)**

The Redo Log Groups editor lets you modify settings and options of existing Redo Log Groups.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to **Open an object editor**, **View and modify database object properties**, and how to **Preview and submit changes**. For detailed information, see **Overview and common usage of Object editors**.

You use this editor to perform the following tasks specific to this object type:

- **Adding or modifying Redo Log group members (Oracle)**

You also use this editor to perform the following common Object editor tasks:

- **View SQL/DDL for a database object**

Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch Logfile</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Take Checkpoint</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Adding or modifying Redo Log group members (Oracle)**

The **Redo Log Members** tab of the Redo Log Groups editor lets you add members or modify members of a Redo Log group.

To **add members or modify members of a redo log group**:

1 **Open an object editor** on the redo log group to be modified.

2 Click the **Redo Log Members** tab.

3 Take one of the following actions:

- Use the **Add** button to open a dialog that lets you add a member to this redo log group.
- Select a member from the list, and click the **Edit** button to open a dialog that lets you modify the **Redo Log Group Name** or **Redo Log Group Member Name** values.
- Select a member from the list, and click the **Drop** button to delete the member.

4 **Preview and submit changes.**
Using the Remote Servers Editor (Microsoft SQL Server, Sybase)

The Remote Servers editor lets you modify settings and options of existing remote servers.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- View identifying information for a Remote Server (Microsoft SQL Server)
- View identifying information for a Remote Server (Sybase)
- Manage logins for a Remote Servers (Microsoft SQL Server, Sybase)

You also use this editor to perform the following common Object editor tasks:

- View SQL/DDL for a database object

View identifying information for a Remote Server (Microsoft SQL Server)

The Information tab of the Remote Servers Editor lets you view identifying information for every remote server on the current datasource. A remote server is a Microsoft SQL Server that you can access with your local Microsoft SQL Server datasource across a network. Instead of logging directly into a remote server, you can log into your local server and execute a remote procedure call to it.

To modify basic identification settings for a remote server:

1. **Open an object editor** on the remote server to be modified.
2. **Click the Information tab.**
3. **Use the following table as a guide to modifying the settings on this tab:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Server</td>
<td>The unique server name that users must supply when executing remote procedure calls.</td>
</tr>
<tr>
<td>Publication Server</td>
<td>Specifies database publishing. When replication is installed, this option permits the tables of a database to be published for replication.</td>
</tr>
<tr>
<td></td>
<td>When selected, this option enables publication. When not selected, this option disables publishing, drops all publications, and unmarks all transactions that were marked for replication in the transaction log.</td>
</tr>
<tr>
<td>Distribution Server</td>
<td>Enables the remote server to be a distribution server.</td>
</tr>
<tr>
<td>Subscription Server</td>
<td>Specifies database subscriptions. When selected, the database can be subscribed for publication. When not selected, the database cannot be subscribed for publication.</td>
</tr>
<tr>
<td>Publisher/Subscriber</td>
<td>Enables the remote server to be a publisher/subscriber server.</td>
</tr>
<tr>
<td>Collation Compatible Server</td>
<td>Indicates Change Distributed Query execution against remote servers. If you select this option, Microsoft SQL Server assumes that all characters in the remote server are compatible with the local server, with regard to character set and collation sequence (or sort order).</td>
</tr>
<tr>
<td>Data Access Server</td>
<td>Indicates whether the target remote server is enabled for distributed query access.</td>
</tr>
<tr>
<td>RPC Out Server</td>
<td>Enables the remote server to execute stored procedures on the local server using a remote procedure call (RPC).</td>
</tr>
</tbody>
</table>

4. **Preview and submit changes.**
View identifying information for a Remote Server (Sybase)

The **Information** tab of the Remote Servers Editor lets you view identifying information for every remote server on the current datasource. A remote server is a Sybase ASE that you can access with your local Sybase ASE across a network. Instead of logging directly into a remote server, you can log into your local server and execute a remote procedure call to it.

To modify basic identification settings for a remote server:

1. **Open an object editor** on the remote server to be modified.
2. **Click the Information tab.**
3. **Use the table below as a guide to understanding and modifying the settings on this tab:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Name</td>
<td>The target remote server’s network name.</td>
</tr>
<tr>
<td>Local Server</td>
<td>The unique server name that users must supply when executing remote procedure calls.</td>
</tr>
<tr>
<td>Time out inactive connections</td>
<td>Indicates if the target server has uses the timeouts option to disable and enable the normal timeout code used by the local server.</td>
</tr>
<tr>
<td>Network Password Encryptions</td>
<td>Indicates whether connections with a remote server are to be initiated with a client-side password encryption handshake or with the usual unencrypted password handshake sequence.</td>
</tr>
<tr>
<td>Security Model A</td>
<td>Indicates if the target remote server uses the Remote Procedure Call (RPC) Security Model A. With this model, Sybase ASE does not support security services such as message confidentiality via encryption between the local and remote servers. For more information on server security, see the Sybase System Administration Guide.</td>
</tr>
<tr>
<td>Security Model B</td>
<td>Indicates if the target remote server uses the RPC Security Model B. With this model, Sybase ASE gets a credential from the security mechanism and uses the credential to establish a secure physical connection with the remote server. With this model, you can choose one or more of the following security services: mutual authentication, message confidentiality, and message integrity. For more information on server security, see the Sybase System Administration Guide.</td>
</tr>
<tr>
<td>Security Mechanism</td>
<td>The name for the security mechanism.</td>
</tr>
<tr>
<td>Mutual Authentication</td>
<td>This option makes the local server authenticate the remote server by retrieving the credential of the remote server and verifying it with the security mechanism. With this option, the credentials of both servers are authenticated and verified. You must select Security Model B to use this option.</td>
</tr>
<tr>
<td>Message Confidentiality</td>
<td>This option means that messages are encrypted when sent to the remote server, and results from the remote server are encrypted. You must select Security Model B to use this option.</td>
</tr>
<tr>
<td>Message Integrity</td>
<td>This option means that messages between the servers are checked for tampering. You must select Security Model B to use this option.</td>
</tr>
</tbody>
</table>

4. **Preview and submit changes.**

Manage logins for a Remote Servers (Microsoft SQL Server, Sybase)

The **Logins** tab of the Remote Servers Editor lets you manage the logins for remote servers on the current server.
To add, modify, or drop logins for a remote server:
1. Open an object editor on the remote server to be modified.
2. Click the Logins tab.
   This tab includes a table listing the remote and local names and password authentication of logins for every remote server.
3. Take one of the following actions:
   - Click Add and use the following table as a guide to setting options for a new login:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Server Name</td>
<td>Displays the remote server name.</td>
</tr>
<tr>
<td>Remote Login Name</td>
<td>Lets you specify a remote login name. Select the All box if you want to use this name for all remote logins.</td>
</tr>
<tr>
<td>Local Login Name</td>
<td>Lets you select a local user name.</td>
</tr>
<tr>
<td>Trusted Remote Login</td>
<td>Lets you specify this remote login as a trusted login.</td>
</tr>
</tbody>
</table>

   - Select a login, click Modify, and use the table above as a guide to modifying login options.
   - Select a login, and click Drop to delete the login.

Using the Roles editor (Microsoft SQL Server, Oracle, Sybase)
The Roles editor lets you modify settings and options of existing roles.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage users of a role (Microsoft SQL Server)
- Manage authentication for a Role (Oracle)
- Manage authentication for a Role (Sybase)
- Manage logins and default roles (Sybase)

You also use this editor to perform the following common Object editor tasks:

- Work with privileges and permissions
- View SQL/DDL for a database object

Grant roles to users or other roles (Oracle)
The Users/Roles tab of the Roles editor lets you grant roles to users or other roles and to revoke those roles.
To grant or revoke roles:
1. Open an object editor on the role to be modified.
2. Click the Users/Roles tab.
3. Take one of the following actions:
   - Click Grant to open the Grant Roles dialog.
   - Select a grantee from the list and click Revoke to open the Revoke Role dialog.

Manage users of a role (Microsoft SQL Server)
The Users tab of the Roles Editor let you manage users for every role on the current datasource. A user becomes associated with an application role after running the target application.

To assign/unassign users:
1. Open an object editor on the role to be modified.
2. Click the Users tab.
3. Use the Join Rule and Leave Role buttons to move users between the Users In Role and Users Not In Role lists.
4. Preview and submit changes.

Manage authentication for a Role (Oracle)
The Authentication tab of the Roles Editor lets you manage role identity. When creating a role, you must establish certain rules governing its use. You can specify whether or not a role must be identified when used.

If you require role identification, you can authenticate the user externally through the operating system, or with a specific password.

To modify authentication settings for a role:
1. Open an object editor on the role to be modified.
2. Click the Authentication tab.
3. In the Identification list, specify whether the role requires identification.
4. If you specified that the role requires identification, provide a Password and specify whether the roles is to be authenticated Globally or Externally.
5. Preview and submit changes.

Manage authentication for a Role (Sybase)
The Authentication tab of the Roles Editor lets you manage role identification for every role on the current server. To identify the role, supply a password.

To manage authentication settings for a role:
1. Open an object editor on the role to be modified.
2. Click the Authentication tab.
3 Use the table below as a guide to modifying values on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>Lets you specify a Password.</td>
</tr>
<tr>
<td>Security</td>
<td>Lets you specify Role Locked.</td>
</tr>
<tr>
<td>Options</td>
<td>Lets you specify Password Expiration, Minimum Password Length, and Maximum Failed Logins.</td>
</tr>
</tbody>
</table>

4 Preview and submit changes.

Manage logins and default roles (Sybase)
The Logins/Roles tab of the Roles Editor lets you grant and revoke permissions for this role to logins or other roles.

To grant or revoke this role to a login or another role:
1 Open an object editor on the role to be modified.
2 Click the Logins/Roles tab.
3 Optionally take one of the following actions:
   • Click Grant to open a dialog that lets you grant this role to a login or another role.
   • Select a role or login and click Revoke.
4 Preview and submit changes.

Using the Rollback Segments Editor (Oracle)
The Rollback Segments editor lets you modify settings and options of existing rollback segments.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

• View Rollback Segment status (Oracle)
• Manage storage for a Rollback Segment (Oracle)
• View activity levels for a Rollback Segments (Oracle)

You also use this editor to perform the following common Object editor tasks:

• View SQL/DDL for a database object

View Rollback Segment status (Oracle)
The Status tab of the Rollback Segments Editor lets you enable and disable a rollback segment and displays status details for the rollback segment.
To manage rollback segment status:
1. **Open an object editor** on the rollback segment to be modified.
2. **Click the Status tab.**
   - The tab displays whether the rollback segment is online or offline and provides the associated **Tablespace**, **Size**, and **No. of Extents**.
3. **Optionally, use the Status list to place the rollback segment online or offline.**
4. **Preview and submit changes.**

Manage storage for a Rollback Segment (Oracle)
The **Storage** tab of the Object Editor lets you manage storage for every rollback segment on the current datasource.

**TIP:** Due to their heavy I/O nature, rollback segment placement is best on server drives that experience little activity. Disks configured with RAID5 are typically poor choices for rollback segments. If you use RAID on the server, a RAID0 or 0+1 configuration is best for rollback segments.

To manage storage settings for a rollback segment:
1. **Open an object editor** on the rollback segment to be modified.
2. **Click the Storage tab.**
3. **Use the following table as a guide to understanding the options on this tab:**

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extents</td>
<td>The unit of space allocated to an object whenever the object needs more space.</td>
</tr>
<tr>
<td></td>
<td><strong>Initial Extent</strong> - The initial space extent (in bytes) allocated to the object.</td>
</tr>
<tr>
<td></td>
<td><strong>Next Extent</strong> - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.</td>
</tr>
<tr>
<td></td>
<td><strong>Optimal Size</strong> - optimal extent size</td>
</tr>
<tr>
<td></td>
<td><strong>Minimum Extents</strong> - The appropriate minimum extents value for the object.</td>
</tr>
<tr>
<td></td>
<td><strong>Maximum Extents</strong> - The appropriate maximum extents value for the object.</td>
</tr>
<tr>
<td>Extent Detail</td>
<td>Displays extent details.</td>
</tr>
</tbody>
</table>

4. **Preview and submit changes.**

View activity levels for a Rollback Segments (Oracle)
The **Activity** tab of the Rollback Segments Editor displays activity levels, and dynamic sizing.

To view activity levels for a rollback segment:
1. **Open an object editor** on the rollback segment to be modified.
2. **Click the Activity tab.**
3 Use the following table as a guide to understanding the options on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Levels</td>
<td>Displays <strong>Active Transactions</strong>, <strong>Writes</strong>, <strong>Gets</strong> and <strong>Waits</strong> values.</td>
</tr>
<tr>
<td>Dynamic Sizing</td>
<td>Displays <strong>High Watermark</strong>, <strong>Extends</strong>, <strong>Shrinks</strong>, and <strong>Wraps</strong> values.</td>
</tr>
</tbody>
</table>

4 **Preview and submit changes**.

Using the Rules editor (Microsoft SQL Server, Sybase)
The Rules editor lets you modify settings and options of existing rules.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to **Open an object editor**, **View and modify database object properties**, and how to **Preview and submit changes**. For detailed information, see **Overview and common usage of Object editors**.

You use this editor to perform the following tasks specific to this object type:

- Modify rule properties (Microsoft SQL Server, Sybase)

You also use this editor to perform the following common Object editor tasks:

- **View SQL/DDL for a database object**
- **Work with object dependencies**

The **Command** menu on the Object editor toolbar provides the following command:

- **Rename**

Modify rule properties (Microsoft SQL Server, Sybase)
The **Properties** tab of the Rules Editor lets you modify the restriction for a rule.

**To modify a rule:**
1 **Open an object editor** on the rule to be modified.
2 Click the **Properties** tab.
3 Modify the rule in the **Restriction** area.
4 If DBArtisan is working with Sybase, specify a rule **Type**.
5 **Preview and submit changes**.

Using the Schemas editor (Microsoft SQL Server 2005)
The Schemas editor lets you modify settings and options of existing schemas.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to **Open an object editor**, **View and modify database object properties**, and how to **Preview and submit changes**. For detailed information, see **Overview and common usage of Object editors**.
You use this editor to perform the following tasks specific to this object type:

- Manage Schema properties
- Managing objects owned by a schema

You also use this editor to perform the following common Object editor tasks:

- Work with privileges and permissions
- View SQL/DDL for a database object

Manage Schema properties

The **Properties** tab of the Schema Editor lets you work with the basic properties of a schema.

**To work with schema properties:**

1. Open an object editor on the schema to be modified.
2. Click the **Properties** tab.
3. Use the following table as a guide to understanding and modifying the settings on this page:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation</td>
<td>Lets you view the <strong>Name</strong> property and lets you set the <strong>Owner</strong> property.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

Managing objects owned by a schema

The **Objects** tab of the Schema Editor lets you work with the database objects associated with a schema.

**To work with schema objects:**

1. Open an object editor on the schema to be modified.
2. Click the **Properties** tab.
   - Objects owned by this schema are grouped by type under folders.
3. Optionally, take one of the following actions:
   - Select an object and click **Edit** to open a object editor on that object.
   - Select an object and click **Drop** to disassociate that object from the schema.
4. Preview and submit changes.

Using the Segments Editor (Sybase)

The Segments editor lets you modify settings and options of existing segments.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.
You use this editor to perform the following tasks specific to this object type:

- **Extend or drop a Segment (Sybase)**
- **Manage database objects associated with a Segment (Sybase)**
- **View usage and space distribution for a Segment (Sybase)**
- **Manage thresholds for a Segment (Sybase)**

You also use this editor to perform the following common Object editor tasks:

- **View SQL/DDL for a database object**

### Extend or drop a Segment (Sybase)

The **Location** tab of the Segments Editor lets you extend or drop a segment on the current datasource.

**To extend or drop a segment:**

1. **Open an object editor** on the segment to be modified.
2. Click the **Location** tab.
3. Select a segment from the list of segments and click the **Drop** or **Extend** button.
4. **Preview and submit changes**.

### Manage database objects associated with a Segment (Sybase)

The **Objects** tab of the Segments Editor lets you manage database objects associated with every segment on the current datasource.

**To view the tables, indexes, or constraints associated with a segment:**

1. **Open an object editor** on the segment to be modified.
2. Click the **Objects** tab.

   DBArtisan organizes the objects in a tree structure with three folders containing all of the following associated objects:
   - Tables
   - Indexes
   - Constraints

   **NOTE:** You can open an object editor on a table, index, or constraint by double-clicking that object on the **Objects** tab.

### View usage and space distribution for a Segment (Sybase)

The **Space** tab of the Segments Editor lets you view segment usage, distribution of segment space, and object Space Usage for every segment on the current datasource.
To view usage and space distribution for a segment:
1. Open an object editor on the segment.
2. Click the Space tab.

Manage thresholds for a Segment (Sybase)
The Threshold tab of the Segments Editor lets you add, edit, or drop thresholds for segments on the current datasource.

To manage thresholds for a segment:
1. Open an object editor on the segment to be modified.
2. Click the Threshold tab.
3. Manage thresholds as follows:
   • Click the Add button to specify the number of free pages and procedure name for a new segment threshold
   • Select an existing threshold and click the Edit button to modify the threshold’s procedure name value
   • Select an existing threshold and click the Drop button to delete that threshold
4. Preview and submit changes.

Using the Sequences editor (IBM DB2 for Linux, Unix, and Windows, Oracle)
The Sequences editor lets you modify settings and options of existing sequences.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:
- Manage parameters for a sequence (IBM DB2 for Linux, Unix, and Windows, Oracle)

You also use this editor to perform the following common Object editor tasks:
- View SQL/DDL for a database object
- Work with privileges and permissions
- Work with object dependencies

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th></th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Alias</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Synonym</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Manage parameters for a sequence (IBM DB2 for Linux, Unix, and Windows, Oracle)

The **Definition** tab of the Sequences Editor lets you manage parameters for every sequence on the current datasource.

**To manage parameters for a sequence:**

1. **Open an object editor** on the sequence to be modified.
2. **Click the** **Definition** **tab.**
3. **Use the following table as a guide to modifying settings on this tab:**

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>Lets you specify increment, minimum value and maximum value settings.</td>
</tr>
<tr>
<td>Current/Next Sequence Numbers</td>
<td>Lets you work with sequence cycle numbers.</td>
</tr>
<tr>
<td>Sequence Datatype (IBM DB2 for Windows, Unix, and Linux)</td>
<td>Lets you specify the datatype and width for the sequence.</td>
</tr>
<tr>
<td>Options</td>
<td>Lets you specify <strong>Cache Size</strong>, <strong>Cycle When Reach Max/Min</strong>, and <strong>Generate Numbers in Order</strong> (useful when you are using the sequence number as a timestamp) values.</td>
</tr>
</tbody>
</table>

4. **Preview and submit changes.**

Using the Stogroups Editor (IBM DB2 for OS/390 and z/OS)

The Stogroups Editor lets you view and modify volumes and manage function privileges.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to **Open an object editor**, **View and modify database object properties**, and how to **Preview and submit changes**. For detailed information, see **Overview and common usage of Object editors**.

You use this editor to perform the following tasks specific to this object type:

- **Add volumes to a Stogroup (IBM DB2 for OS/390 and Z/OS)**

You also use this editor to perform the following common Object editor tasks:

- **View SQL/DDL for a database object**
- **Work with privileges and permissions**
Add volumes to a Stogroup (IBM DB2 for OS/390 and Z/OS)
The Volume Devices tab of the Stogroups editor lets you add one or more volumes to a Stogroup.

To manage volumes for a Stogroup:
1. Open an object editor on the stogroup to be modified.
2. Click the Volume Devices tab.
3. Optimally, take one of the following actions:
   • Click Add to open a dialog that lets you add volumes.
   • Select an volume and click Remove to delete the volume.
4. Preview and submit changes.

Using the Structured Types Editor (IBM DB2 for Linux, Unix, and Windows)
The Structured Types editor lets you modify settings and options of existing structured types.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:
• Manage attributes for Structured Types (IBM DB2 for Linux, Unix, and Windows)
• Manage methods of a Structured Type (IBM DB2 for Linux, Unix, and Windows)
• Manage method bodies of a Structured Type (IBM DB2 for Linux, Unix, and Windows)

You also use this editor to perform the following common Object editor tasks:
• View SQL/DDL for a database object

Manage attributes for Structured Types (IBM DB2 for Linux, Unix, and Windows)
The Attributes tab of the Structured Types Editor lets you manage attributes for structured types on the current datasource.

To manage attributes for a structured type:
1. Open an object editor on the structured type to be modified.
2. Click the Attributes tab.
   • For each structured type attribute, the tab displays the name and datatype for the attribute.
3. Take one of the following actions:
   • Click Add to open a dialog that lets you specify the name and datatype details for a new attribute.
   • Select an attribute and click Drop to delete the attribute.
4. Preview and submit changes.
Manage methods of a Structured Type (IBM DB2 for Linux, Unix, and Windows)

The **Methods** tab of the Structured Types Editor lets you create and manage methods. A database method of a structured type is a relationship between a set of input data values and a set of result values, where the first input value (or subject argument) has the same type, or is a subtype of the subject type (also called the subject parameter), of the method. For example, a method called CITY, of type ADDRESS, can be passed input data values of type VARCHAR, and the result is an ADDRESS (or a subtype of ADDRESS).

**To manage structured type methods:**

1. Open an object editor on the structured type to be modified.
2. Click the **Methods** tab.
   - The tab lists all methods associated with the structured type and for each method, includes name and language details.
3. Optionally, take one of the following actions:
   - Click **Add** to begin the process of adding a new method.
   - Select a method and click **Drop** to delete that method from the structured type.
   - Select a method and click **Create Body** to open the editor’s Method Body tab to provide the body for the method. For details, see Manage method bodies of a Structured Type (IBM DB2 for Linux, Unix, and Windows).
4. **Preview and submit changes**

For more information, see Structured Types Editor.

Manage method bodies of a Structured Type (IBM DB2 for Linux, Unix, and Windows)

The **Method Body** tab of the Structured Types Editor lets you type a method body associated with a method.

**To provide the body for a method:**

1. Open an object editor on the structured type to be modified.
2. Click the **Method Body** tab.
3. Type the text for the method in the **Method Body** area.
4. **Preview and submit changes**

Using the Synonyms editor (IBM DB2 for OS/390 and z/OS, Oracle)

The Synonyms editor lets you modify settings and options of existing synonyms.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- **View synonym properties (IBM DB2 for OS/390 and z/OS, Oracle)**

You also use this editor to perform the following common Object editor tasks:
• Work with object dependencies

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Describe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Rename</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>

View synonym properties (IBM DB2 for OS/390 and z/OS, Oracle)

The Properties tab of the Synonyms Editor lets you view base object information for synonyms of tables, views, procedures, packages, functions, sequences, and snapshots.

To view details on the object of a synonym:
1. Open an object editor on the synonym to be modified.
2. Click the Properties tab.
   The tab displays the type, owner, name, and other details of the object referenced by the synonym.

Using the Tables editor

The Tables editor lets you modify settings and options of existing tables.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

NOTE: Before modifying a DB2 table, you should understand how DBArtisan submits changes to these tables. For details, see About altering DB2 tables.

You use this editor to perform the following tasks specific to this object type:
• Manage properties for a table (All DBMS)
• Manage table columns (IBM DB2, Microsoft SQL Server, Oracle, Sybase)
• Manage table indexes (All DBMS)
• Manage table constraints (All DBMS))
• View space usage details for a table (Microsoft SQL Server, Sybase)
• Manage space usage and distribution for a table (Oracle)
• Manage storage details for a Table (Oracle)
• Manage Table partitions (Sybase)
• Manage Table partitions (IBM DB2 for Windows, Unix, and Linux)
• Managing distribution key columns for a table (IBM DB2 for Windows, Unix, and Linux)
• Managing dimensions for a table (IBM DB2 for Windows, Unix, and Linux)
• Manage tablespaces for tables (IBM DB2 for Windows, Unix, and Linux)
• View LOB column space (Oracle)
• Manage table Partitions (Oracle)

You also use this editor to perform the following common Object editor tasks:

• Work with privileges and permissions
• Add a comment to a database object
• Work with object dependencies
• View SQL/DDL for a database object

Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocate Extent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Analyze</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Build Query</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Convert to Partitioned</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Create Alias</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Insert Statements</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Create Like</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Create Synonym</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create View</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>YES, View Wizard for IBM DB2 for Linux, Unix, and Windows</td>
</tr>
<tr>
<td>DBCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Deallocate Unused Space</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES, For more information, see Describe Window</td>
</tr>
<tr>
<td>Describe</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>YES, For more information, see Describe Window</td>
</tr>
<tr>
<td>Disable Triggers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Edit Data</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Enable Triggers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Estimate Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Extract Data as XML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Import Data From File</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>YES, For more information, see Data Import Wizard for IBM DB2 for Linux, Unix, and Windows</td>
</tr>
<tr>
<td>Indexes</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Lock</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td>For more information, see Lock</td>
</tr>
</tbody>
</table>
About altering DB2 tables

The ALTER TABLE command of Transact-SQL is limited to adding NULL columns to the end of a table and to adding or dropping constraints. Unfortunately, this scenario does not address many requirements of administrators and developers who need to add, delete or modify columns more broadly:

- Add columns anywhere in a table, not just the end
- Add columns that do not permit a NULL value.
- Change the NULL/NOT NULL status of table columns
- Change column datatypes to other compatible types
- Change the length of datatypes
- Delete a column

Due to the limitations of the ALTER TABLE command, the only way to make broader modifications is to write SQL scripts that step through all desired changes. To perform an enhanced table alter, DBArtisan constructs an SQL script that completes the following steps:

1. Renames the existing table so that the original and its data remain intact
2. Builds a CREATE TABLE statement with the new table definition, including declared defaults, primary key and check constraints
3. Builds an INSERT statement to copy data from the original, renamed table to the new one
4. Builds foreign keys on the new table
5. Reapplies any privileges granted on the table
6. Rebuilds all dependencies on the new table, including indexes, triggers, procedures, packages, functions and views. When rebuilding procedures, functions, packages and views, DBArtisan also rebuilds any permissions on them.
For more information, see Columns tab of the Tables Editor.

Manage properties for a table (All DBMS)
The Properties tab of the Tables editor lets you manage the basic property settings for a table.

To manage table properties:
1. Open an object editor on the table to be modified.
2. Click the Properties tab.
3. Use the table below as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table (IBM DB2 for OS/390 and Z/OS)</td>
<td>Lets you view EditProcedure, Table Type, Table Status, Check Flag, DBID, and OBID properties. Lets you set Volatile, Audit, RestrictDrop, Label, and ValidProc properties.</td>
</tr>
<tr>
<td>Table (IBM DB2 for Windows, Unix, and Linux)</td>
<td>Lets you set Percent Free, Lock Size, Append Data, Volatile, Compress, Row Compression, Security Policy, RestrictDrop, LogIndexBuild, and CCSID properties.</td>
</tr>
<tr>
<td>Table (Oracle)</td>
<td>Lets you set Cache, Row Movement, Parallel Degree, and Parallel Instances properties.</td>
</tr>
<tr>
<td>Physical (Oracle)</td>
<td>Lets you view the Row Organization property. Lets you set the Logging property.</td>
</tr>
<tr>
<td>Creation (Microsoft SQL Server)</td>
<td>For Microsoft SQL Server, lets you view the Name and Created properties and modify the Schema property. For Sybase, lets you view the Name and Owner properties.</td>
</tr>
<tr>
<td>Physical Storage (Microsoft SQL Server, Sybase)</td>
<td>For Microsoft SQL Server, lets you view the Text Image Filegroup property and set the Filegroup property. For Sybase, lets you set the Segment Name, Maximum Rows Per Page, Reserve Page Gap, and Identity Gap properties.</td>
</tr>
<tr>
<td>Full Text Index (Microsoft SQL Server)</td>
<td>Lets you view Full Text Indexing Installed and Full Text Indexing Active properties. The full-text index feature provides support for sophisticated word searches in character string data. A full-text index stores information about significant words and their location within a given column. This information is used to quickly complete full-text queries that search for rows with particular words or combinations of words. This functionality is available for Microsoft SQL Server 8.0 or later.</td>
</tr>
<tr>
<td>Text In Row (Microsoft SQL Server)</td>
<td>Lets you view Text In Row Enabled and Text In Row Limit properties.</td>
</tr>
<tr>
<td>Cache Strategy (Sybase)</td>
<td>Lets you set MRU Replacement Strategy and Prefetch Strategy properties.</td>
</tr>
<tr>
<td>Row Locking Strategy (Sybase)</td>
<td>Lets you view the Expected Row Size property and set the Lock Scheme property.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

Manage table columns (IBM DB2, Microsoft SQL Server, Oracle, Sybase)
The Columns tab of the Tables Editor lets you add and delete table columns and modify column attributes.
To manage table columns:

1. Open an object editor on the table to be modified.
2. Click the Columns tab.
   - The list on the left of the tab lists the columns currently included in the table.
3. To modify column details, select a column and use attribute controls on the right side of the tab to change the settings for the column. Use the following table as a guide to understanding the settings:

<table>
<thead>
<tr>
<th>DBMS</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM DB2 for OS/390 and Z/OS</td>
<td>Lets you set <strong>Type</strong>, <strong>Identity Column</strong>, <strong>Allow Nulls</strong>, <strong>Default Value</strong>, and <strong>Comment</strong> properties.</td>
</tr>
<tr>
<td>IBM DB2 for Windows, Unix, and Linux</td>
<td>Lets you view <strong>Security Label</strong>, <strong>Default Value</strong>, and <strong>Express</strong> properties. Lets you set <strong>Type</strong>, <strong>Identity Column</strong>, <strong>Start With</strong>, <strong>Increment By</strong>, <strong>Cache</strong>, <strong>Comment</strong>, <strong>By Default</strong>, <strong>Max Value</strong>, <strong>Min Value</strong>, and <strong>Cycle</strong>, properties.</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>Lets you view <strong>Fill Rows with Default Value</strong>, and <strong>Computed Expression</strong> properties. Lets you set <strong>Name</strong>, <strong>Type</strong>, <strong>Identity Column</strong>, <strong>Allow Nulls</strong>, <strong>Default Value</strong>, <strong>Default Binding</strong>, and <strong>Rule Binding</strong> properties.</td>
</tr>
<tr>
<td>Oracle</td>
<td>Lets you set <strong>Type</strong>, <strong>Width</strong>, <strong>Scale</strong>, <strong>Unused</strong>, <strong>Allow Nulls</strong>, <strong>Default Value</strong>, and <strong>Comment</strong> properties. <strong>NOTE</strong>: For Oracle 9i, DBArtisan supports the XMLType datatype. DBArtisan also supports the new Timestamp datatypes, including Timestamp, Timestamp with Time Zone, and Timestamp with Local Time Zone.</td>
</tr>
<tr>
<td>Sybase</td>
<td>Lets you view <strong>Computed Expression</strong> and <strong>Materialized</strong> properties. Lets you set <strong>Computed</strong>, <strong>Type</strong>, <strong>Identity Column</strong>, <strong>Allow Nulls</strong>, <strong>Default Value</strong>, <strong>Default Binding</strong>, and <strong>Rule Binding</strong> properties.</td>
</tr>
</tbody>
</table>

4. Optionally, take one of the following actions:
   - Click the **Add Column** button to add a new column to the details.
   - Select a column and click the **Delete** button to remove that column from the table.
   - Use the arrow buttons to change the ordering of selected columns.
5. Preview and submit changes.

Manage table indexes (All DBMS)

The **Indexes** tab of the Tables Editor lets you manage indexes for a table.

To manage table indexes:

1. Open an object editor on the table to be modified.
2. Click the **Indexes** tab.
   - The list of indexes for the table is displayed on the tab.
3. Optionally, take one of the following actions:
   - Click **Add** to open a dialog that lets you add a new index to the table.
   - Select an index and click **Edit** to open a dialog that lets you edit index properties.
   - Select an index and click **Drop** to open a dialog that lets you remove the index from the table.
4. Preview and submit changes.
Manage table constraints (All DBMS))
The Constraints tab of the Tables Editor lets you manage primary key, unique key, foreign key, and check constraints for tables on the current datasource.

   NOTE:   For Sybase, DBArtisan also supports logical primary keys and logical foreign keys.

To manage constraints for a table:
1   Open an object editor on the table to be modified.
2   Click the Constraints tab.
   Constraints are grouped by type, under folders
3   Optionally take one of the following actions:
   • Select a constraint type folder and click Add to open a dialog that lets you add a constraint of that type.
   • Select a constraint and click Edit to open a dialog that lets you modify the constraint details.
   • Select a constraint and click Drop to remove the constraint.
4   Preview and submit changes.

View space usage details for a table (Microsoft SQL Server, Sybase)
The Space tab of the Databases Editor lets you view pie charts showing the space usage for tables on the current datasource.

To view table space usage:
1   Open an object editor on the table.
2   Click the Space tab.

Manage space usage and distribution for a table (Oracle)
The Space tab of the Tables Editor lets you view the table usage and the distribution of table space for every table on the current datasource.

To view table space usage:
1   Open an object editor on the table.
2   Click the Space tab.
3   Use the following table as a guide to understanding the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Utilization</td>
<td>Lets you view Size and number of Blocks.</td>
</tr>
<tr>
<td>Row Information</td>
<td>Lets you view the Number of Rows, Average Row Length, and Chain Rows properties.</td>
</tr>
<tr>
<td>Extents</td>
<td>Let you view the Number of Extents and Maximum Extents values.</td>
</tr>
</tbody>
</table>

   NOTE:   This tab provides a listing of details for each extent that includes Extent ID, File ID, Block ID, and number of Blocks.
Manage storage details for a Table (Oracle)
The Storage tab of the Tables Editor lets you manage storage for every table on the current datasource.

To manage storage details for a table:
1. Open an object editor on the table.
2. Click the Space tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab.

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Block Storage</td>
<td>Lets you choose the Tablespace Name, and specify Percent Free, Initial Transactions, and Max Transactions values.</td>
</tr>
<tr>
<td>Extents</td>
<td>Lets you view Initial Extent, Next Extent, Minimum Extents, Maximum Extents, and Percent Increase values.</td>
</tr>
<tr>
<td>Freelists</td>
<td>Lets you specify Freelists and Freelist Groups values.</td>
</tr>
<tr>
<td>Bufferpool</td>
<td>Lets you specify a Buffer Pool.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

Manage Table partitions (Sybase)
The Partitions tab of the Tables Editor lets you manage partitions for tables on the current datasource.

To add or remove table partitions or view partition status for a table:
1. Open an object editor on the table to be modified.
2. Click the Partitions tab.
3. Optionally, use the Partition/Unpartition radio set to change the partitioned status for the table.
4. Optionally, if you chose the Partitioned radio box, specify the Number of Partitions.
5. Preview and submit changes.

Manage Table partitions (IBM DB2 for Windows, Unix, and Linux)
The Partitions tab of the Tables Editor lets you manage partitions for tables on the current datasource.

To manage table partitions:
1. Open an object editor on the table to be modified.
2. Click the Partitions tab.
3. Optionally, use the New or Delete buttons to create or drop a partition column or data partition.
4. Optionally, use the Edit button to edit a data partition.
5 Optionally, use the Commands menu to attach or detach a data partition.

6 Preview and submit changes.

Managing distribution key columns for a table (IBM DB2 for Windows, Unix, and Linux)
The Distribution Key Columns tab of the Tables editor lets you group one or more columns to form a distribution key.

To manage distribution key columns for a table:
1 Open an object editor on the table to be modified.
2 Click the Distribution Key Columns tab.
3 Optionally, take one of the following actions:
   • Click the New button to add a new column to the dimension for the table.
   • Select a column and click the Edit button to modify the dimension column properties.
   • Select a column and click the Delete button to drop a column from the dimension.
4 Preview and submit changes.

Managing index-organized table properties
The IOT Properties tab of the Tables Editor lets you manage basic properties of index-organized tables.

To manage IOT properties for a table:
1 Open an object editor on the table to be modified.
2 Click the IOT Properties tab.
3 Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ungrouped</td>
<td>Lets you specify an overflow segment</td>
</tr>
<tr>
<td>Percent Threshold</td>
<td>Lets you specify the percentage of space reserved for an index-organized table.</td>
</tr>
<tr>
<td>Key Compression</td>
<td>Lets you enable or disable compression and provide a compression value.</td>
</tr>
</tbody>
</table>

4 Preview and submit changes.

Managing dimensions for a table (IBM DB2 for Windows, Unix, and Linux)
The Dimensions tab of the Tables editor lets you group columns to form a dimension.

To manage dimension columns for a table:
1 Open an object editor on the table to be modified.
2 Click the Dimensions tab.
3 Optionally, take one of the following actions:
   • Click the New button to add a new column to the dimension for the table.
   • Select a column and click the Edit button to modify the dimension column properties.
   • Select a column and click the Delete button to drop a column from the dimension.

4 Preview and submit changes.

Manage tablespaces for tables (IBM DB2 for Windows, Unix, and Linux)
The **Tablespaces** tab of the Tables Editor lets you manage tablespaces for tables on the current datasource.

To manage tablespaces for a table:
1 Open an object editor on the table to be modified.
2 Click the **Tablespaces** tab.
3 Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Tablespace, Long Tablespace, and Index Tablespace</td>
<td>Lets you view the <strong>Database partition Group</strong>, <strong>Managed By</strong>, <strong>Page Size</strong>, and <strong>Extent Size</strong> properties. Lets you choose the <strong>Name</strong> of a tablespace.</td>
</tr>
</tbody>
</table>

4 Preview and submit changes.

View LOB column space (Oracle)
The **LOB Column Space** tab of the Tables Editor lets you view space usage details for LOB columns in a table.

To view space usage details for LOB columns in a table:
1 Open an object editor on the table.
2 Click the **Partitions** tab.
3 From the **Column** dropdown, select the LOB column you want to view.
4 Use the following table as a guide to understanding the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOB Column Segment and LOB Column Index</td>
<td>Lets you view the <strong>Segment Name</strong>, <strong>No. of Extents</strong>, <strong>Max Extents</strong>, <strong>Size</strong>, and <strong>Blocks</strong> properties.</td>
</tr>
<tr>
<td>Segment Extents and Index Extents</td>
<td>Lets you view the <strong>Extent ID</strong>, <strong>File ID</strong>, <strong>Block ID</strong> and number of <strong>Blocks</strong> properties for segment and index extents.</td>
</tr>
</tbody>
</table>

Manage table Partitions (Oracle)
The **Partitions** tab of the Tables Editor lets you manage table partitions on the current datasource.
To manage table partitions:

1. Open an object editor on the table to be modified.
2. Click the Partitions tab.
3. Use the table below as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Displays the Partition Type and Subpartition Type for the partition.</td>
</tr>
<tr>
<td>Columns</td>
<td>Displays the column name and datatype for partitioning columns.</td>
</tr>
<tr>
<td>Partition definitions</td>
<td>Displays the Name, Value, and Tablespace for each partition definition.</td>
</tr>
</tbody>
</table>

4. Optionally, use the Edit Partition button or Drop Partitions to edit or drop the partition.

5. Optionally make use of one of the Partition Commands menu items:
   - Allocate Extent
   - Analyze
   - Coalesce
   - Deallocate Unused Space
   - Exchange
   - Mark Local Indexes Unusable
   - Rebuild Unusable Local Indexes
   - Merge Partition
   - Split Partition
   - Truncate

6. Preview and submit changes.

Exchange Data and Index Segments Dialog Box

You can use the Tables Editor to convert a partition into a nonpartitioned table by exchanging the data and index segments. You can also convert a nonpartitioned table into partitions in an existing table. Exchanging the data and index segments is most useful when you have nonpartitioned tables that you want to convert to partitions in a partitioned table.

**CAUTION:** The table and partition being exchanged must have the same number of columns and the column types must match.

The table below describes the options and functionality on the Exchange Data and Index Segments dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition with which to Exchange Table</td>
<td>Select a partition.</td>
</tr>
<tr>
<td>Table with which to Exchange Partition</td>
<td>Select a table owner and a non-partitioned, non-clustered table.</td>
</tr>
<tr>
<td>Include Indexes</td>
<td>Includes indexes in the exchange.</td>
</tr>
<tr>
<td>Validate Proper Collation of Rows</td>
<td>A column’s collation sequence is used in any operation that compares values of the column to each other or to constant values.</td>
</tr>
</tbody>
</table>
Merge Partitions Dialog Box
You can use the Tables Editor to merge the contents of two adjacent partitions from a range or composite partitioned table into one. If you want to roll up similar partitions into larger partitions, which act as archives, you can merge them. For example, you can make partitions for the data from each week which you can then merge into a partition for the month.

**NOTE:** You cannot merge hash partitions.

**NOTE:** The partition that results from the merge inherits the higher upper bound of the two partitions.

The table below describes the options and functionality on the Merge Partitions dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition Definition</td>
<td>Lets you enter a new name to contain the merged partitions. Select two partitions to merge. To select multiple partitions, click SHIFT+click or CTRL+click.</td>
</tr>
<tr>
<td>Segment</td>
<td>Displays tablespace parameters.</td>
</tr>
<tr>
<td>Physical</td>
<td>Displays physical parameters.</td>
</tr>
<tr>
<td>Storage</td>
<td>Displays storage parameters.</td>
</tr>
</tbody>
</table>

Using the Tablespaces editor (DB2, Oracle)
The Tablespaces editor lets you modify settings and options of existing tablespaces.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage containers for a Tablespace (IBM DB2 for Windows, Unix, and Linux)
- Manage Tablespace performance (IBM DB2 for Windows, Unix, and Linux)
- Manage Tablespace quotas (Oracle)
- Manage Tablespace partitions (IBM DB2 for OS/390 or Z/OS)
- View Tablespace status (IBM DB2 for OS/390 or Z/OS)
- View table usage and space distribution for a Tablespace (IBM DB2, Oracle)
- View information on database objects on a Tablespace (IBM DB2 for OS/390 or Z/OS, Oracle)
- Manage Tablespace datafiles (Oracle)
- Manage Tablespace storage (Oracle)
- View Tablespace contents (Oracle)

You also use this editor to perform the following common Object editor tasks:

- Add a comment to a database object
- View SQL/DDL for a database object
• Work with privileges and permissions

Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th></th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Status</td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coalesce</td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorganize</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set Undo Tablespace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>

Manage containers for a Tablespace (IBM DB2 for Windows, Unix, and Linux)

The **Container** tab of the Tablespace Editor lets you add or modify containers of a tablespace.

**NOTE:** This option is only available for database managed tablespaces.

To manage containers for a tablespace:
1. Open an object editor on the tablespace to be modified.
2. Click the **Container** tab.
3. Take one of the following actions:
   • Click **Add** to set up a new container for the tablespace. For details, see Add a Container to a tablespace.
   • Select one of the containers displayed on the tab and click **Edit** to modify details for that container. For details, see Change Container Size.
4. Preview and submit changes.

Add a Container to a tablespace

You can increase the size of a tablespace by adding one or more containers to the tablespace.

To add a container to a tablespace:
1. Open an object editor on the tablespace to be modified.
2. Click the **Container** tab.
3. Click **Add** to open the Add Container for Tablespace dialog.
4. Use the following table as a guide to providing details on this dialog:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>Lets you select the node(s).</td>
</tr>
</tbody>
</table>
5 Click **Add**.

6 **Preview and submit changes**.

### Modify a container stripe set

When a tablespace is created, its tablespace map is created and all of the initial containers are lined up such that they all start in stripe 0. This means that data will be striped evenly across all of the table space containers until the individual containers fill up.

The Tablespace editor lets you add a container to an existing table space or extend a container to increase its storage capacity. For more information, see [Using the Tablespaces editor (DB2, Oracle)](#).

Adding a container which is smaller than existing containers results in an uneven distribution of data. This can cause parallel I/O operations, such as prefetching data, to perform less efficiently than they otherwise could on containers of equal size.

When adding a new container, the placement of that container within the new map depends on its size and the size of the other containers in its stripe set. If the container is large enough such that it can start at the first stripe in the stripe set and end at (or beyond) the last stripe in the stripe set, then it will be placed that way. If the container is not large enough to do this, it will be positioned in the map such that it ends in the last stripe of the stripe set. This is done to minimize the amount of data that needs to be rebalanced.

### Manage Tablespace performance (IBM DB2 for Windows, Unix, and Linux)

The **Performance** tab of the Tablespaces Editor lets you manage performance for every tablespace on the current datasource.

**To manage performance for a tablespace:**

1. **Open an object editor** on the tablespace to be modified.

2. **Click the Performance tab.**
3 Use the following table as a guide to understanding and modifying settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Setting</td>
<td>Lets you view Page Size and Extent Size settings and modify the Prefetch Size setting.</td>
</tr>
<tr>
<td>I/O Setting</td>
<td>Let you modify the Overhead and Transfer Rate settings.</td>
</tr>
<tr>
<td>Dropped Table Settings</td>
<td>Lets you view the Recovery Status.</td>
</tr>
<tr>
<td>Defaults</td>
<td>Lets you view the Nodegroup and specify a Bufferpool.</td>
</tr>
</tbody>
</table>

4 Preview and submit changes.

Manage Tablespace quotas (Oracle)
Oracle limits the amount of space that can be allocated for storage of a user’s objects within the specified tablespace to the amount of the quota. Users with privileges to create certain types of objects can create those objects in the specified tablespace. The Quotas tab of the Tablespace editor lets you manage user space quotas for tablespaces on the current datasource:

To manage user quotas for a tablespace:
1 Open an object editor on the tablespace to be modified.
2 Click the Quotas tab.
3 Optionally, take one of the following actions:
   • Click Add or select a user and click Edit to assign a user unlimited or a specific space usage quota on the tablespace
   • Select an existing user and click Drop to delete the quota for that user
4 Preview and submit changes.

Manage Tablespace partitions (IBM DB2 for OS/390 or Z/OS)
The Partition tab of the Tablespace Editor lets you partition tablespace on the current datasource.

To manage partitions for a tablespace:
1 Open an object editor on the tablespace to be modified.
2 Click the Partition tab.
   The tab displays a list of partitions for the tablespace.
3 Select a partition and click Edit to open a dialog that lets you modify properties for that partition.
4 Preview and submit changes.

View Tablespace status (IBM DB2 for OS/390 or Z/OS)
The Status tab of the Tablespaces Editor lets you view the current status of the following items:
To view tablespace status:
1. Open an object editor on the tablespace to be modified.
2. Click the Status tab.
3. From the Display Status list, choose one of the following status types:
   - WEPR
   - Claimers
   - Locks
   - LPL
   - Use

Manage Tablespace properties (IBM DB2 for OS/390 or Z/OS)
The Definition tab of the Tablespaces Editor lets you manage the parameters for the tablespace.

To manage properties for a tablespace:
1. Open an object editor on the tablespace to be modified.
2. Click the Definition tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sizing</td>
<td>Lets you work with the Buffer Pool and Max rows per page properties.</td>
</tr>
<tr>
<td>Locking</td>
<td>Lets you work with the Lock Size, Maximum Locks, Use System Value, and Selective Partition Locking properties.</td>
</tr>
<tr>
<td>Logging</td>
<td>Lets you enable and disable locking.</td>
</tr>
<tr>
<td>Text Default</td>
<td>Lets you work with the Encoding Scheme property.</td>
</tr>
<tr>
<td>Dataset Handling</td>
<td>Lets you work with the Close Rule and Erase Rule properties.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.
For more information, see Tablespace Editor.

View table usage and space distribution for a Tablespace (IBM DB2, Oracle)
The Space tab of the Tablespace Editor lets you view the table usage and the distribution of space for every tablespace on the current datasource.

To manage space for a tablespace:
1. Open an object editor on the tablespace.
2. Click the Space tab.
3. Optionally, if working with Oracle, you can Coalesce the tablespace. For more information, see Coalesce.
4. Preview and submit changes.
View information on database objects on a Tablespace (IBM DB2 for OS/390 or Z/OS, Oracle)

The **Objects** tab of the Tablespace Editor lets you view database objects stored on a tablespace.

**To view objects stored on a tablespace:**
1. Open an object editor on the tablespace.
2. Click the **Objects** tab.
   
   Database objects are grouped by object type, under folders.
3. Optionally, if you are working with Oracle, take one of the following actions:
   - Specify a logging option using the **Log Changes When Scheme/Data is Modified?** radio set.
   - Select an object under one of the object folders and click **Edit** to open an object editor on that object.
4. Preview and submit changes.

Manage Tablespace datafiles (Oracle)

The **Datafile** tab of the Tablespace Editor lets you manage datafiles for tablespaces on the current datasource.

**TIP:** You can view and manage the Auto-Undo option on the Datafile Tab.

**To add or edit a datafile for a tablespace:**
1. Open an object editor on the tablespace to be modified.
2. Click the **Datafile** tab.
   
   The tab lists details for each datafile on the tablespace.
3. Either click **Add** or select a datafile and click **Edit**. For more information, see **Add or Modify Datafile**.
4. Preview and submit changes.

Manage Tablespace storage (Oracle)

The **Storage** tab of the Tablespace Editor lets you view storage details for tablespaces on the current datasource.

**TIP:** Always create tablespaces for user data and never place user tables and indexes in the SYSTEM tablespace. Placing user objects in the SYSTEM tablespace can degrade performance and introduce space-related headaches to the database.

Oracle8i or later supports locally managed tablespaces, which can all but eliminate the problem of tablespace fragmentation. It totally does away with the storage parameters of MINEXTENTS, MAXEXENTS, PCTINCREASE, and NEXT. With locally managed tablespaces you either specify the initial extent size and let Oracle automatically size all other extents, or specify a uniform extent size for everything.

For users using a version earlier than Oracle 8i and locally managed tablespaces, there are manual methods can employ to assist in the fight against tablespace fragmentation. They include:

- Setting PCTINCREASE to zero for all tablespaces and objects to promote same-sized extents.
- Specifying equal-sized allotments for your INITIAL and NEXT object storage parameters.
• Grouping objects with like growth and storage needs together in their own tablespaces.

**TIP:** One of the best ways to avoid fragmentation in a tablespace is to pre-allocate the space that your objects will use. If possible, plan for one to two years’ growth for each object and allocate your space accordingly. Having initial empty objects will not affect table scan times as Oracle only scans up to the high-water mark (the last used block) in a table.

Of all your tablespaces, you want to avoid fragmentation problems in your SYSTEM tablespace the most as this is the major hotbed tablespace for Oracle activities. The easiest way to avoid this is to not allow any user (even the default DBA ID's SYS and SYSTEM) to have access to it. There are three ways to do this:

• Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.

• Ensure no user has a quota set for SYSTEM.

Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

**To view storage settings for a tablespace:**
1. Open an object editor on the tablespace to be modified.
2. Click the Storage tab.
3. Use the following table as a guide to understanding the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extents</td>
<td>Lets you view Initial Extent, Next Extent, Minimum Extent Size, Minimum Extents, Maximum Extents, and Percent Increase values.</td>
</tr>
<tr>
<td>Segment Management</td>
<td>Lets you view whether segment management is Automatic or Manual, and if manual, the TableSpace Block Size.</td>
</tr>
<tr>
<td>Extent Details</td>
<td>Lists details for each extent.</td>
</tr>
<tr>
<td>Extent Management</td>
<td>Lets you view whether the Type of extent management is LOCAL or DICTIONARY, whether the Allocation Type is AUTO, UNIFORM, or USER, and the Extent Size.</td>
</tr>
</tbody>
</table>

**View Tablespace contents (Oracle)**
The Map tab of the tablespace editor lets you view a tablespace map. Options on the tab let you perform tasks such as scanning for honeycomb or bubble space fragmentation.

**To work with a tablespace map:**
1. Open an object editor on the tablespace.
2. Click the Map tab.
3. Use the following table as a guide to understanding and working with the tablespace map:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Info</td>
<td>Provides size details.</td>
</tr>
<tr>
<td>Map</td>
<td>Displays the tablespace map.</td>
</tr>
<tr>
<td>Object Demographics</td>
<td>Provides details on each block of the tablespace</td>
</tr>
</tbody>
</table>
4 Optionally, take one of the following actions:
   • Select a block on the map and click Reorganize to Rebuild Indexes.
   • Click Legend to view a description of the map.
   • Right-click on a block of the map to open a context menu with viewing options.
   • Hover the mouse over a block of the map to view details for that block.
   • Click a block in the Object Demographics group to highlight that block in the map.

5 Preview and submit changes.

Using the Triggers editor
The Triggers editor lets you modify settings and options of existing triggers.

   NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:
   • Modify a trigger definition (IBM DB2, Microsoft SQL Server, Sybase)
   • Modify the Trigger action (Oracle)
   • View trigger options (IBM DB2)
   • Modify trigger properties (Oracle)
   • View the SQL for a Trigger (IBM DB2 for OS/390 and Z/OS)

You also use this editor to perform the following common Object editor tasks:
   • Add a comment to a database object (IBM DB2 for Windows, Unix, and Linux, Oracle)
   • Work with object dependencies
   • View SQL/DDL for a database object

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Compile</td>
<td>Compile</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disable Triggers</td>
<td>Disable</td>
<td>Disable</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enable Triggers</td>
<td>Enable</td>
<td>Enable</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td>Rename</td>
<td>Rename</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Modify a trigger definition (IBM DB2, Microsoft SQL Server, Sybase)
The Definition tab of the Triggers Editor lets you modify the CREATE TRIGGER body for a trigger. To modify a trigger, DBArtisan must drop then create the trigger.
To modify a trigger definition:
1. Open an object editor on the trigger to be modified.
2. Click the Definition tab.
3. Edit the text of the trigger body in the Trigger Text area.
4. Preview and submit changes.

Modify the Trigger action (Oracle)
The Action tab of the Triggers Editor lets you modify the trigger action PL/SQL block for any trigger on the datasource.

To modify a trigger action:
1. Open an object editor on the trigger to be modified.
2. Click the Action tab.
3. Modify the body of the trigger in the Trigger Action (PL/SQL Block) area.
4. Preview and submit changes.

View trigger options (IBM DB2)
The Properties tab of the Triggers Editor lets you modify trigger options for triggers on the datasource.

To modify trigger options:
1. Open an object editor on the trigger to be modified.
2. Click the Properties tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Lets you view the Trigger Timing, Trigger Events, Trigger Type, Object Status, Definer, and Function Path properties.</td>
</tr>
</tbody>
</table>

Modify trigger properties (Oracle)
The Type tab of the Triggers Editor lets you modify trigger options for triggers on the datasource.

To modify trigger options:
1. Open an object editor on the trigger to be modified.
2. Click the Type tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status (Oracle only)</td>
<td>Controls the status (Enable/Disable) the trigger.</td>
</tr>
<tr>
<td>Event</td>
<td>Specifies that the trigger executes on Insert, Update, or Delete.</td>
</tr>
</tbody>
</table>
4 Preview and submit changes.

**View the SQL for a Trigger (IBM DB2 for OS/390 and Z/OS)**
The Type tab of the Triggers Editor lets you view the SQL code for every trigger on the current datasource.

**To view SQL for a trigger:**
1. Open an object editor on the trigger.
2. Click the Type tab.

**Using the Unique Keys editor**
The Unique Keys editor lets you modify settings and options of existing unique keys.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage Unique Key columns (All DBMS)
- Work with Unique Key properties (IBM DB2 for Windows, Unix, and Linux, Oracle, Microsoft SQL Server, Sybase)
- Manage Unique Key storage (Oracle)
- Manage Unique Key partitions (Oracle)
- View page and row statistics for a Unique Key (Sybase, Microsoft SQL Server)
- View space allocations for a unique key (IBM DB2 for Linux, Unix, and Windows, Oracle)

You also use this editor to perform the following common Object editor tasks:

- Add a comment to a database object
- View SQL/DDL for a database object
Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th></th>
<th>DB2</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocate Extent</td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Deallocate Unused Space</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Rebuild Unusable Local Indexes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Rename</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>

**Manage Unique Key columns (All DBMS)**

The **Columns** tab of the Unique Keys Editor lets you manage columns for unique keys on the current datasource.

**To manage the columns of a unique key:**

1. **Open an object editor** on the unique key to be modified.
2. Click the **Columns** tab.
   - The tab provides a listing of columns for the unique key.
3. Optionally, take one of the following actions:
   - Click the New button to add a new column to the unique key.
   - Select a column and click the Delete key to remove that column from the unique key.
   - **NOTE:** Use the **Sort** control to specify a sort option for a column.
4. **Preview and submit changes**.

**Work with Unique Key properties (IBM DB2 for Windows, Unix, and Linux, Oracle, Microsoft SQL Server, Sybase)**

The **Properties** tab of the Unique Keys Editor lets you view or modify basic unique key set properties.

**To work with basic properties of a unique key:**

1. **Open an object editor** on the unique key.
2. Click the **Properties** tab.
3. Use the following table as a guide to understanding the settings on this tab:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enforcing Index</strong> (Oracle only)</td>
<td>Lets you view <strong>User-Defined</strong>, <strong>Index Owner</strong>, and <strong>Index Name</strong> settings.</td>
</tr>
</tbody>
</table>
Manage Unique Key storage (Oracle)
The Storage tab of the Unique Keys Editor lets you manage storage for every unique key on the current datasource.

To manage storage for a unique key:
1. Open an object editor on the unique key to be modified.
2. Click the Storage tab.
3. Use the following table as a guide to understanding and modifying the settings on this tab.

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
</table>
| Attributes             | For IBM DB2 for Windows, Unix, and Linux, lets you view the Defined By property.  
                          | For Oracle, lets you set No Sort, Logging, Reverse, Validate, Deferrable, Deferred, Enabled, Cascade, Rely, and Update Date properties.  
                          | For Microsoft SQL Server, lets you set the Clustered property.  
                          | For Sybase, lets you set Clustered and Maximum Rows Per Page properties.  |
| Storage                | For IBM DB2 for Windows, Unix, and Linux, lets you view the Percent Free and Minimum Percent Used property.  
                          | For Microsoft SQL Server, lets you set the File Group and Fill Factor properties.  
                          | For Sybase, lets you set the Reserve Page Gap, Segment Name, and Fill Factor properties.  |

4. **Preview and submit changes.**

Manage Unique Key partitions (Oracle)
The Partitions tab of the Unique Keys Editor lets you partition unique keys on the current datasource.

To manage partitions for a unique key:
1. Open an object editor on the unique key to be modified.
2. Click the Partitions tab.

**NOTE:** If the unique key is not currently partitioned, this tab shows a Partition button you can use to partition it. For more information, see Convert to Partitioned.
3. Use the following table as a guide to understanding the settings on this page:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Lets you view the <strong>Locality</strong> (Global/Local), <strong>Alignment</strong> (Prefixed/Non-Prefixed), <strong>Partition Type</strong> (including Range-Hash Composite or Range-List Composite), and <strong>Subpartition type</strong> properties.</td>
</tr>
<tr>
<td>Columns</td>
<td>Displays partitioning columns.</td>
</tr>
<tr>
<td>Partition Definitions</td>
<td>Displays details for each partition.</td>
</tr>
</tbody>
</table>

4. **Preview and submit changes.**

View page and row statistics for a Unique Key (Sybase, Microsoft SQL Server)

The **Statistics** tab of the Unique Keys Editor lets you view the page and row statistics for every unique key on the current datasource.

**To view page and row statistics:**

1. [Open an object editor] on the unique key.
2. Click the **Statistics** tab.
3. Use the following table as a guide to understanding the settings displayed:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Statistics</td>
<td><strong>Data Pages</strong>, <strong>Pages Reserved</strong>, <strong>Used Pages</strong>, <strong>OAM Page Ratio</strong> (Sybase only), <strong>Index Page Ratio</strong> (Sybase only), and <strong>Total Pages Modified</strong> (Microsoft SQL Server only)</td>
</tr>
<tr>
<td>Row Statistics</td>
<td><strong>Maximum Rows Per Page</strong> (Sybase only), <strong>Minimum Rows Per Page</strong> (Sybase only), <strong>Maximum Row Size</strong>, <strong>Max Size of Non-Leaf Index Row</strong>, and <strong>Total Rows Modified Since Last</strong> (Microsoft SQL Server only)</td>
</tr>
</tbody>
</table>

4. **Preview and submit changes.**

View space allocations for a unique key (IBM DB2 for Linux, Unix, and Windows, Oracle)

The **Space** tab of the Unique Keys Editor lets you view space details for a unique keys.

**To view space allocations for a unique key:**

1. [Open an object editor] on the unique key to be modified.
2. Click the **Space** tab.
3. Use the following table as a guide to understanding the settings displayed:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Utilization (Oracle)</td>
<td><strong>Size and number of Blocks</strong></td>
</tr>
</tbody>
</table>

4. **Preview and submit changes.**
Using the User Datatypes editor (IBM DB2, Microsoft SQL Server, Sybase)

The User Datatypes editor lets you modify settings and options of existing user datatypes.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Modify base user datatype properties (IBM DB2, Microsoft SQL Server, Sybase)
- Display column usage information for a User Datatype (Microsoft SQL Server, Sybase)

You also use this editor to perform the following common Object editor tasks:

- Add a comment to a database object
- View SQL/DDL for a database object

Lastly, the Command menu on the Object editor toolbar provides the following command:

- Rename

Modify base user datatype properties (IBM DB2, Microsoft SQL Server, Sybase)

The Properties tab of the User Datatypes Editor lets you modify a user datatype to change its base datatype properties.

**To modify the definition of a user datatype:**

1. Open an object editor on the user datatype to be modified.
2. Click the Properties tab.
3. Use the following table as a guide to modifying a user datatype for the DBMS DBArtisan is working with:

<table>
<thead>
<tr>
<th>Group</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics</td>
<td>For IBM DB2 for Windows, Unix, and Linux, lets you view Index Level, Cluster Ration, Cluster Factor, Leaf Pages, Sequential Pages, and Density settings. For Oracle, lets you view Index Level, Distinct Keys, Cluster Factor, Leaf Blocks, Leaf Blks/Key and Data Blks/Key settings.</td>
</tr>
<tr>
<td>Cards (IBM DB2 for Windows, Unix, and Linux)</td>
<td>Lets you view First Key, First 2 Keys, First 3 Keys, First 4 Keys and Full Keys settings.</td>
</tr>
<tr>
<td>Extents (Oracle)</td>
<td>Lets you view Number of Extents and Maximum Extents properties.</td>
</tr>
</tbody>
</table>
Display column usage information for a User Datatype (Microsoft SQL Server, Sybase)

The Usage tab of the User Datatypes Editor displays all objects referring to or referenced by a user datatype.

To view columns defined as using a user datatype:
1. Open an object editor on the user datatype.
2. Click the Usage tab.

Using the User Messages Editor (Microsoft SQL Server, Sybase)

The User Messages editor lets you modify settings and options of existing user messages.

NOTE: Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:
- View, add, and edit User Messages (Microsoft SQL Server, Sybase)
- Manage object bindings for User Messages (Sybase)

You also use this editor to perform the following common Object editor tasks:
- View SQL/DDL for a database object

View, add, and edit User Messages (Microsoft SQL Server, Sybase)

The Information tab of the User Messages Editor lets you view, add, and edit user message on the current datasource. The Information tab displays the language and contents of user messages.

To manage user message content:
1. Open an object editor on the user message to be modified.
2. Click the Information tab.
3 Take one of the following actions:

- Click Add to open a dialog that lets you create a user message or select an existing message and click Edit to open a dialog that lets you modify the content or language for that message. For more information, see Create or Modify User Message Text.
- Select a message and click Delete to delete that message.

4 Preview and submit changes.

Manage object bindings for User Messages (Sybase)

Messages can be bound to foreign key constraints or check key constraints. When a message is bound to a constraint, when that constraint value is breached, users receive the message you create. For information on setting up user messages, see View, add, and edit User Messages (Microsoft SQL Server, Sybase).

The Bindings tab of the User Messages Editor lets you manage object bindings for user messages on the current datasource. Sybase ASE lets you bind a user message to user-defined datatypes and to columns in any table in a database (if the default value is appropriate to the datatype of the column.)

To manage bindings for a user messages:

1 Open an object editor on the user message to be modified.
2 Click the Bindings tab.

Any existing bindings are displayed in the list on the tab, and grouped according to whether the binding is to a foreign key constraint or a check key constraint.

3 Optionally you can take one of the following actions:

- Create a new binding by clicking the Bind button to open the Bind User Message dialog box then choosing the constraint type, selecting one or more constraints of that type, and clicking the Execute button.
- Remove existing bindings by selecting one or more bindings below the Check Constraints or Foreign Key Constraints folders, clicking the Unbind button to open the Unbind Constraints from User Message dialog, and then clicking the Execute button.

Using the Users editor

The Users editor lets you modify settings and options of existing users.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- Manage database objects associated with a User (IBM DB2 for OS/390 and Z/OS, Microsoft SQL Server, Oracle, Sybase)
- Manage user roles (Microsoft SQL Server, Oracle)
- Manage user properties (All DBMS)

You also use this editor to perform the following common Object editor tasks:

- Work with privileges and permissions
• View SQL/DDL for a database object
• Work with object dependencies

Lastly, the Command menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze Schema</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td>For more information, see Analyze.</td>
</tr>
<tr>
<td>Change Password</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compile Schema</td>
<td></td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td>For more information, see Compile.</td>
</tr>
<tr>
<td>Create Like</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td>YES</td>
<td></td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Manage database objects associated with a User (IBM DB2 for OS/390 and Z/OS, Microsoft SQL Server, Oracle, Sybase)

The Objects tab of the Users Editor lets you manage database objects associated with every user on the current datasource. DBArtisan organizes the objects in a tree structure with folders containing the objects.

To manage objects associated with a user:
1. Open an object editor on the user.
2. Click the Objects tab.
3. Optionally, take one of the following actions:
   • Select an object and click Edit to open an object editor on the selected object.
   • Select an object and click Drop to initiate dropping the selected object.
4. Preview and submit changes.

Manage user roles (Microsoft SQL Server, Oracle)

The Roles tab of the Users Editor lets you assign roles to a user.

To assign roles to a user:
1. Open an object editor on the user to be modified.
2. Click the Roles tab.
3. Select the check box for each role to be assigned to this user:
4. Preview and submit changes.

Manage user properties (All DBMS)

The Properties tab of the Users Editor lets you manage basic user properties.
To modify properties for a user:

1. Open an object editor on the user to be modified.
2. Click the Definition tab.
3. Use the following table to understand and modify the user properties on this tab:

<table>
<thead>
<tr>
<th>DBMS</th>
<th>User Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM DB2</td>
<td>Lets you view the user Name for the user.</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>Lets you view the user Name and Login Name for the user.</td>
</tr>
<tr>
<td>Oracle</td>
<td>Lets you view the Name and External Name properties. Let you set the Default Tablespace, Temporary Tablespace, Profile, Identified By, Password, Account Locked, and Password Expired properties.</td>
</tr>
<tr>
<td>Sybase</td>
<td>Lets you view the user Name and Login Name for the user. Let you specify a Group Name for the user.</td>
</tr>
</tbody>
</table>

4. Preview and submit changes.

Using the Views editor

The Views editor lets you tailor and otherwise manage other details of views.

**NOTE:** Prior to working with this editor, you should be familiar with basic Object editor usage. You must know how to Open an object editor, View and modify database object properties, and how to Preview and submit changes. For detailed information, see Overview and common usage of Object editors.

You use this editor to perform the following tasks specific to this object type:

- View SQL for a View (IBM DB2, Microsoft SQL Server, Oracle, Sybase)
- Display columns for a view (All DBMS)

You also use this editor to perform the following common Object editor tasks:

- Add a comment to a database object
- Work with privileges and permissions
- Work with object dependencies
Lastly, the **Command** menu on the Object editor toolbar provides a set of utilities for working with objects of this type. The following table lists the utility, availability by DBMS system, and provides any useful notes:

<table>
<thead>
<tr>
<th>Utility</th>
<th>DB2 (W/U/L)</th>
<th>DB2 (OS/390 and Z/OS)</th>
<th>SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Query</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>For more information, see Describe Window.</td>
</tr>
<tr>
<td>Compile</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Alias</td>
<td>YES</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Synonym</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Describe</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>For more information, see Describe Window.</td>
</tr>
<tr>
<td>Indexes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select * From</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>For more information, see Viewing Schema.</td>
</tr>
<tr>
<td>Schema</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Triggers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** For IBM DB2 for Windows, Unix, and Linux (version9), you can update statistics for a view. For details, see Updating statistics for views (IBM DB2 for Windows, Unix, and Linux).

**View SQL for a View (IBM DB2, Microsoft SQL Server, Oracle, Sybase)**

The **Definition** tab of the Views editor lets you view the SQL code for every view on the current datasource.

**To view SQL for a view:**
1. Open an object editor on the view.
2. Click the **Definition** tab.

**Display columns for a view (All DBMS)**

The **Properties** tab of the Views editor lets you view column information for every view on the current datasource.

**To view column details for a view:**
1. Open an object editor on the view to be modified.
2. Click the **Properties** tab.

The tab displays all columns for the view. Details for each column include the **Column Name**, the **Datatype** (and if applicable, with the precision in parentheses), and whether or not **Nulls** are allowed for that column.

**NOTE:** If DBArtisan is working with IBM DB2 for Windows, Unix, and Linux, you can also set the **EnableQueryOptimization** property.
## Functionality

The table below lists functionality available in DBArtisan.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>IBM DB2 for Windows/Unix</th>
<th>IBM DB2 for OS/390 and z/OS</th>
<th>Microsoft SQL Server</th>
<th>Oracle</th>
<th>Sybase ASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate Logging</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add, Insert, or Modify Partition</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Add or Modify Datafile</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Add or Modify Cluster Column</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allocate Extent</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backup (see Data Unload Wizard for Microsoft SQL Server and Sybase ASE)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bind Data Cache</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Bind Package</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bind Plan</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build Query</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Change Category</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Container Size</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Change Database Comment</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Password</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Change Status</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check Index</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checkpoint</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Clone Partition</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clone Table</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coalesce</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compile</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection Editor</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Convert to Partitioned</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Copy Object Names</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Copy Schema</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy to Copy</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Alias</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Insert Statements</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create Like</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Functionality</td>
<td>IBM DB2 for Windows/Unix</td>
<td>IBM DB2 for OS/390 and z/OS</td>
<td>Microsoft SQL Server</td>
<td>Oracle</td>
<td>Sybase ASE</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------</td>
<td>-----------------------------</td>
<td>----------------------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>Create or Edit User Accounts</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create or Modify Constraint</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create or Modify User Message Text</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create Synonym</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataLink Options</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBCC</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deallocate Unused Space</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detach/Attach</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disable/Enable Job Queues</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disable/Enable Trigger</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk Resize</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drop</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Edit Data</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Error</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Estimate Size</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extract</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Extract Data as XML</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flush Cache</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Free Packages</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Free Plans</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grant Roles</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Impact Analysis</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Load Java</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lob Storage Definition</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Lock</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merge Copy</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Migrate</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Modify Constraint</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Recovery</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Statistics</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Open Control Files</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partitioned Columns</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partition Upper Bound</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functionality</td>
<td>IBM DB2 for Windows/Unix</td>
<td>IBM DB2 for OS/390 and z/OS</td>
<td>Microsoft SQL Server</td>
<td>Oracle</td>
<td>Sybase ASE</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------</td>
<td>----------------------------</td>
<td>----------------------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>Place</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Preview: Create</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Preview</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Quiesce Database</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quiesce Instance</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quiesce Tablespaces</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reassign by Category</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rebind Packages</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rebind Plans</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rebuild Index</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rebuild Outlines</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recompile</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redistribute</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refit</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refresh Table</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reinitialize</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Reorganize</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorganize Tablespace</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Restart</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revoke Role</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Runstats Index</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Runstats Tablespace</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Select * From</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Set Integrity</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set Online/Offline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Set Tablespaces Quota</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set UNDO Tablespace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shrink</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Database</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Stop Database</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summary Definition</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switch Online</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Completing a Dialog Box

NOTE: DBArtisan includes Wizards for creating new objects. For more information see Creating an Object.

1. On the Datasource Explorer, select the target object node.

   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select the target functionality.

   OR

   In the right pane of the application, right-click the target object, and then select the target functionality.

   DBArtisan opens the functionality dialog box.

3. Complete the dialog box.

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Preview]</td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td>![Schedule]</td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td>![Execute]</td>
<td>Executes the task.</td>
</tr>
<tr>
<td>![Analysis]</td>
<td>Opens the Impact Analysis dialog box. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see Functionality.

Creating an Object

DBArtisan offers easy-to-use wizards and dialog boxes for creating objects. Platform-by-platform discussions of wizards are discussed as follows:

IBM DB2 for Linux, Unix, and Windows Object Wizards
IBM DB2 for OS/390 and z/OS Object Wizards

**NOTE:** For enhanced functionality of IBM DB2 for OS/390 and z/OS versions 7 and 8, some components must be installed on the DB2 server system. Please review Installation Helpful Hints, System Requirements, and DBArtisan Server Components Installation Overview.

Microsoft SQL Server Object Wizards
Oracle Object Wizards
Sybase ASE Object Wizards
Completing a Dialog Box

Overview and common usage of object wizards
An object wizard lets you create new database and server objects on a datasource. For example, a Tables wizard lets you create columns, set up permissions to work with that table, provide details on physical storage for the table, and so on. Each pane on an object wizard lets you perform a logical task or collection of logical tasks for that object type. For example:

![Create Index Wizard - Step 1 of 6](image)

In order to work with object wizards, you must know how to:

- [Open an object wizard](#) for a server or database object.
- Navigate and set properties in an object wizard
- Create the new object.

Open an object wizard
Object wizards are accessed from the Datasource Explorer pane of the DBArtisan main window.

To open an Object wizard for a server or database object
1. Connect to the datasource where you want to create a new resource. For more information, see Datasource Management.
2. On the Datasource Explorer, expand the target datasource.
3. Continue to expand folders under the datasource until the type of object you want to create is visible.
4. On the Datasource Explorer, right-click on the specific type of object that you want to create and select New from the context menu.

DBArtisan opens the object wizard for that object type.

Navigate and set properties in an object wizard
When you invoke an object wizard, it opens on the first pane of the wizard, typically a Properties pane. As you select options or provide details, you use the navigation buttons at the bottom of the window and the pane controls at the left of the window to navigate through the wizard.
• Use the **Next** button to move to the next pane of the wizard.

**NOTE:** In some cases, the **Next** button is not activated until required information is provided. Similarly, some panes of a wizard do not become accessible until activated by choice of an option or property on a previous pane. For example, a **Partitions** tab in a table wizard may only become available if the **Clustered** and **Number of Partitions** options on a prior tab are set accordingly.

• Use the **Back** button to move to the previous pane of the wizard.

• Use the pane controls at the left of the window to move to a specific pane of the wizard.

### Create the new object

The final pane of a wizard is most commonly a **DDL View** or **Definition** pane. That pane lets you view the SQL or DDL generated to create the object with which you are working. In most cases you arrive at that pane by navigating through panes, choosing options as you proceed. Alternatively, if you have provided all required information for that object type, you can click **Finish** to view the generated SQL or DDL.

You can inspect the SQL or DDL in a **DDL View** or **Definition** pane and if necessary navigate to other panes to modify your option choices. In some **Definition** panes, you have to complete the DDL or SQL to create the object.

Lastly, when your option selections are complete, use the **Execute** button to create the new object.

### IBM DB2 for Linux, Unix, and Windows Object Wizards

<table>
<thead>
<tr>
<th>Alias Wizard</th>
<th>Bufferpool Wizard</th>
<th>Create Check Constraint Dialog Box</th>
<th>Database Wizard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Monitor Wizard</td>
<td>Foreign Key Wizard</td>
<td>Function Wizard</td>
<td>Group Wizard</td>
</tr>
<tr>
<td>Index Wizard</td>
<td>Materialized Query Table Wizard</td>
<td>Node Group Wizard</td>
<td>Create Primary Key Constraint Dialog Box</td>
</tr>
<tr>
<td>Procedure Wizard</td>
<td>Schema Wizard</td>
<td>Sequence Wizard</td>
<td>Table Wizard</td>
</tr>
<tr>
<td>Tablespace Wizard</td>
<td>Trigger Wizard</td>
<td>Type Wizard</td>
<td>Unique Key Wizard for IBM DB2 for Windows, Unix, and Linux</td>
</tr>
<tr>
<td>User Datatype Wizard</td>
<td>User Wizard</td>
<td>View Wizard</td>
<td>Set Integrity</td>
</tr>
</tbody>
</table>

### Alias Wizard for IBM DB2 for Linux, Unix, and Windows

An alias offers you security and convenience so that you can refer to an object without revealing who owns it or what database it belongs to. You can create aliases for tables, views, and even other aliases. The Alias Wizard lets you create an alias without knowing the underlying commands. As you complete the Alias Wizard process, DBArtisan constructs the necessary CREATE ALIAS statement based on the information that you supply.
To create a new alias using a wizard:
1. Open an object wizard for an alias.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>Select the schema that is to own the alias.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the alias.</td>
</tr>
<tr>
<td>Target Owner</td>
<td>Select the owner of the object to which you are creating an alias.</td>
</tr>
<tr>
<td>Target Type</td>
<td>Select the type of object to which you want to create an alias.</td>
</tr>
<tr>
<td>Target Name</td>
<td>Select the specific object to which you are creating an alias.</td>
</tr>
<tr>
<td>Comment</td>
<td>Optionally link a comment to the alias you are creating.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>

### Buffer Pool Wizard for IBM DB2 for Linux, Unix, and Windows

The Buffer Pool Wizard lets you create a bufferpool (an area of memory where data pages are read, altered, and held during an operation) without knowing the underlying commands. As you complete the Bufferpool Wizard, DBArtisan constructs the necessary `CREATE BUFFERPOOL` statement based on the information that you supply. Note that the bufferpool you’re creating to store the data from the tables in selected database won’t be active until you restart the database.

The Bufferpool Wizard lets you:

- Name the tablespace and specify the size of the bufferpool.
- Set storage options for UNIX systems.
- Set Bufferpool disbursement parameters.
- Override the default bufferpool size for partitions (or nodes).

### To Open the Bufferpool Wizard
1. On the **Explorer**, find the database where you want to add the new bufferpool.
2. On the **Buffer Pools** branch of the **Performance** node, right-click and select **New**. OR
   - Click the **Buffer Pools** branch and then select **New** from the toolbar.

The table below describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the bufferpool?</td>
<td>Give the new bufferpool it's name.</td>
</tr>
</tbody>
</table>
UNIX and Partitions

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the size of the bufferpool?</td>
<td>Indicate the number of pages you want to include in the new bufferpool. If the database is partitioned database, the default size is of all the partitions (or nodes) where the bufferpool exists. A value of -1, the default, means the size will be taken from the database configuration parameter (BUFFPAGE).</td>
</tr>
<tr>
<td>Pagesize (in KB)</td>
<td>The page size, in KB, is available in the increments 4, 8, 16, or 32.</td>
</tr>
</tbody>
</table>

Partition Size

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Extended Storage</td>
<td>Select if you are using a UNIX system and want to store migrated pages from the bufferpool in extended storage.</td>
</tr>
<tr>
<td>All Nodes</td>
<td>Disperses the bufferpool across all nodes.</td>
</tr>
<tr>
<td>Nodes in checked nodegroups</td>
<td>Specifies nodegroups where the bufferpool should reside.</td>
</tr>
</tbody>
</table>

Database Wizard for IBM DB2 for Linux, Unix, and Windows

The Database Wizard lets you create a database (a structured collection of data that can be updated or queried) without knowing the underlying commands. Databases can be simple, that is one file with many records and the same fields, or much more complicated with multiple files with different fields.

To Open the IBM DB2 for Linux, Unix, and Windows Database Wizard

On the Registration tool bar, click New UDB Database.

OR

On the Datasource Explorer, right-click an instance node or the Datasources node, and then click New UDB Database.

The tables below describe the fields you may encounter as you complete the wizard.
## Naming the New Database

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the database?</td>
<td>Each database requires a unique name. The database name should be between 1 - 8 characters and must begin with a letter A-Z. It's wise to avoid the symbols @, #, and $ if the database will exist in a communications environment or be summoned from an area where those characters do not exist in the language (and hence, on a keyboard).</td>
</tr>
<tr>
<td>What is the datasource name for the new database?</td>
<td>Each datasource should have a unique name. This is the name that will appear on the Explorer.</td>
</tr>
<tr>
<td>What is the alias name of the database.</td>
<td><strong>OPTIONAL</strong>: A database alias is the local synonym given to the database you are creating, and must be unique. If you don’t assign an alias, the database name becomes the default alias. Note that a DB2 client can have connections to two different databases with the same name if those databases live on different servers and each database has its own alias.</td>
</tr>
<tr>
<td>What is the comment used for the database?</td>
<td><strong>OPTIONAL</strong>: Lets you type a comment up to 30 characters. Any comment you enter can be changed later.</td>
</tr>
</tbody>
</table>

## Drive Path and Parameters

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On what drive/path will the database reside?</td>
<td><strong>OPTIONAL</strong>: Leave blank if you want DBArtisan to create the database using the DFTBPATH (default database path configuration) parameter.</td>
</tr>
<tr>
<td>What default tablespace parameters should be used?</td>
<td><strong>OPTIONAL</strong> Extent size: This is the number of pages of table data that will be written before data is written to the next container. Number of segments: The default is 0. The number you designate specifies the number of extended memory segments available for the database to use. You should only set this number if you have a large amount of real memory.</td>
</tr>
<tr>
<td>What global default parameters should the database use?</td>
<td>Territory: The territory where the database was created. This is purely informational, for example, en_US and is related to region-specific support. Codeset: IBM-1252 or UTF-8 are the options for language support. Collating Sequence: Compatibility, Identity, System are your choices for comparing character data. <strong>Note</strong>: The collation sequence can’t be changed after the database has been created.</td>
</tr>
<tr>
<td>Finish</td>
<td>Opens the <a href="#">Preview SQL dialog box</a></td>
</tr>
</tbody>
</table>
Add Catalog/Temporary/User Containers
There are three distinct wizard panels, one for each container type, that enable you to customize your database. Click **Next** or **Finish** depending on how you want to proceed.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What container(s) will be used to store the catalog/temporary tables?</td>
<td><strong>Use System Managed Space</strong>: If you select this option, the operating system’s file system manager allocates and manages the space where the catalog tables are stored. This is the default. <strong>Use Database Managed Space</strong>: If you select this option, the database manager controls the storage space for catalog tables. Add - Click to open the <a href="#">Add Container for Catalog/Temp/User Tablespace</a> dialog box. Edit - Click to open the <a href="#">Edit Container for Catalog/Temp/User Tablespace</a> dialog box.</td>
</tr>
<tr>
<td>What optional default storage parameters should be used.</td>
<td>Optionally identify values for extent size, pre-fetch size, transfer rate, and overhead.</td>
</tr>
<tr>
<td>Finish</td>
<td>Opens the <a href="#">Preview SQL dialog box</a>.</td>
</tr>
</tbody>
</table>

Add/Edit Container for Tablespace
The table below describes the options and functionality on the Add and Edit Container for Tablespace dialog box:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container Parameters</td>
<td>These parameters are enabled for modification only if you have opted to use database managed space. When enabled, you can either indicate file size parameters or specify device information.</td>
</tr>
<tr>
<td>Directory</td>
<td>Identify the directory where the catalog table container will be stored.</td>
</tr>
<tr>
<td>File Name</td>
<td>Identify a name for the file where the catalog table container will be stored.</td>
</tr>
<tr>
<td>Device Information</td>
<td>Enable either raw partition and type a value, or raw drive, and select a value.</td>
</tr>
</tbody>
</table>

Event Monitor Wizard for IBM DB2 for Windows/Unix
An event monitor is used to collect information about a particular database and connected appliances when one or more events you specify take(s) place. Where a snapshot monitor captures information for a given moment in time, the event monitor lets you trace the activity you specify. The Event Monitor Wizard lets you:

- Select the event owner, name, and type.
- Specify the target of the event monitor data.
- Specify any conditions for the three event types (connections, statements, and transactions.)
- Specify the definition of the event monitor.
- Select which partition(s) the event monitor runs against.
To Open the Event Monitor Wizard

1. On the **Explorer tree**, expand the **Performance** node, and select **Event Monitor**.
2. Right-click **Event Monitor** and select **New**.
   
   OR
   
   Select **New** from the toolbar.

The tables below describe the fields you may encounter as you move through the Event Monitor Wizard.

### Naming the Monitor and Selecting the Event(s)

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the event monitor?</td>
<td>Type a name for the monitor you are creating.</td>
</tr>
<tr>
<td>Do you want to autostart the event monitor when the database starts?</td>
<td>Yes or No.</td>
</tr>
<tr>
<td>Select the types of events you would like to monitor</td>
<td>From the check list, choose one, a few, or all monitorable events. The information available for each event type follows.</td>
</tr>
<tr>
<td></td>
<td><strong>Bufferpools</strong>: Counters for the buffer pool, page cleaners, prefetches, direct I/O.</td>
</tr>
<tr>
<td></td>
<td><strong>Connections</strong>: Application-level counters.</td>
</tr>
<tr>
<td></td>
<td><strong>Database</strong>: Database-level counters.</td>
</tr>
<tr>
<td></td>
<td><strong>Deadlocks</strong>: Involved applications, locks in contention. <strong>With Details</strong>: More comprehensive information including participating statements and a list of locks held. This option can affect performance when deadlocks occur.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE</strong>: Deadlocks with details is only available for IBM DB2 for Windows/Unix 8.</td>
</tr>
<tr>
<td></td>
<td><strong>Statements</strong>: Start or stop time, CPU used, text of dynamic SQL, and other metrics. For <strong>partitioned</strong> databases, CPU consumed, time of execution, and table queue information is also recorded.</td>
</tr>
<tr>
<td></td>
<td><strong>Tables</strong>: Row reads and writes for each table.</td>
</tr>
<tr>
<td></td>
<td><strong>Tablespaces</strong>: Bufferpool counters, prefetchers, page cleaners, and direct I/O.</td>
</tr>
<tr>
<td></td>
<td><strong>Transactions</strong>: Unit of work start/stop time, CPU consumed, and locking and logging metrics.</td>
</tr>
</tbody>
</table>

**NOTE:** Whenever you create a new DB2 database, an event monitor named DB2DETAILDEADLOCK is automatically created and starts when you activate the database. You can drop it if you want to avoid the overhead this event monitor incurs.
Event Monitor Target and Specifications

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where is the target for the Event Monitor?</td>
<td>Specify the target of the event monitor data. There are three types of target: to file, to pipe, and to table. A table event monitor streams event records to SQL tables. A pipe event monitor sends event records directly to a named pipe. A file event monitor sends records to a series of files with an .evt extension, although the data should be treated as a single logical file. <strong>NOTE:</strong> To Table option is only available for IBM DB2 for Windows/Unix 8.</td>
</tr>
<tr>
<td>When the event monitor is full do you want the database agents to block until the event data is written to the disk?</td>
<td>Specify a blocking option. A file type event monitor will keep saving event records to the specified file until the file reaches its limit. If it runs out of disk space, the event monitor shuts down and logs an error message. You can avoid this by removing full files.</td>
</tr>
<tr>
<td>Specify the buffer size for the Event Monitor</td>
<td><strong>OPTIONAL:</strong> Lets you specify buffer size in pages. When an event monitor writes to a file or table, the process buffers the records. Larger buffers reduce the number of disk accesses.</td>
</tr>
<tr>
<td>Specify the maximum number of files and the maximum file size for the Event Monitor.</td>
<td><strong>OPTIONAL:</strong> Lets you specify maxfiles and maxfilesize in pages. This option is available only for File output type.</td>
</tr>
<tr>
<td>How do you want the Event Monitor output written to the file?</td>
<td>Lets you specify an output option. <strong>Append:</strong> This is the default. When this is selected, the event records start writing at the end of the file it wrote to most recently. If you have removed that file, the number starts where the prior file left off. <strong>Replace:</strong> When you select this option, the existing event files are deleted and the numbering restarts.</td>
</tr>
</tbody>
</table>

Specifying Event Conditions

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Type</td>
<td>If you want to create an event monitor for the connections, statements, and/or transactions types, choose that type here.</td>
</tr>
<tr>
<td>Event Condition</td>
<td>Type a clause specific to the event type to set a filter. Only events that meet the criteria set by the filter are recorded.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the database partition where the event monitor runs and writes the events</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Where do you want this Event Monitor to collect data?</td>
<td>This is enabled if the only event type you want to monitor is deadlock events. The monitoring scope is local or global and is relevant only to file or pipe output. <strong>Local:</strong> Events are monitored as they occur locally on the monitored partition. <strong>Global:</strong> Activity from all partitions running on the DB2 database manager.</td>
</tr>
</tbody>
</table>
Foreign Key Wizard for IBM DB2 for Linux, Unix, and Windows

Foreign keys are unique values that refer to specific columns of other tables. Thus, a foreign key links two tables together. Embarcadero DBArtisan's Foreign Key Wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data. The Foreign Key Wizard lets you create a foreign key without knowing the underlying commands.

The Foreign Key Wizard lets you:

- Name the foreign key constraint
- Identify the parent table and the referenced constraint.
- Map the column pairs between the parent and child tables.

To create a new foreign key using a wizard:

1. Open an object wizard for a foreign key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td><strong>Table Schema</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Table Name</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Enabled</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Delete Rule</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Update Rule</strong></td>
</tr>
<tr>
<td><strong>Column Mapping</strong></td>
<td><strong>Under Referenced Table</strong>, choose the <strong>Owner</strong> and then the <strong>Name</strong> of the referenced, or parent, table. Under the <strong>Main Table</strong>, select checkboxes corresponding to the columns that are to reference columns in the referenced table. Then, under <strong>Referenced Table</strong>, select the corresponding column check boxes.</td>
</tr>
<tr>
<td><strong>Comment</strong></td>
<td>Optionally link a comment to the alias you are creating.</td>
</tr>
<tr>
<td><strong>DDL View</strong></td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to Create the new object.</td>
</tr>
</tbody>
</table>
Function Wizard for IBM DB2 for Linux, Unix, and Windows

To create a relationship between one set of values and another, DBArtisan offers the Function Wizard. You can develop reusable subroutines so you can control, access, and manipulate the data that underlies an object. As you complete the Function Wizard process, DBArtisan constructs the necessary CREATE FUNCTION statement based on the information that you supply. The Function Wizard lets you create a function without knowing the underlying commands.

**NOTE:** To create a user-defined function, you need CREATE ANY privileges or IMPLICIT_SCHEMA authority on the database if the schema does not already exist.

The Function Wizard lets you:

- Specify the function name, owner, and type.
- Specify the datatypes for the parameters of the new function.
- Specify special options of how and when the function should act.

To create a new function using a wizard:

1. Open an object wizard for a foreign key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>Select the owner of the function.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the function.</td>
</tr>
<tr>
<td>Specific Name</td>
<td>Provide the Specific name to be used by some SQL statements and DB2 commands for this function.</td>
</tr>
<tr>
<td>Function Type</td>
<td>Select the type of function:</td>
</tr>
<tr>
<td></td>
<td><strong>External Scalar</strong> - written in a programming language and the value returned is scalar.</td>
</tr>
<tr>
<td></td>
<td><strong>External Table</strong> - written in a programming language and the value returned is a complete table.</td>
</tr>
<tr>
<td></td>
<td><strong>OLEDB</strong> - accesses OLE DB data in user-defined OLE DB external tables.</td>
</tr>
<tr>
<td></td>
<td><strong>Sourced</strong> - another function is invoked to implement the function you are creating.</td>
</tr>
<tr>
<td></td>
<td><strong>SQL Language</strong> - written in SQL and returns a table, scalar value, or single row.</td>
</tr>
<tr>
<td></td>
<td><strong>Template</strong>: - this is a partial function and can only be invoked from a federated datasource.</td>
</tr>
<tr>
<td></td>
<td>For more information, see About function types.</td>
</tr>
<tr>
<td>Language</td>
<td>If you chose a Function Type of EXTERNAL SCALAR or EXTERNAL TABLE, specify a language of C, JAVA, or OLE.</td>
</tr>
<tr>
<td></td>
<td>For more information, see About function types.</td>
</tr>
<tr>
<td>Return Type</td>
<td>If you chose a Function Type of SQL, select ROW, TABLE, or SCALAR.</td>
</tr>
<tr>
<td></td>
<td>For other Function Type choices, this indicates the default return type for that choice.</td>
</tr>
<tr>
<td>External Name</td>
<td>Provide the External Name of the function.</td>
</tr>
</tbody>
</table>
**About function types**

**External Scalar/Table/OLE DB Function**
External scalar user-defined functions are implemented in an external programming language. The functions are executed on the server and can read SQL data but cannot make changes to the data. These functions are often used to extend the set of built-in functions for DB2, perform logic inside a SQL query that SQL can’t perform on its own, and, encapsulate a scalar query that is often used as a subquery in SQL statements, for example, if given an ingredient, search a table for a recipe that uses that ingredient.

When specifying the **Specific Name** for one of these function types:

- If you are using C language, specify the full library path and the function name, otherwise IBM DB2 Database Manager assumes the function is under the IBM DB2 library.

- If you are using Java script, specify the Class ID and the function name, otherwise IBM DB2 Database Manager assumes the function is under the IBM DB2 library.

- If you are using OLE language, specify the full library path and the function name, otherwise IBM DB2 Database Manager assumes the function is under the IBM DB2 library.

**Sourced Functions**
When you create a sourced function, the new function you are creating will be implemented by a preexisting (source) function that’s known to the database manager. The source function can be a built-in function or a previously created user-defined scalar function.
When you select the appropriate schema, you are really specifying an implicit schema privilege. In other words, you’re selecting a schema/function that belongs to a user with DBAdmin privileges. If you want to use a built-in function, you must specify the function’s specific name.

**SQL Language Function**
The function you are creating is written in SQL. A table, scalar value, or single row is returned.

**Template Function**
A template function is better thought of as a template for a function. It’s a partial function that has no executable code. You create a function template for mapping it to a datasource function. When the mapping is created, you can specify the function template be used in queries submitted to the federated server. When the query is processed, the federated server invokes datasource function where the template is mapped and returns the appropriate values.

**Group Wizard for IBM DB2 for Windows/Unix**
The Group Wizard lets you define new groups and to assign the appropriate object and system level privileges. Any user who is a part of the group you create will inherit the authorities and privileges assigned to the group.

**To Open the Group Wizard**
1. On the Explorer, find the datasource where you want to add the new group.
2. Expand the Security branch, right-click Groups, and select New.

The table that follows describes the fields you may encounter as you complete the Group Wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the group?</td>
<td>Lets you enter the name of the group.</td>
</tr>
<tr>
<td>What database authority(ies) do you</td>
<td>Select all or some of the authorities:</td>
</tr>
<tr>
<td>want to give this group?</td>
<td>BINDADD: User can create new database packages</td>
</tr>
<tr>
<td></td>
<td>CONNECT: Gives users access to the database</td>
</tr>
<tr>
<td></td>
<td>CREATETAB: A user can create new tables in the database</td>
</tr>
<tr>
<td></td>
<td>CREATE_NOT_FENCED: Allows a user to create user-defined functions</td>
</tr>
<tr>
<td></td>
<td>IMPLICIT_SCHEMA: Enables any user to implicitly create a schema by creating an object with a CREATE statement using a schema name that doesn’t already exist.</td>
</tr>
<tr>
<td></td>
<td>DBADM: Gives users administrative privileges.</td>
</tr>
<tr>
<td></td>
<td>LOAD: Users can load data into a table.</td>
</tr>
<tr>
<td><strong>NOTE</strong>: The authorities listed here</td>
<td>The authorities listed here are the system privileges. You can specify object privileges after the new group has been created using the Using the Groups editor (IBM DB2 for Windows, Unix, and Linux, Oracle, Sybase).</td>
</tr>
</tbody>
</table>

**Index Wizard for IBM DB2 for Linux, Unix, and Windows**
Comparable to an index in a book, an index gives you speedy access to particular records in a table. The Index Wizard lets you create an index without knowing the underlying commands.
To create a new index using a wizard:

1. Open an object wizard for an index.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td><strong>Table Schema and Table Name</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Schema and Name</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Index Type</strong></td>
</tr>
<tr>
<td><strong>Clustered</strong></td>
<td>Specifies that the index is the clustering index of the table. The cluster factor of a clustering index is maintained or improved dynamically as data is inserted into the associated table, by attempting to insert new rows physically close to the rows for which the key values of this index are in the same range.</td>
</tr>
<tr>
<td><strong>Percent Free</strong></td>
<td>Lets you type or select the percentage of each index page to leave as free space when building the index, from 0 to 99.</td>
</tr>
<tr>
<td><strong>Minimum Percentage Used</strong></td>
<td>Lets you type or select the minimum percentage of space used on an index leaf page. If, after a key is removed from an index leaf page, the percentage of space used on the page is at or below integer percent, an attempt is made to merge the remaining keys on this page with those of a neighboring page. If there is sufficient space on one of these pages, the merge is performed and one of the pages is deleted. The value of integer can be from 0 to 99.</td>
</tr>
<tr>
<td><strong>Columns and Include Columns</strong></td>
<td>Index columns can be segregated into unique key columns (Columns pane) and Include columns that are to be part of the index but do not form part of the unique key. The steps in completing the panes for the two column types are identical. From the Column dropdown, select a column for the index and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
</tr>
<tr>
<td><strong>Comment</strong></td>
<td>Type a comment, associated with this index in the Comment area.</td>
</tr>
<tr>
<td><strong>DDL View</strong></td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

**Materialized Query Table Wizard for IBM DB2 for Linux, Unix, and Windows**

A materialized query table (MQT) is a table based on the result of a query. An MQT contains information that is summarized from other tables and can save time when processing dynamic SQL queries. The Materialized Query Table Wizard lets you create a table without knowing the underlying commands.

**To Open the Materialized Query Wizard**

1. On the Explorer, find the datasource where you want to add the new materialized query table.

2. Expand the Schema branch, right-click Materialized Query Tables, and select New.

The table that follows describes the fields you may encounter as you complete the Materialized Query Table Wizard.

**NOTE:** These options are only available if the tablespace you selected is a database managed tablespace.
<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who owns the table?</td>
<td>Choose the owner of the table you’re creating from the drop-down list.</td>
</tr>
<tr>
<td>What is the name of the table?</td>
<td>Type the name of the materialized query table you are creating.</td>
</tr>
<tr>
<td>Select a tablespace on which to place the table:</td>
<td><strong>OPTIONAL:</strong> No selection is the default. But you can select a tablespace that belongs to the new table’s owner if you want.</td>
</tr>
<tr>
<td>Specify separate tablespaces for index and long data</td>
<td><strong>OPTIONAL:</strong> Lets you separate indexes or long data from the table.</td>
</tr>
<tr>
<td>Specify the query on which this table is based</td>
<td>Write the query you want to use to create the table.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Every select list element must have a name (use the AS clause for expressions)</td>
</tr>
<tr>
<td>Add the columns belonging to this table</td>
<td>Click Add, Insert or Edit to add or modify table columns. Click Drop to delete a selected column.</td>
</tr>
<tr>
<td>Add the columns belonging to the partition key</td>
<td></td>
</tr>
<tr>
<td>Do you want the table replicated across database partitions?</td>
<td>The default is no, but check the box if you do want to replicate the table. Replicated materialized query tables can help you improve query performance by enabling collocation between tables. Replication is especially helpful when you have joins between large fact tables and small dimension tables. It’s best if tables that are to be replicated are infrequently updated.</td>
</tr>
<tr>
<td>Definition Only</td>
<td>Lets you select definition options, <strong>Include Column Defaults</strong> and <strong>Include Identity Column Attributes</strong>.</td>
</tr>
<tr>
<td></td>
<td>When you select the Definition Only option, the new table is treated as if it was a base table rather than a MQT.</td>
</tr>
<tr>
<td></td>
<td>After you complete the wizard, DBArtisan opens the Tables Editor.</td>
</tr>
<tr>
<td>Refreshable</td>
<td>Lets you select refresh options: <strong>Immediate:</strong> The table you are creating will be refreshed automatically when changes are made to the base table(s).</td>
</tr>
<tr>
<td></td>
<td><strong>Deferred:</strong> Static SQL will not be optimized. Changes to the base table(s) will not be reflected.</td>
</tr>
<tr>
<td></td>
<td>Query Optimization:</td>
</tr>
<tr>
<td></td>
<td><strong>Enabled:</strong> Queries will be routed to the MQT.</td>
</tr>
<tr>
<td></td>
<td><strong>Disabled:</strong> This is the default.</td>
</tr>
<tr>
<td></td>
<td>Maintained by:</td>
</tr>
<tr>
<td></td>
<td><strong>System:</strong> This is the default</td>
</tr>
<tr>
<td></td>
<td><strong>User</strong></td>
</tr>
<tr>
<td></td>
<td>After you complete the wizard, DBArtisan opens the Material Query Tables Editor.</td>
</tr>
<tr>
<td>Would you like extra information regarding SQL changes to this table to be written to the log?</td>
<td>If you opted to replicate the table, you must make a selection here. <strong>Yes</strong> means you want to receive error messages issued by the DB2 replication programs. <strong>Include Longvar Columns</strong> means you want to receive error messages for these columns if long varchar data is a part of the table you’re replicating. <strong>No</strong></td>
</tr>
</tbody>
</table>
Node Group Wizard for IBM DB2 for Linux, Unix, and Windows

A nodegroup is a named subset of one or more database partitions. The Node Group Wizard lets you create a node group without knowing the underlying commands. When you create a nodegroup, the wizard simply asks you to name the nodegroup and select the partitions to include in the nodegroup.

To Open the Nodegroup Wizard

1. On the Explorer, find the datasource where you want to add the new Nodegroup.
2. Expand the Storage branch, right-click Nodegroups, and select New.

Primary Key Wizard for IBM DB2 for Linux, Unix, and Windows

A primary key is a column or group of columns that you can use to identify or access one or more specific rows in a table. A primary key is ‘constrained’ in that no values can be null and no two values are equal. You can only create one primary key for any table. The Create Primary Key Constraint dialog box lets you create a primary key without knowing the underlying commands.

When you create a primary key, specify the table owner and the table on which you want to place the primary key constraint. You name the constraint and select the column(s) you want to include.
To create a new primary key using a wizard:

1. Open an object wizard for a primary key.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Table Schema and Table Name</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td>Columns</td>
<td>From the Column dropdown, select a column for the primary key and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
</tr>
<tr>
<td>Comment</td>
<td>Type a comment, associated with this index in the Comment area.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

Procedure Wizard for IBM DB2 for Linux, Unix, and Windows

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

To create a new procedure using a wizard:

1. Open an object wizard for a procedure.

2. Use the following table as a guide to understanding the settings in this wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Schema</td>
<td>Select the owner for the procedure.</td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>Provide the name of the function.</td>
</tr>
<tr>
<td></td>
<td>Specific Name</td>
<td>Optionally, provide the unique name of the procedure.</td>
</tr>
<tr>
<td></td>
<td>Language</td>
<td>Select among C, JAVA, COBOL, OLE, or SQL. The database manager will call the procedure accordingly.</td>
</tr>
<tr>
<td></td>
<td>External Name</td>
<td>Provide the external name of the procedure.</td>
</tr>
</tbody>
</table>
| SQL Access Level | Select an option:  
MODIFIES SQL DATA - the procedure can support any SQL statement except those that cannot be supported in procedures.  
CONTAINS SQL DATA - only SQL statements that neither modify nor read SQL data can be executed in the procedure.  
READS SQL DATA - some SQL statements that don’t modify SQL data can be included in the procedure. |
Sequence Wizard for IBM DB2 for Linux, Unix, and Windows

A sequence allows the automatic generation of values, something well-suited to the generation of unique key values. Sequences are not tied to particular table columns. The Sequence Wizard lets you create a sequence without knowing the underlying commands. As you complete the Sequence Wizard, DBArtisan constructs the necessary CREATE SEQUENCE statement from the information that you have supplied. When finished, you can instruct DBArtisan to compile the sequence on the target DBArtisan database or to write a script file containing the CREATE SEQUENCE statement.

The Sequence Wizard lets you:

- Specify the name and owner of the sequence.

<table>
<thead>
<tr>
<th>Step</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced</td>
<td>Results Sets</td>
<td>Indicate the estimated upper bound of returned result sets. 0 is the default.</td>
</tr>
<tr>
<td></td>
<td>External Action</td>
<td>Select the External Action option.</td>
</tr>
<tr>
<td></td>
<td>New Save Point</td>
<td>Specify whether the procedure is to be created for the procedure.</td>
</tr>
<tr>
<td></td>
<td>Threadsafe</td>
<td>Specify whether the procedure is safe to run within the same process as other routines.</td>
</tr>
<tr>
<td></td>
<td>Fenced</td>
<td>If you select yes, you are saying you do not want the procedure to run in the manager operating system environment. This means the database management system will protect its internal resources from the procedure. This option can affect the procedure’s operation. To run a procedure as not fenced, or a No selection, you must have SYSADMIN or DBADMIN privileges because of the potential to compromise data if the procedure has not been adequately tested.</td>
</tr>
<tr>
<td></td>
<td>Parameter Style</td>
<td>Lets you select an option: DB2DARI, DB2GENERAL, DB2SQL, GENERAL, GENERAL WITH NULLS, JAVA, and SQL. DB2GENERAL is for Java Language only. DB2SQL is for C, COBOL, or OLE Language only. GENERAL is for C Language only. GENERAL WITH NULLS is for C or COBOL Language only. JAVA is for Java Language only. SQL is for C, COBOL, or OLE Language only.</td>
</tr>
<tr>
<td></td>
<td>Program Type</td>
<td><strong>MAIN</strong>: valid for C or COBOL Language and Parameter Style GENERAL, GENERAL WITH NULLS, SQL, or DB2SQL only. In this case, parameters will be passed as an argument counter or argument vector. <strong>SUBROUTINE</strong>: the procedure expects the parameters to be passed as separate arguments.</td>
</tr>
<tr>
<td></td>
<td>DBINFO</td>
<td>Specific information contains such information as the database name, application ID, database code page, and so on.</td>
</tr>
<tr>
<td></td>
<td>Deterministic</td>
<td>Enabling this feature specifies the procedure will always return the same result for given argument values. Disabling it means there are state values that affect the results and so the same result will not always be returned when you use identical inputs to invoke the procedure.</td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
<td>For each parameter for this function, use the New button to add a new parameter, provide a name for the parameter, and in the Attributes area, select a Type, specify a Parameter Mode of INPUT, OUTPUT, or INPUT_OUTPUT, and if appropriate, the Precision, Scale, and Size options.</td>
</tr>
<tr>
<td>DDL View</td>
<td></td>
<td>Preview then execute to create the new alias.</td>
</tr>
</tbody>
</table>
• Set both the value of the sequence, and an interval and ranges for incrementing it.

• Cache the sequence, cycle the sequence when it reaches its minimum or maximum values, and guarantee that DBArtisan generates sequence numbers in the order of request.

**NOTE:** To create a sequence, it must belong to your schema or you need CREATE SEQUENCE privileges.

**To Open the Sequence Wizard**

1. On the **Explorer**, find the datasource where you want to add the new **Sequence**.
2. Expand the **Schema** branch, right-click **Sequences**, and select **New**.

The table that follows describes what you may encounter as you complete the Sequence Wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who owns the sequence?</td>
<td>You decide.</td>
</tr>
<tr>
<td>What is the sequence name?</td>
<td>Your choice.</td>
</tr>
<tr>
<td>What numeric datatype should the Sequence use?</td>
<td>Choose among BIGINT (big integer), decimal (choose width as well), integer, small integer.</td>
</tr>
<tr>
<td>What is the first sequence number to be generated?</td>
<td>Starting with 1 is the default.</td>
</tr>
<tr>
<td>What is the interval between sequence numbers?</td>
<td>Increment by 1 is the default.</td>
</tr>
<tr>
<td>What is the sequence’s minimum value?</td>
<td>Choose none or set a value</td>
</tr>
<tr>
<td>What is the sequence’s maximum value?</td>
<td>Choose none or set a value</td>
</tr>
<tr>
<td>Should DB2 preallocate sequence numbers and cache them for faster access?</td>
<td>Preallocating and storing values in the cache reduces synchronous I/O to the log when values are generated for the sequence. If Yes, give number of values No</td>
</tr>
<tr>
<td>Should the sequence continue to generate values after reaching either its maximum or minimum value?</td>
<td>Lets you make the sequence cycle and continue to generate numbers. Yes or No</td>
</tr>
<tr>
<td>Should the sequence numbers be generated in the order of request?</td>
<td>Select to generate sequence numbers in the order of request. The ORDER option is useful when you are using the sequence number as a timestamp. Yes or No</td>
</tr>
</tbody>
</table>

**Schema Wizard for IBM DB2 for Linux, Unix, and Windows**
The Schema Wizard lets you create the structure of a database system including database objects.

**To Open the Schema Wizard**

1. On the **Explorer**, find the datasource where you want to add the new **Schema**.
2. Right-click **Schema** and select **New**.

All you need to do when the single-panel wizard opens is to give a unique name to the schema you’re creating.
Table Wizard for IBM DB2 for Linux, Unix, and Windows

The Table Wizard lets you create a table without knowing the underlying commands.

The Table Wizard lets you:

- Name the table and owner
- Lets you set logging options for the table and to provide table comments.

To create a new table using a wizard:

1. Open an object wizard for a table.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Schema</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td>Percent Free</td>
<td>Lock Size</td>
</tr>
<tr>
<td>Append Data</td>
<td>Enable or Disable appending data to the end of the table.</td>
</tr>
<tr>
<td>Volatile</td>
<td>Enable this feature if a table contents may fluctuate from empty to very large. The access plan will not depend on the existing statistics for that table.</td>
</tr>
<tr>
<td>Compress</td>
<td>Enable or disable value compression</td>
</tr>
<tr>
<td>Row Compression</td>
<td>Enable or disable row compression.</td>
</tr>
<tr>
<td>Security Policy</td>
<td>Lets you add a security policy to a table.</td>
</tr>
<tr>
<td>RestrictDrop</td>
<td>When this feature is enabled, you can only drop the table and associated tablespace or database using the</td>
</tr>
<tr>
<td>Log Index Build</td>
<td>Enables this level of logging when creating, recreating, or reorganizing an index.</td>
</tr>
<tr>
<td>CCSID</td>
<td>Specify ASCII or UNICODE or leave unspecified. If specified, this is the encoding scheme for string data. If unspecified, CCSID default encoding is used.</td>
</tr>
<tr>
<td>Tablespace</td>
<td>Select a tablespace.</td>
</tr>
<tr>
<td>Index Tablespace</td>
<td>Select an index tablespace.</td>
</tr>
<tr>
<td>Long Data Tablespace</td>
<td>Select a tablespace for Long or LOB table columns.</td>
</tr>
<tr>
<td>Do not initially log</td>
<td>If enabled, all changes to the table will be flushed out at commit time. This also means that if a statement fails, the unit of work will rollback. If you are concerned about recoverability, disable this feature.</td>
</tr>
<tr>
<td>Data Capture</td>
<td>Specify additional information logged by selecting DATACAPTURE NONE, DATACAPTURE CHANGES, or DATACAPTURE CHANGES INCLUDE LONGVAR.</td>
</tr>
<tr>
<td>Columns</td>
<td>For each column in the table, click the New button to create a column and provide a name for the column. Then, in the Column Attributes area, provide details for the column. Use the Delete button to drop a selected column.</td>
</tr>
</tbody>
</table>
### Tablespace Wizard for IBM DB2 for Linux, Unix, and Windows

Tablespaces establish connections between the physical storage devices of your database system and the logical containers or tables being used to store data. In essence, a tablespace is a storage structure that can hold tables, indexes, large objects, and long data. The Tablespace Wizard lets you create a tablespace without knowing the underlying commands.

The Tablespace Wizard lets you:

- Specify data type to be stored on the tablespace.
- Specify what type of tablespace to create.
- Add containers to the tablespace.
- Specify how to manage the growth of the tablespace and select a bufferpool on which the tablespace should reside.

#### To Open the Tablespace Wizard

1. On the **Explorer**, find the datasource where you want to add the new **Tablespace**.
2. Expand the **Schema** branch, right-click **Tablespaces**, and select **New**.

The table that follows describes what you may encounter as you complete the Tablespace Wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>Under <strong>Partition Columns</strong>, for each partition column, click the New button and then choose a column name from the dropdown. To add a <strong>Data Partition</strong>, click the New button to open a dialog that lets you add a partition.</td>
</tr>
<tr>
<td>Tablespaces</td>
<td>For each Data Tablespace or Long Tablespace, click the New button and then choose a tablespace from the dropdown.</td>
</tr>
<tr>
<td>Dimensions</td>
<td>For each column that is to make up a dimension, click the New button to open a dialog that lets you add a column.</td>
</tr>
<tr>
<td>Distribution Key Columns</td>
<td>For each column that is to make up the distribution key, click the New button and then select a column from the dropdown</td>
</tr>
<tr>
<td>Indexes</td>
<td>Click <strong>Add</strong> to open the Index wizard. For more information, see <strong>Index Wizard for IBM DB2 for Linux, Unix, and Windows</strong>.</td>
</tr>
<tr>
<td>Constraints</td>
<td>Selecting a constraint type and clicking <strong>Add</strong> opens the object wizard for that object type. For details see: <strong>Primary Key Wizard for IBM DB2 for Linux, Unix, and Windows</strong>, <strong>Unique Key Wizard for IBM DB2 for OS/390 and z/OS</strong>, <strong>Foreign Key Wizard for IBM DB2 for OS/390 and z/OS</strong>.</td>
</tr>
<tr>
<td>Comment</td>
<td>Optionally link a comment to the alias you are creating.</td>
</tr>
<tr>
<td>Permissions</td>
<td>For each specific permission to be granted to a login or group, select the cell corresponding to the login/group and specific permission, and click the <strong>Grant</strong> button. To revoke a privilege, select a cell showing a Granted permission and click <strong>Revoke</strong>.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>
### Required Information

<table>
<thead>
<tr>
<th>Description</th>
<th>Required Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What is the name of the tablespace?</strong></td>
<td>You decide</td>
</tr>
<tr>
<td><strong>Choose the type of data to store in the tablespace</strong></td>
<td>Regular/Indexes only</td>
</tr>
<tr>
<td></td>
<td>Long: will the table have columns with LONG data values?</td>
</tr>
<tr>
<td></td>
<td>Temporary: Temporary tables</td>
</tr>
<tr>
<td></td>
<td>Use Temporary: A preexisting temporary tablespace</td>
</tr>
<tr>
<td><strong>Specify whether this tablespace will be managed by the operating system or the database</strong></td>
<td><strong>Operating system managed:</strong> Uses your operating system to manage physical storage and growth of the database. This type of tablespace can be used for small databases that do not experience heavy access. You cannot add containers to a system managed tablespace after it has been created.</td>
</tr>
<tr>
<td></td>
<td><strong>Database managed:</strong> Relies on IBM DB2 to manage physical storage and the growth of the database. This type of tablespace lets you have more control over storage options such as physical location, size, type of storage, fetch rate and placement of indexes and tables. This type of tablespace should be used for large databases where performance is an issue. Additionally, a database managed tablespace can be backed up, restored or tuned separately since you have control over individual space.</td>
</tr>
<tr>
<td><strong>Select a nodegroup to assign to the tablespace</strong></td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td><strong>Add Container for Tablespace</strong></td>
<td>The Add a Container to a tablespace opens.</td>
</tr>
<tr>
<td><strong>Specify Page Size (Kbytes)</strong></td>
<td>The default page size is 4KB. Note that the page size for the bufferpool must match the page size you designate for the tablespace.</td>
</tr>
<tr>
<td><strong>Specify the extent and prefetch sizes</strong></td>
<td>Extent size - Number of pages of table data that will be written to a container before data will be written to the next container.</td>
</tr>
<tr>
<td></td>
<td>Prefetch size - Number of pages the prefetcher gets from the disk at a time. This value must be a multiple of the extent size.</td>
</tr>
<tr>
<td><strong>Specify the overhead and transfer fields</strong></td>
<td>The overhead and transfer rates help determine the best access path to use when the SQL statement is compiled.</td>
</tr>
<tr>
<td></td>
<td><strong>Overhead</strong> - The amount of time in milliseconds that a container needs before reading any data into memory.</td>
</tr>
<tr>
<td></td>
<td><strong>Transfer rate</strong> - The amount of time, in milliseconds, that it takes to read one page of data into memory.</td>
</tr>
<tr>
<td></td>
<td>Both of these values are important to keep in mind when considering the I/O rate.</td>
</tr>
<tr>
<td><strong>Select the buffer pool for this tablespace</strong></td>
<td>Each tablespace is linked to a specific and existing bufferpool. IBMDEFAULTBP is the default.</td>
</tr>
<tr>
<td></td>
<td>In general, having more than one bufferpool allows you to better configure the memory used by the database.</td>
</tr>
<tr>
<td><strong>Drop Table recovery status</strong></td>
<td>ON/OFF</td>
</tr>
</tbody>
</table>

### Trigger Wizard for IBM DB2 for Linux, Unix, and Windows

A trigger defines a set of actions that take place in conjunction with, or are triggered, for a particular base table, with an insert, update, or delete statement. Triggers are dandy ways to validate input data, read from other tables for cross-referencing purposes, and other similar purposes. The Trigger Wizard lets you create a trigger without requiring you to know any of the underlying commands.
The Trigger Wizard lets you:

- Specify the trigger name and the table on which it fires.
- Specify trigger timing, event action, and applicable table columns for an Update trigger.
- Specify the granularity of the trigger and correlation names for the new rows.

To create a new trigger using a wizard:

1. Open an object wizard for a trigger.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Choose the owner and name of the table for which the trigger is being created.</td>
</tr>
<tr>
<td>Schema and Name</td>
<td>Choose the owner and name of the trigger being created.</td>
</tr>
<tr>
<td>Trigger Timing</td>
<td>BEFORE: These triggers serve as extensions to the constraint subsystem and are most often used to validate input data, generate values for newly inserted rows, and read from other tables for cross-reference purposes. Note: Before triggers must be created as a For Each Row. AFTER: Such a trigger is run after the integrity constraint validations; they can be used to modify operations in the database or be used for activities beyond the database, like supporting an alert notification.</td>
</tr>
<tr>
<td>Trigger Events</td>
<td>An INSERT trigger must be associated with an INSERT statement. For example, if a data load operation doesn't include an INSERT statement, the trigger won't be invoked. An UPDATE trigger can be associated with specific columns of the base table and will only be activated if those columns are updated.</td>
</tr>
<tr>
<td>Trigger Type</td>
<td>STATEMENT: (only fires once). ROW (fires for each affected row): The trigger runs as many times as there are rows in the affected section. If the set of affected rows is empty, the trigger doesn't run.</td>
</tr>
<tr>
<td>Old Table Alias</td>
<td>Type the name of a temporary table of rows as they exist before they're updated or deleted.</td>
</tr>
<tr>
<td>New Table Alias</td>
<td>Type a name for a temporary table of rows as they exist after they're inserted or updated.</td>
</tr>
<tr>
<td>Old Row Alias</td>
<td>Type a name for the rows as they are before they've been deleted or updated.</td>
</tr>
<tr>
<td>New Row Alias</td>
<td>Type a name for the rows as they are after they've been inserted or updated.</td>
</tr>
<tr>
<td>Column Selection</td>
<td>If you chose UPDATE as the Trigger Event, select the columns, select the check box beside each column that is to fire the trigger.</td>
</tr>
<tr>
<td>Definition</td>
<td>Complete the CREATE TRIGGER outline provided by typing or pasting the body of the trigger. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>
Type Wizard for IBM DB2 for Linux, Unix, and Windows

Structured types come in handy when an object has a well-defined structure made up of attributes. For example, a person’s name and address describe an instance of a person in a database. Or, a shape may be made up of value coordinates on a Cartesian plane. Structured types can be stored as rows in a table or object view, or as values in a column. Creating an object type is a two-step process. You create the object type specification, and then create the type body. The Structured Type Wizard lets you create an object type specification and body. A template is created using the CREATE OR REPLACE TYPE or TYPE BODY statements.

To Open the Type Wizard
1. On the Explorer, find the datasource where you want to add the new Structured Type.
2. Expand the Schema branch, right-click Structured Type, and select New.

The table that follows describes what you may encounter as you complete the Type Wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who owns the type to be created?</td>
<td>Choose your favorite from the drop-down list.</td>
</tr>
<tr>
<td>What is the name of the type?</td>
<td>Type a unique name.</td>
</tr>
<tr>
<td>Is this structured type a subtype?</td>
<td>If you choose this option, the two options below become active.</td>
</tr>
<tr>
<td>Supertype Schema</td>
<td></td>
</tr>
<tr>
<td>Supertype Name</td>
<td></td>
</tr>
<tr>
<td>Add Attribute:</td>
<td>Attribute Name: This must be unique.</td>
</tr>
<tr>
<td>Datatype: An existing datatype must be used, but a datatype cannot be defined so that one of its attribute types directly or indirectly uses itself.</td>
<td></td>
</tr>
<tr>
<td>Width/Scale</td>
<td></td>
</tr>
<tr>
<td>LOB options: Logged specifies that changes made to the data are to be written to the log. Compact specifies that LOB values should take up minimal disk space.</td>
<td></td>
</tr>
<tr>
<td>File Link Options</td>
<td></td>
</tr>
<tr>
<td>For Bit data: Specifies that the data should be treated as binary data.</td>
<td></td>
</tr>
<tr>
<td>Instantiate</td>
<td>Instantiable/Not instantiable</td>
</tr>
<tr>
<td>If you choose not instantiable, no constructor function can be generated, nor can a non-instantiable type be used as the type of a table or view.</td>
<td></td>
</tr>
<tr>
<td>Inline length</td>
<td>The value you set here indicates the maximum size of an instance of a structured type in a column. If the size of a structured type instance is less than the defined maximum, the data is stored inline with the other values in the row. If the size of the structured type is larger than the defined maximum, the structured type data is stored outside of the table (like LOBs).</td>
</tr>
<tr>
<td>Cast function (Source as Ref)/Cast function (Ref as Source)</td>
<td>Here you have the opportunity to name the cast function, although one will be created with the default name of the structured type you are creating. The cast function “casts” a value between the reference type and the representation type in both directions.</td>
</tr>
<tr>
<td>Representation Type</td>
<td>Defines the built-in data type used as the underlying data type for the structured type you are creating and all its subtypes. So, for example, the reference type for a PERSON type, might be VARCHAR.</td>
</tr>
<tr>
<td>Width/Scale</td>
<td></td>
</tr>
<tr>
<td>With Function Access</td>
<td>If selected, all methods of the type you are creating, and those you will create, can be accessed using functional notation. Some applications prefer functional notation over method invocation.</td>
</tr>
</tbody>
</table>
Unique Key Wizard for IBM DB2 for Windows, Unix, and Linux

A unique key constraint is a key for which no two of its values can be equal and no values can be null. A table can have a number of unique constraints, but it cannot have more than one unique constraint on the same set of columns. If you are creating a unique key constraint on a table that already exists (as opposed to creating a unique key at the time the table is first generated), a unique index must already exist on the columns of the unique key you want to constrain. If no unique index exists, the Index Wizard will open as you complete the Create Unique Key Constraint dialog box.

To create a new unique key using a wizard:

1. Open an object wizard for a unique key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Comparisons</td>
<td>This indicates that there are no comparison functions supported for instances of the structured type.</td>
</tr>
<tr>
<td>Not final</td>
<td>Not final indicates that the structured type can be used as a supertype.</td>
</tr>
</tbody>
</table>

User Datatype Wizard for IBM DB2 for Linux, Unix, and Windows

User defined datatypes allow column definitions to be consistent across a database. They let you associate frequently used datatype information to be associated with a specified function and take advantage of performance features available to built-in datatypes including indexing and parallel database queries.
To create a new user datatype using a wizard:

1. Open an object wizard for a user datatype.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Select the owner of the user datatype.</td>
</tr>
<tr>
<td>Datatype</td>
<td>Provide a name for the datatype.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the base datatype.</td>
</tr>
<tr>
<td>Size</td>
<td>Provide the size of the datatype.</td>
</tr>
<tr>
<td>Allow Bit Data</td>
<td>The option is only available for certain datatypes. A check means you want to store the data in a bit format.</td>
</tr>
<tr>
<td>Comment</td>
<td>Type a comment, associated with this unique key in the Comment area.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute buttons to Create the new object.</td>
</tr>
</tbody>
</table>

User Wizard for IBM DB2 for Windows, Unix, and Linux

Users have authorization to use a database and its objects, and the User Wizard gives you an easy way to add new ones.

To create a new user using a wizard:

1. Open an object wizard for a user

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Provide the user name.</td>
</tr>
<tr>
<td>System Permissions</td>
<td>Set up the system permissions for this user</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

View Wizard for IBM DB2 for Linux, Unix, and Windows

A view gives you an alternative way to look at data in one or more tables. You can customize how you see a table’s contents.
To create a new view using a wizard:

1. Open an object wizard for a view.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>Select the owner of the view. The owner of the view must have SELECT privileges for the tables in the CREATE view statement or DBADM authority on the database that contains the table.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the view.</td>
</tr>
<tr>
<td>Check Type</td>
<td></td>
</tr>
<tr>
<td>CHECK_NONE</td>
<td>- No search conditions must be satisfied for insert or update operations.</td>
</tr>
<tr>
<td>CHECK_LOCAL</td>
<td>- Update and insert operations on view must satisfy the search conditions of the view and underlying views that are defined with a check option. Furthermore, every updatable view that is directly or indirectly defined on view inherits those search conditions (the search conditions of view and all underlying views of that are defined with a check option) as a constraint on insert or update operations.</td>
</tr>
<tr>
<td>CHECK_CASCADED</td>
<td>- Update and insert operations on the view must satisfy the search conditions of view and all underlying views, regardless of whether the underlying views were defined with a check option. Furthermore, every updatable view that is directly or indirectly defined on view inherits those search conditions (the search conditions of view and all underlying views) as a constraint on insert or update operations.</td>
</tr>
<tr>
<td>Definition</td>
<td></td>
</tr>
<tr>
<td>Create the new object</td>
<td></td>
</tr>
</tbody>
</table>

IBM DB2 for OS/390 and z/OS Object Wizards

Alias Wizard       Bufferpool Wizard       Create Check Constraint Dialog Box       Database Wizard
Foreign Key Wizard Function Wizard       Index Wizard       Create Primary Key Constraint Dialog Box
Procedure Wizard   Stogroup Wizard       Synonym Wizard       Table Wizard
Tablespace Wizard   Trigger Wizard       Create Unique Key Constraint Dialog Box User Datatype Wizard
View Wizard         DBArtisan Server Components Installation Wizard (Bind) Plan Wizard (Bind) Packages Wizard

Alias Wizard for IBM DB2 for OS/3910 and z/OS
An alias offers you security and convenience so that you can refer to an object without revealing who owns it or what database it belongs to. You can create aliases for tables or views. The Alias Wizard lets you create an alias without knowing the underlying commands. As you complete the Alias Wizard process, DBArtisan constructs the necessary CREATE ALIAS statement based on the information that you supply. To create an alias, you must have CREATE ALIAS privileges or sysadmin or sysctrl authority.
To create a new alias using a wizard:

1. Open an object wizard for an alias.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>Select the schema that is to own the alias.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the alias.</td>
</tr>
<tr>
<td>Target Owner</td>
<td>Select the owner of the object to which you are creating an alias.</td>
</tr>
<tr>
<td>Target Type</td>
<td>Select the type of object (TABLE, VIEW) to which you are creating an alias.</td>
</tr>
<tr>
<td>Target Name</td>
<td>Select the specific object to which you are creating an alias.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>

**Database Wizard for IBM DB2 for OS/390 and z/OS**

The Database Wizard lets you create a database (a structured collection of data that can be updated or queried) without knowing the underlying commands. Databases can be simple, that is one file with many records and the same fields, or much more complicated with multiple files with different fields.

To create a new database using a wizard:

1. Open an object wizard for a database.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Provide a unique name for the database.</td>
</tr>
</tbody>
</table>
| Type      | **Workfile**: This option is only available if the server is configured in IBM DB2 for OS/390 and z/OS to allow sharing. For more information, contact your System administrator.  
**Temp**: This option indicates the database is only for declared temporary tables. A temp database cannot be shared. |
| Group Member | Specifies the member for which this database is being created. Use this only in a shared environment. |
| Tablespace Buffer Pool | Select the default buffer pool to be used for tablespaces created within the database. |
| Index Buffer Pool  | Select the default buffer pool name to be used for indexes created within the database. |
| Storage Group | Select the default storage group to support the DASD space requirements for tablespaces and indexes within the database. |
| Encoding Scheme | Select an encoding schema of DEFAULT, ASCII, EBCDIC, or UNICODE.  
**NOTE**: To change the encoding scheme for a database after it has been created to use a different coded character set identifier (CCSID) that supports the Euro symbol, all data must be unloaded and reloaded. For more information regarding the encoding scheme, contact your System administrator. |
Foreign Key Wizard for IBM DB2 for OS/390 and z/OS

Foreign keys are unique values that refer to specific columns of other tables. Thus, a foreign key links two tables together. Embarcadero DBArtisan’s Foreign Key Wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data. The Foreign Key Wizard lets you create a foreign key without knowing the underlying commands.

The Foreign Key Wizard lets you:

- Name the foreign key constraint
- Identify the parent table and the referenced constraint.
- Map the column pairs between the parent and child tables.

To create a new foreign key using a wizard:

1. Open an object wizard for a foreign key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permissions</td>
<td>Set up the user permissions for this database.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

Function Wizard for IBM DB2 for OS/390 and z/OS

To create a relationship between one set of values and another, DBArtisan offers the Function Wizard. You can develop reusable subroutines so you can control, access, and manipulate the data that underlies an object. As you complete the Function Wizard process, DBArtisan constructs the necessary CREATE FUNCTION statement based on the information that you supply.

NOTE: To create a user-defined function, you need CREATE ANY privileges or IMPLICIT_SCHEMA authority on the database if the schema does not already exist.
To create a new foreign key using a wizard:

1. Open an object wizard for a foreign key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td><strong>Schema</strong> Select the owner of the function.</td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong> Provide a name for the function.</td>
</tr>
<tr>
<td></td>
<td><strong>Specific Name</strong> Provide the Specific name to be used by some SQL statements and DB2 commands for this function.</td>
</tr>
<tr>
<td>Function Type</td>
<td><strong>External Scalar</strong>: This allows you to extend the function by adding your own or another party’s definition for the function. <strong>External Table</strong>: Use this to create a function that is written in ASSEMBLE, C, COBOL, or PLI to return a table after it is deployed. <strong>Sourced</strong>: Here you are creating a function that is based on an existing scalar or table function with an application server. <strong>SQL</strong>: This type of function returns a single value when the function is invoked if the SQL statement that defines it is valid.</td>
</tr>
<tr>
<td>Language</td>
<td>If you chose a Function Type of EXTERNAL SCALAR or EXTERNAL TABLE, specify a language of ASSEMBLE, C, COBOL, or PLI.</td>
</tr>
<tr>
<td>Return Type</td>
<td>Identifies the return type of the function.</td>
</tr>
<tr>
<td></td>
<td><strong>External Name</strong> Provide the External Name of the function.</td>
</tr>
<tr>
<td>SQL Access Level</td>
<td>Indicates whether the function can execute SQL statements.</td>
</tr>
<tr>
<td></td>
<td>CONTAINS SQL: Statements that don’t read or modify SQL can be executed.</td>
</tr>
<tr>
<td></td>
<td>NO SQL: No SQL statements can be executed.</td>
</tr>
<tr>
<td></td>
<td>READS SQL: Statements that cannot modify SQL can be executed.</td>
</tr>
<tr>
<td>WLM Environment</td>
<td>Specify a Workload Management Environment (Required if Language is JAVA/COMPJAVA/REXX, the Procedure contains a LOB parameter, Security is set to ‘USER’ or ‘DEFINER’, or program type is ‘SUB’).</td>
</tr>
<tr>
<td></td>
<td><strong>WLM For Nested</strong> Self-explanatory</td>
</tr>
<tr>
<td>Source</td>
<td>Only available when you choose a Function Type of SOURCED. Select the Schema, Name, and Specific Name of the source function.</td>
</tr>
<tr>
<td>Parameters</td>
<td>For each parameter for this function, use the New button to add a new parameter, provide a name for the parameter, and in the Attributes area, select a Type, and if appropriate, the Precision, Scale, Size, and As Locator options.</td>
</tr>
<tr>
<td>Return Scalar</td>
<td>Under Return Datatype, select a Type and depending on your choice, provide or select Precision, Scale, Size, and As Locator options. To make use of a CAST FROM clause, under Cast Datatype set Enabled to True, select a Type, and if appropriate, the Scale, Size, and As Locator options</td>
</tr>
<tr>
<td>Return Columns</td>
<td>Only available when you choose a Function Type of EXTERNAL TABLE. For each column returned by this function, use the New button to add a new parameter, provide a name for the parameter, and in the Attributes area, select a Type, and if appropriate, the Precision, Scale, Size, and As Locator options</td>
</tr>
<tr>
<td>Body</td>
<td>Enter the return statement for the function.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>
Index Wizard for IBM DB2 for OS/390 and z/OS

Like the index in a book, a database index makes it easier for you to access information quickly. An index lists the location of rows, sorted by the contents of one or more columns.

The Index Wizard lets you:

- Specify the table owner and name.
- Specify the index owner and name.
- Create a unique index and provide a comment for the index.
- Identify the index location and order.
- Select a bufferpool and determine if the tablespace is managed by the user or DB2.
- Specify free-block and gbpcache-block index parameters.
- Specify other index parameters.

To create a new index using a wizard:

1. Open an object wizard for an index.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Table Schema</td>
<td>Choose the owner and name of the table in which the index is being created.</td>
</tr>
<tr>
<td>and Table Name</td>
<td></td>
</tr>
<tr>
<td>Schema and Name</td>
<td>Choose the owner and name of the index being created.</td>
</tr>
<tr>
<td>Index Type</td>
<td></td>
</tr>
<tr>
<td>Unique</td>
<td>Prevents the selected table from having two or more rows with the same value of the index key. The uniqueness is enforced at the end of the SQL statement update. Also, null values are treated like any other values, so a column cannot contain more than one null. If you later elect to partition the index, the columns specified for the unique key must include all columns for the partitioning key. Non-Unique (default)</td>
</tr>
<tr>
<td>Clustered</td>
<td>Enable or disable clustering.</td>
</tr>
<tr>
<td></td>
<td>Unless you specifically select the CLUSTER option when you create an index, the first index you create on a table will be bestowed with that distinction. Each table can only have one clustering index at a time. The clustering index tells DB2 to insert rows in the table in the order of the clustering key values. Inserted rows will be stored contiguously in sequence when possible. Columns frequently searched with a range of values BETWEEN, less than, greater than, and LIKE, or GROUP BY, etc., are good candidates for a clustering index.</td>
</tr>
<tr>
<td>Number of Partitions</td>
<td>If you enabled Clustering, specify the number of partitions.</td>
</tr>
<tr>
<td>Buffer Pool</td>
<td>Provide the buffer pool in which this index should reside</td>
</tr>
<tr>
<td>Defer, Close, Copy, and Define</td>
<td>Enable or disable these DB2 options as required.</td>
</tr>
<tr>
<td>Piece Size</td>
<td>The maximum addressability of the dataset for a non-partitioned index.</td>
</tr>
<tr>
<td>Step</td>
<td>Settings and tasks</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
</tr>
<tr>
<td><strong>Columns</strong></td>
<td>From the Column dropdown, select a column for the index and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
</tr>
</tbody>
</table>
| **Storage - Edit button (manage datasets)** | Choose a data set management option:  

**DB2 will define and manage the data sets on a volume of the default storage group of the database**  

**DB2 will define and manage the data sets on a volume of the specified storage group**  
Select a storage group (a storage group is made up of disk device volumes): Each data set will be defined on a volume listed in the storage group you select.  
Minimum primary space allocation: 12 kilobytes is the default.  
Minimum secondary space allocation: 12 kilobytes is the default.  
NOTE: If the primary and (118 x) secondary space allocations combine to be greater than 2 gigabytes, more than one data set may eventually be used.  
Erase data sets when index dropped? If choose this option, DB2 will overwrite all data with zeros before they are deleted as a security measure.  
**User will manage the data sets on a specified VCAT catalog-name** Enter or select the VCAT. Do not select this option for an index on a declared temporary table. |
| **Storage - Attributes (Availability differs according to the dataset management options you chose)** | The ability to set these options depends on the dataset management options you chose. |
| **Free Page** | One free page exists for every x pages. The x specifies how often to leave a page of free space when index entries are created from executing a DB2 utility or creating an index for a table with pre-existing rows. (0-255) |
| **Percent Free** | The percentage of free space you want to leave in every page when entries are added to an existing index. The default is 10%. |
| **GBP Cache** | This option is available only in a data-sharing environment.  
ALL: As pages are read, all of them will be cached in the group buffer pool.  
CHANGED: Updated pages are cached to the group buffer pool.  
NONE: No pages will be cached. |
| **Partitions** | Displays the default settings for the number of partitions you specified on the Properties pane. Select a partition and click the Edit button to modify details for that partition. |
| **Comment** | Type a comment, associated with this index in the Comment area. |
| **DDL View** | Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object. |
Plan Wizard for IBM DB2 for OS/390 and z/OS
A plan, also known as the application plan, is a control structure that is used to process the SQL statements DB2 encounters when it is executing those SQL statements. The Plan Wizard, really, the Bind Plan Wizard, creates the structure that is used in the bind process—the process by which the output from the SQL precompiler is converted into usable form. Some authorization checking is necessary.

To Open the Bind Plan Wizard
1. On the Explorer, find the database where you want to add the new bind plan.
2. On the Plan branch, right-click and select Bind.
   OR
   Click the Plan branch and then select Bind from the toolbar.

For complete information, see Bind Plan.

Primary Key Wizard for IBM DB2 for OS/390 and z/OS
A primary key is a unique key that is part of a table’s definition. There can be only one primary key for each table, and the columns contained in a primary key cannot have null values. A primary key constraint forbids duplicate values in one or more columns. A table with a primary key will be considered the parent table to a table with a foreign key, which becomes the dependent table.

**NOTE:** A nullable column cannot be a part of a primary key.

To create a new primary key using a wizard:
1. Open an object wizard for a primary key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Table Schema</td>
<td>Choose the owner and name of the table in which the primary key is being created.</td>
</tr>
<tr>
<td>and Table Name</td>
<td></td>
</tr>
<tr>
<td>Schema and Name</td>
<td>Choose the owner and name of the primary key being created.</td>
</tr>
<tr>
<td>Clustered</td>
<td>Enable or disable clustering.</td>
</tr>
<tr>
<td>Number of Partitions</td>
<td>If you enabled Clustering, specify the number of partitions.</td>
</tr>
<tr>
<td>Buffer Pool</td>
<td>Provide the buffer pool in which this primary key should reside</td>
</tr>
<tr>
<td>Defer, Close, Copy, and Define</td>
<td>Enable or disable these DB2 options as required.</td>
</tr>
<tr>
<td>Piece Size</td>
<td>The maximum addressability of the dataset for a non-partitioned index.</td>
</tr>
<tr>
<td>Columns</td>
<td>From the Column dropdown, select a column for the primary key and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
</tr>
</tbody>
</table>
The primary key has the same name as the unique index.

**Procedure Wizard for IBM DB2 for OS/390 and z/OS**

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

---

**Step | Settings and tasks**
---|---
**Storage - Edit button (manage datasets)** | Choose a data set management option:
- **DB2 will define and manage the data sets on a volume of the default storage group of the database**
- **DB2 will define and manage the data sets on a volume of the specified storage group**
  
Select a storage group (a storage group is made up of disk device volumes): Each data set will be defined on a volume listed in the storage group you select.

Minimum primary space allocation: 12 kilobytes is the default.

Minimum secondary space allocation: 12 kilobytes is the default.

**NOTE:** If the primary and (118 x) secondary space allocations combine to be greater than 2 gigabytes, more than one data set may eventually be used.

Erase data sets when index dropped? If choose this option, DB2 will overwrite all data with zeros before they are deleted as a security measure.

**User will manage the data sets on a specified VCAT catalog-name** Enter or select the VCAT. Do not select this option for an index on a declared temporary table.

**Storage - Attributes (Availability differs according to the dataset management options you chose)** | **Storage Group, Primary Space Allocation, Secondary Space Allocation, Erase, and VCAT catalog**
---|---

The ability to set these options depends on the dataset management options you chose.

**Free Page** | One free page exists for every x pages. The x specifies how often to leave a page of free space when index entries are created from executing a DB2 utility or creating an index for a table with pre-existing rows. (0-255)

**Percent Free** | The percentage of free space you want to leave in every page when entries are added to an existing index. The default is 10%.

**GBP Cache** | This option is available only in a data-sharing environment.

  - **ALL:** As pages are read, all of them will be cached in the group buffer pool.
  - **CHANGED:** Updated pages are cached to the group buffer pool.
  - **NONE:** No pages will be cached.

**Partitions** | Displays the default settings for the number of partitions you specified on the **Properties** pane. Select a partition and click the **Edit** button to modify details for that partition.

**Comment** | Type a comment, associated with this index in the **Comment** area.

**DDL View** | Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the **Execute** button to **Create the new object**.
To create a new procedure using a wizard:

1. Open an object wizard for a procedure.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>Select the schema that is to own the procedure.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the procedure</td>
</tr>
<tr>
<td>Specific Name</td>
<td>Provide the Specific Name for the procedure</td>
</tr>
<tr>
<td>Language</td>
<td>The database manager will call the procedure accordingly assuming the program is designed to run in the server’s environment.</td>
</tr>
<tr>
<td></td>
<td>Assemble: a stored procedure written in Assembler</td>
</tr>
<tr>
<td></td>
<td>C: a stored procedure written in C or C++</td>
</tr>
<tr>
<td></td>
<td>COBOL: a stored procedure written in COBOL</td>
</tr>
<tr>
<td></td>
<td>CompJAVA: CompJAVA is no longer supported. Stored procedures should alternatively be written in JAVA.</td>
</tr>
<tr>
<td></td>
<td>JAVA</td>
</tr>
<tr>
<td></td>
<td>PLI: A stored procedure written in PL/I.</td>
</tr>
<tr>
<td></td>
<td>REXX (Restructured Extended Executor Language) - Don’t use this language when SQL parameter style is in effect. To specify REXX, the general parameter style or general with nulls.</td>
</tr>
<tr>
<td></td>
<td>SQL</td>
</tr>
<tr>
<td>SQL Access Level</td>
<td>MODIFIES SQL DATA (Default): The procedure can support any SQL statement except those that cannot be supported in procedures.</td>
</tr>
<tr>
<td></td>
<td>CONTAINS SQL DATA: Only SQL statements that neither modify nor read SQL data can be executed in the procedure.</td>
</tr>
<tr>
<td></td>
<td>READS SQL DATA Some SQL statements that don’t modify SQL data can be included in the procedure</td>
</tr>
<tr>
<td></td>
<td>NO SQL: Only SQL statements with a data access classification of NO SQL can be executed. Don’t select this option for a JAVA procedure that uses a .jar.</td>
</tr>
<tr>
<td>WLM Environment</td>
<td>Specify a Workload Management Environment (Required if Language is JAVA/COMPJAVA/REXX, the Procedure contains a LOB parameter, or Security is set to ‘USER’ or ‘DEFINER’).</td>
</tr>
<tr>
<td>WLM For Nested</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Parameters</td>
<td>For each parameter for this function, use the New button to add a new parameter, provide a name for the parameter, and in the Attributes area, select a Type, specify a Parameter Mode of INPUT, OUTPUT, or INPUT_OUTPUT, and if appropriate, the Precision, Scale, and Size options.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

Important Notes

- If you are creating a SQL routine procedure, you must have the latest DB2 fixpack installed on your OS/390 Client. If you do not have the latest fixpack installed, the SQL routine procedure creation will fail.
# Stogroup Wizard for IBM DB2 for OS/390 and z/OS

Stogroups are storage groups. You create them on the current server. Storage from the named sets of volumes you create can be allocated at a later date for tablespaces or index spaces.

## To Open the Stogroup Wizard

1. On the Explorer, find the datasource where you want to add the new Storage Group.
2. Expand the Storage branch, right-click Stogroup, and select New.

The table below describes the fields you will encounter as you complete the Stogroup Wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the Stogroup?</td>
<td>Enter a name for the storage group.</td>
</tr>
<tr>
<td>VCAT</td>
<td>This is the integrated catalog facility catalog, or volume catalog. Name the catalog or choose one from the drop-down list if it is available.</td>
</tr>
<tr>
<td>Select the volumes in the Stogroup</td>
<td>Specify a set of volumes that may exist on the system but may not be in use by other storage groups.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all listed volumes.</td>
</tr>
<tr>
<td>Unselect All</td>
<td>Unselects all listed volumes.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Add Volume dialog box.</td>
</tr>
<tr>
<td>Remove</td>
<td>Deletes all selected volumes from the list.</td>
</tr>
</tbody>
</table>

## Add Volume Dialog Box

The table below describes the options and functionality on the Add Volume dialog box:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter one or more volumes to add to the stogroup</td>
<td>Type the names of the volumes (separated by spaces) to add to the stogroup.</td>
</tr>
<tr>
<td>Check</td>
<td>Click to see if any additional information is available about the volumes you typed. Opens the Volumes Info dialog box.</td>
</tr>
<tr>
<td>Or select volumes for the list</td>
<td>Lets you select volumes.</td>
</tr>
</tbody>
</table>

## Volumes Info Dialog Box

The table below describes the options and functionality on the Volume Info dialog box:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volumes</td>
<td>Lets you select volumes.</td>
</tr>
<tr>
<td>OK</td>
<td>Click to add volumes to Stogroup Wizard.</td>
</tr>
</tbody>
</table>
Synonym Wizard for IBM DB2 for OS/390 and z/OS

A synonym is an alternative name you can create for a table or view. Using a synonym can make it easier for you to remember that table or view instead of having to think about the possibly cumbersome formal name (for example, a show dog may have a formal name that incorporates all his ancestors, but will answer to the name Spot around the house).

To create a new synonym using a wizard:
1. Open an object wizard for a synonym.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>Select the schema that is to own the synonym.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the synonym.</td>
</tr>
<tr>
<td>Referenced Object Owner</td>
<td>Select the owner of the object to which you are creating a synonym.</td>
</tr>
<tr>
<td>Referenced Object Type</td>
<td>Select the type of object (TABLE, VIEW) to which you are creating a synonym.</td>
</tr>
<tr>
<td>Referenced Object Name</td>
<td>Select the specific object to which you are creating a synonym.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

Table Wizard for IBM DB2 for OS/390 and z/OS

All data in a database is stored in a tabular format, that is a collection of rows and columns. Tables, therefore are fundamental to whatever database you are administering.

To Open the Table Wizard
1. On the Explorer, find the datasource where you want to add the new Table.
2. Expand the Schema branch, right-click Tables, and select New.

The table that follows describes the fields you may encounter as you complete the Table Wizard.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Tasks and settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Select the Schema, provide a Name, and provide or select other table properties.</td>
</tr>
<tr>
<td>Columns</td>
<td>For each column in the table, click the Add Column button to create a column, provide a Name for the column and provide or select the remaining column attributes. Use the Delete button to drop a selected column. Use the arrow buttons to reposition the columns.</td>
</tr>
<tr>
<td>Indexes</td>
<td>Click Add to open the Index Wizard for IBM DB2 for OS/390 and z/OS.</td>
</tr>
</tbody>
</table>
Tablespace Wizard for IBM DB2 for OS/390 and z/OS

Tablespaces establish connections between the physical storage devices of your database system and the logical containers or tables being used to store data. In essence, a tablespace is a storage structure that can hold tables, indexes, large objects, and long data.

To Open the Tablespace Wizard

1. On the Explorer, find the datasource where you want to add the new Tablespace.
2. Expand the Storage branch, right-click Tables, and select New.

The table that follows describes the fields you may encounter as you complete the Tablespace Wizard.

NOTE: To change the encoding scheme for a database after it is created to utilize a different coded character set identifier (CCSID) that supports the Euro symbol, all data must be unloaded and reloaded. For more information regarding the encoding scheme, contact your System administrator.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>Selecting a constraint type and clicking Add opens the object wizard for that object type. For details see: Foreign Key Wizard for IBM DB2 for OS/390 and z/OS Primary Key Wizard for IBM DB2 for OS/390 and z/OS Unique Key Wizard for IBM DB2 for OS/390 and z/OS Create or Modify Constraint</td>
</tr>
<tr>
<td>Comment</td>
<td>Type a comment, associated with this table in the Comment area.</td>
</tr>
<tr>
<td>Permissions</td>
<td>For each specific permission to be granted, select the cell corresponding to the name and specific permission, and click the Grant button. To revoke a privilege, select a cell showing a Granted permission and click Revoke.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

Required Information

<table>
<thead>
<tr>
<th>Description</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the tablespace?</td>
<td>Self-explanatory</td>
<td>Choose Workfile or Temp from the drop-down list. Note that a workfile database cannot be specified if you want to create a LOB tablespace. Select the database from the drop-down list</td>
</tr>
<tr>
<td>Select the database that contains the tablespace</td>
<td>Non partitioned (formerly simple): Multiple tables can be combined in one tablespace, but segregating rows from different tables can make scans unwieldy. Processes move row by row. Partitioned (all types of table objects): A partitioned tablespace lets you control fragmentation of data when table size is larger that 4GB. Segmented: Pages are grouped by segment, and one segment belongs to one table only. Tablespace scans are efficient and mass deletions are possible. LOB (long text, audio, video, or images): This signifies the tablespace will be used to hold large object values.</td>
<td></td>
</tr>
</tbody>
</table>
Trigger Wizard for IBM DB2 for OS/390 and z/OS

All data in a database is stored in a tabular format, that is a collection of rows and columns. Tables, therefore are fundamental to whatever database you are administering.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 will define and manage the data sets on a volume of the default storage group of the database</td>
<td>When you specify this option, DB2 not only uses the default storage group of the database, but also uses the defaults for PRIQTY (minimum primary space allocation), SECQTY (minimum secondary space allocation), and ERASE (indicates whether the DB2-managed data sets are to be erased when they are deleted during the execution of a SQL statement that drops the tablespace).</td>
</tr>
<tr>
<td>DB2 will define and manage the data sets on a volume of the specified storage group</td>
<td>The storage group you choose, must exist on the server where you place the tablespace.</td>
</tr>
<tr>
<td>Select a storage group</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Minimum primary space allocation</td>
<td>KB: If you do not specify a value, DB2 uses the default. If you use a value other than -1, the space allocation is at least n kilobytes where n is an integer. For a 4KB page, if the integer is less than 12, n is 12. For an 8 KB page, if integer is less than 24, then n is 24....</td>
</tr>
<tr>
<td>Minimum secondary space allocation</td>
<td>KB: If you specify a minimum secondary space allocation and do not specify a value of -1, DB2 specifies the secondary space allocation to access services using the smallest multiple of p KB not less than n, where p is the page size of the table space. The allocated space can be greater than the amount of space requested by DB2</td>
</tr>
<tr>
<td>Erase data sets when tablespace dropped?</td>
<td>Not erasing the data sets is the default. If you choose to erase data sets, DB2 overwrites all the data with zeros as a security measure before enacting the deletion.</td>
</tr>
<tr>
<td>User will manage the data sets on a specified VCAT catalog name</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Enter or Select the VCAT</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Select the buffer pool the tablespace should use</td>
<td>Self-explanatory. The buffer pool must be active and the privilege set must include SYSADM or SYSCTRL authority.</td>
</tr>
<tr>
<td>Number of partitions</td>
<td>(1-254). Select the number.</td>
</tr>
<tr>
<td>Partition Size (DSSIZE)</td>
<td>G: This is a value in gigabytes that indicates the maximum size for each partition or for LOB table spaces.</td>
</tr>
<tr>
<td>Segment Size</td>
<td>pages/segment</td>
</tr>
<tr>
<td>Free space portion of each page</td>
<td>%</td>
</tr>
<tr>
<td>Logging</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Encoding Scheme</td>
<td>ASCII, Unicode, EBCDIC</td>
</tr>
<tr>
<td>Data set handling</td>
<td>Close rule: Yes/No</td>
</tr>
<tr>
<td>Lock Size</td>
<td>Self-evident.</td>
</tr>
<tr>
<td>Maximum Locks/Use System Value</td>
<td>Self-evident</td>
</tr>
<tr>
<td>Selective Partition Locking</td>
<td>Locks are obtained at the partition level and individual partitions are locked as they are accessed.</td>
</tr>
<tr>
<td>Free page frequency</td>
<td>(0-256): This identifies what percentage of each page in a tablespace is left free when you are loading or reorganizing data. The free space is used when you insert or update data.</td>
</tr>
</tbody>
</table>
To create a new table using a wizard:

1. Open an object wizard for a table.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Schema</td>
<td>Select the schema that is to own the table.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the table.</td>
</tr>
<tr>
<td>Database Name</td>
<td>Select the database that is to contain this table.</td>
</tr>
<tr>
<td>Volatile</td>
<td>Enable this feature for tables that vary widely in size.</td>
</tr>
<tr>
<td>Audit</td>
<td>Specify an AUDIT Clause option of ALL, NONE, or CHANGES.</td>
</tr>
<tr>
<td>RestrictDrop</td>
<td>Enables the DB2 Restrict on Drop attribute.</td>
</tr>
<tr>
<td>Columns</td>
<td>For each column in the table, click the New button to create a column and provide a name for the column. Then, in the Column Attributes area, provide details for the column. Use the Delete button to drop a selected column.</td>
</tr>
<tr>
<td>Indexes</td>
<td>Specify any indexes on the table.</td>
</tr>
<tr>
<td>Constraints</td>
<td>Selecting a constraint type and clicking Add opens the object wizard for that object type. For details see:</td>
</tr>
<tr>
<td></td>
<td>Primary Key Wizard for IBM DB2 for OS/390 and z/OS</td>
</tr>
<tr>
<td></td>
<td>Unique Key Wizard for IBM DB2 for OS/390 and z/OS</td>
</tr>
<tr>
<td></td>
<td>Foreign Key Wizard for IBM DB2 for OS/390 and z/OS</td>
</tr>
<tr>
<td>Comment</td>
<td>Type a comment, associated with this index in the Comment area.</td>
</tr>
<tr>
<td>Permissions</td>
<td>Set up the user permissions for this database.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Schedule or Execute buttons to Create the new object.</td>
</tr>
</tbody>
</table>

Unique Key Wizard for IBM DB2 for OS/390 and z/OS

A unique key constraint is a key for which no two of its values can be equal and no values can be null. A table can have a number of unique constraints, but it cannot have more than one unique constraint on the same set of columns. If you are creating a unique key constraint on a table that already exists (as opposed to creating a unique key at the time the table is first generated), a unique index must already exist on the columns of the unique key you want to constrain. If no unique index exists, the Index Wizard will open as you complete the Create Unique Key Constraint dialog box.
To create a new unique key using a wizard:

1. **Open an object wizard** for a unique key.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td></td>
</tr>
<tr>
<td>Table Schema and Table Name</td>
<td>Choose the owner and name of the table in which the unique key is being created.</td>
</tr>
<tr>
<td>Schema and Name</td>
<td>Choose the owner and name of the unique key being created.</td>
</tr>
<tr>
<td>Clustered</td>
<td>Enable or disable clustering.</td>
</tr>
<tr>
<td>Number of Partitions</td>
<td>If you enabled Clustering, specify the number of partitions.</td>
</tr>
<tr>
<td>Buffer Pool</td>
<td>Provide the buffer pool in which this primary key should reside</td>
</tr>
<tr>
<td>Defer, Close, Copy, and Define</td>
<td>Enable or disable these DB2 options as required.</td>
</tr>
<tr>
<td>Piece Size</td>
<td>The maximum addressability of the dataset for a non-partitioned index.</td>
</tr>
<tr>
<td><strong>Columns</strong></td>
<td></td>
</tr>
<tr>
<td>From the Column dropdown, select a column for the primary key and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
<td></td>
</tr>
<tr>
<td><strong>Storage - Edit button (manage datasets)</strong></td>
<td>Choose a data set management option:</td>
</tr>
<tr>
<td>DB2 will define and manage the data sets on a volume of the default storage group of the database</td>
<td></td>
</tr>
<tr>
<td>DB2 will define and manage the data sets on a volume of the specified storage group</td>
<td>Select a storage group (a storage group is made up of disk device volumes): Each data set will be defined on a volume listed in the storage group you select. Minimum primary space allocation: 12 kilobytes is the default. Minimum secondary space allocation: 12 kilobytes is the default. NOTE: If the primary and (118 x) secondary space allocations combine to be greater than 2 gigabytes, more than one data set may eventually be used. Erase data sets when index dropped? If choose this option, DB2 will overwrite all data with zeros before they are deleted as a security measure.</td>
</tr>
</tbody>
</table>
User Datatype Wizard for IBM DB2 for OS/390 and z/OS

A datatype is a named set of valid values that can be manipulated by a set of operations. There are intrinsic datatypes, which are predefined and always available, and derived datatypes. A derived datatype is a user-defined datatype, which can include both intrinsic and previously derived datatypes. The User Datatype Wizard lets you create a derived datatype without knowing the underlying commands.

The User Data Type Wizard lets you:

- Name the user datatype.
- Define the base datatype properties.

### Step | Settings and tasks
--- | ---
**Storage - Attributes** (Availability differs according to the dataset management options you chose) | Storage Group, Primary Space Allocation, Secondary Space Allocation, Erase, and VCAT catalog (The ability to set these options depends on the dataset management options you chose.)
Free Page | One free page exists for every x pages. The x specifies how often to leave a page of free space when index entries are created from executing a DB2 utility or creating an index for a table with pre-existing rows. (0-255)
Percent Free | The percentage of free space you want to leave in every page when entries are added to an existing index. The default is 10%.
GBP Cache | This option is available only in a data-sharing environment. ALL: As pages are read, all of them will be cached in the group buffer pool. CHANGED: Updated pages are cached to the group buffer pool. NONE: No pages will be cached.
Partitions | Displays the default settings for the number of partitions you specified on the Properties pane. Select a partition and click the Edit button to modify details for that partition.
Comment | Type a comment, associated with this unique key in the Comment area.
DDL View | Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.
To create a new user datatype using a wizard:

1. Open an object wizard for a user datatype.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Select the owner of the user datatype.</td>
</tr>
<tr>
<td>Datatype</td>
<td>Provide a name for the datatype.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the base datatype.</td>
</tr>
<tr>
<td>Size</td>
<td>Provide the size of the datatype.</td>
</tr>
<tr>
<td>For Data</td>
<td>Select the MIXED, SBCS, or BIT option for the datatype.</td>
</tr>
<tr>
<td>CCSID</td>
<td>Select the NONE, ASCII, EBCDIC, or UNICODE option for the datatype.</td>
</tr>
<tr>
<td>Comment</td>
<td>Type a comment, associated with this unique key in the Comment area.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

View Wizard for IBM DB2 for OS/390 and z/OS

A view gives you a new way of looking at data in a results table. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables. Columns added to the base table(s) after the view is created are not included in the result set. Views are thus handy tools for controlling access to a table. You can allow someone to see portions of data without allowing that user to see the table in its entirety. For example, you can create a view that will permit a user to see employee names in a table without allowing access to the Social Security numbers of that same table.

The wizard itself is a single panel. After you complete the wizard, the View Editor opens so you can complete the definition of the view, choose the columns to show in the view, the dependencies, and access privileges to the view.

To create a new view using a wizard:

1. Open an object wizard for a view.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Select the owner of the view. The owner of the view must have SELECT privileges for the tables in the CREATE view statement or DBADM authority on the database that contains the table.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the view.</td>
</tr>
</tbody>
</table>
**DBArtisan Server Components Installation Wizard for IBM DB2 for OS/390 and z/OS**

DBArtisan provides enhanced functionality for IBM DB2 for OS/390 and z/OS version 6 and IBM DB2 for OS/390 and z/OS version 7. A portion of this functionality is implemented by components that must be installed on the DB2 server system. The following describes the system requirements for the components and the process for installing and enabling them.

For more information, see:

- [Installation Helpful Hints](#)
- [DBArtisan Server Components System Requirements](#)
- [DBArtisan Server Components Installation Overview](#)
- [DBArtisan Server Components Installation Wizard](#)

## Installation Helpful Hints

Before launching the DBArtisan Server Components Installation Wizard for IBM DB2 for OS/390 and z/OS, do the following:

- Obtain a User ID and password for mainframe access. This is for the FTP process and for creating datasets and running jobs.
- Grant RACF Authority on the mainframe to allow the User ID the authority to:
  - Create datasets using the pre-determined high level qualifier.
  - Read authority on DB2 and the language environment datasets.
  - Grant authority to access the DB2 startup JCL and view the address space through SDSF.
- Grant RACF Authority to access CEE. Datasets.
- Obtain the TCP/IP address of the mainframe.
- Obtain high-level qualifiers for Language Environment libraries, the SDSNEXIT library and SDSNLOAD libraries.

### Table: Check Type

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
</table>
| Check Type | **CHECK_NONE** - No search conditions must be satisfied for insert or update operations.  
**CHECK_LOCAL** - Update and insert operations on view must satisfy the search conditions of the view and underlying views that are defined with a check option. Furthermore, every updatable view that is directly or indirectly defined on view inherits those search conditions (the search conditions of view and all underlying views of that are defined with a check option) as a constraint on insert or update operations.  
**CHECK CASCADED** - Update and insert operations on the view must satisfy the search conditions of view and all underlying views, regardless of whether the underlying views were defined with a check option. Furthermore, every updatable view that is directly or indirectly defined on view inherits those search conditions (the search conditions of view and all underlying views) as a constraint on insert or update operations. |
| Definition | Complete the CREATE VIEW statement by typing or pasting in AS, SELECT, FROM, and WHERE statements or clauses. Finally, use the Schedule or Execute buttons to Create the new object. |
• Obtain the name of the workload manager environment used for running the IBM supplied DSNUTLIB.

• Obtain the name of the workload manager environment that will be used for running the DBArtisan REXX stored procedures.

• Obtain the name of an additional workload manager environment to run the remaining DBArtisan stored procedures (this may need to be created if one does not already exist.)

• Embarcadero’s utility execution stored procedures make extensive use of declared global temporary tables. The DBArtisan 8.0 z/OS server component installation instructions do not guide the user through the steps involved with defining the needed objects to allow for the declaration of these temporary tables. The following statements can be used to define the databases needed for proper execution of the affected features:

  • In a non-data sharing environment, run these statements, replacing xxxx with the target subsystem ID:

    CREATE DATABASE xxxxTEMP AS TEMP;
    CREATE TABLESPACE xxxxTMP1 IN xxxxTEMP SEGSIZE 32
    PRIQTY 12000 SECQTY 12000 BUFFERPOOL BP0;
    CREATE TABLESPACE xxxxTMP2 IN xxxxTEMP SEGSIZE 32
    PRIQTY 12000 SECQTY 12000 BUFFERPOOL BP0;

  • In a data sharing environment, run these statements one time for each member, replacing xxxx with the member name:

    CREATE DATABASE xxxxTEMP AS TEMP FOR xxxx;
    CREATE TABLESPACE xxxxTMP1 IN xxxxTEMP SEGSIZE 32
    PRIQTY 12000 SECQTY 12000 BUFFERPOOL BP0;
    CREATE TABLESPACE xxxxTMP2 IN xxxxTEMP SEGSIZE 32
    PRIQTY 12000 SECQTY 12000 BUFFERPOOL BP0;

• Proceed to the DBArtisan Server Components System Requirements.

System Requirements
The system requirements for the installation of DBArtisan server components for DB2 for OS/390 and z/OS are the following:

Software requirements
The following are software requirements for the DBArtisan server components:

  • IBM DB2 for OS/390 and z/OS Version 2 Release 10 or z/OS Version 1 Release 1 or higher
  • IBM DB2 for OS/390 and z/OS Version 6 or IBM DB2 for OS/390 and z/OS Version 7 or higher

Hardware requirements
There are no specific hardware requirements beyond those for the required software.

DASD
The server components require approximately 278 3390 tracks.
Environmental requirements

In order to automatically upload installation data sets, the Install Wizard requires an FTP server to be active on the target system, and the installing user ID must have access to it. For manual installation, the FTP server is not required.

In order to automatically submit installation jobs, the Install Wizard requires an FTP server to be active on the target system, with access to JES for job submission.

The server component program objects require a PDSE or PDS for storage.

If the SYSPROC.DSNWZP stored procedure runs in the DB2-established stored procedure address space (xxxxSPAS), which is the case by default for DB2 subsystem version 6 and version 7, the EMB.SUBSYSTEM_INFO stored procedure must run there also. This requires a PDSE or PDS in the STEPLIB concatenation in the SPAS startup procedure, with approximately 15 3390 tracks of available space.

With DB2 version 8, SYSPROC.DSNWZP runs in a WLM-established stored procedure address space and so EMB.SUBSYSTEM_INFO must also.

The EMB.UTILITY_PROC_O stored procedure has several specific requirements:

It must run in a WLM-established stored procedure address space. This requires a PDSE or PDS in the STEPLIB concatenation in the SPAS startup procedure, with approximately 15 3390 tracks of available space.

It must run APF-authorized, which requires that all libraries in the STEPLIB concatenation be APF-authorized.

Due to its specific resource requirements, the WLM SPAS in which it runs must use the NUMTCB=1 parameter.

The EMB.BIND_X and EMB.DATA_SET_INFO_X stored procedures are written in REXX, and must run in a WLM-established SPAS that is properly configured for running REXX language stored procedure. This requires a PDS (not necessarily a PDSE) in the SYSEXEC concatenation for the WLM SPAS startup procedure, with approximately 3 3390 tracks of available space.

The remaining stored procedures must run in a WLM-established SPAS. This requires a PDSE or PDS in the STEPLIB concatenation in the SPAS startup procedure. This WLM SPAS does not require APF authorization or NUMTCB=1.

DBArtisan Server Components Installation Overview

The Install Wizard provides a step-by-step interface that allows you to complete the required installation and configuration of the server components. DBArtisan automatically invokes the Install Wizard when it detects that the server components have not yet been installed. You can also invoke the wizard by selecting the DB2 z/OS Server Component Installer command on the Tools menu. You can also invoke the wizard directly, outside the context of DBArtisan, by executing the zOSInstaller.exe program installed with DBArtisan.

The Install Wizard gathers required information from you in a step-by-step interface, and uses the FTP server on the target system to:

• Upload data sets
• Link program objects
• Bind packages
• Copy program objects and REXX execs to stored procedure libraries.

If the FTP server is not available, or if you wish to use a different mechanism to upload data sets and submit installation jobs, you can still use the Install Wizard in a manual mode to walk through the installation process.

The installation process proceeds in two distinct phases, per-system installation and per-subsystem installation.
Per-system installation
In the first phase of the installation, you install the server component data sets on a target MVS system that hosts one or more DB2 subsystems. In a sysplex environment, if you have DASD shared across two or more members of the sysplex, you need to install the data sets only one time. After doing so, the data sets will automatically be available to the other sysplex members sharing that DASD.

Per-subsystem installation
After you install the server component data sets, you must make the server components available to stored procedure environments for the target DB2 subsystem you will access, and you must define the server component stored procedures in that subsystem. You must repeat this process one time for each subsystem that you will access with DBArtisan.

DBArtisan Server Components Installation Wizard
DBArtisan includes a DBArtisan Server Components Installation Wizard that helps you install the necessary components. This wizard consists of 26 panels. For the first panel, click here.

Welcome Panel
The Install Wizard opens with the following page describing the installation process, substantially repeating the information available in this document:

After reviewing the displayed information, click Next to continue the installation process. The Overview – Per-System Installation page displays.

Overview – Per-System Installation Panel
The Overview – Per-System Installation page provides an overview of the part of the installation that must be completed once per MVS system or sysplex:

After reviewing the steps, click Next to continue.

Overview – Per-Subsystem Installation Panel
The Overview – Per-Subsystem Installation page provides an overview of the part of the installation that must be completed once per DB2 subsystem:

Review the steps.

Select Skip these overview instructions in the future if you do not want the Overview – Per-System Installation and Overview – Per-Subsystem Installation pages to display the next time you run the Install Wizard.

Click Next to continue.

Installation Mode Panel
You use the Installation Mode page to select how the installation process should proceed:

Select Automatic Installation to use the Install Wizard to automatically upload data sets and submit installation jobs. This option requires access to an FTP server running on the target system, including the ability JES access to submit jobs and query job status.

Select Manual Installation to use the Install Wizard to describe the required steps in the installation process, as well as to generate JCL for the installation jobs. Use this option if an FTP server is not available on the target system.
After selecting the desired installation mode, click Next to continue. If you selected Automatic Installation, the FTP Server page displays.

**FTP Server Panel**
You use the FTP Server page to provide information about the FTP server running on the target system:

Enter the values in the displayed fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server name</td>
<td>Enter the IP host name or address of the target system.</td>
</tr>
<tr>
<td>Port number</td>
<td>Enter the port number on which the FTP server on the target system is listening. In most cases, there is no need to change the value from the default, 21.</td>
</tr>
<tr>
<td>User ID</td>
<td>Enter a user ID with which to log on to the target system. This user ID must have authority to create data sets, bind packages, and copy files to SPAS libraries.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password for the specified user ID.</td>
</tr>
</tbody>
</table>

After entering the required parameters, click Next to continue. The Install Wizard connects to the FTP server, validating the information you have entered. If an error occurs, an error message displays, and you remain on this page. Otherwise, the Installation Parts page displays.

**Installation Parts Panel**
You use the Installation Parts page to specify which parts of the installation process to perform:

Select Per-system followed by per-subsystem installation to use the Install Wizard to upload the required data sets to the target system and link the program objects, and then to configure the server components for one or more DB2 subsystems on that system.

Select Per-subsystem installation only if you have previously uploaded the required data sets, and wish only to configure the server components for one or more additional DB2 subsystems.

After selecting the installation parts, click Next to continue. If you selected Per-system followed by per-subsystem installation, the Installation Data Sets page displays. If you selected Per-subsystem installation only, the Data Sets page displays.

**Installation Data Sets Panel**
You use the Installation Data Sets page to provide information about how to create data sets on the target system to receive the server components (in distributed format):

Enter the values in the displayed fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>High level qualifier</td>
<td>Enter the desired high level qualifier for the installation data sets. Four data sets will be created using this HLQ: hlq.CNTL.XMIT, hlq.DBRMLIB.XMIT, hlq.EXEC.XMIT, and hlq.OBJ.XMIT.</td>
</tr>
<tr>
<td>Device type</td>
<td>Enter the device type to be used when allocating the installation data sets. 3390 is recommended.</td>
</tr>
<tr>
<td>Volume</td>
<td>Optionally, enter the volume to be used when allocating the installation data sets. If no volume is specified here, none will be specified when allocating the data sets.</td>
</tr>
<tr>
<td>Management class</td>
<td>Optionally, enter the desired SMS management class for the allocated data sets.</td>
</tr>
</tbody>
</table>
After entering the required parameters, click Next to continue. The Production Data Sets page displays.

**Production Data Sets Panel**
You use the Production Data Sets page to provide information about how to create data sets on the target system to store the server components (in production format):

Enter the values in the displayed fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage class</td>
<td>Optionally, enter the desired SMS storage class for the allocated data sets.</td>
</tr>
</tbody>
</table>

After entering the required parameters, click Next to continue. The Transfer Files page displays.

**Transfer Files Panel**
You use the Transfer Files page to either transfer the installation data sets to the target system (if you are using automatic mode) or to describe and confirm the transfer (if you are using manual mode):

If you are using automatic mode to install, simply click Next to have the Install Wizard connect to the FTP server and transfer the files.

If you are using manual mode to install, use the information displayed in the window to manually transfer the files, and click Next when you have finished.

*TIP:* For a successful Receive job, use the IND$FILE program with the PUT command and no options to transfer the files.

The Receive Job page displays.

**Receive Job Panel**
You use the Receive Job page to provide information about how to generate JCL for a job to receive the uploaded installation data sets into the production data sets:

Enter the values in the displayed fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job cards</td>
<td>Enter appropriate job cards for executing a batch job on the target system. In order for the Install Wizard to function correctly, it must be able to obtain job status information for jobs that it submits from the FTP server. Depending on the version of the FTP server and how it is configured, this may be possible only if the job name consists of your user ID followed by a single character.</td>
</tr>
</tbody>
</table>
After entering the required parameters, click Next to continue. The Review Receive Job page displays.

**Review Receive Job Panel**

You use the Review Receive Job page to review and execute the generated JCL to receive the installation data sets into the production data sets.

The generated JCL is displayed in the Receive JCL field. If necessary, you can modify this JCL to adhere to site-specific requirements.

**NOTE:** Do not remove the DSNTYPE=LIBRARY parameter from the LOAD DD statement. The EMB.V730.LOAD library must be created as a PDSE or PDS.

If you are using automatic mode to install, simply click Next to have the Install Wizard submit the job (including any changes you make) and wait for it to complete.

If you are using manual mode to install, you must submit the displayed job before continuing. You can either select the text in the Receive JCL field and press CTRL+C to copy it to the Windows Clipboard, or you can click the Save button to save it in a local file. Click Next when the job has completed successfully.

The Link Job page displays.

**Link Job Panel**

The last step in creating the production data sets is to link the supplied object code into executable program objects. You use the Link Job page to provide information about how to generate JCL for a job to perform the link:

Enter the value in the displayed field:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>High level qualifier</td>
<td>Several Language Environment data sets are required during the link. You must enter the high level qualifier of these data sets. The generated names of the data sets will be included in the SYSLIB concatenation of the link step.</td>
</tr>
</tbody>
</table>

After entering the required parameter, click Next to continue. The Review Link Job page displays.

**Review Link Job Panel**

You use the Review Link Job page to review and execute the generated JCL to link the supplied object code into executable program objects:

The generated JCL is displayed in the Link JCL field. If necessary, you can modify this JCL to adhere to site-specific requirements.

If you are using automatic mode to install, simply click Next to have the Install Wizard submit the job (including any changes you make) and wait for it to complete.

If you are using manual mode to install, you must submit the displayed job before continuing. You can either select the text in the Link JCL field and press CTRL+C to copy it to the Windows Clipboard, or you can click the Save button to save it in a local file. Click Next when the job has completed successfully.

The Per-System Installation Complete page displays.

**Per-System Installation Complete Panel**

The Per-System Installation Complete page informs you that you have successfully completed the first part of the installation process:
Click Next to proceed with the second part of the installation process, the per-subsystem installation. The Subsystem Configuration page displays.

**Subsystem Configuration Panel**
You use the Subsystem Configuration page to provide information required to connect to the DB2 subsystem you wish to configure:

Enter the values in the displayed fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsystem ID</td>
<td>Enter the 1-4 character subsystem ID for the desired DB2 subsystem.</td>
</tr>
<tr>
<td>Load library 1</td>
<td>Optionally, enter the name of the first load library to allocate to STEPLIB in order to access the specified subsystem. If all of the subsystem libraries are in the system linklist, you can leave this field blank.</td>
</tr>
<tr>
<td>Load library 2</td>
<td>Optionally, enter the name of an additional load library to allocate to STEPLIB in order to access the specified subsystem. If all of the subsystem libraries are in the system linklist, you can leave this field blank.</td>
</tr>
<tr>
<td>Load library 3</td>
<td>Optionally, enter the name of an additional load library to allocate to STEPLIB in order to access the specified subsystem. If all of the subsystem libraries are in the system linklist, you can leave this field blank.</td>
</tr>
</tbody>
</table>

After entering the required parameters, click Next to continue. The Bind Job page displays.

**Bind Job Panel**
You use the Bind Job page to provide information required to bind the server component packages and plans:

Enter the value in the displayed field:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection ID</td>
<td>Enter the collection ID in which the server component packages should be bound. You must have the BINDADD (or similar) privilege in this collection. See the DB2 Command Reference manual for more information on requirements for binding packages and plans.</td>
</tr>
</tbody>
</table>

After entering the required parameter, click Next to continue. The Review Bind Job page displays.

**Review Bind Job Panel**
You use the Review Bind Job page to review and execute the generated JCL to bind the server component packages and plans:

The generated JCL is displayed in the Bind JCL field. If necessary, you can modify this JCL to adhere to site-specific requirements.

If you are using automatic mode to install, simply click Next to have the Install Wizard submit the job (including any changes you make) and wait for it to complete.

If you are using manual mode to install, you must submit the displayed job before continuing. You can either select the text in the Bind JCL field and press CTRL+C to copy it to the Windows Clipboard, or you can click the Save button to save it in a local file. Click Next when the job has completed successfully.

The SPAS Environment for EMB.SUBSYSTEM_INFO page displays.
SPAS Environment for EMB.SUBSYSTEM_INFO Panel
You use the SPAS Environment for EMB.SUBSYSTEM_INFO page to provide information about the environment in which the stored procedure will run:

Enter the value in the displayed field:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2-established</td>
<td>Select this value if the SYSPROC.DSNWZP stored procedure runs in the DB2-established stored procedure address space (xxxxSPAS). The EMB.SUBSYSTEM_INFO stored procedure will be created to run in this same environment.</td>
</tr>
<tr>
<td>WLM-established</td>
<td>Select this value if the SYSPROC.DSNWZP stored procedure runs in a WLM-established stored procedure address space. This is true for DB2 for z/OS version 8, and also for DB2 for OS/390 and z/OS version 7 if APAR PQ12345 is applied. The EMB.SUBSYSTEM_INFO stored procedure will be created to run in a WLM environment.</td>
</tr>
<tr>
<td>Load library</td>
<td>If you selected DB2-established, enter the name of a PDS or PDSE in the STEPLIB concatenation in the startup procedure for the DB2-established SPAS for the selected subsystem. The load module for the EMB.SUBSYSTEM_INFO stored procedure will be copied to this library. You must have RACF update authority on this library. If instead, the EMB.V800.LOAD data set has been added to STEPLIB, leave this field blank, and no copy will be performed.</td>
</tr>
</tbody>
</table>

After entering the required parameter, click Next to continue. The WLM Environment for EMB.UTILITY_PROC_O page displays.

WLM Environment for EMB.UTILITY_PROC_O Panel
You use the WLM Environment for EMB.UTILITY_PROC_O page to provide information about the WLM environment in which the stored procedure will run:

Enter the value in the displayed field:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment name</td>
<td>Enter the name of the WLM application environment in which the EMB.UTILITY_PROC_O stored procedure will run. All of the data sets in the STEPLIB concatenation in the startup procedure for the environment must be APF-authorized, and the environment must be started with the NUMTCB=1 parameter.</td>
</tr>
<tr>
<td>Load library</td>
<td>Enter the name of a PDS or PDSE in the STEPLIB concatenation in the startup procedure for the specified WLM-established SPAS. The load module for the EMB.UTILITY_PROC_O stored procedure will be copied to this library. You must have RACF update authority on this library. If instead, the EMB.V800.LOAD data set has been added to STEPLIB, leave this field blank, and no copy will be performed.</td>
</tr>
</tbody>
</table>

After entering the required parameters, click Next to continue. The WLM Environment for REXX Stored Procedures page displays.

WLM Environment for REXX Stored Procedures Panel
You use the WLM Environment for REXX Stored Procedures page to provide information about the WLM application environment in which the stored procedures written in REXX will run:
Enter the values in the displayed fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment name</td>
<td>Enter the name of the WLM application environment in which the REXX stored procedures will run. The startup procedure for the environment must be properly configured to run REXX stored procedures.</td>
</tr>
<tr>
<td>Exec library</td>
<td>Enter the name of a PDS or PDSE in the SYSEXEC concatenation in the startup procedure for the specified WLM-established stored procedure address space. The execs for the REXX stored procedures will be copied to this library. If the EMB.V800.EXEC data set has been added to SYSEXEC, leave this field blank, and no copies will be performed.</td>
</tr>
</tbody>
</table>

After entering the required parameters, click Next to continue. The WLM Environment for Other Stored Procedures page displays.

**WLM Environment for Other Stored Procedures Panel**

You use the WLM Environment for Other Stored Procedures page to provide information about the WLM application environment in which the remaining stored procedures will run:

Enter the values in the displayed fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment name</td>
<td>Enter the name of the WLM application environment in which the remaining stored procedures will run. There are no special requirements for the configuration of this environment.</td>
</tr>
<tr>
<td>Load library</td>
<td>Enter the name of a PDSE or PDS in the STEPLIB concatenation in the startup procedure for the specified WLM-established stored procedure address space. The program objects for the remaining stored procedures will be copied to this library. If the EMB.V800.LOAD data set has been added to STEPLIB, leave this field blank, and no copies will be performed.</td>
</tr>
</tbody>
</table>

After entering the required parameters, click Next to continue. If you entered the names of any load libraries or exec libraries for the DB2- or WLM-established stored procedures address spaces, the Review Copy Job page displays. Otherwise, the Review Create Job page displays.

**Review Copy Job Panel**

You use the Review Copy Job page to review and execute the generated JCL to copy the program objects and REXX execs to the appropriate stored procedure address space libraries:

The generated JCL is displayed in the Copy JCL field. If necessary, you can modify this JCL to adhere to site-specific requirements.

If you are using automatic mode to install, simply click Next to have the Install Wizard submit the job (including any changes you make) and wait for it to complete.

If you are using manual mode to install, you must submit the displayed job before continuing. You can either select the text in the Copy JCL field and press CTRL+C to copy it to the Windows Clipboard, or you can click the Save button to save it in a local file. Click Next when the job has completed successfully.

The Review Create Job page displays.

**Review Create Job Panel**

You use the Review Create Job page to review and execute the generated JCL to create the stored procedure definitions in the DB2 subsystem:
The generated JCL is displayed in the Create JCL field. If necessary, you can modify this JCL to adhere to site-specific requirements. You can review the SQL statements by scrolling down in the Create JCL field.

If you are using automatic mode to install, simply click Next to have the Install Wizard submit the job (including any changes you make) and wait for it to complete.

If you are using manual mode to install, you must submit the displayed job before continuing. You can either select the text in the Copy JCL field and press CTRL+C to copy it to the Windows Clipboard, or you can click the Save button to save it in a local file. Click Next when the job has completed successfully.

The Temporary Database page displays.

**Temporary Database Panel**
The Temporary Database page informs you of an additional DBArtisan requirement related to the use of declared global temporary tables.

As described on the page, the DBArtisan stored procedures use DB2 declared global temporary tables (DGTTs). You must ensure that a temporary database has been created in each subsystem (or each member of a data sharing group) accessed by DBArtisan. After completing the Install Wizard, you can use DBArtisan itself to create the temporary database, if it does not already exist. Alternatively, you can use the following SQL statements to create the temporary database.

In a non-data sharing environment, use these statements, replacing xxxx with the subsystem ID:

```
CREATE DATABASE xxxxTEMP AS TEMP;
CREATE TABLESPACE xxxxTMP1 IN xxxxTEMP SEGSIZE 32 PRIQTY 12000 SECQTY 12000 BUFFERPOOL BP0;
CREATE TABLESPACE xxxxTMP2 IN xxxxTEMP SEGSIZE 32 PRIQTY 12000 SECQTY 12000 BUFFERPOOL BP0;
```

In a data sharing environment, use these statements one time for each member of the data sharing group, replacing xxxx with the member name: `CREATE DATABASE xxxxTEMP AS TEMP FOR xxxx;`

```
CREATE TABLESPACE xxxxTMP1 IN xxxxTEMP SEGSIZE 32 PRIQTY 12000 SECQTY 12000 BUFFERPOOL BP0;
CREATE TABLESPACE xxxxTMP2 IN xxxxTEMP SEGSIZE 32 PRIQTY 12000 SECQTY 12000 BUFFERPOOL BP0;
```

After reviewing the information about temporary databases, click Next to continue. The Per-Subsystem Configuration Complete page displays.

**Per-Subsystem Installation Complete Panel**
The Per-System Installation Complete page informs you that you have successfully completed the second part of the installation process:

Click Next if you wish to configure another DB2 subsystem to be accessed by DBArtisan. The Subsystem Configuration page displays.

Click Finish if you have finished configuring all of the DB2 subsystems to be accessed by DBArtisan. The Install Wizard closes.
Microsoft SQL Server Object Wizards

Create Check Constraints Dialog Box  Database Wizard  Default Wizard  Dump or Backup Device Wizard

Foreign Key Wizard  Function Wizard  Index Wizard  Linked Server Wizard

Login Wizard  Create Primary Key Constraint Dialog Box  Procedure Wizard  Remote Server Wizard

Role Wizard  Rule Wizard  Table Wizard  Trigger Wizard

Create Unique Key Constraint Dialog Box  User Message Wizard  User Wizard  User Datatype Wizard

View Wizard

Database Wizard for Microsoft SQL Server

The Database Wizard presents you with a different set of options based on your server version to create the database accurately on each platform.

The Database Wizard lets you:

• Name the database.

• Specify how and where you want to place the physical storage of the database.

  **TIP:** Microsoft SQL Server recommends that you do not create any user objects, such as tables, views, stored procedures, or triggers, in the master database. The master database includes the system tables that store the system information used by SQL Server, such as configuration option settings.

• Attach an existing set of operating system files.

• Name the filegroup and the database on file name.

• Lets you add the transaction log file.

• Lets you specify database options.
To create a new database using a wizard:

1. **Open an object wizard** for a database.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Options</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the database.</td>
</tr>
<tr>
<td>Attach existing OS files</td>
<td>To create a database from an existing set of operating system files, there must be a &lt;filespec&gt; entry for the first PRIMARY file. The PRIMARY filegroup contains all the database system tables. Primary files have a .mdf extension.</td>
</tr>
<tr>
<td>Compatible Level</td>
<td>Select a version compatibility level.</td>
</tr>
<tr>
<td>Properties group</td>
<td>Select the following settings: ANSI null default, ANSI nulls, ANSI padding, ANSI warnings, auto create statistics, auto update statistics, autoclose, autoshrink, concat null yields null, cursor close on commit, arithabort, db chaining, dbo use only, default to local cursor, merge publish, numeric roundabout, offline, published, quoted identifier, read only, recursive triggers, select into/bulkcopy/pllsort, single user, subscribed, torn page detection, and trunc log on chkpt.</td>
</tr>
<tr>
<td>Placement</td>
<td>Indicate the file where you want the database to live. For example, a new Books database could include author and title filegroups. By default, when you open the Wizard and click the Placement tab, a filegroup definition, using the name you provided for the database and default settings, is displayed. For each filegroup to be added, click the New button, provide a Device File Name for the filegroup, and use the File Group Properties and Device File Properties groups to provide the attributes of the filegroup. Use the Delete button to delete a selected filegroup.</td>
</tr>
<tr>
<td>Transaction Log</td>
<td>The transaction log file is a required file for each database. This file holds the log information to recover the database. There can be multiple log files for a database, but there has to be at least one. Traditionally the logfile extension has been .ldf. By default, when you open the Wizard and click the Transaction Log tab, a transaction log file definition, using the name derived from the name you provided for the database and with default settings, is displayed. For each file to be added, click the New button, provide a Device File Name, and use the Log Device Properties group to provide the attributes of the file. Use the Delete button to delete a selected file.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

**NOTE:** As you complete the wizard, be aware that the Primary file contains startup information for the database and is also used to store data. The transaction log files hold the information used to recover the database.

**Default Wizard for Microsoft SQL Server**

When bound to a column or user-defined object, a default ensures that a specific value will be inserted into the column where the object will be bound if no explicit value is given.

The Default Wizard lets you name the default and specify its value.
To create a new default using a wizard:

1. Open an object wizard for a default.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td><strong>Setting</strong></td>
</tr>
<tr>
<td>Owner</td>
<td>Select the schema that is to own the default.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the default.</td>
</tr>
<tr>
<td>Value</td>
<td>Provide the value of the default.</td>
</tr>
<tr>
<td><strong>Dependencies</strong></td>
<td><strong>Setting</strong></td>
</tr>
<tr>
<td>Type</td>
<td>Choose Column or Datatype, and if you chose Column, choose a Table from the Table dropdown. The list on the left is populated with candidate columns or datatypes. To move a candidate from the list on the left to the dependencies column on the right, select the candidate and click Add. Remove columns or datatypes from the dependencies list on the right by selecting the column or datatype and clicking Remove.</td>
</tr>
</tbody>
</table>

**DDL View**

Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the **Execute** button to Create the new object.

---

**Backup Device Wizard for Microsoft SQL Server**

The Backup Device Wizard lets you link to a backup device. Backup devices store backups of databases and transaction logs.

Before you can dump a database or its transaction log, first link to a backup device that stores the resulting backup file. Microsoft SQL Server lets you create tape, disk and diskette dump devices.

**To Open the Backup Device Wizard**

1. On the Explorer, connect to the server where you want to link a backup device.
2. Expand the Storage node.
3. Right-click Backup Devices and select New.

The table below describes the fields you may encounter as you complete the Dump/Backup Device Wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the device type?</td>
<td>Pick the device that suits you. Disk- Fast and convenient mechanism for backups. This represents a disk file created on the hard disk. Diskette - Facilitates off site storage, but only for very small databases. Pipe - A temporary connection between Microsoft SQL Server and another application--a named pipe. Tape- A tape drive to facilitate off site storage for enhanced disaster recovery. Skip ANSI Labels - Select for tape if you want to skip American National Standards Institute (ANSI) labels.</td>
</tr>
<tr>
<td>What is the dump/backup device Name?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>What is the physical name?</td>
<td>Specify a device or file name that is recognized by the operating system (e.g., a:sqtable.dat).</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).
Extended Procedure Wizard for Microsoft SQL Server

Extended Procedures are dynamic link libraries that can be used to load and execute application routines written in other programming languages, such as C or Visual Basic. Extended Procedures function and appear in the same manner as SQL Server stored procedures in that you can pass parameters to them and obtain results.

Extended stored procedures provide a method for calling procedural language functions from within the Adaptive Server.

NOTE: Extended procedures can only be created in the Master database.

To create a new Extended procedure using a wizard:
1. Open an object wizard for an extended procedure.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Select the owner of the extended procedure.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the extended procedure.</td>
</tr>
<tr>
<td>Library</td>
<td>Provide the name of the DLL containing the extended procedure.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

Foreign Key Wizard for Microsoft SQL Server

Foreign keys are unique values that refer to specific columns of other tables. Thus, a foreign key links two tables together. Embarcadero DBArtisan’s Foreign Key Wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data. The column in the initial table, the parent table, is called the primary key. The corresponding column in the (child) table that references the primary key, is the foreign key. Foreign keys can also refer to columns within the same table.

The Foreign Key Wizard lets you:

• Name the foreign key constraint

• Identify the parent table and the referenced constraint.

• Map the column pairs between the parent and child tables.

To create a new Foreign Key using a wizard:
1. Open an object wizard for a foreign key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Table Owner and Table Name</td>
<td>Select the owner and name of the table for which the foreign key is being created.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the foreign key.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enables or disables the foreign key.</td>
</tr>
</tbody>
</table>
Function Wizard for Microsoft SQL Server

Functions are subroutines that you define so you can reuse code without having to reinvent the wheel each time. You can use functions to determine the best methods for controlling access and manipulation of the underlying data contained in an object. A function returns a value, unlike a stored procedure, which does not.

- To create a user-defined function, you need CREATE ANY privileges or IMPLICIT_SCHEMA authority on the database if the schema does not already exist.

To create a new function using a wizard:

1. **Open an object wizard** for a function.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

### Step Settings and tasks

<table>
<thead>
<tr>
<th><strong>Not For Replication</strong></th>
<th>Replication copies and distributes data and database objects from one database to another and then synchronizes information between databases for consistency.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete Rule</strong></td>
<td>If you choose the CASCADE option, all rows containing data involved with the foreign key will be deleted after a delete operation.</td>
</tr>
<tr>
<td><strong>Update Rule</strong></td>
<td>If you choose the CASCADE option, all rows containing data involved with the foreign key will be deleted after an update operation.</td>
</tr>
<tr>
<td><strong>Column Mapping</strong></td>
<td>Under <strong>Referenced Table</strong>, choose the <strong>Owner</strong> and then the <strong>Name</strong> of the referenced, or parent, table. Under the <strong>Main Table</strong>, select checkboxes corresponding to the columns that are to reference columns in the referenced table. Then, under <strong>Referenced Table</strong>, select the corresponding column check boxes.</td>
</tr>
<tr>
<td><strong>DDL View</strong></td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>

**Function Wizard for Microsoft SQL Server**

Functions are subroutines that you define so you can reuse code without having to reinvent the wheel each time. You can use functions to determine the best methods for controlling access and manipulation of the underlying data contained in an object. A function returns a value, unlike a stored procedure, which does not.

- To create a user-defined function, you need CREATE ANY privileges or IMPLICIT_SCHEMA authority on the database if the schema does not already exist.

**Index Wizard for Microsoft SQL Server**

Like an index in a book, a table index helps you get at the data you want without having to read through the whole table. Indexes can exist on single column or on multiple columns. Indexes appear in the form of B-trees. And, as for books, you can have multiple indexes for a single table. You can also create indexes for a view.
To create a new index using a wizard:

1. Open an object wizard for an extended procedure.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td><strong>Parent Type</strong> Select TABLE or VIEW.</td>
</tr>
<tr>
<td></td>
<td><strong>Parent Owner</strong> Select the owner of the table or view.</td>
</tr>
<tr>
<td></td>
<td><strong>Parent Name</strong> Select the specific table or view containing the columns you want to index.</td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong> Provide a name for the index.</td>
</tr>
<tr>
<td></td>
<td><strong>Index Type</strong> Select UNIQUE or NONUNIQUE. An index is unique when no two rows are permitted to have the same index value. <em>(Note: A clustered index on a view must be unique.)</em> If an INSERT or UPDATE statement creates a duplicate value, the operation may fail.</td>
</tr>
<tr>
<td></td>
<td><strong>Clustered</strong> A clustered index is one in which the physical order of rows matches the order of indexed rows. A table or view can only have one clustered index at a time. In a nonclustered index, the physical order of rows is independent of the indexed order of rows. For an indexed view, you can only create a nonclustered index if there is already a clustered index extant.</td>
</tr>
<tr>
<td></td>
<td><strong>Ignore Duplicate Key</strong> This option controls what happens when an attempt is made to insert a duplicate key value into a column that is part of a unique clustered index. If the option is selected and an INSERT statement that creates a duplicate key is executed, SQL Server issues a warning and ignores the duplicate row. If not selected, SQL Server issues an error message and rolls back the entire INSERT statement.</td>
</tr>
<tr>
<td></td>
<td><strong>Statistics Recompute</strong> Enabling this feature means queries involving the table run at the optimal level as distribution statistics are updated automatically when the index is created. If you disable this option, you can compromise query performance.</td>
</tr>
<tr>
<td></td>
<td><strong>File Group</strong> <strong>Fill Factor</strong> Fillfactor specifies a percentage (0-100) that indicates how full each leaf level of each index page during index creation should be. When an index page fills up, time is needed to split the index page to make room for more rows. This can affect performance, so choose the fillfactor value carefully. When a fillfactor is given, SQL server rounds up the number of rows on each index page. If no value is given, the default is zero.</td>
</tr>
<tr>
<td></td>
<td><strong>Pad Index</strong> Enable or disable padding of index pages.</td>
</tr>
<tr>
<td></td>
<td><strong>Sort in TempDB</strong> Select to store the intermediate index sort results in tempdb. This option may reduce the time needed to create an index if tempdb is on a different set of disks than the user database, but it increases the amount of disk space used to create an index. In addition to the space required in the user database to create the index, tempdb must have about the same amount of additional space to hold the intermediate sort results.</td>
</tr>
<tr>
<td></td>
<td><strong>Columns</strong> From the Column dropdown, select a column for the index and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
</tr>
<tr>
<td></td>
<td><strong>DDL View</strong> Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>
Linked Server Wizard for Microsoft SQL Server

A linked server object represents a database system other than Microsoft SQL Server. The links are created through OLE DB data sources. OLE DB is a COM-based application programming interface (API) for accessing data. OLE DB supports accessing data stored in any format (databases, spreadsheets, text files, and so on) for which an OLE DB provider is available. With a linked server you can execute commands against OLE DB data sources on different servers. Linked servers let you issue distributed queries, commands, and so on.

What the wizard is essentially asking you to do is to use its interface to provide much of the same information you give when you initially register a server.

To Open the Linked Server Wizard

1. Expand the Datasource menu on the toolbar and scroll down to Objects.
2. On the Objects list, select Linked Servers.
3. Click New on the toolbar.

The table that follows describes the fields you may encounter as you complete the wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the linked server?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>If this is a SQL Server, you can use the default values for</td>
<td>Any tables retrieved from the server are from the default database defined for the login on the linked server.</td>
</tr>
<tr>
<td>a linked SQL Server</td>
<td></td>
</tr>
<tr>
<td>What is the provider name?</td>
<td>Pick the appropriate provider from the drop-down list.</td>
</tr>
<tr>
<td>What is the product name of the OLE DB datasource you are</td>
<td>If the server is not an instance Microsoft SQL Server, click the provider list, and then click the name of the OLE DB provider managing access to the specified linked server. An OLE DB provider is a software component that exposes OLE DB interfaces. Each OLE DB provider exposes data from a particular type of data source (for example SQL Server databases, Access databases, or Excel spreadsheets).</td>
</tr>
<tr>
<td>linking to?</td>
<td></td>
</tr>
<tr>
<td>What is the datasource as interpreted by the OLE DB</td>
<td>This is the datasource to be used when the linked server is made through a connection to the OLE DB server. For example, in MS SQL server, it is the database name.</td>
</tr>
<tr>
<td>provider?</td>
<td></td>
</tr>
<tr>
<td>What is the location as interpreted by the OLE DB provider?</td>
<td>This is the physical location of the server, which is not often revealed by the server name. Use this to add a descriptor, like basement or third floor.</td>
</tr>
<tr>
<td>What is the OLE DB provider-specific connection string that</td>
<td>This is helpful if there are complicated connection strings and is information specific to the OLE DB provider. The connection string is an alternative to the datasource and provider name properties, but isn’t necessarily required.</td>
</tr>
<tr>
<td>identifies a unique data source?</td>
<td></td>
</tr>
<tr>
<td>What is the catalog to be used when making a connection to</td>
<td>A catalog, or data dictionary, holds the definition of a database structure. To connect to a linked server, the ODBC driver calls catalog functions. So, to call a linked server, you can name the catalog to which that database driver belongs</td>
</tr>
<tr>
<td>the OLE DB provider?</td>
<td></td>
</tr>
</tbody>
</table>
Login Wizard for Microsoft SQL Server

Your login account controls access to the server and all of the databases within it. Only the System Administrator or System Security Officer can create logins. Once you can log into a server, you need additional privileges to access user databases. Specifically, each database owner adds the login as a user or alias to the database.

To Open the Login Wizard

1. On the **Explorer**, open the database where you want to add a login.
2. Expand the **Security** node.
3. Right-click **Logins** and then click **New**.

The table that follows describes the fields you may encounter as you complete the wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server Options</strong></td>
<td><strong>RPC Out server</strong>: If checked, specifies that remote procedure calls with output parameters can be executed.</td>
</tr>
<tr>
<td></td>
<td><strong>RPC</strong>: Specifies whether the linked server supports remote procedure calls.</td>
</tr>
<tr>
<td></td>
<td><strong>Collation Compatible</strong>: Select if the character set and sort order in the datasource corresponding to the linked server is the same as the local server.</td>
</tr>
<tr>
<td></td>
<td><strong>Use Remote Collation</strong>: Select to use the collation information of character columns from the linked server.</td>
</tr>
<tr>
<td></td>
<td><strong>Collation name</strong>: Select the collation to be used for character data from the linked server if the datasource is not a SQL Server datasource.</td>
</tr>
<tr>
<td></td>
<td><strong>Lazy Schema Validation</strong>: If selected, the schema used by the linked server data is validated whether the local instance of SQL Server is idle or not.</td>
</tr>
<tr>
<td></td>
<td><strong>Query Timeout</strong>: Timeout for queries to the linked server in seconds.</td>
</tr>
<tr>
<td></td>
<td><strong>Connection Timeout</strong>: Timeout for connecting to the linked server in seconds.</td>
</tr>
<tr>
<td><strong>Required Information</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>What is the name of the login?</td>
<td>You can use either the name of an individual, a role, or whatever you please.</td>
</tr>
<tr>
<td>Specify whether this login account will be</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>for a SQL Server Login or a Windows NT User</td>
<td></td>
</tr>
<tr>
<td>or NT Group</td>
<td></td>
</tr>
<tr>
<td>What should be the login’s default database?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>(Master is not recommended)</td>
<td>Master is not recommended to prevent users from creating objects there by mistake.</td>
</tr>
<tr>
<td>What is the login’s default language?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>The server’s default is used if you don’t</td>
<td></td>
</tr>
<tr>
<td>make a selection</td>
<td></td>
</tr>
<tr>
<td>Select databases to create users</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>Database roles</td>
<td><strong>Note</strong>: The options on the panel vary by version of MS SQL Server.</td>
</tr>
</tbody>
</table>
Primary Key Wizard for Microsoft SQL Server

Primary key constraints make sure that no duplicate values or NULLS are entered in the columns you specify. You can use primary key constraints to enforce uniqueness and referential integrity. A table can only have a single primary key constraint.

The dialog box lets you specify the owner and table on which you want to place the primary key constraint.

To create a new primary key using a wizard:

1. Open an object wizard for a primary key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td><strong>Table Owner</strong> and <strong>Table Name</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Provide a name of the primary key being created.</td>
</tr>
<tr>
<td><strong>Clustered</strong></td>
<td>Enable or disable clustering.</td>
</tr>
<tr>
<td><strong>File Group</strong></td>
<td>If you do not specify a filegroup, Microsoft SQL Server creates the index in the default filegroup.</td>
</tr>
<tr>
<td><strong>Fill Factor</strong></td>
<td>This specifies how full each index page that's storing data should be. The fill factor is a percentage value between 0 and 100.</td>
</tr>
</tbody>
</table>

| **Columns** | From the Column dropdown, select a column for the primary key and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns. |
| **DDL View** | Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the **Execute** button to **Create the new object**. |

Procedure Wizard for Microsoft SQL Server

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

The Procedure Wizard lets you:

- Name the procedure and specify its body.
- Specify any execution options and you can encrypt the stored procedure text in syscomments.
To create a new procedure using a wizard:

1. Open an object wizard for a procedure.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Select the owner of the procedure.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the procedure.</td>
</tr>
<tr>
<td>Procedure Number</td>
<td>Optionally, provide a procedure number. By using a number, you can group procedures of the same name together. This also enables you to drop them using only one DROP PROCEDURE statement. So, the procedures bill;1, bill;2, bill;3, etc. will be dropped simultaneously when the time comes.</td>
</tr>
<tr>
<td>Replication</td>
<td>This option creates a procedure that is used as a stored procedure filter and is executed only during replication.</td>
</tr>
<tr>
<td>Repcompile</td>
<td>The plan for this procedure will not be cached and the procedure is recompiled when it is run. This option is appropriate when you're using atypical or temporary values and you don't want to override the execution plan cached in memory.</td>
</tr>
<tr>
<td>Encryption</td>
<td>If you select this option, SQL Server will encrypt the syscomments table entry containing the text of the CREATE PROCEDURE statement. It keeps the procedure from being published as part of replication.</td>
</tr>
<tr>
<td>Definition</td>
<td>Complete the CREATE PROCEDURE outline provided by typing or pasting the body of the procedure. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

Remote Server Wizard for Microsoft SQL Server

Remote Servers are servers on a network that you can access through your local server. Instead of logging directly into a remote server, you can log into your local server and execute a remote procedure call to it.

**NOTE:** Only stored procedures are allowed against remote servers. You can use a linked server to execute distributed queries and stored procedures.

Before beginning

Before you can set up a remote server, you first configure both the local and remote servers to accept remote access. To configure the Microsoft SQL Servers for remote access, you need to log into each server. The table below describes the parameters you need to configure on each server:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Access</td>
<td>1 (enabled)</td>
</tr>
<tr>
<td>Remote Connections</td>
<td>Number of remote connections required</td>
</tr>
<tr>
<td>Remote Logins</td>
<td>Number of remote logins required</td>
</tr>
<tr>
<td>Remote Sites</td>
<td>Number of remote sites required</td>
</tr>
</tbody>
</table>

After setting these server configuration parameters, shutdown and restart the server so the new values can take effect.
To Open the Remote Server Wizard
1. On the Explorer expand the database where you want to connect a remote server.
2. Expand the Security node, right-click Remote Servers and select New.

The table that follows describes the fields you may encounter as you complete the Remote Server Wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the remote server?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>Is the remote server the local or remote server?</td>
<td>The local server routes all commands to the remote server. To communicate with a remote server, you log into a local server and submit your remote procedure calls for remote servers. The remote server receives remote procedure calls from the local server and processes them accordingly.</td>
</tr>
</tbody>
</table>

Server Options
- **OPTIONAL** (options available to you vary by SQL Server version)
  - **Publication Server** - Select if the server is to publish data for replication.
  - **Subscription Server** - This option applies if you want to deploy a subscription management application from the remote server.
  - **Distribution Server** - Select if the server is to manage the distribution database.
  - **Publisher/Subscriber** - Select if the server is both a subscriber and publisher of replicated data.
  - **DSN Server** - Select if the server is to receive replicated data via ODBC.
  - **Collation Compatible** - Select if the server is to be collation compatible.
  - **Data Access Server** - Select if the server is to serve as a data access server.
  - **RPC Out Server** - Select if the server is to serve as a RPC Out server.

Role Wizard for Microsoft SQL Server
Roles are sets of user privileges you associate with access to objects within a database. Roles streamline the process of granting permissions. You can use roles to grant sets of permissions and privileges to users and groups. Roles can help you comply with Sarbanes Oxley regulations by limiting which users can have access to what privileges, for example a Human Resources Role versus an Accounting Role.

To Open the Role Wizard
1. On the Explorer expand the database node where you want to add the Role.
2. Right-click the Roles node and select New.

The single page wizard asks you to name the role you’re creating, assign ownership, and select whether you want to use the standard SQL Server role authorization or Application Role authorization where you will create a password so the application can determine a user’s authenticity.

After you click Finish and the role has been created, the Using the Roles editor (Microsoft SQL Server, Oracle, Sybase) opens where you can assign object and system privileges to the role and determine which users can take part in the role.
Rule Wizard for Microsoft SQL Server

Rules promote data integrity by allowing you to validate the values supplied to a table column. They are reusable objects that you can bind to table columns or user datatypes. Check constraints are similar to rules, and are in fact the preferred way of restricting data. A column or user-defined data type can have only one rule bound to it, but a column can have both a rule and one or more check constraints associated with it. Not that a rule cannot apply to data already existing in the database at the time you're creating the rule and can't be bound to a system-created data type. If you create a new rule when one already exists, the new rule will override the previous one.

To create a new rule using a wizard:

1. Open an object wizard for a rule.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Select the Owner of the rule.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the rule.</td>
</tr>
<tr>
<td>Restriction</td>
<td>Type the condition. The rule restriction is the condition that defines the rule and can be any expression valid in a WHERE clause and can include such elements as arithmetic operators, relational operators, and predicates (for example, IN, LIKE, BETWEEN).</td>
</tr>
<tr>
<td>Dependencies</td>
<td>From the Type dropdown, choose Column or Datatype, and if you chose Column, choose a Table from the Table dropdown. The list on the left is populated with candidate columns or datatypes. To move a candidate from the list on the left to the dependencies column on the right, select the candidate and click Add. Remove columns or datatypes from the dependencies list on the right by selecting the column or datatype and clicking Remove.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

Table Wizard for Microsoft SQL Server

A table is a column-based arrangement of data in which the content of one column has a bearing on the other column(s). So, for example, a table might have a column for authors, another column for the books each author has written, and a third for the number of copies each title by a given author has sold. The data moves across the columns in rows.

You must have CREATE TABLE permissions to generate a new table.
To create a new table using a wizard:
1. Open an object wizard for a table.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td><strong>Schema</strong> Select the owner of the table.</td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong> Provide a name for the table.</td>
</tr>
<tr>
<td></td>
<td><strong>ANSI_NULLS option</strong> By setting this option, you are setting ANSI_NULLS, ANSI_PADDING, ANSI_WARNINGS, and QUOTED_IDENTIFIER to on so the table can be used in an Indexed View.</td>
</tr>
<tr>
<td></td>
<td><strong>Filegroup</strong> A filegroup categorizes the OS files containing data from a single SQL Server database to simplify some tasks, like backing up. A single database can contain more than one filegroup but a filegroup can’t contain more than the OS files for a single database. Pick the filegroup you want to use from the drop-down list. If you make no selection, the default is used.</td>
</tr>
<tr>
<td></td>
<td><strong>Text Image Filegroup</strong> Unless you make a selection, the text and image columns will be stored on the same filegroup as the table. Because these datatypes may contain lots of data, putting them on a different filegroup can help performance.</td>
</tr>
<tr>
<td>Columns</td>
<td>For each column in the table, click the Add Column button to create a column, provide a Name for the column and provide or select the remaining column attributes. Use the <strong>Delete</strong> button to drop a selected column. Use the arrow buttons to reposition the columns. <strong>NOTE:</strong> For SQL Server 2000, if you create a table with a Column datatype = text., you can set the storage and image values on the Storage Tab of the Tables Editor Storage Tab. When you have a text datatype, the Storage Tab displays a Text In Row box where you can specify the maximum size to be stored. <strong>NOTE:</strong> Because the smalldatetime datatype stores dates and time with less precision than the datetime datatype, before outputting you use the CAST or CONVERT functions to convert any boxes with the smalldatetime datatype to either VARCHAR or datetime datatypes. For more information, see SQL Server Books Online, Transact-SQL Reference.</td>
</tr>
<tr>
<td>Indexes</td>
<td>Click Add to open the <strong>Index Wizard for IBM DB2 for OS/390 and z/OS</strong>.</td>
</tr>
<tr>
<td>Constraints</td>
<td>Selecting a constraint type and clicking Add opens the object wizard for that object type. For details see: <strong>Primary Key Wizard for Microsoft SQL Server</strong> <strong>Unique Key Wizard for Microsoft SQL Server</strong> <strong>Foreign Key Wizard for Microsoft SQL Server</strong> Create or Modify Constraint</td>
</tr>
<tr>
<td>Permissions</td>
<td>For each specific permission to be granted, select the cell corresponding to the name and specific permission, and click the <strong>Grant</strong> button. To revoke a privilege, select a cell showing a Granted permission and click Revoke.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to Create the new object.</td>
</tr>
</tbody>
</table>
Trigger Wizard for Microsoft SQL Server

Triggers are a special type of procedure that automatically fire when defined data modification operations (insert, update, or delete) occur on a target table or view. Triggers fire after an insert, update or delete, but belong to the same transaction as the data modification operation. Triggers can be implemented to enforce business rules or referential data integrity.

Important Notes

- For more information on the syntax for Trigger bodies, consult the Microsoft SQL Server Transact-SQL Documentation.

To create a new trigger using a wizard:

1. Open an object wizard for a trigger.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Select the TABLE or VIEW on which the trigger is to be created.</td>
</tr>
<tr>
<td>Parent Type</td>
<td></td>
</tr>
<tr>
<td>Parent Schema</td>
<td>Select the owner of the table or view on which the trigger is to be created.</td>
</tr>
<tr>
<td>Parent Name</td>
<td>Select the specific table or view in which the trigger is to be created.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the trigger.</td>
</tr>
<tr>
<td>Trigger Timing</td>
<td>INSTEAD OF: This is the only option for a View trigger. An INSTEAD OF trigger fires in place of the triggering statement and will not make changes to the data unless the conditions of the INSTEAD OF statement are met first. So, your UPDATE execution statement is replaced by an INSTEAD OF UPDATE statement as a way to enforce particular business rules you establish. AFTER: An AFTER trigger fires following the successful completion of the triggering action. So, for example, the trigger would fire after an UPDATE statement has executed and after constraints have been checked and verified.</td>
</tr>
<tr>
<td>Fire On Insert</td>
<td>An INSERT trigger must be associated with an INSERT statement. For example, if a data load operation doesn’t include an INSERT statement, the trigger won’t be invoked.</td>
</tr>
<tr>
<td>Fire On Update</td>
<td>An UPDATE trigger can be associated with specific columns of the base table and will only be activated if those columns are updated.</td>
</tr>
<tr>
<td>Fire On Delete</td>
<td>A DELETE trigger is associated with a DELETE operation.</td>
</tr>
<tr>
<td>Encrypted</td>
<td>If you choose to encrypt the trigger, the trigger can’t be published as part of SQL Server replication.</td>
</tr>
</tbody>
</table>

Definition

Complete the CREATE TRIGGER outline provided by typing or pasting the body of the trigger. Finally, use the Schedule or Execute buttons to Create the new object.

You can use the Trigger Editor opens to create dependencies or alter the trigger statement.
Unique Key Wizard for Microsoft SQL Server

Unique keys can enforce logical keys that are not chosen as the primary key. In other words, you can use a unique key to ensure no duplicate values are entered in specific columns that are not a part of the primary key. Although you can only attach one primary key to a table, you can attach multiple unique keys. Also, you can use unique keys on columns that allow null values.

To create a new unique key using a wizard:

1. Open an object wizard for a unique key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Table Owner and Table Name Choose the owner and name of the table in which the unique key is being created.</td>
</tr>
<tr>
<td></td>
<td>Name Provide a name of the unique key being created.</td>
</tr>
<tr>
<td></td>
<td>Clustered Enable or disable clustering.</td>
</tr>
<tr>
<td></td>
<td>File Group If you do not specify a filegroup, Microsoft SQL Server creates the index in the default filegroup.</td>
</tr>
<tr>
<td></td>
<td>Fill Factor This specifies how full each index page that’s storing data should be. The fill factor is a percentage value between 0 and 100.</td>
</tr>
<tr>
<td>Columns</td>
<td>From the Column dropdown, select a column for the primary key and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

User Message Wizard for Microsoft SQL Server

A user message lets you write the error message users will see when a user-defined event transpires. You have the option of saving the message to the Windows NT Event log. You can also create messages in multiple languages, but you have to create the message in US English before you can write it in other languages.

Important Notes

• The user messages node only displays under the master database.

To Open the User Message Wizard

1. On the Explorer, expand the datasource where you want to write a user message.
2. Open the Master database, right-click the User Messages Node, and then click New.

The table that follows describes the fields you will encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the message number?</td>
<td>A message number must be greater than 50,000. The wizard automatically loads with the next unclaimed message number, but you can use another number if you prefer.</td>
</tr>
<tr>
<td>What is the severity?</td>
<td>The security level can be any level between 1 and 25. The levels and their meanings are indicated in the User Message Editor.</td>
</tr>
</tbody>
</table>
User Wizard for Microsoft SQL Server

The User Wizard lets you create a user who will then have access to the database where you are registering him or her. You can also identify the appropriate user group and the system privileges you want to assign to the new user.

To create a new user using a wizard:
1. Open an object wizard for a user.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Properties</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>Select the server login associated with this user.</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Provide the user name.</td>
<td></td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute buttons to Create the new object.</td>
<td></td>
</tr>
</tbody>
</table>

User Datatype Wizard for Microsoft SQL Server

User datatypes promote domain consistency by streamlining the definition of commonly used table columns in a database. You can build a customized datatype from system datatypes and bind defaults and rules to it to enhance integrity. When you reference the user datatype in a column, the column assumes all of the properties of the user datatype.

To create a new user datatype using a wizard:
1. Open an object wizard for a user datatype.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Select the owner of the user datatype.</td>
</tr>
<tr>
<td>Datatype</td>
<td>Provide a name for the datatype.</td>
</tr>
<tr>
<td>Type</td>
<td>Select the base datatype.</td>
</tr>
<tr>
<td>Size</td>
<td>Provide the size of the datatype.</td>
</tr>
<tr>
<td>Allow Nulls</td>
<td>Null has no explicitly assigned value. Null is not equivalent to zero or blank. A value of null is not considered to be greater than, less than, or equivalent to any other value, including another value of null.</td>
</tr>
<tr>
<td>Default Binding</td>
<td>Defaults promote data integrity by supplying a default value to a column if the user does not explicitly provide one. They are reusable objects that you can bind to user datatypes.</td>
</tr>
</tbody>
</table>
View Wizard for Microsoft SQL Server

Views are SQL queries stored in the system catalog that customize the display of data contained in one or more tables. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables.

To create a new view using a wizard:
1. Open an object wizard for a view.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Binding</td>
<td>Rules promote data integrity by allowing you to validate the values supplied to a column. They are reusable objects that you can bind to user datatypes.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute buttons to Create the new object.</td>
</tr>
</tbody>
</table>

Oracle Object Wizards

<table>
<thead>
<tr>
<th>Create Check Constraint Dialog Box</th>
<th>Cluster Wizard</th>
<th>Database Link Wizard</th>
<th>Directory Wizard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreign Key Wizard</td>
<td>Function Wizard</td>
<td>Index Wizard</td>
<td>Job Queue Wizard</td>
</tr>
<tr>
<td>Library Wizard</td>
<td>Materialized View Wizard (formerly Snapshot Wizard)</td>
<td>Materialized View Log Wizard (formerly Snapshot Log Wizard)</td>
<td>Package Wizard</td>
</tr>
<tr>
<td>Create Primary Key Constraint Dialog Box</td>
<td>Procedure Wizard</td>
<td>Profile Wizard</td>
<td>Redo Log Group Wizard</td>
</tr>
</tbody>
</table>
Cluster Wizard for Oracle

The Cluster Wizard lets you create a cluster. A cluster is a schema object that has one or more tables that all have one or more common columns. Rows of one or more tables that share the same value in these common columns are stored together in the database. The related columns of tables stored in a cluster are known as the cluster key.

Important Notes

- To create a cluster, you need the CREATE CLUSTER or CREATE ANY CLUSTER system privilege.

To Open the Cluster Wizard

1. On the Explorer, find the schema where you want to add the new cluster.
2. On the Cluster branch, right-click and select New.

OR

1. On the main toolbar, click Datasource and scroll to Objects
2. Click Clusters and then click New from the toolbar.

The tables below describe the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who owns the cluster?</td>
<td>Pick an owner</td>
</tr>
<tr>
<td>What is the name of the cluster?</td>
<td>Type a unique name</td>
</tr>
<tr>
<td>On which tablespace do you want to create the cluster?</td>
<td>Self-explanatory</td>
</tr>
</tbody>
</table>
| Add columns that are in this cluster | Add Button - Opens the Add Cluster Column dialog box.
Edit Button - Open the Modify Cluster Column dialog box.
Drop Button - Drops the column. |
<p>| What is the size of this cluster? | This is the estimated number of bytes/KB/MB required by an average cluster key and its associated rows. Do not exceed the size of a data block. |
| What is the cluster type? | Index: Rows having the same cluster key value are stored together. Each separate cluster key is stored only once in each data block. An indexed cluster is helpful if your clustered tables might grow unpredictably. Hash: Rows with the same hash key value are stored together. This is helpful if the tables are static. |
| If this is a hash cluster, what is the number of hash keys? | Type the number of hash keys. Oracle will round the value up to the nearest prime number. |</p>
<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If this is a hash cluster, what is the hash function?</td>
<td>Oracle uses a hash function to generate a distribution of numeric values, called hash values, which are based on specific cluster key values. The key of a hash cluster, like the key of an index cluster, can be a single column or composite key (multiple column key). To find or store a row in a hash cluster, Oracle applies the hash function to the row's cluster key value. The resulting hash value corresponds to a data block in the cluster, which Oracle then reads or writes on behalf of the issued statement. <strong>Default</strong> is the Oracle internal hash function, otherwise specify the hash expression you want to use.</td>
</tr>
<tr>
<td>How many transaction entries are allowed for each data block in the cluster?</td>
<td>Each transaction that updates a data block requires a transaction entry. <strong>Initial (1-255)</strong>: The initial parameter ensures that a minimum number of concurrent transactions can update a data block, avoiding the overhead of allocating a transaction entry dynamically. <strong>Maximum (1-255)</strong>: The maximum parameter limits concurrency on a data block.</td>
</tr>
<tr>
<td>What is the percent of space reserved for future updates?</td>
<td>Percent Free (0-99): This sets the percentage of a data block to be reserved for possible row updates that are included in the block. The value you set is the percent kept free.</td>
</tr>
<tr>
<td>What is the minimum percentage of used space that Oracle maintains for each data block?</td>
<td>The storage parameter lets you tune performance by minimizing the occurrence of row migration and chaining caused by update operations that extend the length of rows stored on the data block. Percent Used (1-99)</td>
</tr>
<tr>
<td>How large are the cluster's extents?</td>
<td>The unit of space allocated to an object whenever the object needs more space. <strong>Initial Extent</strong> - The initial space extent (in bytes) allocated to the object. <strong>Next Extent</strong> - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.</td>
</tr>
<tr>
<td>Specify the number of free lists</td>
<td>Free lists are lists of data blocks that have space available for inserting rows. Identifying multiple free lists can reduce contention for free lists when concurrent inserts take place and potentially improve the performance of the cluster. <strong>Free Lists</strong>: The default and minimum value is 1; this option should be set higher if multiple processes access the same data block.</td>
</tr>
<tr>
<td>Specify the number of free list groups (specify only if you are using the parallel server option)</td>
<td>This is the number of groups of free lists for the database objects being created.</td>
</tr>
<tr>
<td>Define a default buffer pool for this cluster</td>
<td>Default - Select to retain the default. <strong>Keep</strong> - Select to retain the object in memory to avoid I/O conflicts. <strong>ORACLE 8i ONLY</strong>: Recycle - Select to rid data blocks from memory as soon as they are no longer in use.</td>
</tr>
<tr>
<td>Oracle’s parallel query option</td>
<td>The parallel server query option lets you process queries using many query server processes running against multiple CPUs, which provides substantial performance gains such as reducing the query completion time.</td>
</tr>
<tr>
<td>Choosing Cache</td>
<td><strong>Cache</strong>: This keeps the data block in memory by placing it at the most recently used end. This option is useful for small lookup tables. <strong>No Cache</strong></td>
</tr>
</tbody>
</table>

---
Database Link Wizard for Oracle

A database link specifies a communication path from one database to another. If you’re creating a link to a remote database, a database session is established in the remote database on behalf of the local application request. By creating either a public or private database link, you can determine which schema on the remote database the link will establish connections to by creating fixed, current, and connected database links. By creating a link you can reuse connectivity instructions each time you connect to the remote database.

To Open the Database Link Wizard

1. On the Explorer, find the schema where you want to add the new database link.
2. On the Database Links branch, right-click and select New.

OR

1. On the main toolbar, click Datasource and scroll to Objects
2. Click Database Links and then click New from the toolbar.

The table that follows describes the fields you will encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the database link?</td>
<td>Create a unique name</td>
</tr>
<tr>
<td>Should the database link be made public?</td>
<td>A public link is a link on a local database that's accessible to all users on that database. If you keep the database link private by selecting No, a link is created in a specific schema of the local database and only the owner of the link can use it to access database objects in the corresponding remote database.</td>
</tr>
<tr>
<td>What is the name of the remote user?</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>What is the remote user’s password?</td>
<td>Create a password for the remote user.</td>
</tr>
<tr>
<td>What is the connection string?</td>
<td>Self-explanatory</td>
</tr>
</tbody>
</table>

Directory Wizard for Oracle

A directory object specifies an alias for a directory on the server file system where external binary file LOBs and external table data are located. The wizard completes a CREATE DIRECTORY statement from the information you supply. The Directory Wizard prompts you to name the directory and provide the full-qualified directory path.

To Open the Directory Wizard

1. On the Explorer, find the Datasource where you want to create a directory and expand the Storage node.
2. Right-click Directories and select New.

The table that follows describes the fields you will encounter as you complete the wizard.
After you have created the Directory, you can give other users Read or Write privileges by opening the new Directory on the Directories node, and making changes at the Privileges tab.

Foreign Key Wizard for Oracle

A foreign key value in one table (child table) refers to a primary key value in another table (parent table). For example, the Author Name column in a publisher’s database may be the primary key for a table of addresses that includes the Author Name column. If an author isn’t included in the parent table, you can’t add the address to the dependent address table. So foreign keys enforce referential integrity between tables by verifying the existence of foreign key values in the parent table before letting you insert or update foreign key values in the child table. In other words, a foreign key is an integrity constraint that requires each value in one table’s column to match a value in a related table’s data.

To create a new foreign key using a wizard:

1. Open an object wizard for a foreign key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td></td>
</tr>
<tr>
<td>Table Owner</td>
<td>The owner of the table where the foreign key is being created.</td>
</tr>
<tr>
<td>Table Name</td>
<td>This is the table where the foreign key link originates—the child table.</td>
</tr>
<tr>
<td>Name</td>
<td>Lets you select a constraint name.</td>
</tr>
<tr>
<td></td>
<td>System Generated Name - DB2 automatically generates a name.</td>
</tr>
<tr>
<td></td>
<td>User Specified Constraint Name - You type the name.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enable or disable the foreign key. Enabled ensures that all data modifications to a given table (or tables) satisfy the conditions of the constraints. When disabled, the constraint is temporarily not operational.</td>
</tr>
<tr>
<td>Delete Rule</td>
<td>Select an action:</td>
</tr>
<tr>
<td></td>
<td>NO ACTION - ensures that referenced values cannot be updated or deleted if to do so would violate referential integrity.</td>
</tr>
<tr>
<td></td>
<td>CASCADE permits a referenced row in a child table to be deleted/updated if it is deleted/updated in the parent table. A row in the child table is SET NULL when rows in the parent table are deleted/updated.</td>
</tr>
<tr>
<td><strong>Column Mapping</strong></td>
<td></td>
</tr>
<tr>
<td>Under Referenced Table, choose the Owner and then the Name of the referenced, or parent, table.</td>
<td></td>
</tr>
<tr>
<td>Under the Main Table, select checkboxes corresponding to the columns that are to reference columns in the referenced table. Then, under Referenced Table, select the corresponding column check boxes.</td>
<td></td>
</tr>
</tbody>
</table>
Function Wizard for Oracle

Functions are subroutines that you define and are useful for reusing application logic. You can use functions to determine the best methods for controlling access and manipulation of the underlying data contained in an object. A function returns a value, unlike a procedure, which does not.

To create a new function using a wizard:
1. Open an object wizard for a function.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>

Index Wizard for Oracle

Indexes are optional structures associated with tables. You can create indexes specifically to speed SQL statement execution on a table. When properly used, indexes are the primary means of reducing disk I/O. Indexes are logically and physically independent of the data in the associated table. Unique Indexes guarantee that no two rows of a table have duplicate values in the columns that define the index.

The Index Wizard lets you specify:

- Index owner.
- Table indexed.
- Index name.
- Index properties.
- The table columns that participate in the index.
- The tablespace on which to place the index.
- How Oracle should store the index on blocks.
- Oracle's Parallel Server options.
- How Oracle should allocate additional extents as the index grows.
- Storage parameters for the index.
- The partitioning columns.
• An ordered list of partitions by adding, inserting, editing, or dropping.

**NOTE:** The Index Wizard varies slightly in content based on the version of Oracle to which you are connected.

**NOTE:** To create indexes in your own schema, you need INDEX privileges on the target table. To create indexes in other schema, you need CREATE ANY INDEX privileges.

**NOTE:** For Oracle 8i or later, you can place a unique key constraint on an Index-Organized table.

**TIP:** Index-organized tables take up less storage space and quickly access table rows. Index-organized tables stores rows in primary key order reducing the amount of storage space needed.

**TIP:** An advantage of using index-organized tables is that the tables use less memory because key columns are not duplicated in the table and index. DBArtisan stores the remaining non-key columns in the index structure.
To create a new index using a wizard:

1. Open an object wizard for an index.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Table Owner and Table Name</strong></td>
<td>Choose the owner and name of the table in which the index is being created.</td>
</tr>
<tr>
<td><strong>Owner and Name</strong></td>
<td>Choose the owner and provide the name of the index being created.</td>
</tr>
</tbody>
</table>
| **Index Type** | NONUNIQUE - In a non-unique index, the ROWID is treated as part of the key. Oracle treats a constraint as deferrable.  
UNIQUE - Select if the index is a unique constraint. The values in the indexed columns must be distinct.  
BITMAP - Widely used in data warehousing environments. The environments typically have large amounts of data and ad hoc queries, but a low level of concurrent DML transactions. |
| **No Sort** | Enable this feature if the rows in the table already stored in ascending order. This increases the speed of the index creation process. Oracle does not sort the rows. |
| **Logging** | Enabling logs this operation to the redo file. |
| **Reverse** | Enabling this feature stores the bytes of the index block in reverse order and excludes the ROWID. The ROWID is a globally unique identifier for a row in a database. It is created at the time the row is inserted into a table, and destroyed when it is removed from a table. |
| **Function-Based** | Permits the results of known queries to be returned much more quickly. When you select this option, you are asked for the expression that governs the function-based index you are creating. |
| **No Parallel Execution** | The parallel server query option lets you process queries, using many query server processes, running against multiple CPUs. This option provides substantial performance gains such as reduction of the query completion time.  
After creation, ALTER INDEX for NOPARALLEL execution - when you use multiple query servers and you select this option, the parallel query option remains in place, but parallel processing will be removed. If, for example, multiple users on numerous nodes are modifying the same small set of data, the cost of synchronization from the parallel processing may have an unnecessarily large drag on throughput. |
| **Parallel Degree** | The value you select indicates the number of query server processes that should be used in the operation. |
| **Parallel Instances** | The value you select indicates how you want the parallel query partitioned between the Parallel Servers. |
| **Columns** | From the Column dropdown, select a column for the index and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns. |
Job Queue Wizard for Oracle

Job Queues are built-in mechanisms that let you schedule a variety of SQL-based or command-line driven tasks.

The Job Queue Wizard lets you:
• Specify the PL/SQL code that will run in the job.
• Specify when the job will run, if it will run again, and if it should be enabled to run.

To Open the Job Queue Wizard
1. On the Explorer, find the datasource where you want to create an Oracle job and expand the Instance node.
2. Right-click the Oracle Job Queue node, and select New.

The table that follows describes the fields you may encounter using this wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter the PL/SQL code you would like submitted</td>
<td>Type PL/SQL code or retrieve a previously saved PL/SQL script.</td>
</tr>
<tr>
<td>No Parse/Parse</td>
<td>Parsing can be expensive resource-wise, so think carefully before you choose the parse option. For a parse call, Oracle checks the syntactic and semantic validity of the SQL statement, whether the process has the correct permissions, and allocates a private SQL area for the statement. Oracle will also check to see whether the statement exists in a shared library cache.</td>
</tr>
<tr>
<td>When would you like for the job to begin execution?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>Would you like to have the job run on an ongoing basis?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>Would you like to have the job submitted as disabled?</td>
<td>Self-explanatory.</td>
</tr>
</tbody>
</table>

For more information, see:
Completing a Dialog Box

Library Wizard for Oracle
Libraries are an object type introduced in Oracle8 that represent a call to an operating system shared library cache. After the call is made, libraries can be used by SQL or PL/SQL to link to external procedures or functions. Libraries are only to be used on operating systems that support shared libraries and dynamic linking. Libraries serve as pointers or aliases to physical operating system shared library files and do not exist as a physical object; rather they rely on the physical existence of the files in the external operating system library to which they refer. To access the function or procedures stored in the library, you need execute privileges at the operating system level where the shared library resides.

• To create a library in your own schema, you need CREATE ANY LIBRARY privileges. To use the functions or procedures stored in the library, you need object EXECUTE privileges on the library.

To Open the Library Wizard
1. On the Explorer, find the datasource where you want to create a Library and expand the Schema node.
2. Right-click the Libraries node, and select New.

The table that follows describes the fields you may encounter using this wizard:
A materialized view gives you indirect access to table data by storing a query’s results in a separate schema object. Essentially, a materialized view is a database object that contains the results of a query.

The Materialized View Wizard lets you:

- Specify the materialized view owner and to name the materialized view.
- Specify the materialized view's refresh configuration.
- Place the materialized view on a tablespace and specify the query that should be used to populate the materialized view.
- Specify how Oracle should allocate data blocks to store the materialized view.
- Specify how Oracle should manage the growth of the materialized view.
- Specify if Oracle updates the materialized view, register an existing table, and specify how to populate a materialized view.
- Specify if the data for the materialized view is cached, if you want the updates logged, and to specify a number of threads for a parallel operation.
- Specify rollback segments, and enable query rewrites.

To Open the Materialized View Wizard

1. On the Explorer, find the datasource where you want to create a Materialized View and expand the Schema node.
2. Right-click the Materialized Views node, and select New.

The table that follows describes the fields you may encounter as you complete the wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who owns the materialized view?</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>What is the name of the materialized view?</td>
<td>Self-explanatory</td>
</tr>
</tbody>
</table>
### Required Information

<table>
<thead>
<tr>
<th>How should the materialized view be refreshed?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast - Using the information logged on the materialized view logs or a partition maintenance operation, the refresh applies incremental changes. See Fast Refresh Requirements for more information. Complete - This refresh recalculate...</td>
<td>Fast - Using the information logged on the materialized view logs or a partition maintenance operation, the refresh applies incremental changes. See Fast Refresh Requirements for more information. Complete - This refresh recalculate...</td>
</tr>
</tbody>
</table>

| Choose a refresh mechanism | On Demand - This option requires that all refreshes be manually executed. On Commit - Select to refresh the materialized view whenever Oracle processes a transaction. Only select this option for materialized views on single table aggregates and materialized views containing joins. Automatically - Select to refresh the materialized view automatically. In the On this date: boxes select a time and date, and then select a refresh amount and a unit of time. |

| Where do you want to place the materialized view? | Select the tablespace where you want the materialized view placed. |

| What is the materialized view query? | Type the SQL query to be used to populate and to refresh the materialized view. |

| Select a refresh method | Primary Key - A primary key's values uniquely identify the rows in a table. Changes are propagated according to row changes as identified by the primary key value of the row. Only one primary key can be defined for each table. The primary key of the master table is the basis for this refresh option, which is the default option. ROWID - A globally unique identifier for a row in a database based on the physical row identifiers. A RowID is created at the time the row is inserted into a table, and destroyed when it is removed from a table. ROWID materialized views cannot contain distinct or aggregate functions or GROUP BY subqueries, joins and set operations. |

| How many transaction entries are allowed for each datablock in the materialized view? | A transaction is a logical unit of work that contains one or more SQL statements. Each transaction that updates a data block requires a transaction entry. Initial (1-255) - Ensures that a minimum number of concurrent transactions can update a data block, avoiding the overhead of allocating a transaction entry dynamically. Maximum (1-255) - Limits concurrency on a data block. |

| What is the percent of space reserved for future updates? | Percent Free (0-99) - This sets the percentage of a data block to be reserved for possible row updates that are included in the block. The value you set is the percent kept free. Percent Used (0-99) - Set the amount of space to be used for each datablock. NOTE: The sum of percent free and the percent used cannot exceed 100. |

| What is the minimum percentage of used space that Oracle maintains for each datablock? | The unit of space allocated to an object whenever the object needs more space. An extent is a specific number of contiguous data blocks set aside for storing a specific type of information. Initial Extent (KB) - The initial space extent (in bytes) allocated to the object. Next Extent - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required. |

| How large are the materialized views extents? | Minimum Extents - The appropriate minimum extents value for the object. Maximum Extents - The appropriate maximum extents value for the object. |

| How many extents should be allocated to the materialized view? | Percent Increase - Magnifies how an object grows and can materially affect available free space in a tablespace. Select a value in the corresponding box. |

<p>| Can the materialized view be updated? | Yes/No |</p>
<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want to register a prebuilt table to the view?</td>
<td>Yes/No. This option is particularly useful for registering large materialized views in a data warehousing environment.</td>
</tr>
<tr>
<td>Should the materialized view be immediately filled?</td>
<td>Yes/No: Select Yes if you want the materialized view populated immediately or during the next refresh operation.</td>
</tr>
<tr>
<td>Should data for the materialized view be cached?</td>
<td>Yes/No: Select if you want Oracle to put data you access frequently at the most recently used end of the list in the buffer cache when a full table scan is performed. This option is useful for small lookup tables.</td>
</tr>
<tr>
<td>Do you want updates to be logged?</td>
<td>Yes/No.</td>
</tr>
<tr>
<td>Do you want to specify the number of threads used in a parallel operation?</td>
<td>Parallel processes means that multiple processes work simultaneously to run a single statement. This can cut the amount of time it takes to get a response. Specify the degree of parallelism if you so desire. If you leave the default at 1, the operation will not be “parallelized.”</td>
</tr>
<tr>
<td>Would you like to specify rollback segments to be used for the materialized view refresh?</td>
<td>A rollback segment temporarily stores old data that has changed in a SQL statement transaction until it is committed. The “before” image of the database, as it were. Local Rollback Segment - Default indicates that Oracle will select the rollback segment to use on the local machine. Master Rollback Segment - Specify the remote rollback segment used at the remote master site for the individual materialized view.</td>
</tr>
<tr>
<td>Is the materialized view eligible for query rewrite?</td>
<td>Select to enable the materialized view for query rewrite. Only enable query rewrite if expressions in the statement are repeatable.</td>
</tr>
<tr>
<td>Do you want to partition this materialized view?</td>
<td>Yes/No Partitioning methods available are: Range: Data is mapped to partitions based on ranges of column values. This is the default. Composite: Based on the range method of partitioning, you can create subpartitions within each partition. Hash: Data is distributed evenly over a specified number of partitions. Data need not fit into a logical range. List: You control explicitly how rows map to partitions. List partitions allow you to group and organize unrelated sets of data.</td>
</tr>
<tr>
<td>Do you want to enable Row Movement?</td>
<td>Yes/No Enabling row movement allows you to specify whether Oracle can move a table row when you are compressing a table or performing an update on partitioned data.</td>
</tr>
<tr>
<td>Select the partitioning columns</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Select the subpartitioning method</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Select the subpartitioning columns</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Hash Partitioning methods</td>
<td>None Partition Definition: Specify number of partitions and (optionally) tablespaces Specify individual partitions by name and (optionally) tablespaces</td>
</tr>
</tbody>
</table>
| Create list/ordered list of partitions                   | Self-explanatory Add Partition dialog box may open
### Fast Refresh Requirements

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify number of subpartions</td>
<td>Self-explanatory Click Add, Insert, or Edit to open a dialog box that lets you work with subpartition properties.</td>
</tr>
<tr>
<td>Select the default tablespaces to contain the subpartitions (optional)</td>
<td>Self-explanatory</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>When the Materialized View has:</th>
<th>Only Joins</th>
<th>Joins and Aggregates</th>
<th>Aggregate on a Single Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only Joins</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joins and Aggregates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aggregate on a Single Table</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detail tables only</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Single table only</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Table Appears only once in the FROM list</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>No non-repeating expressions like SYSDATE and ROWNUM</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>No references to RAW or LONG RAW</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>No GROUP BY</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rowids of all the detail tables must appear in the SELECT list of the query</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expressions are allowed in the GROUP BY and SELECT clauses provided they are the same</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aggregates allowed but cannot be nested</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG with COUNT</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SUM with COUNT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARIANCE with COUNT and SUM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STDDEV with COUNT and SUM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WHERE clause includes join predicates which can be ANDed bit not ORed.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No WHERE clause</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>No HAVING or CONNECT BY</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>No subqueries, inline views, or set functions like UNION or MINUS</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Materialized view logs are tables that maintain a history of modifications to the master table, and they are used to refresh simple materialized views. When you create a materialized view log, Oracle automatically creates a log table to track data changes in the master table and a log trigger to maintain the data in the log table. A log can refresh the materialized view incrementally and is therefore often less time-consuming than a complete refresh.

The Materialized View Log Wizard lets you:

- Specify the materialized view log owner and master table.
- Select refresh types and select column filters.
- Specify how Oracle should allocate data blocks to store the materialized view log.
- Specify how Oracle should manage the growth of the materialized view.
- Specify if you want the data for the materialized view log cached, if you want updates logged, and to enable parallel query.
- Specify if you want the log to hold new values.

To Open the Materialized View Log Wizard

1. On the Explorer, find the datasource where you want to create a Materialized View Log and expand the Schema node.
2. Right-click the Materialized View Log node, and select New.

The table that follows describes the fields you may encounter as you complete the wizard:
<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who owns the materialized view log’s master table?</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Which table will serve as the materialized view log’s master table?</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>On which tablespace do you want to place the log?</td>
<td>Self-explanatory</td>
</tr>
</tbody>
</table>
| Which refresh types would you like to use?               | **Primary Key** - The log records changes to the master table based on the primary key of affected rows. A primary key’s values uniquely identify the rows in a table. Changes are propagated according to row changes as identified by the primary key value of the row. Only one primary key can be defined for each table. The primary key of the master table is the basis for this refresh option, which is the default option.  
**ROWID** - The log records changes to the master table based on the RowID of the affected rows. A ROWID is a globally unique identifier for a row in a database based on the physical row identifiers. A ROWID is created at the time the row is inserted into a table, and destroyed when it is removed from a table. ROWID materialized views cannot contain distinct or aggregate functions or GROUP BY subqueries, joins and set operations. |
| Optional: Select any filter column(s) to be recorded in the materialized view log. | A filter column is a column whose values you want to be recorded in the materialized view log for any rows that are changed. You can specify only one primary key, one ROWID, and one filter column list per materialized view log. |
| How many transaction entries are allowed for each data block in the materialized view log? | A transaction is a logical unit of work that contains one or more SQL statements. Each transaction that updates a data block requires a transaction entry.  
Initial (1-255) - Ensures that a minimum number of concurrent transactions can update a data block, avoiding the overhead of allocating a transaction entry dynamically.  
Maximum (1-255) - Limits concurrency on a data block. |
| What is the percent of space reserved for future updates? | **Percent Free** (0-99) - This sets the percentage of a data block to be reserved for possible row updates that are included in the block. The value you set is the percent kept free. |
| What is the minimum percentage of used space that Oracle maintains for each data block? | **Percent Used** (0-99) - Set the amount of space to be used for each datablock.  
**NOTE**: The sum of percent free and the percent used cannot exceed 100. |
| How large are the materialized view log’s extents?        | **Initial Extent** (KB) - The default is the value specified for the tablespace where the materialized view log resides. |
| Should the data for a materialized view log be cached?    | **Yes/No** - Select **Yes** if you want Oracle to put data you access frequently at the most recently used end of the least recently used list in the buffer cache when a full table scan is performed. This option is useful for small lookup tables. No indicates that your most frequently accessed data blocks are put at the least recently used end of the least recently used list of the buffer cache. |
| Do you want updates to be logged?                         | **Yes/No**. The integer is number of parallel threads used in the parallel operation.  
The Parallel server query option lets you process queries using many query server processes running against multiple CPUs. This option provides substantial performance gains such as reduction of the query completion time. |
| Do you want to enable parallel query for the log?         | **Degree**. The integer is number of parallel threads used in the parallel operation.  
The Parallel server query option lets you process queries using many query server processes running against multiple CPUs. This option provides substantial performance gains such as reduction of the query completion time. |
**Outline Wizard for Oracle**

Outlines are a set of results for the execution plan generation of a particular SQL statement. When you create an outline, plan stability examines the optimization results using the same data used to generate the execution plan. That is, Oracle uses the input to the execution plan to generate an outline, and not the execution plan itself.

**NOTE:** To create an outline, you must have CREATE ANY OUTLINE system privileges.

**To create a new outline using a wizard:**

1. Open an object wizard for an outline.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Stored Outline Name</strong> Provide a unique name for the outline.</td>
</tr>
<tr>
<td></td>
<td><strong>Category</strong> The category is a name you want to use to group stored outlines.</td>
</tr>
<tr>
<td></td>
<td>(You can type over the word Default that appears automatically.)</td>
</tr>
<tr>
<td>Definition</td>
<td><strong>Type the SQL statement you want to store as an outline.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> The only SQL statements possible with stored outlines are SELECT, DELETE,</td>
</tr>
<tr>
<td></td>
<td>UPDATE, INSERT...SELECT, and CREATE TABLE...AS SELECT.</td>
</tr>
<tr>
<td>DDL View</td>
<td>**Preview the DDL and if necessary navigate backward through the steps to make</td>
</tr>
<tr>
<td></td>
<td>corrections. Finally, use the <em>Execute</em> button to <em>Create the new object</em>.</td>
</tr>
</tbody>
</table>

**Package Wizard for Oracle**

A package is a collection of related program objects stored together in the database. A package specification or package header, declares variables, constants, etc., that are visible outside the package’s immediate scope. The package body defines objects declared by the specification but are not visible to applications outside the package. Packages contain all the information needed to process SQL statements from a single source file. You can use packages to process and call batches of SQL.

**To Open the Package Wizard**

1. On the Explorer, find the datasource where you want to create a package and expand the **Schema** node.
2. Right-click the **Packages** node, and select **New**.

You’re asked to name an owner for the package and give it a name. When you click Finish, the Packages editor opens to the header tab where you indicate any package specifications and create the package body. For more information, see [Using the Package Bodies Editor (Oracle)](#).
Primary Key Wizard for Oracle

Primary key (constraint)s are a set of table columns that can uniquely identify every row of a table. No fields that are a part of the primary key can have null values, and each table can have only one primary key.

To create a new primary key using a wizard:
1. Open an object wizard for a primary key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td><strong>Table Owner</strong> and <strong>Table Name</strong> Choose the owner and name of the table in which the primary key is being created.</td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong> Provide a name for the primary key being created.</td>
</tr>
<tr>
<td></td>
<td><strong>No Sort</strong> Enable this feature if the rows in the table already stored in ascending order. This increases the speed of the index creation process. Oracle does not sort the rows.</td>
</tr>
<tr>
<td></td>
<td><strong>Logging</strong> Enabling logs this operation to the redo file.</td>
</tr>
<tr>
<td></td>
<td><strong>Reverse</strong> Enabling this feature stores the bytes of the index block in reverse order and excludes the ROWID. The ROWID is a globally unique identifier for a row in a database. It is created at the time the row is inserted into a table, and destroyed when it is removed from a table.</td>
</tr>
<tr>
<td></td>
<td><strong>Validate</strong> Enabling this option indicates that existing data is checked against the constraint when the primary key is enabled. Leaving it disabled indicates that only new data is to be checked against the constraint.</td>
</tr>
<tr>
<td></td>
<td><strong>Deferrable</strong> Dictates whether constraint checking can be deferred until the end of a transaction.</td>
</tr>
<tr>
<td></td>
<td><strong>Deferred</strong> This option is enabled only if you enabled the Deferrable option. Select IMMEDIATE to have the constraint checked at the end of every DDL statement. Select DEFERRED to have the constraint checked only at the end of a transaction.</td>
</tr>
<tr>
<td></td>
<td><strong>Enabled</strong> Enables or disables the primary key.</td>
</tr>
<tr>
<td>Columns</td>
<td>From the Column dropdown, select a column for the index and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
</tr>
</tbody>
</table>
## Storage

<table>
<thead>
<tr>
<th>Settings and tasks</th>
<th>Data Block Storage group</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Storage</strong></td>
<td>Select the DEFAULT Tablespace only if you are creating a local partitioned primary key and want the partitions in the same tablespace as the partitions in the underlying table. (Each partition of a local primary key is associated with one partition of the table. Oracle can then keep the primary key partitions in sync with table partitions.)</td>
</tr>
<tr>
<td></td>
<td>A transaction entry is needed for each INSERT, UPDATE, DELETE, etc. statement that accesses one or more rows in the block. Transaction entries in many operating systems require approx. 23 bytes.</td>
</tr>
<tr>
<td></td>
<td><strong>Percent Free</strong> identifies how much space you want to allocate for new rows or updates to existing rows.</td>
</tr>
<tr>
<td></td>
<td><strong>Initial Transactions</strong> ensures that a minimum number of concurrent transactions can update a primary key block, avoiding the overhead of allocating a transaction entry dynamically.</td>
</tr>
<tr>
<td></td>
<td><strong>Maximum Transactions</strong> limits concurrency on a primary key block.</td>
</tr>
</tbody>
</table>

## Extents group

<table>
<thead>
<tr>
<th>Settings and tasks</th>
<th>Extents group</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extents</strong></td>
<td>An extent is the unit of space allocated to an object whenever the object needs more space.</td>
</tr>
<tr>
<td></td>
<td><strong>Initial Extent</strong> - The initial space extent (in bytes) allocated to the object.</td>
</tr>
<tr>
<td></td>
<td><strong>Next Extent</strong> - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.</td>
</tr>
<tr>
<td></td>
<td><strong>Percentage Increase</strong> - Lets you type the percentage.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> You should be careful when setting Percent Increase because it magnifies how an object grows and, therefore, can materially affect available free space in a tablespace.</td>
</tr>
<tr>
<td></td>
<td><strong>Minimum Extents</strong> - For a dictionary managed tablespace, this is the total number of extents to be allocated when the index is first created. For a locally managed tablespace, this is simply the initial amount of space allocated.</td>
</tr>
<tr>
<td></td>
<td><strong>Maximum Extents</strong> - For a dictionary managed tablespace, this is the total number of extents that can ever be allocated to the index. In a locally managed tablespace, the database will automatically manage the extents.</td>
</tr>
</tbody>
</table>

## Freelists group

<table>
<thead>
<tr>
<th>Settings and tasks</th>
<th>Freelists group</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Freelists</strong></td>
<td><strong>Free lists</strong> let you manage the allocation of data blocks when concurrent processes are issued against the primary key. You can potentially improve the performance of the primary key by identifying multiple free lists, which can reduce contention for free lists when concurrent inserts take place.</td>
</tr>
<tr>
<td></td>
<td>The default and minimum value is 1. You should increase this number if multiple processes access the same data block.</td>
</tr>
<tr>
<td></td>
<td><strong>Free List Groups</strong> is the number of groups of free lists.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> This option is only applicable for the parallel server option.</td>
</tr>
</tbody>
</table>

## Buffer Pool

<table>
<thead>
<tr>
<th>Settings and tasks</th>
<th>Buffer Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Buffer Pool</strong></td>
<td><strong>DEFAULT</strong> - Choose this if you want to use the default bufferpool.</td>
</tr>
<tr>
<td></td>
<td><strong>KEEP</strong> - Use this to retain the object in memory to avoid I/O conflicts. This type of bufferpool stores frequently referenced data blocks in a separate cache.</td>
</tr>
<tr>
<td></td>
<td><strong>RECYCLE</strong> - Select this option to save cache space by ridding data blocks from memory as soon as they are no longer in use.</td>
</tr>
</tbody>
</table>

## Partition

<table>
<thead>
<tr>
<th>Settings and tasks</th>
<th>Partition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Partition</strong></td>
<td>Clicking Create Partition opens the Add, Insert, or Modify Partition wizard.</td>
</tr>
</tbody>
</table>

## DDL View

<table>
<thead>
<tr>
<th>Settings and tasks</th>
<th>DDL View</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DDL View</strong></td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to Create the new object.</td>
</tr>
</tbody>
</table>
Procedure Wizard for Oracle

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly. Procedures do not return values, unlike functions.

**NOTE:** To create a procedure in your own schema, you need CREATE PROCEDURE privileges. To create a procedure in someone else's schema, you need CREATE ANY PROCEDURE privileges.

To create a new procedure using a wizard:

1. Open an object wizard for a procedure.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td><strong>Owner</strong> Select the owner of the procedure.</td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong> Provide a name for the procedure</td>
</tr>
<tr>
<td>Definition</td>
<td><strong>Complete the CREATE PROCEDURE outline provided by typing or pasting the body of</strong></td>
</tr>
<tr>
<td></td>
<td><strong>the procedure. Finally, use the Execute buttons to Create the new object.</strong></td>
</tr>
</tbody>
</table>

Profile Wizard for Oracle

Profiles are a mechanism for allocating system and database resources to users. In essence, a profile is a set of limits with a name attached to them. If the profile is active, Oracle will limit use and resources to the limits defined by the profile.

The Profile Wizard lets you:

- Name the profile.
- Set composite limit.
- Set session limits for SGA shared pool.
- Set limits on total connection time per session and Idle time per session.
- Set limits on concurrent sessions per user, CPU time per session, and data blocks read per session.
- Set limits on CPU time per call, and number of data blocks read for a call to process an SQL statement.
- Set the number of failed login attempts, and the days an account locks.

**NOTE:** To create a profile, you need the CREATE PROFILE system privilege.

The Default option is subject to the limit specified for that particular resource. The default profile initially permits unlimited resources. Limits to the default profile can be made using an Alter statement.

The Unlimited option allows the user with that profile to take use unlimited amounts of that resource.

The Other options take on the resource limits that you indicate.

To Open the Profile Wizard

1. On the Explorer, find the datasource where you want to create a procedure and expand the Security node.
2. Right-click the Profile node, and select New.
The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the profile?</td>
<td>Write a name that’s 30 characters or less</td>
</tr>
<tr>
<td>What is the composite limit on resources per session?</td>
<td>You can set a single composite limit for all resource limits in a profile in addition to setting specific resource limits for a profile. Explicit and composite limits can peaceably coexist. The limit that is reached first stops the session’s activity. Service units are a weighted sum of CPU per session, connect time, logical reads per session, and private SGA. Default/Unlimited/Other (service units)</td>
</tr>
<tr>
<td>What is the limit on the amount of private space a session can allocate in the shared pool of the SGA?</td>
<td>This limit only applies if you are using Shared Server architecture. Default/Unlimited/Other (KB)</td>
</tr>
<tr>
<td>What is the limit on the total connection time per session?</td>
<td>The total elapsed time limit for a session Default/Unlimited/Other (Minutes)</td>
</tr>
<tr>
<td>What is the limit on idle time per session?</td>
<td>Permitted periods of continuous inactive time during a session, expressed in minutes. Long-running queries and other operations are not subject to this limit. Default/Unlimited/Other (Minutes)</td>
</tr>
<tr>
<td>What is the limit on concurrent sessions per user?</td>
<td>Default/Unlimited/Other</td>
</tr>
<tr>
<td>What is the limit on CPU time per session?</td>
<td>Default/Unlimited/Other (Hundredths of a Second)</td>
</tr>
<tr>
<td>What is the limit on data blocks read per session?</td>
<td>Default/Unlimited/Other</td>
</tr>
<tr>
<td>What is the limit on CPU time per call?</td>
<td>The limit on a parse, execute, or fetch. Default/Unlimited/Other (Hundredths of a Second)</td>
</tr>
<tr>
<td>What is the limit on the number of data blocks read for a call to process a SQL statement?</td>
<td>This is the number of logical reads per call. Default/Unlimited/Other</td>
</tr>
<tr>
<td>How many failed login attempts will be allowed before an account is locked?</td>
<td>Default/Unlimited/Other</td>
</tr>
<tr>
<td>How long will an account be locked after the specified number of failed login attempts?</td>
<td>Default/Unlimited/Other (Days)</td>
</tr>
<tr>
<td>What is the lifetime of the password?</td>
<td>Default/Unlimited/Other (Days)</td>
</tr>
<tr>
<td>How many days must pass before a password can be reused?</td>
<td>Default/Unlimited/Other (Days)</td>
</tr>
<tr>
<td>How many password changes are required before the current password can be reused?</td>
<td>Default/Unlimited/Other</td>
</tr>
<tr>
<td>What is the grace period allowed for a password to be changed without expiring?</td>
<td>Default/Unlimited/Other (Days)</td>
</tr>
</tbody>
</table>
**Redo Log Group Wizard for Oracle**

A redo log makes it possible to replay SQL statements because a redo log stores all changes made to a database as they occur. If something unanticipated happens to a datafile, you can restore a backed-up datafile and use the redo log to bring the datafile back to the state it was in before the unanticipated activity. The redo log can consist of two parts: the online redo log and the archived redo log. In archive log mode, redo log files will not be overwritten unless they have been archived somewhere. Each log is a member of the group.

**NOTE:** Redo should not be confused with Undo, which is a transaction based process.

If you find that you have problems with redo log group availability and Oracle consistently has to wait for redo log groups to become free, you can create additional redo log groups. DBArtisan lets you create additional redo log groups to ensure that there is always a recycled redo log group available.

The Redo Log Group Wizard lets you:

- Assign a number to the new redo log group and determine the file size for the group members.
- Choose a thread for the redo log group for Oracle Parallel Server.
- Add redo log members with the Add Redo Log Member dialog box.

**To Open the Redo Log Wizard**

1. On the **Explorer**, find the datasource where you want to create a redo log and expand the **Storage** node.
2. Right-click the **Redo Log Groups** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want to use Oracle Managed Files (OMF) for the redo log group? You must be set up to use OMF to use this option</td>
<td>Lets you specify to use Oracle Managed Files to automatically create the redo log group. Oracle automatically creates and removes operating system files as you create and drop storage objects. <strong>NOTE:</strong> If you select OMF, you do not have to enter a redo log group number or file size.</td>
</tr>
<tr>
<td>What is the group number for the redo log group?</td>
<td>When numbering your redo log groups, it is better not to skip numbers (e.g., 5, 10, 15). Skipping numbers causes Oracle to consume extra space in the control files of the database.</td>
</tr>
<tr>
<td>What file size should the group’s redo log members be? Note: All members of the group must be the same. With OMF, you can let Oracle assign a size or enter it manually.</td>
<td>size in KB or MB</td>
</tr>
</tbody>
</table>
Role Wizard for Oracle

As you complete the Role Wizard, DBArtisan constructs the necessary CREATE ROLE statement from the information that you have supplied. The Role Wizard lets you specify a name for the role and whether or not the role should be identified.

**NOTE:** To create a role, you need the CREATE ROLE system privilege.

**To Open the Role Wizard**

1. On the Explorer, find the datasource where you want to create a role and expand the Security node.
2. Right-click the Roles node, and select New.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To which thread should the redo log group be added?</td>
<td><strong>OPTIONAL:</strong> If you are using Oracle Parallel Server, type the thread. A thread is an operating system component that allows the logic of multiuser applications to be performed as several separate, asynchronous execution paths. Threads help ensure that work is being performed for some user connections even when other connections are blocked (for example, when waiting for a disk read or write operation to complete).</td>
</tr>
<tr>
<td>Add redo log group members</td>
<td>The name of each group member should be a file name. For example, D:\ORACLE\ORADATA\FINANCEDB\REDO002.LOG is a valid group member name. If you do not specify a full path, DBArtisan creates files in either the default or current directory of the database server.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).
Rollback Segment Wizard for Oracle

Rollback segments manage all transactions in your Oracle databases. A transaction is a read, modify, and write cycle for an Oracle database. A rollback entry is made for all transactions unless a particular clause is specified. So, a rollback segment is a transaction log that consists of a pre-update image value and transaction status, among other things. The rollback segments maintain read consistency among concurrent users in a database and if the transaction fails for any reason, the old image is taken from the rollback segment. By maintaining a history of data changes, rollback segments can rollback uncommitted transactions so that data is rolled back to the prior state. SYS owns all rollback segments no matter who created them and are not accessible to users, just Oracle.

Oracle, it should be mentioned, strongly recommends that you use automatic undo management to simplify managing databases. Tuning rollback segments is a manual process that has largely been deprecated by Oracle, but it is supported for backward compatibility reasons.

The Rollback Segment Wizard lets you:

- Name the rollback segment and to place it online or off-line.
- Place the rollback segment on a tablespace.
- Specify the initial next and optimal extent size as well a the minimum and maximum number of extents that should be allocated to the rollback segment.

**NOTE:** This wizard is not available if auto-UNDO management is enabled.

**TIP:** Make sure enough rollback segments exist on a database to handle the imposed workload. One rule of thumb is to create one rollback segment for every four concurrent users.

To Open the Rollback Segment Wizard

1. On the **Explorer**, find the datasource where you want to create a rollback segment and expand the **Storage** node.

2. Right-click the **Roleback Segments** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the rollback segment?</td>
<td>Self-explanatory.</td>
</tr>
</tbody>
</table>
| Should this rollback segment be made public?               | Yes - A public rollback segment can be brought online by any instance in a parallel server. Public rollback segments form a pool of rollback segments that can be used by any instance that needs one. 
No - This is the default. A private rollback segment can only be acquired by the instance specifying the segment in its initialization file. |
| Do you want to place the rollback segment to be online following its creation? | Online - To be useful, a rollback segment must be online. 
Offline - You may want to take rollback segments offline if you want to take a tablespace offline and it contains rollback segments that you want to keep from being used. |
| On which tablespace do you want to place this rollback segment? | Self-explanatory. 
Oracle suggests that you create one or more tablespaces specifically to hold all rollback segments. This way, the data contained in the rollback segments is held apart from other data types. |
Sequence Wizard for Oracle

Sequences are programmable database objects that provide numbers in sequence for input to a table. A sequence can be used to automatically generate primary key values for tables. Once defined, a sequence can be made available to many users. When you create a sequence, you can define its initial value, increment interval, and maximum value.

The Sequence Wizard lets you:

- Specify the name and owner of the sequence.
- Set both the value of the sequence, and an interval and ranges for incrementing it.
- Cache the sequence, cycle the sequence when it reaches its minimum or maximum values, and guarantee that Oracle generates sequence numbers in the order of request.

**NOTE:** To create a sequence, it must belong to your schema or you need CREATE SEQUENCE privilege.

To Open the Sequence Wizard

1. On the Explorer, find the datasource where you want to create a rollback segment and expand the **Schema** node.
2. Right-click the **Sequences** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who owns the sequence?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>What is the sequence name?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>What is the first sequence number to be generated?</td>
<td>Start with: Pick an integer.</td>
</tr>
<tr>
<td>What is the interval between sequence numbers?</td>
<td>Increment by: Positive numbers will generate ascending values, and a negative number will generate descending numbers</td>
</tr>
<tr>
<td>What is the sequence’s minimum value?</td>
<td>The default is 1 for ascending sequences; This integer value can have 28 or fewer digits. None Minimum value - Identify how low the sequence can go.</td>
</tr>
</tbody>
</table>
Snapshot and Snapshot Log Wizards for Oracle

Oracle has replaced the snapshot functionality with materialized views. Refer to the Materialized View Wizard for Oracle and Materialized View Log Wizard for Oracle.

Synonym Wizard for Oracle

Synonyms are alternate names for database objects to be used as a reference by users or applications. A synonym offers you security and convenience so that you can refer to an object without revealing who owns it or what database it belongs to. Synonyms Depending on the platform, you can define synonyms on tables, views, sequences, procedures, functions, packages, and materialized views. If an underlying object needs to be renamed or moved, it’s easy enough to redefine the synonym without having to modify any applications based on the synonym.

The Synonym Wizard lets you:

- Specify the object type to be referenced by the synonym.
- Identify the base database object for which you are creating the synonym.

NOTE: To create a private synonym, you need CREATE SYNONYM privileges. To create a public synonym, you need CREATE PUBLIC SYNONYM privileges.
To create a new synonym using a wizard:

1. Open an object wizard for a synonym.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td><strong>Owner and Name</strong> Choose the owner and provide a name for the synonym being created.</td>
</tr>
<tr>
<td></td>
<td><strong>Referenced Object Owner</strong> Select the owner of the object to which the synonym is to refer.</td>
</tr>
<tr>
<td></td>
<td><strong>Referenced Object Type</strong> Select the type of the object to which the synonym is to refer.</td>
</tr>
<tr>
<td></td>
<td><strong>Referenced Object Name</strong> Select the specific object to which the synonym is to refer.</td>
</tr>
<tr>
<td>Database Link</td>
<td>If the object resides on a remote database, select a database link.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

### Table Wizard for Oracle

Tables are the most basic data storage units for Oracle. As you might expect, data is stored in rows and columns. By completing the Table Wizard, DBArtisan constructs the necessary CREATE TABLE statement from the information that you supply. The Table Wizard varies slightly in content based on the version of Oracle on the target datasource. But in all cases, you name columns and determine column width or precision and scale depending on the column’s data type. A row collects the column information that corresponds to a single record.

You can set rules for the columns to live by, and these are called integrity constraints. For example, if you select NOT NULL, that column will have to have a value in each row.

Also, before beginning, consider what kind of table you want to create as the wizard will ask you to choose:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heap organized table</td>
<td>This is a basic table where data is stored as an unordered collection, i.e., heap.</td>
</tr>
<tr>
<td>Index-organized table</td>
<td>A B-tree index structure stores data, sorted by primary key. Nonkey column values are stored too.</td>
</tr>
<tr>
<td>Partitioned table</td>
<td>Data is broken down into smaller, more manageable pieces called partitions or subpartitions. Each partition can be managed individually and operate independent of the other partitions.</td>
</tr>
<tr>
<td>Clustered table</td>
<td>This is a table that, once created, is part of a cluster. A cluster is a group of tables that share some data blocks and columns and are often used together. To create a cluster, use the Cluster Wizard for Oracle.</td>
</tr>
</tbody>
</table>

The Table Wizard lets you:

- Specify the table owner, name the table and place it on a tablespace.
- Specify partitioning.
- Specify how Oracle should allocate data blocks to store the table.
- Specify Oracle’s Parallel Server options.
• Provide a table comment and/or description.

**NOTE:** The table wizard panels differ depending on what options you select.

**NOTE:** To simplify the process of creating a table, the Table Wizard focuses on creating the basic table definition with a primary key constraint. After you create the basic table definition you can add unique and foreign keys to the table on the Constraints Tab of the Tables Editor.

**To create a new table using a wizard:**

1. **Open an object wizard** for a table.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Select the owner of the table.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the table</td>
</tr>
<tr>
<td>Cache</td>
<td>Enabling this feature keeps a block in memory by placing it at the most recently used end. This option is useful for small lookup tables.</td>
</tr>
<tr>
<td>Row Movement</td>
<td>Enabling this option permits the migration of a row to a new partition if its key is updated.</td>
</tr>
<tr>
<td>Parallel Degree</td>
<td>A value indicating the number of query server processes that should be used in the operation.</td>
</tr>
<tr>
<td>Parallel Instances</td>
<td>A value indicating how you want the parallel query partitioned between the Parallel Servers.</td>
</tr>
</tbody>
</table>
| Physical group | Choose a Row Organization of INDEX, HEAP, or EXTERNAL.  
If you chose INDEX or HEAP, enable or disable Logging.  
If you chose EXTERNAL, provide an External Type, Default Directory, Access Parameters, Reject Limit, and Location. |
| Logging | Redo logs minimize the loss of data in the event that an uncontrolled shutdown happens. |
| **Columns** | For each column in the table, click the New button to create a column and provide a name for the column. Then, in the Column Attributes area, provide details for the column.  
Use the Delete button to drop a selected column. |
| **Indexes** (not available with a Row Organization of EXTERNAL) | Click Add to open the Index Wizard for Oracle. |
| **Constraints** (not available with a Row Organization of EXTERNAL) | Selecting a constraint type and clicking Add opens the object wizard for that object type. For details see:  
Primary Key Wizard for Oracle  
Unique Key Wizard for Oracle  
Foreign Key Wizard for Oracle |
## Storage (not available with a Row Organization of EXTERNAL)

**Data Block Storage** group
- **Select the DEFAULT Tablespace** only if you want the partitions in the same tablespace as the partitions in the underlying table.
- **Percent Free** identifies how much space you want to allocate for new rows or updates to existing rows.
- **Initial Transactions** ensures that a minimum number of concurrent transactions can update a primary key block, avoiding the overhead of allocating a transaction entry dynamically.
- **Maximum Transactions** limits concurrency on a primary key block.

## Extents group
- An extent is the unit of space allocated to an object whenever the object needs more space.
  - **Initial Extent** - The initial space extent (in bytes) allocated to the object.
  - **Next Extent** - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.
  - **Percentage Increase** - Lets you type the percentage.
  - **NOTE:** You should be careful when setting Percent Increase because it magnifies how an object grows and, therefore, can materially affect available free space in a tablespace.
  - **Minimum Extents** - For a dictionary managed tablespace, this is the total number of extents to be allocated when the index is first created. For a locally managed tablespace, this is simply the initial amount of space allocated.
  - **Maximum Extents** - For a dictionary managed tablespace, this is the total number of extents that can ever be allocated to the index. In a locally managed tablespace, the database will automatically manage the extents.

## Freelist group
- **Free lists** let you manage the allocation of data blocks when concurrent processes are issued against the primary key. You can potentially improve the performance of the primary key by identifying multiple free lists, which can reduce contention for free lists when concurrent inserts take place.
  - The default and minimum value is 1. You should increase this number if multiple processes access the same data block.
  - **Free List Groups** is the number of groups of free lists.
  - **NOTE:** This option is only applicable for the parallel server option.

## Buffer Pool
- **DEFAULT** - Choose this if you want to use the default bufferpool.
- **KEEP** - Use this to retain the object in memory to avoid I/O conflicts. This type of bufferpool stores frequently referenced data blocks in a separate cache.
- **RECYCLE** - Select this option to save cache space by ridding data blocks from memory as soon as they are no longer in use.

## IOT Properties (not available with a Row Organization of EXTERNAL)
- Provide compression and space details for an index-organized table.

## Partition (not available with a Row Organization of EXTERNAL)
- Prior to working with partitions, you should be familiar with the material in [DBArtisan and Oracle Partitioning](#).
- Click Create Partition to [Partition a table](#).

## Comment
- Optionally, add a comment for the table.

## Permissions
- Set up the user permissions for this table.
DBArtisan and Oracle Partitioning

Partitioning your tables lets you get around the problem of supporting large tables. Partitioning lets you break large tables into smaller pieces, which are called partitions. Partitions make the data in your table easier to manage and analyze. Your SQL statements can access the partitions rather than the entire table. Partitions are most useful in data warehouse applications, which store large amounts of data.

The table below describes the types of partitions in Oracle:

<table>
<thead>
<tr>
<th>Partition Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Use range partitioning to map rows to partitions based on ranges of column values. This type of partitioning is useful when dealing with data that has logical ranges into which it can be distributed; for example, months of the year. Performance is best when the data evenly distributes across the range. If partitioning by range causes partitions to vary dramatically in size because of unequal distribution, you may want to consider one of the other methods of partitioning.</td>
</tr>
<tr>
<td>Hash</td>
<td>Use hash partitioning if your data does not easily lend itself to range partitioning, but you would like to partition for performance and manageability reasons. Hash partitioning provides a method of evenly distributing data across a specified number of partitions. Rows are mapped into partitions based on a hash value of the partitioning key. Creating and using hash partitions gives you a highly tunable method of data placement, because you can influence availability and performance by spreading these evenly sized partitions across I/O devices (striping).</td>
</tr>
<tr>
<td>Composite</td>
<td>In Oracle 8i, Oracle introduced both hash and composite partitioning. Hash partitions the table according to a hash function. Composite partitions use both range and hash types, first partitioning the data by a range of values, and then further dividing the partitions into subpartitions by way of a hash function. This option is not available for index-organized tables.</td>
</tr>
<tr>
<td>List</td>
<td>Use list partitioning when you require explicit control over how rows map to partitions. You can specify a list of discrete values for the partitioning column in the description for each partition. This is different from range partitioning, where a range of values is associated with a partition, and from hash partitioning, where the user has no control of the row to partition mapping.</td>
</tr>
</tbody>
</table>

Partition a table

The Add partition wizard lets you set up partitions for a table. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Select a Partition Type and optionally, a Subpartition Type.</td>
</tr>
<tr>
<td>Columns</td>
<td>For each column, click the New button and select a name from the Column dropdown. Use the Delete button to drop a selected column.</td>
</tr>
<tr>
<td>Subpartition Columns (only available with a Partition Type of RANGE and Subpartition Type of HASH or LIST)</td>
<td>For each column, click the New button and select a name from the Column dropdown. Use the Delete button to drop a selected column.</td>
</tr>
</tbody>
</table>
Add a partition definition

Use the following table as a guide in completing the settings in this wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpartitions (only available with a</td>
<td>Specify a <strong>Default number of partitions</strong>. For each partition, click the New</td>
</tr>
<tr>
<td>Subpartition Type of HASH)</td>
<td>button and then select a tablespace from the dropdown.</td>
</tr>
<tr>
<td>Range Definitions (only available with a</td>
<td>Click the New button to <strong>Add a partition definition</strong>.</td>
</tr>
<tr>
<td>Partition Type of RANGE)</td>
<td></td>
</tr>
<tr>
<td>Partition Definition (only available with</td>
<td>To specify a partition method other than <strong>None</strong>, take one of the following</td>
</tr>
<tr>
<td>a Partition Type of HASH)</td>
<td>actions: (1) Select the <strong>Number Of Partitions</strong> radio box, specify the **Number</td>
</tr>
<tr>
<td></td>
<td>Of Partitions**, and for each partition, click the New button and choose a</td>
</tr>
<tr>
<td></td>
<td>tablespace from the dropdown, or (2) select the <strong>By Partition Name</strong> radio box</td>
</tr>
<tr>
<td></td>
<td>and for each partition, click the New button provide a name and then choose a</td>
</tr>
<tr>
<td></td>
<td>tablespace from the dropdown.</td>
</tr>
<tr>
<td>List Definitions (only available with a</td>
<td>Click New to <strong>Add a partition definition</strong>.</td>
</tr>
<tr>
<td>Partition Type of List)</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** When you split a range-list partition, you cannot specify the new partitions’ subpartition information.

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition Definition</td>
<td>Name Provide a name.</td>
</tr>
<tr>
<td></td>
<td>Tablespace Select a tablespace from the dropdown</td>
</tr>
<tr>
<td></td>
<td>Logging Enable or disable logging.</td>
</tr>
<tr>
<td>Subpartitions</td>
<td>To specify a partition method other than <strong>None</strong>, select the <strong>By Subpartition Name</strong> radio box and for each partition, click the New button to open a dialog that lets you provide subpartition values. <strong>NOTE:</strong> When you split a range-list partition, you cannot specify the new partitions’ subpartition information.</td>
</tr>
<tr>
<td>Storage</td>
<td>Provide or select Data Block Storage, Extents, Freelists, and Buffer Pool values.</td>
</tr>
</tbody>
</table>

**Tablespace Wizard for Oracle**

Tablespaces are logical storage structures that act as partitions for the database. Each tablespace consists of one or more datafiles which are the physical structures holding the data. You can create a tablespace to store table data and other objects related to table performance such as indexes or large object data. Tablespaces are used to manage large complex databases. Once you have created a tablespace, you can place objects on it.

The Tablespace Wizard lets you:

- Name the tablespace, and specify space management.
- Specify what types of objects are stored on the tablespace, and place the tablespace online or offline.
- Add the datafiles that comprise the tablespace and specify the parameters for the datafiles.
- Specify how Oracle should manage the growth of the tablespace.
Important Notes

- For auto-UNDO management to be in effect, set init.ora parameter to undo_management. When set to MANUAL (the default), it disables auto-UNDO management. When set to AUTO, auto-UNDO management is enabled.

- To determine if the undo_management parameter is set to AUTO, use the following query:
  ```sql
  SELECT VALUE
  FROM   SYS.V_$PARAMETER
  WHERE NAME = 'undo_management'
  
  NOTE: This parameter cannot be set dynamically via the ALTER SYSTEM or ALTER SESSION.
  ```

For users using a version earlier than Oracle 8i and locally managed tablespaces, there are manual methods you can use to assist in the fight against tablespace fragmentation. They include:

- Setting PCTINCREASE to zero for all tablespaces and objects to promote same-sized extents.
- Specifying equal-sized allotments for your INITIAL and NEXT object storage parameters.
- Grouping objects with like growth and storage needs together in their own tablespaces.

One of the best ways to avoid fragmentation in a tablespace is to pre-allocate the space that your objects will use. If possible, plan for one to two years’ growth for each object and allocate your space accordingly. Having initial empty objects will not affect table scan times as Oracle only scans up to the high-water mark (the last used block) in a table.

Of all your tablespaces, you want to avoid fragmentation problems in your SYSTEM tablespace the most as this is the major hotbed tablespace for Oracle activities. The easiest way to avoid this is to not allow any user (even the default DBA ID's SYS and SYSTEM) to have access to it. There are three ways to do this:

- Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
- Ensure no user has a quota set for SYSTEM.
- Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

To Open the Tablespace Wizard

1. On the Explorer, find the datasource where you want to create a tablespace and expand the Storage node.
2. Right-click the Tablespace node, and select New.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the tablespace</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>What type of tablespace would you like to create?</td>
<td><strong>Permanent</strong>: Contains schema objects you want to keep on an ongoing basis. The objects are stored in datafiles. <strong>Temporary</strong>: Schema objects will last only as long as your session continues. <strong>Undo</strong>: This is a kind of permanent tablespace that holds undo data if your database is operating in automatic undo mode. Oracle recommends the automatic undo mode as the wiser choice than using rollback segments to undo.</td>
</tr>
<tr>
<td>Do you want space management to be performed through the data dictionary or managed locally?</td>
<td><strong>Dictionary Managed</strong>: The dictionary tables in the SYS schema will track allocated and free extents. Oracle does not recommend this option because it can be slow. <strong>Locally Managed</strong>: Each tablespace manages its own free and used space so locally managed tablespaces are generally much more efficient.</td>
</tr>
<tr>
<td>Required Information</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **Will the extents be allocated by the system or do you want to choose a uniform extent size?** | **Autoallocate extents:** Users cannot specify an extent size because the tablespace is system managed. This option is not available for temporary tablespaces.  
**Uniform extents:** The default extent size is 1 megabyte. Extents for temporary tablespaces are uniform in size, but if you want to select a size other than the default, you can. Uniform is not an option for undo tablespaces. |
| **If you want to set a size for uniform extents, then enter it below, otherwise the default size will be used** | The default extent size is 1 megabyte. |
| **Should the tablespace be placed online following its creation?** | **NOTE:** This option is not available for temporary tablespaces.  
**Online:** This specifies the tablespace will be available to users immediately after it is created and while the database is up. This is the default.  
**Offline:** This is a tablespace that’s unavailable to users even when the database is open. The datafiles in the tablespace are offline too. |
| **Do you want redo logging when Schema/Data is modified?** | Self-explanatory. Logging is the default. The option is only available for permanent tablespaces. |
| **Add Datafiles for the tablespace** | See [Add or Modify Datafile](#) |
| **What type of segment management should be used?** | This is an option available only to permanent, locally-managed tablespaces.  
**Automatic (Oracle bitmaps):** Free space segments will be managed by bitmap. Subsequent storage specifications for objects in the tablespace will be ignored if you make this selection (which Oracle recommends).  
**Manual (Oracle free lists):** The database will manage the segment free space using free lists. |
| **What block size should the tablespace use?** | With this option, you can specify a nonstandard block size. The integer must correspond with the setting of one DB-nK_CACHE_SIZE parameter setting. This isn’t available if you are creating a temporary tablespace. |
| **Specify the default storage parameter for any objects created on the tablespace** | **Initial extent:** The default value is the size of 5 data blocks. For manually managed space segments, the minimum value is 2 data blocks plus 1 data block for each free list group you specify.  
**Next extent:** This is the size of the subsequent extent to be allocated after the initial extent is filled.  
**Minimum extent:** The total number of extents to allocate at the object’s creation. The default is 1, which is equivalent to the initial extent. |
| **How many extents should be allocated to objects placed on the tablespace** | **Minimum extents:** The total number of extents to allocate at the object’s creation. The default is 1, which is equivalent to the initial extent.  
**Maximum extents:** This is the total number of extents, including the first, that can be allocated to an object. |
| **What should be the growth rate for sizing additional extents** | **Percent increase:** The percent by which the third and following extents grow beyond the prior extent. Thus, if give a value of 20, the next extent will be 20% larger than the one that preceded it. |

**Trigger Wizard for Oracle**

A trigger is a special type of procedure that automatically fires when defined data modification operations (insert, update, or delete) occur on a designated table or view. Triggers fire after an insert, update or delete, but belong to the same transaction as the data modification operation. A stored procedure, on the other hand, is activated explicitly by the user or application (or trigger). Triggers should only be used for centralized, global operations that should be initiated by the triggering statement and should not be based on whether a particular user or database application issues the statement.
The Trigger Wizard lets you:

- Specify the trigger name and the table on which it fires.
- Specify the granularity of the trigger.
- Define the trigger body.
- Specify the trigger timing, event and applicable table columns for an update trigger.
- Define the PL/SQL block that should execute when the trigger fires.

**Important Notes**

- To create triggers in your own schema, you need CREATE TRIGGER privileges. To create triggers in other schemas, you need CREATE ANY TRIGGER privileges.

To create a new trigger using a wizard:

1. Open an object wizard for a trigger.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td><strong>Parent Type, Parent Owner, and Parent Name</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Enabled</strong></td>
</tr>
<tr>
<td><strong>Trigger Timing</strong></td>
<td><strong>BEFORE</strong>: These triggers serve as extensions to the constraint subsystem and are most often used to validate input data, generate values for newly inserted rows, and read from other tables for cross-reference purposes. <strong>Note</strong>: Before triggers must be created as a For Each Row.</td>
</tr>
<tr>
<td></td>
<td><strong>AFTER</strong>: Such a trigger is run after the integrity constraint validations; they can be used to modify operations in the database or be used for activities beyond the database, like supporting an alert notification.</td>
</tr>
<tr>
<td><strong>Fire On Insert, Fire On Update, and Fire On Delete</strong></td>
<td>Enable the events that fire the trigger.</td>
</tr>
<tr>
<td></td>
<td>An <strong>INSERT</strong> trigger must be associated with an INSERT statement. For example, if a data load operation doesn’t include an INSERT statement, the trigger won’t be invoked.</td>
</tr>
<tr>
<td></td>
<td>An <strong>UPDATE</strong> trigger can be associated with specific columns of the base table and will only be activated if those columns are updated.</td>
</tr>
<tr>
<td></td>
<td>A <strong>DELETE</strong> trigger fires automatically after items in the table are deleted.</td>
</tr>
<tr>
<td><strong>Trigger Type</strong></td>
<td><strong>STATEMENT</strong>: (only fires once).</td>
</tr>
<tr>
<td></td>
<td><strong>ROW</strong> (fires for each affected row): The trigger runs as many times as there are rows in the affected section. If the set of affected rows is empty, the trigger doesn’t run.</td>
</tr>
<tr>
<td><strong>Old Table Alias</strong></td>
<td>Type the name of a temporary table of rows as they exist before they’re updated or deleted.</td>
</tr>
<tr>
<td><strong>New Table Alias</strong></td>
<td>Type a name for a temporary table of rows as they exist after they’re inserted or updated.</td>
</tr>
<tr>
<td><strong>When Clause</strong></td>
<td>Type a WHEN clause or open a WHEN CLAUSE wizard to further qualify the trigger behavior.</td>
</tr>
</tbody>
</table>
Object Type Wizard for Oracle

Types define an abstract data type or object composed of a collection of similar types of data. For example, you can create an object type that defines a full address rather than the pieces of an address, such as city, state and postal code. An object type stores the pieces of an address in a single type, storing them in the same location, and allowing the full address to be accessed and manipulated as a single unit rather than multiple units.

To Open the Object Type Wizard

1. On the Explorer, find the datasource where you want to create a Type and expand the Schema node.

2. Right-click the Type node, and select New.

The single page wizard asks that you pick an owner for the type and name the type. The fun begins after you make your choices and click Finish. The Object Editor, where you can finalize the object type’s creation, opens.

The Object Type Editor’s tabs are:

- **Header**: Here you flesh out the specification for the type you’re creating including any methods (a subprogram), attributes, and any parameters or return types for a function.

- **Body**: For every specification, the object type body defines the code for the method. If the object type header declares only attributes, not a method, the body is unnecessary.

You need to create the object type before you can see the following tabs. After you’ve added what you need to the Header and Body pages of the Editor, click Create on the toolbar.

- **Information**: Read-only page that indicates the vital statistics for the object type.

- **Dependencies**: A tree structure displays the objects that depend on the type you have created.

- **Privileges**: Displays individual, group, and role permissions associated with the object type.

Unique Key Wizard for Oracle

A unique key constraint requires that every value in a column or set of columns be unique. Thus, no two rows of a table have duplicate values in the column or set of columns you identified. So, for example, you can use a unique key constraint to make sure you haven’t duplicated a social security number in a list of employees.
To create a new unique key using a wizard:

1. **Open an object wizard** for a unique key.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties</strong></td>
<td><strong>Table Owner and Table Name</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td></td>
<td><strong>No Sort</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Logging</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Reverse</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Validate</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Deferrable</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Deferred</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Enabled</strong></td>
</tr>
<tr>
<td><strong>Columns</strong></td>
<td>From the Column dropdown, select a column for the index and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
</tr>
</tbody>
</table>
User Wizard for Oracle

Whoever you add as a user will have access to the Oracle database. You also can set up the means by which the database recognizes the user.

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
</table>
| Storage       | **Data Block Storage group**

Select the DEFINITABLE Tablespace only if you are creating a local partitioned unique key and want the partitions in the same tablespace as the partitions in the underlying table. (Each partition of a local unique key is associated with one partition of the table. Oracle can then keep the primary key partitions in synch with table partitions.)

A transaction entry is needed for each INSERT, UPDATE, DELETE, etc. statement that accesses one or more rows in the block. Transaction entries in many operating systems require approx. 23 bytes.

**Percent Free** identifies how much space you want to allocate for new rows or updates to existing rows.

**Initial Transactions** ensures that a minimum number of concurrent transactions can update a primary key block, avoiding the overhead of allocating a transaction entry dynamically.

**Maximum Transactions** limits concurrency on a primary key block.

| Extents group | An extent is the unit of space allocated to an object whenever the object needs more space.

**Initial Extent** - The initial space extent (in bytes) allocated to the object.

**Next Extent** - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.

**Percentage Increase** - Lets you type the percentage.

**NOTE:** You should be careful when setting Percent Increase because it magnifies how an object grows and, therefore, can materially affect available free space in a tablespace.

**Minimum Extents** - For a dictionary managed tablespace, this is the total number of extents to be allocated when the index is first created. For a locally managed tablespace, this is simply the initial amount of space allocated.

**Maximum Extents** - For a dictionary managed tablespace, this is the total number of extents that can ever be allocated to the index. In a locally managed tablespace, the database will automatically manage the extents.

| Freelist group | **Free lists** let you manage the allocation of data blocks when concurrent processes are issued against the primary key. You can potentially improve the performance of the primary key by identifying multiple free lists, which can reduce contention for free lists when concurrent inserts take place.

The default and minimum value is 1. You should increase this number if multiple processes access the same data block.

**Free List Groups** is the number of groups of free lists.

**NOTE:** This option is only applicable for the parallel server option.

| Buffer Pool | **DEFAULT** - Choose this if you want to use the default bufferpool.

**KEEP** - Use this to retain the object in memory to avoid I/O conflicts. This type of bufferpool stores frequently referenced data blocks in a separate cache.

**RECYCLE** - Select this option to save cache space by ridding data blocks from memory as soon as they are no longer in use.

| Partition | Clicking Create Partition opens the Add, Insert, or Modify Partition wizard.

| DDL View | Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object. |
The User Wizard lets you:

- Set up the user name.
- Identify the user by password, externally, or globally.
- Grant locked or unlocked access to the user's account.
- Assign tablespaces and a profile to the user.
- Assign tablespace quotas to the user.
- Grant roles to the user.

Important Notes

- To create a user, you need the CREATE USER system privilege.

To create a new user using a wizard:

1. Open an object wizard for a user.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>Default Tablespace and Temporary Tablespace</td>
</tr>
<tr>
<td></td>
<td>Profile</td>
</tr>
<tr>
<td></td>
<td>Identified By</td>
</tr>
<tr>
<td></td>
<td>Password</td>
</tr>
<tr>
<td></td>
<td>External Name</td>
</tr>
<tr>
<td></td>
<td>Account Locked</td>
</tr>
<tr>
<td>Roles</td>
<td></td>
</tr>
<tr>
<td>Quotas</td>
<td></td>
</tr>
<tr>
<td>DDL View</td>
<td></td>
</tr>
</tbody>
</table>
View Wizard for Oracle

Views are SQL queries stored in the system catalog that customize the display of data contained in one or more tables. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables. You can use views to help enforce corporate security policies by creating a view that limits information a user can see.

Important Notes

• To create a view in your own schema, you need CREATE VIEW privileges. To create a view in someone else's schema, you need CREATE ANY VIEW privileges.

To create a new view using a wizard:

1. Open an object wizard for a view.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td><strong>Owner</strong> Select the owner of the view. The owner of the view must have SELECT privileges for the tables in the CREATE view statement or DBADM authority on the database that contains the table. Name Provide a name for the view.</td>
</tr>
<tr>
<td>Definition</td>
<td>This is where you build the query that will display the data you're interested in seeing. The template is CREATE VIEW &lt;name&gt; AS SELECT: Identify the columns you want to show in the view FROM: Identify the table(s) you want to draw the data from WHERE: Write the equivalence you want to view. You can use the Query Builder to help you write the appropriate SQL statement. Finally, use the Execute buttons to Create the new object.</td>
</tr>
</tbody>
</table>

Sybase ASE Object Wizards

- Alias Wizard
- Create Check Constraint Dialog Box
- Create Check Constraint Dialog Box
- Default Wizard
- Dump Device Wizard
- Extended Procedure Wizard
- Foreign Key Wizard
- Group Wizard
- Index Wizard
- Login Wizard
- Create Primary Key Constraint Dialog Box
- Procedure Wizard
- Remote Server Wizard
- Role Wizard
- Rule Wizard
- Segment Wizard
- Table Wizard
- Trigger Wizard
- View Wizard
- User Datatype Wizard
- User Message Wizard
- User Wizard
Alias Wizard for Sybase ASE

The Alias Wizard lets you map a login to an existing user in the database. You can set up aliases so that multiple users log in to the same account and therefore have the same privileges. You can also set up an alias based on individual log ins and give those users access to the same alias with the advantage that you can track their activity within the database.

To create a new alias using a wizard:

1. **Open an object wizard** for an alias.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Properties and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the alias being created.</td>
</tr>
<tr>
<td>User</td>
<td>Select the user to which the alias refers. The user has to have a valid account on SQL Server but cannot be a user in the current database.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>

Database Device Wizard for Sybase ASE

A database device is part or all of a disk device used for storing a database and its transaction log.

The Database Device Wizard lets you:

- Name the device and indicate if it should become the default device.
- Specify the virtual device number for and size of the new device.
- Override many obscure, rarely-used parameters of the DISK INIT statement.

To Open the Database Device Wizard

1. On the **Explorer**, find the datasourse where you want to add a database device and expand the **Storage** node.
2. Right-click the **Database Devices** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the logical name of the device?</td>
<td>This is the abstract name you want to assign to the device. If for any reason the physical device is down, you can redefine the logical name to a new device.</td>
</tr>
<tr>
<td>What is the physical name of the device?</td>
<td>You need to specify the full path for the file, including the hard drive letter or machine name, directory and full file name with the *.DAT extension (for example, D:\SYB\DATA\TEST.DAT).</td>
</tr>
</tbody>
</table>
Database Wizard for Sybase ASE

The Database Wizard lets you create a database (a structured collection of data that can be updated or queried) without knowing the underlying commands. Databases can be simple, that is one file with many records and the same fields, or much more complicated with multiple files with different fields.

To create a new database using a wizard:

1. Open an object wizard for a database.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want to specify this device as a default for database storage?</td>
<td>The default device is where all new DBArtisan databases are placed unless the user specifies an exact device in the CREATE DATABASE statement.</td>
</tr>
<tr>
<td>What is the virtual device number?</td>
<td>DBArtisan automatically calculates the next open device number and defaults its value in the box. Numbering is sequential.</td>
</tr>
<tr>
<td>What is the size of the device in megabytes?</td>
<td>Make sure that you have enough free space on the hard disk to accommodate the device file.</td>
</tr>
<tr>
<td>If you want to choose a specific starting page number, enter it here.</td>
<td>Lets you set the starting panel number, type a value in the box. Normally, Sybase ASE defaults this value to 0.</td>
</tr>
<tr>
<td>What is the controller number?</td>
<td>Normally, Sybase ASE uses 0.</td>
</tr>
<tr>
<td>Do you want to force the database file to be created contiguously?</td>
<td>NOTE: This option is only available on Open VMS systems.</td>
</tr>
</tbody>
</table>

### Database Wizard for Sybase ASE Properties and tasks

<table>
<thead>
<tr>
<th>Step</th>
<th>Properties and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Database Name</td>
</tr>
<tr>
<td>For Load</td>
<td>This option speeds loading by eliminating the step for pre-initializing panels. The for load option is appropriate when creating a database for recovering from media failure or moving a database from one machine to another.</td>
</tr>
<tr>
<td>With Override</td>
<td>Enable this option to specify whether logs and data are to be kept on the same logical device</td>
</tr>
<tr>
<td>Type</td>
<td>Optionally, use the TEMP option to designate this as a temporary database.</td>
</tr>
<tr>
<td>Options group</td>
<td>Enable or disable the following Sybase database options: abort tran on log full, allow nulls by default, auto identity, dbo use only, ddl in tran, identity in nonunique index, no chkpt on recovery, no free space acctg, read only, select into/bulkcopy/pllsort, single user, trunc log on chkpt, and unique auto_identity index.</td>
</tr>
<tr>
<td>Placement</td>
<td>In the Fragment Properties area provide or select values for the default fragment. Device Name, Size for the fragment, and a Device Type value of data only, log only, or data and log. If necessary use the New button to add a new fragment and repeat the steps above to provide details for that fragment. Use the Delete button to drop a fragment.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>
Data Cache Wizard for Sybase ASE
The data cache holds the log, data, and index pages that are currently being used or were recently accessed. There's a default cache loaded by Sybase, but you can add additional caches without reducing the size of the default. You can bind a table, index, image, or text to the cache you are creating.

To Open the Data Cache Wizard
1. On the Explorer, find the datasource where you want to add a database device and expand the Storage node.
2. Right-click the Data Caches node, and select New.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the data cache?</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>What is the size of the data cache box?</td>
<td>Select the unit: megabytes, kilobytes, gigabytes or pages.</td>
</tr>
<tr>
<td>What is the cache type?</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>What is the cache replacement policy?</td>
<td>A relaxed cache replacement policy can improve performance when there's little or no buffer replacement. This is the least recently used replacement policy. <strong>Strict</strong> is the default.</td>
</tr>
</tbody>
</table>

Default Wizard for Sybase ASE
Here you create a default for table column or user-defined datatype in the event that no value is available when the data is inserted. The default value you specify will be inserted only in the current database. You can then bind the default to a specific column or user-datatype.

To create a new default using a wizard:
1. Open an object wizard for a default.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td><strong>Owner</strong> Select the schema that is to own the default.</td>
</tr>
<tr>
<td></td>
<td><strong>Name</strong> Provide a name for the default.</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong> Provide the value of the default.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>From the <strong>Type</strong> dropdown, choose Column or Datatype, and if you chose <strong>Column</strong>, choose a Table from the <strong>Table</strong> dropdown. The list on the left is populated with candidate columns or datatypes. To move a candidate from the list on the left to the dependencies column on the right, select the candidate and click <strong>Add</strong>. Remove columns or datatypes from the dependencies list on the right by selecting the column or datatype and clicking <strong>Remove</strong>.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>
Dump Device Wizard for Sybase ASE

Your system writes its memory image to a dump device so you can retrieve data if your system crashes. All load and dump operations require that you name a device before the operation will complete. When you create a new dump device, ASE adds a row to the sysdevices table in the master database. Sybase recommends that you back up the master database before you create a new device and again after you have created the device. Be sure you have enough disk space on the physical drive to perform these operations.

To Open the Dump Device Wizard

1. On the Explorer, find the datasource where you want to add a dump device and expand the Security node.
2. Right-click the Dump Devices node, and select New.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the device type?</td>
<td>Tape or disk.</td>
</tr>
<tr>
<td>What is the dump device Name?</td>
<td>This is the device’s logical name.</td>
</tr>
<tr>
<td>What is the physical name?</td>
<td>The device or file name that is recognized by the operating system. (For example, a:sqltable.dat).</td>
</tr>
</tbody>
</table>

Extended Procedure Wizard for Sybase ASE

Extended stored procedures provide a method for calling external procedural language functions from within the Adaptive Server. A procedural language is a language capable of calling a C language function or manipulating C language datatypes.

To create a new extended procedure using a wizard:

1. Open an object wizard for an procedure.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Properties and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Select the owner of the extended procedure.</td>
</tr>
<tr>
<td>Name</td>
<td>Provide the name of the procedure.</td>
</tr>
<tr>
<td>Library Name</td>
<td>Provide the name of the library containing the procedure.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

Foreign Key Wizard for Sybase ASE

Foreign keys are used to relate information from one table to another. Foreign keys are unique values that refer to specific columns of other tables. Thus, a foreign key links two tables together. Embarcadero DBArtisan’s Foreign Key Wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data. The column in the initial table, the parent table, is called the primary key. The corresponding column in the (child) table that references the primary key, is the foreign key. Foreign keys can also refer to columns within the same table.
To create a new foreign key using a wizard:

1. Open an object wizard for a foreign key.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Properties and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Table Owner</td>
<td>Select the owner of the referring, or child, table.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Select the name of the referring, or child, table.</td>
</tr>
<tr>
<td>Match Full</td>
<td>Specify a referential integrity option.</td>
</tr>
<tr>
<td>Column Mapping</td>
<td>Under <strong>Referenced Table</strong>, choose the <strong>Owner</strong> and then the <strong>Name</strong> of the referenced, or parent, table. Under the <strong>Main Table</strong>, select checkboxes corresponding to the columns that are to reference columns in the referenced table. Then, under <strong>Referenced Table</strong>, select the corresponding column check boxes.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Execute</strong> button to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>

**Group Wizard for Sybase ASE**

A group is a collection of privileges that the DBA assigns to designated users.

**To Open the Group Wizard**

1. On the **Explorer**, find the datasource where you want to add a foreign key constraint and expand the **Databases** node.

2. Right-click **Groups**, and select **New**.

This single page wizard simply asks that you give the group a name. After you click **Finish** and **Execute**, the Group Editor opens where you assign users to the new group and give them particular system and object privileges.

**Index Wizard for Sybase ASE**

Like an index in a book, a table index helps you get at the data you want without having to read through the whole table. Indexes can exist on single column or on multiple columns. Indexes appear in the form of B-trees. And, as for books, you can have multiple indexes for a single table.
To create a new index using a wizard:
1. Open an object wizard for an index.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Properties and tasks</th>
</tr>
</thead>
</table>
| Properties  | Select or provide the following:  
|             | • The owner of the table for which the index is being created (Table Owner).  
|             | • The table for which the index is being created (Table Name).  
|             | • A Name for the index.  
|             | Under Attributes, provide or select values for the following Sybase options: Index Type, Clustered, Ignore Duplicate Key, Ignore Duplicate Rows, and Maximum Rows Per Page properties.  
|             | Under Storage, provide or select values for the following Sybase options: Reserve Page Gap, Segment Name, Fill Factor, Prefetch Strategy, and MRU Replacement Strategy.  
| Columns     | Select a column from the Columns dropdown and specify a Sort option.  
|             | Use the New button to add more columns to the index. Use the Delete button to remove selected columns from the index.  
| DDL View    | Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Schedule or Execute buttons to Create the new object.  

Login Wizard for Sybase ASE
Your login account controls access to the server and all of the databases within it. Only the System Administrator or System Security Officer can create logins. Once you can log into a server, you need additional privileges to access user databases. Specifically, each database owner adds the login as a user or alias to the database.

To Open the Login Wizard
1. On the Explorer, open the database where you want to add a login.
2. Expand the Security node.
3. Right-click Logins and then click New.

The table that follows describes the fields you may encounter as you complete the wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the ID for the login</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>What is the password for the login?</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>What is the full name of the login?</td>
<td>Optional (e.g., John Smith)</td>
</tr>
<tr>
<td>What should be the login’s default database?</td>
<td>Self-explanatory. Be cautious about using Master to avoid the inadvertent creation/deletion of objects.</td>
</tr>
<tr>
<td>What is the login’s default language?</td>
<td>If you do not select one, DBArtisan uses the server’s default language.</td>
</tr>
</tbody>
</table>
Primary Key Wizard for Sybase ASE

Primary key constraints make sure that no duplicate values or NULLS are entered in the columns you specify. You can use primary key constraints to enforce uniqueness and referential integrity. A table can only have a single primary key constraint.

The dialog box lets you specify the owner and table on which you want to place the primary key constraint.

To create a new primary key using a wizard:
1. Open an object wizard for a primary key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select any roles that you wish to grant to the new login</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Do you want to lock the new login after creating it?</td>
<td>Locking a login is an efficient way to disable a login on the master database without deleting it. Deleting a login can be complicated if the user owns any objects or has granted use of those objects to other users. Implementing roles is another reason to lock a login to prevent any single user from having all privileges and to be able to trace actions.</td>
</tr>
<tr>
<td>Select databases to create users</td>
<td>Select the check boxes that correspond to the databases or click the Check All button.</td>
</tr>
<tr>
<td></td>
<td>User Type - Lets you select the user type you want to add.</td>
</tr>
<tr>
<td></td>
<td>If you select Alias in the Alias box, click the User Name list, and then click the user name.</td>
</tr>
<tr>
<td></td>
<td>If you select dbo and you want to transfer aliases and permissions, click the Transfer Aliases and Permissions check box.</td>
</tr>
<tr>
<td></td>
<td>If you select User, in the User box, click the User Name list, and then click the user name, then click the Group list, and then click the user group.</td>
</tr>
<tr>
<td>Enter the password expiration, if any</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Enter the minimum password length, if any</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Enter the maximum number of login attempts, if any</td>
<td>Self-explanatory</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step</th>
<th>Properties and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Provide or select the following:</td>
</tr>
<tr>
<td></td>
<td>• The <strong>Table Owner</strong> and <strong>Table Name</strong> of the table for which to create the index.</td>
</tr>
<tr>
<td></td>
<td>• A <strong>Name</strong> for the Index.</td>
</tr>
<tr>
<td></td>
<td>• Whether the index is <strong>Clustered</strong>.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>Maximum Rows Per Page</strong>.</td>
</tr>
<tr>
<td></td>
<td>• A Reserve Page Gap.</td>
</tr>
<tr>
<td></td>
<td>• A <strong>Segment Name</strong>.</td>
</tr>
<tr>
<td></td>
<td>• A <strong>Fill Factor</strong> value.</td>
</tr>
</tbody>
</table>
Procedure Wizard for Sybase ASE

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

The Procedure Wizard lets you:

- Name the procedure and specify its body.
- Specify any execution options and you can encrypt the stored procedure text in syscomments.

To create a new procedure using a wizard:

1. Open an object wizard for a procedure.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Properties and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>Use the Column dropdown to choose a column name and choose a Sort option for that column. Use the New button to add an additional column to the index or use the Drop button to delete selected columns.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Schedule or Execute buttons to Create the new object.</td>
</tr>
</tbody>
</table>

Remote Server Wizard for Sybase ASE

Remote Servers are servers on a network that you can access through your local server. Instead of logging directly into a remote server, you can log into your local server and execute a remote procedure call to it.

**NOTE:** Only stored procedures are allowed against remote servers. You can use a linked server to execute distributed queries and stored procedures.

**Before beginning**

- Before you can set up a remote server, you first configure both the local and remote servers to accept remote access. To configure the Microsoft SQL Servers for remote access, you need to log into each server. The table below describes the parameters you need to configure on each server:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Access</td>
<td>1 (enabled)</td>
</tr>
</tbody>
</table>
After setting these server configuration parameters, shut down and restart the server so the new values can take effect. When you have configured both servers, proceed to the steps below.

To Open the Remote Server Wizard

1. On the Explorer, expand the server where you want to add a remote server.
2. Expand the Security node.
3. Right-click the Remote Servers node and click New.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Connections</td>
<td>Number of remote connections required</td>
</tr>
<tr>
<td>Remote Logins</td>
<td>Number of remote logins required</td>
</tr>
<tr>
<td>Remote Sites</td>
<td>Number of remote sites required</td>
</tr>
</tbody>
</table>

Role Wizard for Sybase ASE

Roles are sets of user privileges you associate with access to objects within a database. Roles streamline the process of granting permissions. You can use roles to grant sets of permissions and privileges to users and groups. Roles can help you comply with Sarbanes Oxley regulations by limiting which users can have access to what privileges, for example a Human Resources Role versus an Accounting Role.
To Open the Role Wizard
1. On the Explorer, expand the server where you want to add a remote server.
2. Expand the Security node.
3. Right-click the Roles node and click New.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the role?</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Enter the password, if any, that must be entered to activate the role</td>
<td>Seven characters is the default.</td>
</tr>
<tr>
<td>Enter the password expiration if any</td>
<td>OPTIONAL Self-explanatory</td>
</tr>
<tr>
<td>Enter the minimum password length, if any, if any</td>
<td>OPTIONAL Self-explanatory</td>
</tr>
<tr>
<td>Enter the minimum number of login attempts, if any</td>
<td>OPTIONAL Self-explanatory</td>
</tr>
</tbody>
</table>

After you click EXECUTE and the Role is successfully created, the Role Editor opens where you can go about the serious business of assigning object and system privileges.

Rule Wizard for Sybase ASE
Rules promote data integrity by allowing you to validate the values supplied to a table column. They are reusable objects that you can bind to table columns or user datatypes. Check constraints are similar to rules, and are in fact the preferred way of restricting data. A column or user-defined data type can have only one rule bound to it, but a column can have both a rule and one or more check constraints associated with it. Not that a rule cannot apply to data already existing in the database at the time you're creating the rule and can't be bound to a system-created data type. If you create a new rule when one already exists, the new rule will override the previous one.

To create a new rule using a wizard:
1. Open an object wizard for a rule.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Properties and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td><strong>Owner and Name</strong> Select an Owner and provide a Name for the rule.</td>
</tr>
<tr>
<td>Restrictions</td>
<td>Type the condition. The rule restriction is the condition that defines the rule and can be any expression valid in a WHERE clause and can include such elements as arithmetic operators, relational operators, and predicates (for example, IN, LIKE, BETWEEN).</td>
</tr>
<tr>
<td>Type</td>
<td>Choose STANDARD_RULE, AND_ACCESS_RULE, or OR_ACCESS_RULE.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>From the Type dropdown, choose Column or Datatype, and if you chose Column, choose a Table from the Table dropdown. The list on the left is populated with candidate columns or datatypes. To move a candidate from the list on the left to the dependencies column on the right, select the candidate and click Add. Remove columns or datatypes from the dependencies list on the right by selecting the column or datatype and clicking Remove.</td>
</tr>
</tbody>
</table>
Segment Wizard for Sybase ASE
Segments allow you to control the placement of objects on database storage devices. A segment is a subset of the database device on which a specific database is stored. Each database can contain as many as 32 segments. Each database includes system, logsegment, and default segments. Others can be added using the Segment Wizard. By judiciously placing large tables and nonclustered indexes on segments on different devices or segments of specific sizes, you can improve I/O throughput or control space usage.

To Open the Segment Wizard
1. On the Explorer expand the database node where you want to add the Segment.
2. Right-click the Segments node and select New.

The table that follows describes the fields you may encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the name of the segment?</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>On which devices do you wish to place the segment?</td>
<td>Self-explanatory</td>
</tr>
</tbody>
</table>

When you click Execute, DBArtisan opens the Segments Editor. Here you can extend the segment and see the objects stored in the segment.

Table Wizard for Sybase ASE
A table is a column-based arrangement of data in which the content of one column has a bearing on the other column(s). So, for example, a table might have a column for authors, another column for the books each author has written, and a third for the number of copies each title by a given author has sold. The data moves across the columns in rows.

**NOTE:** You must have CREATE TABLE permissions to generate a new table.

To create a new table using a wizard:
1. Open an object wizard for a table.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Owner</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>Segment Name</td>
</tr>
</tbody>
</table>
## Trigger Wizard for Sybase ASE

Triggers are a special type of procedure that automatically fire when defined data modification operations (insert, update, or delete) occur on a target table or view. Triggers fire after an insert, update or delete, but belong to the same transaction as the data modification operation. Triggers can be implemented to enforce business rules or referential data integrity.

### Important Notes
- For more information on the syntax for Trigger bodies, consult the Microsoft SQL Server Transact-SQL Documentation.

### Step Settings and tasks

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Maximum Rows Per Page</strong></td>
<td>Specifying a number allows you to override the default. The default, 0, creates indexes with full pages and nonclustered indexes with full leaf pages. This number can be changed at any time.</td>
</tr>
<tr>
<td><strong>Reserve Page Gap</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Identity Gap</strong></td>
<td>Specify the size of the identity gap for the table.</td>
</tr>
<tr>
<td><strong>MRU Replacement Strategy</strong></td>
<td>When enabled, new pages are read into the least recent end of the page chain. When pages reach the most recent end of the chain, the pages are flushed.</td>
</tr>
<tr>
<td><strong>Prefetch Strategy</strong></td>
<td>Enabling this feature allows you to fetch as many as eight 2K data pages simultaneously instead of one at a time (the default).</td>
</tr>
<tr>
<td><strong>Lock Scheme</strong></td>
<td>Select a locking scheme of ALLPAGES, DATAPAGES, or DATAROWS.</td>
</tr>
<tr>
<td><strong>Expected Row Size</strong></td>
<td>If you specified a <strong>Lock Scheme</strong> of DATAROWS or DATAPAGES, provide an expected row size.</td>
</tr>
<tr>
<td><strong>Columns</strong></td>
<td>For each column in the table, click the <strong>Add Column</strong> button to create a column, provide a <strong>Name</strong> for the column and provide or select the remaining column attributes. Use the <strong>Delete</strong> button to drop a selected column. Use the arrow buttons to reposition the columns. <strong>NOTE:</strong> Because the smalldatetime datatype stores dates and time with less precision than the datetime datatype, before outputting you use the CAST or CONVERT functions to convert any boxes with the smalldatetime datatype to either VARCHAR or datetime datatypes.</td>
</tr>
<tr>
<td><strong>Indexes</strong></td>
<td>Click Add to open the <strong>Index Wizard for Sybase ASE</strong>.</td>
</tr>
<tr>
<td><strong>Constraints</strong></td>
<td>Selecting a constraint type and clicking <strong>Add</strong> opens the object wizard for that object type. For details see:</td>
</tr>
<tr>
<td></td>
<td><strong>Primary Key Wizard for Sybase ASE</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Unique Key Wizard for Sybase ASE</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Foreign Key Wizard for Sybase ASE</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Create or Modify Constraint</strong></td>
</tr>
<tr>
<td><strong>Permissions</strong></td>
<td>Set up the user permissions for this table.</td>
</tr>
<tr>
<td><strong>DDL View</strong></td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the <strong>Schedule</strong> or <strong>Execute</strong> buttons to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>
To create a new trigger using a wizard:

1. Open an object wizard for a trigger.

2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td></td>
</tr>
<tr>
<td>Table Owner,</td>
<td>Select the owner and specific table for which the trigger is being created.</td>
</tr>
<tr>
<td>Table Name</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Provide a name for the trigger being created.</td>
</tr>
<tr>
<td>Fire On Insert, Fire On Update, and Fire On Delete</td>
<td>Enable the events that fire the trigger.</td>
</tr>
<tr>
<td></td>
<td>An <strong>INSERT</strong> trigger must be associated with an INSERT statement. For example, if a data load operation doesn't include an INSERT statement, the trigger won't be invoked.</td>
</tr>
<tr>
<td></td>
<td>An <strong>UPDATE</strong> trigger can be associated with specific columns of the base table and will only be activated if those columns are updated.</td>
</tr>
<tr>
<td></td>
<td>A <strong>DELETE</strong> trigger fires automatically after items in the table are deleted.</td>
</tr>
<tr>
<td>Definition</td>
<td>Complete the CREATE TRIGGER statement by typing or pasting in content. Finally, use the <strong>Execute</strong> button to <strong>Create the new object</strong>.</td>
</tr>
</tbody>
</table>

**NOTE:** For more information on the syntax for Trigger bodies, consult the Sybase ASE Documentation.

Use the Trigger Editor to create dependencies or alter the trigger statement.

**Unique Key Wizard for Sybase ASE**

Unique keys can enforce logical keys that are not chosen as the primary key. In other words, you can use a unique key to ensure no duplicate values are entered in specific columns that are not a part of the primary key. Although you can only attach one primary key to a table, you can attach multiple unique keys. Also, you can use unique keys on columns that allow null values.
To create a new unique key using a wizard:
1. Open an object wizard for a unique key.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Table Owner and Table Name: Choose the owner and name of the table in which the unique key is being created.</td>
</tr>
<tr>
<td></td>
<td>Name: Provide a name for the unique key being created.</td>
</tr>
<tr>
<td></td>
<td>Clustered: Select to create a unique index (one in which no two rows are permitted to have the same index value) on a table. <strong>NOTE:</strong> Only one clustered constraint or index is possible for each table.</td>
</tr>
<tr>
<td></td>
<td>Maximum Rows Per Page: Specify a maximum.</td>
</tr>
<tr>
<td></td>
<td>Reserve Page Gap: If you do not specify a segment, Sybase ASE creates the index on the default segment.</td>
</tr>
<tr>
<td></td>
<td>Segment Name: If you do not specify a segment, Sybase ASE creates the index on the default segment.</td>
</tr>
<tr>
<td></td>
<td>Fill Factor: This specifies how full each index page that's storing data should be. The fill factor is a percentage value between 0 and 100.</td>
</tr>
<tr>
<td>Columns</td>
<td>From the Column dropdown, select a column for the primary key and specify a Sort option. To add more columns, click the New button and then follow the steps in the last instruction. Use the Delete button to drop columns.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute buttons to Create the new object.</td>
</tr>
</tbody>
</table>

**Important Notes**
- If you are creating a non-clustered index constraint, you should place it on a separate segment from the target table.

**User Datatype Wizard for Sybase ASE**
User datatypes promote domain consistency by streamlining the definition of commonly used table columns in a database. You can build a customized datatype from system datatypes and bind defaults and rules to it to enhance integrity. When you reference the user datatype in a column, the column assumes all of the properties of the user datatype.

To create a new user datatype using a wizard:
1. Open an object wizard for a user datatype.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Datatype: Provide a name for the datatype.</td>
</tr>
<tr>
<td></td>
<td>Type: Select the base datatype.</td>
</tr>
<tr>
<td></td>
<td>Size: Provide the size of the datatype.</td>
</tr>
</tbody>
</table>
User Message Wizard for Sybase ASE
A user message lets you write the error message users will see when a user-defined event transpires. You have the option of saving the message to the event log. You can also create messages in multiple languages, but you have to create the message in US English before you can write it in other languages.

To Open the User Message Wizard
1. On the Explorer, expand the datasource where you want to write a user message.
2. Open the database, right-click the User Messages Node, and then click New.

The table that follows describes the fields you will encounter as you complete the wizard.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the message number?</td>
<td>A message number must be greater than 2,000. The wizard automatically loads with the next unclaimed message number, but you can use another number if you prefer.</td>
</tr>
<tr>
<td>Add the text for this message</td>
<td>Click Add and write the message you want to attach to the message. Otherwise, what's the point of using the wizard?</td>
</tr>
</tbody>
</table>

When you have successfully created the new message, the User Messages Editor opens where you can bind the message to foreign key or check constraints as well as edit the message text.

User Wizard for Sybase ASE
The User Wizard lets you create a user who will then have access to the database where you are registering him or her. You can also identify the appropriate user group and the system privileges you want to assign to the new user.
To create a new user using a wizard:
1. Open an object wizard for a user
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Login Name: Choose a login ID from the drop-down list.</td>
</tr>
<tr>
<td></td>
<td>Name: Provide the user name.</td>
</tr>
<tr>
<td></td>
<td>Group Name: Specify a group for the user.</td>
</tr>
<tr>
<td>DDL View</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute buttons to Create the new object.</td>
</tr>
</tbody>
</table>

You can use the User Editor to assign System and Object privileges as well as create an alias.

View Wizard for Sybase ASE
Views are SQL queries stored in the system catalog that customize the display of data contained in one or more tables. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables. Views can only be

To create a new view using a wizard:
1. Open an object wizard for a view.
2. Use the following table as a guide to setting properties and performing tasks as you pass through the steps of the wizard:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Owner: Select the owner of the view.</td>
</tr>
<tr>
<td></td>
<td>Name: Provide a name for the view.</td>
</tr>
<tr>
<td></td>
<td>Check Type: When enabled, when a row is modified through a view, this option makes sure the data remains visible through the view after the modification is committed.</td>
</tr>
<tr>
<td>Definition</td>
<td>Complete the CREATE VIEW statement by typing or pasting in the SELECT, FROM, and WHERE statements. Finally, use the Execute buttons to Create the new object.</td>
</tr>
</tbody>
</table>

Activate Logging
NOTE: This functionality is available for IBM DB2 for Linux, Unix, and Windows only.

The Activate Logging dialog box lets you store the most recent log records, when the table is not logging initially.

Completing the Activate Logging Dialog Box
To complete this dialog box, do the following:
1. On the Datasource Explorer, select the target object node.
   DBArtisan displays the target objects in the right pane of the Explorer window.
2  On the Datasource Explorer tool bar, click Command, and then select Activate Logging.
   DBArtisan opens the Active Logging dialog box.

3  Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

Add, Insert, or Modify Partition

**NOTE:** This functionality is available for Oracle.

You can open the dialog box in the following wizards:

- Oracle Index Wizard - Panel 10
- Oracle Materialized View Wizard - Panel 10 (Composite Partitioning)
- Oracle Materialized View Wizard - Panel 9 (Range Partitioning)
- Oracle Table Wizard - Composite Partitioning
- Oracle Table Wizard - Range Partitioning

The table below describes the options and functionality on the Add, Insert or Modify Partition dialog box.

**NOTE:** Options differ by platform and wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Lets you type the name of the new partition. Do not use single quotation marks ('') in the name.</td>
</tr>
<tr>
<td># of Partitions</td>
<td>Lets you specify the number of partitions.</td>
</tr>
<tr>
<td>Set Upper Bound Button</td>
<td>Lets you set the non-inclusive upper bound for the partitioning columns. Not available for the local index. Opens the Partition Upper Bound dialog box. The partitioning columns you chose in Step 1 appear in the column grid, with a default upper bound of MAXVALUE.</td>
</tr>
<tr>
<td>Tablespace</td>
<td>Lets you select the tablespace on which the partition should be placed.</td>
</tr>
<tr>
<td>No Logging</td>
<td>Select to make the partition log.</td>
</tr>
<tr>
<td>Physical</td>
<td>Lets you set the percent free, initial transactions and maximum transactions.</td>
</tr>
<tr>
<td>Space Attributes</td>
<td>Click View to view space attributes.</td>
</tr>
<tr>
<td>Storage</td>
<td>Lets you set any storage parameters by clicking the corresponding lists or typing the values in the corresponding boxes.</td>
</tr>
</tbody>
</table>
Completing the Add, Insert or Modify Partition Dialog Box

To complete this dialog box, do the following:

1. In the **Partition Definition** box, type the name of the new partition.
   
   **NOTE:** Do not use single quotation marks (') in the name.

   - If you do not want to set the non-inclusive upper bound for the partitioning columns, proceed to step 13.
   - To set the non-inclusive upper bound for the partitioning columns, proceed to step 8.

2. Click the column in the grid and then click the **Set Upper Bound** button.

   DBArtisan opens the **Partition Upper Bound dialog box**.

   **NOTE:** The partitioning columns you chose in Step 1 appear in the column grid, with a default upper bound of MAXVALUE.

The following table describes when you should enable editing of the upper bound:

<table>
<thead>
<tr>
<th>Partition Type</th>
<th>Index</th>
<th>Table</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash</td>
<td>N/A</td>
<td>N/A</td>
<td>Does not have upper bound.</td>
</tr>
<tr>
<td>Range Local</td>
<td>Disable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>Range Global</td>
<td>Enable all but the last partition</td>
<td>Enable</td>
<td>The upper bound button will be disabled if there is only one partition.</td>
</tr>
<tr>
<td>Composite Global</td>
<td>Enable all but the last partition</td>
<td>Enable</td>
<td>The upper bound button will be disabled if there is only one partition.</td>
</tr>
</tbody>
</table>

3. Click the **Partitioning Column** list and then click the partitioning column.

4. In the **Upper Bound** list, click the default or type the value of the upper bound. All rows in the partition will have partitioning keys that compare to less than the partition bound and greater than or equal to the partition bound for the preceding partition.

5. Click **Set**.

   DBArtisan establishes the upper bound.

6. Click **OK**.

   DBArtisan closes the DBArtisan message.
When you finish setting the upper bound for the partitioning columns, click Close.

DBArtisan returns to the Add Partition dialog box.

In the Segment box, click the Tablespace list and then click the tablespace on which the partition should be placed.

To make the partition log, click the No Logging check box to clear this option.

In the Physical box, set the percent free, initial transactions and maximum transactions by clicking the corresponding lists or typing the values in the corresponding boxes.

In the Storage Parameters box, set any storage parameters by clicking the corresponding lists or typing the values in the corresponding boxes.

Click Add.

DBArtisan adds the partition.

When you finish adding partitions, click the Close button.

• To insert another partition into the list, click the Insert button, and then click Insert to insert the column.
• To edit a partition, select the partition in the grid and then click the Edit button, and then click Modify to modify the column.
• To drop a partition, click the Drop button and then click Yes.

DBArtisan confirms the drop, closes the DBArtisan message, and omits the partition from the list.

When you finish specifying the partitioning attributes, click Finish.

For more information, see:

Add, Insert or Modify Partition
Completing a Dialog Box

Add or Modify Cluster Column
The Add or Modify Cluster Column dialog box lets you manage cluster columns. You can open the dialog box in the Oracle Cluster Wizard - Panel 2.

The table below describes the options and functionality on the Add or Modify Cluster Column dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Name</td>
<td>Lets you type the column name.</td>
</tr>
<tr>
<td>Datatype</td>
<td>Lets you select the datatype for the cluster.</td>
</tr>
<tr>
<td></td>
<td>If you select CHAR, RAW or VARCHAR2, in the Width box, type the width value.</td>
</tr>
<tr>
<td></td>
<td>If you select NUMBER, in the Width box, type the width value and in the Scale box, type the scale value.</td>
</tr>
</tbody>
</table>
Completing the Add or Modify Cluster Column Dialog Box
To complete this dialog box, do the following:

1. In the **Add Cluster Column** dialog box, in the **Column Name** box, type the column name.
2. Click the **Datatype** list, click the datatype for the cluster.
   - If you clicked **CHAR, RAW or VARCHAR2**, in the **Width** box, type the width value.
   - If you clicked **NUMBER**, in the **Width** box, type the width value and in the **Scale** box, type the scale value.
3. Click the **Add** button.
4. To continue adding columns to the cluster, repeat steps 1-3.
5. When you finish adding columns, click **Close**.

DBArtisan closes the Add Cluster Column dialog box.

For more information, see:
- Add or Modify Cluster Column
- Completing a Dialog Box

---

Add or Modify Datafile

**NOTE:** This functionality is available for all platforms.

The Add/Modify Datafile dialog box lets you add a datafile to a tablespace, or edit an existing one. You can open the dialog box in the **Oracle Tablespace Wizard**.

The table below describes the options and functionality on the Add/Modify Datafile dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datafile Name</td>
<td>Type the name of the datafile in the box.</td>
</tr>
<tr>
<td>Datafile Size</td>
<td>Type the datafile size and then click the list to indicate MB or KB.</td>
</tr>
<tr>
<td>Reuse Existing File?</td>
<td>The default is No.</td>
</tr>
<tr>
<td>Autoextend</td>
<td>Select this check box if you want to autoextend the datafile when more extents are required. This enables the options below.</td>
</tr>
<tr>
<td>Disk Space to Allocate to the Datafile When More Extents are Required</td>
<td>Type the disk space to allocate to the datafile when more extents are required by typing it and then click the list to indicate MB or KB.</td>
</tr>
<tr>
<td>Maximum Disk Space Allowed for Allocation to the Datafile</td>
<td>Specify the maximum disk space allowed for allocation to the datafile. Select Unlimited or Other. In the Other box, type the amount and then click the list to indicate MB or KB.</td>
</tr>
</tbody>
</table>

Completing the Add/Modify Datafile Dialog Box
To complete this dialog box, do the following:

1. In the **Datafile Name** box, type the datafile name.
2. In the **Datafile Size** box, type the datafile size and then click the list to indicate **MB or KB**.
3 **Reuse Existing File?**
   - To reuse the existing file, click the **Yes** option button.
   - If you do not want to reuse the existing file, click the **No** option button.

4 Select the **Autoextend** check box.
   - DBArtisan auto extends the datafile.
   - In the **Disk Space to Allocate to the Datafile When More Extents are Required** box, type the disk space to allocate to the datafile when more extents are required by typing it and then click the list to indicate **MB** or **KB**.
   - Specify the maximum disk space allowed for allocation to the datafile by clicking the **Unlimited** option button or the **Other** option button.
     - **NOTE:** If you click Other, in the **Other** box, type the amount and then click the list to indicate **MB** or **KB**.

5 Click **OK**.
   - DBArtisan adds the datafile to the tablespace definition.

6 Click **Close**.
   - DBArtisan closes the Add Datafile dialog box.

For more information, see:
- **Add or Modify Datafile**
- **Completing a Dialog Box**

### Allocate Extent

**NOTE:** This functionality is available for Oracle only.

The Allocate Extent dialog box lets you explicitly allocate extents for clusters, tables, and indexes in Oracle. Though Oracle dynamically allocates extents when additional space is required, explicit allocation of additional extents can be useful if you know that an object grows.

Explicit allocation of additional extents can be particularly helpful when using Oracle Parallel Server. When using Oracle Parallel Server and allocating additional extents, you can allocate an extent explicitly to a specific instance in order to minimize contention for free space among multiple instances.

**Important Notes**

For composite-partitioned tables, you can allocate extents to subpartitions as well as partitions.

The table below describes the options and functionality on the Allocate Extent dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Describe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extent Size</td>
<td>Refers to a datafile in the tablespace of the table, index, or cluster. You can choose a specific datafile from which to take space for the added extent. If you choose (Default), Oracle takes the space from any accessible datafile in the tablespace containing the table, index, or cluster.</td>
</tr>
<tr>
<td>Datafile</td>
<td>Lets you select the new datafile.</td>
</tr>
</tbody>
</table>
Completing the Allocate Extent Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the target object node.
   - DBArtisan displays the target objects in the right pane of the Explorer window.
2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Allocate Extent**.
   - **OR**
   - In the right pane of the application, right-click the target object, and then select **Allocate Extent**.
   - DBArtisan opens the Allocate Extent dialog box.
3. In the **Extent Size** box, type the new value of the extent, click the list to indicate the extent size: **MB** or **KB**.
4. Click the **Datafile** list, click the new datafile.
5. In the **Instance** box, specify the instance for Oracle to use when allocating the extent; this option only applies for Oracle with the Parallel Server in parallel query mode.
6. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Preview.png" alt="Preview" /></td>
<td>Opens the <strong>Preview dialog box</strong>.</td>
</tr>
<tr>
<td><img src="Schedule.png" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <strong>Scheduling</strong>.</td>
</tr>
<tr>
<td><img src="Execute.png" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="Analysis.png" alt="Analysis" /></td>
<td>Opens the <strong>Impact Analysis dialog box</strong>, This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see:

- **Allocate Extent**
- **Completing a Dialog Box**
Analyze
The Analyze dialog box lets you analyze the selected partitions. Analyzing a partition is helpful when you are trying to gather statistics about that partition, or if you are trying to validate the structure of that partition. If you are using cost-based optimization techniques, you can use the statistics to determine the most efficient use of SQL statements when accessing the partitions. In the case of hardware or system failure, analyze your partitions to make sure that their structure is not corrupt. Gathering information on chained rows lets you see if you have enough room for updates to rows in the partition.

Important Notes
For composite-partitioned tables, you can analyze subpartitions as well as partitions.

The table below describes the options and functionality on the Analyze dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition Box</td>
<td>Lists all the partitions available for analysis. Click the target partition. To select all the partitions, use the Select All button.</td>
</tr>
<tr>
<td>Analysis Option</td>
<td>Lets you select the type of analysis. The table below describes the analysis types and any additional information that you need to supply for each type.</td>
</tr>
<tr>
<td>Tables Only:</td>
<td>Perform the core calculations for a width-balanced histogram. For each row, Oracle returns the number of the histogram bucket appropriate for the data.</td>
</tr>
<tr>
<td>Histogram Option</td>
<td>Click to open the Histogram Statistics dialog box.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analysis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute Statistics</td>
<td>Scans the partition in its entirety to gather the information and computes exact statistics. An analysis that computes statistics takes longer than an estimation of statistics.</td>
</tr>
<tr>
<td>Delete Statistics</td>
<td>Deletes statistics for a partition from the data dictionary. Deletes statistics not used when cost-based optimization is used for SQL statements accessing the partition.</td>
</tr>
<tr>
<td>For Indexes:</td>
<td>Scans the partition and gathers representative information from portions of the partition. Estimating statistics is not as accurate as computing statistics but the analysis is less time consuming.</td>
</tr>
<tr>
<td>Estimate Statistics</td>
<td>If you choose Estimate Statistics, you must also type in a percentage or a row count in Sample. Use the drop-down menu to choose Percent or Row Count. The higher the percentage or the row count, the better the estimation.</td>
</tr>
</tbody>
</table>
Completing the Analyze Dialog Box
To complete this dialog box, do the following:

1. On the Datasource Explorer, select the target object node.
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select Analyze.
   OR
   In the right pane of the application, right-click the target object, and then select Analyze.
   DBArtisan opens the Analyze dialog box.

3. Select the target partition(s).
   TIP: To select all partitions in the index, click Select All.

4. In Analysis Option, select the type of analysis. The table below describes the analysis types and any additional information that you need to supply for each type.

<table>
<thead>
<tr>
<th>Analysis Option</th>
<th>Description</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>For Tables:</strong> Estimate Statistics</td>
<td>Scans the partition and gathers representative information from portions of the partition. Estimating statistics is not as accurate as computing statistics but the analysis is less time consuming.</td>
<td>If you choose Estimate Statistics, you must also type in a percentage or a row count in Sample. Use the drop-down menu to choose Percent or Row Count. The higher the percentage or the row count, the better the estimation. If you choose Estimate Statistics, you can also choose to generate a histogram in Histogram Option. If you choose to create a histogram for Selected Columns, you must also choose the columns in Selected Histogram Columns. If there are no available columns, click the Add button.</td>
</tr>
<tr>
<td><strong>For Indexes:</strong> Validate Structure</td>
<td>Verifies the integrity of the partition. If the partition is corrupt, Oracle returns an error message. If the partition is valid, no error is returned.</td>
<td>None</td>
</tr>
<tr>
<td><strong>For Tables:</strong> Validate Structure</td>
<td>Verifies the integrity of the partition. If the partition is corrupt, Oracle returns an error message. If the partition is valid, no error is returned. Oracle also verifies that the rows belongs to the correct partition. If rows do not collate correctly, they are considered invalid and Oracle lists them in a table. You must specify a table in which to put any invalid rows. If you do not have a table prepared to accept the invalid rows, you can check the Create Target Table if it does not exist box to have DBArtisan create a table for you. Clicking Cascade validates the structure of associated objects such as indexes.</td>
<td></td>
</tr>
<tr>
<td><strong>For Tables:</strong> List Chained Rows</td>
<td>Lists the chained rows of the partition. The results are stored in the named table.</td>
<td>You must specify a table in which to list the chained rows. If you do not have a table prepared to accept the results of the List Chained Rows command, DBArtisan will prompt you to create the target table after you click Execute.</td>
</tr>
</tbody>
</table>
5  Do one of the following:

<table>
<thead>
<tr>
<th>Analysis Option</th>
<th>Description</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute Statistics</td>
<td>Scans the partition in its entirety to gather the information and computes exact statistics. An analysis that computes statistics takes longer than an estimation of statistics.</td>
<td>None</td>
</tr>
<tr>
<td>Delete Statistics</td>
<td>Deletes statistics for a partition from the data dictionary. Deletes statistics not used when cost-based optimization is used for SQL statements accessing the partition.</td>
<td>None</td>
</tr>
<tr>
<td>Estimate Statistics</td>
<td>Scans the partition and gathers representative information from portions of the partition. Estimating statistics is not as accurate as computing statistics but the analysis is less time consuming.</td>
<td>If you choose Estimate Statistics, you must also type in a percentage or a row count in Sample. Use the drop-down menu to choose Percent or Row Count. The higher the percentage or the row count, the better the estimation.</td>
</tr>
<tr>
<td>Validate Structure</td>
<td>Verifies the integrity of the partition. If the partition is corrupt, Oracle returns an error message. If the partition is valid, no error is returned.</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the Impact Analysis dialog box. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see:

Analyze
Completing a Dialog Box

Histogram Statistics Dialog Box
The Histograms Statistics dialog box lets you specify options for histograms.

Important Notes
You can only generate a histogram for tables.
The table below describes the options and functionality on the Histogram Statistics dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Number of bands, or sections, into which the column data is sorted. Each section refers to a range of values and Oracle fits each piece of data into one of those sections. For example, in the image above, the column data for the NUMBER_SEATS column is split into 10 sections.</td>
</tr>
</tbody>
</table>

**Completing the Histogram Statistics Dialog Box**

To complete this dialog box, do the following:

1. Click the option button that corresponds to the level at which you want to capture histogram statistics.
2. Type any appropriate values in the corresponding boxes.
3. If you click **Selected Columns**, click the columns you want and then click the **Select** or **Unselect** button.
4. Click **OK**.

For more information, see:

Histogram Statistics Dialog Box
Completing a Dialog Box

**Bind Data Cache**

*NOTE:* This functionality is available for Sybase ASE only.

The data cache holds the data, index, and log pages currently in use and pages used recently by Sybase ASE. The Bind Data Cache dialog box lets you bind the following items to the data cache:

- Entire Database
- Index(es)
- Table(s)
- Text and Image Columns from Tables

**Important Notes**

- The data cache must be active before you can bind an object to it. To activate a new data cache, Sybase ASE requires that you restart the server. The bindings take effect immediately after the server restart.
- You can bind the syslogs table to a logonly cache.

For more information, see Completing a Dialog Box.

**Bind Package**

*NOTE:* This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Bind Package Wizard lets you set package parameters, add environments, and set package properties.

For more information, see Bind Package Wizard - Panel 1.
Bind Package Wizard - Panel 1

The table below describes the options and functionality on the first panel of the Bind Package wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Lets you select the name of the location to connect to.</td>
</tr>
<tr>
<td>Collection</td>
<td>Lets you select the location of the DBMS where the package binds and where the description of the package resides.</td>
</tr>
<tr>
<td>New package - PDS name</td>
<td>Lets you select a name.</td>
</tr>
<tr>
<td>Member</td>
<td>Lets you select what database request modules (DBRMs) to include in the package.</td>
</tr>
<tr>
<td>Copy from collection.</td>
<td>Lets you select a collection.</td>
</tr>
<tr>
<td>Package</td>
<td>Lets you select a package.</td>
</tr>
<tr>
<td>Version</td>
<td>Lets you select a version of the package.</td>
</tr>
<tr>
<td>Options</td>
<td>Lets you select an option.</td>
</tr>
</tbody>
</table>

**TIP:** To filter the Owner and Object lists, click the Filter button next to each list. DBArtisan opens the Filter dialog box.

For more information, see Bind Package.

Bind Package Wizard - Panel 2

The table below describes the options and functionality on the second panel of the Bind Package wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Lets you select a package owner.</td>
</tr>
<tr>
<td>Qualifier</td>
<td>Lets you select a qualifier, the package creator.</td>
</tr>
<tr>
<td>Action</td>
<td>Lets you select an action.</td>
</tr>
<tr>
<td>Version</td>
<td>Lets you select a version of the package.</td>
</tr>
</tbody>
</table>

For more information, see Bind Package.

Bind Package Wizard - Panel 3

The table below describes the options and functionality on the third panel of the Bind Package wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isolation</td>
<td>Determines how far to isolate an application from the effects of other running applications.</td>
</tr>
</tbody>
</table>
For more information, see Bind Package.

Bind Package Wizard - Panel 4
The table below describes the options and functionality on the fourth panel of the Bind Package wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explain</td>
<td>Obtains information about how SQL statements in the package are to execute, and then inserts that information into the table owner.PLAN_TABLE, where owner is the authorization ID of the owner of the plan or package. This option does not obtain information for statements that access remote objects.</td>
</tr>
<tr>
<td>Reopt(VARS)</td>
<td>Re-determines the access path at run time.</td>
</tr>
</tbody>
</table>
## For more information, see Bind Package.

### Bind Package Wizard - Panel 5
The table below describes the options and functionality on the fifth panel of the Bind Package wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare</td>
<td>Prepares dynamic SQL statements that refer to remote objects.</td>
</tr>
<tr>
<td>ImmedWrite</td>
<td>Immediate writes will be done for updates made to group buffer pool dependent pagesets or partitions.</td>
</tr>
<tr>
<td>Opthint</td>
<td>Query optimization hints are used for static SQL.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Lets you select type of language for the package.</td>
</tr>
<tr>
<td>Path</td>
<td>Lets you select a path that DB2 uses to resolve unqualified user-defined distinct types, functions, and stored procedure names (in CALL statements).</td>
</tr>
<tr>
<td>Flag</td>
<td>Lets you select a flags, messages to display:</td>
</tr>
<tr>
<td></td>
<td>- All informational, warning, error, and completion messages</td>
</tr>
<tr>
<td></td>
<td>- Only warning, error, and completion messages</td>
</tr>
<tr>
<td></td>
<td>- Only error and completion messages</td>
</tr>
<tr>
<td></td>
<td>- Only completion messages</td>
</tr>
</tbody>
</table>

For more information, see Bind Package.

### Bind Plan

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Bind Plan Wizard lets you set plan parameters, add packages, and set bind properties.

For more information, see Bind Plan Wizard - Panel 1.

### Bind Plan Wizard - Panel 1
The table below describes the options and functionality on the first panel of the Bind Plan wizard:
To filter the Owner and Object lists, click the Filter button next to each list. DBArtisan opens the Filter dialog box.

For more information, see Bind Plan.

# Bind Plan Wizard - Panel 2

The table below describes the options and functionality on the second panel of the Bind Plan wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Name</td>
<td>Lets you select the member name.</td>
</tr>
<tr>
<td>PDS Name</td>
<td>Lets you select the partitioned data set.</td>
</tr>
<tr>
<td>Add</td>
<td>Click to enter each member and PDS name.</td>
</tr>
</tbody>
</table>

For more information, see Bind Plan.
Bind Plan Wizard - Panel 3
The table below describes the options and functionality on the third panel of the Bind Plan wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Lets you select the name of the location to connect to.</td>
</tr>
<tr>
<td>Collection</td>
<td>Lets you select the location of the DBMS where the plan binds and</td>
</tr>
<tr>
<td></td>
<td>where the description of the plan resides.</td>
</tr>
<tr>
<td>Package</td>
<td>Lets you select a package.</td>
</tr>
</tbody>
</table>

For more information, see Bind Plan.

Bind Plan Wizard - Panel 4
The table below describes the options and functionality on the fourth panel of the Bind Plan wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isolation</td>
<td>Determines how far to isolate an application from the effects of other running applications.</td>
</tr>
<tr>
<td>Keep Dynamic</td>
<td>Specifies that DB2 keeps dynamic SQL statements after commit points. The application does not need to prepare an SQL statement after every commit point. DB2 keeps the dynamic SQL statement until the application process ends, a rollback operation occurs or the application executes an explicit PREPARE statement with the same statement identifier. If the prepared statement cache is active, DB2 keeps a copy of the prepared statement in the cache. If the prepared statement cache is not active, DB2 keeps only the SQL statement string past a commit point. DB2 then implicitly prepares the SQL statement if the application executes an OPEN, EXECUTE, or DESCRIBE operation for that statement.</td>
</tr>
<tr>
<td>Current Data</td>
<td>Determines whether to require data currency for read-only and ambiguous cursors when the isolation level of cursor stability is in effect. It also determines whether block fetching can be used for distributed, ambiguous cursors.</td>
</tr>
<tr>
<td>Degree</td>
<td>Determines whether to attempt to run a query using parallel processing to maximize performance. Lets you select an option.</td>
</tr>
<tr>
<td>Dynamic Rules</td>
<td>Determines what values apply at run time for the following dynamic SQL attributes:</td>
</tr>
<tr>
<td></td>
<td>The authorization ID that is used to check authorization</td>
</tr>
<tr>
<td></td>
<td>The qualifier that is used for unqualified objects</td>
</tr>
<tr>
<td></td>
<td>The source for application programming options that DB2 uses to parse and semantically verify dynamic SQL statements</td>
</tr>
<tr>
<td></td>
<td>Whether dynamic SQL statements can include GRANT, REVOKE, ALTER, CREATE, DROP, and RENAME statements</td>
</tr>
<tr>
<td>Release</td>
<td>Determines when to release resources that a program uses, either at each commit point or when the program terminates.</td>
</tr>
<tr>
<td></td>
<td>Commit - Releases resources at each commit point.</td>
</tr>
<tr>
<td></td>
<td>Deallocate - Releases resources only when the program terminates.</td>
</tr>
</tbody>
</table>
For more information, see [Bind Plan](#)

**Bind Plan Wizard - Panel 5**
The table below describes the options and functionality on the fifth panel of the Bind Plan wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validate</td>
<td>Determines whether to recheck, at run time, errors found during bind. The option has no effect if all objects and needed privileges exist. Bind - If not all objects or needed privileges exist at bind time, the wizard displays an error messages, and does not bind the package. Run - If not all objects or privileges exist at bind time, the process issues warning messages, but the bind succeeds. DB2 checks existence and authorization again at run time for SQL statements that failed those checks during bind. The checks use the authorization ID of the plan owner.</td>
</tr>
</tbody>
</table>

For more information, see [Bind Plan](#)

**Bind Plan Wizard - Panel 6**
The table below describes the options and functionality on the sixth panel of the Bind Plan wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable or Disable</td>
<td>Lets you select system connection types that can use the plan or package.</td>
</tr>
<tr>
<td>System</td>
<td>Lets you select a system.</td>
</tr>
<tr>
<td>Cname</td>
<td>Lets you select an option.</td>
</tr>
</tbody>
</table>
For more information, see Bind Plan.

### Build Query

**NOTE:** This functionality is available for all platforms.

**Query Builder** is a database productivity tool that lets you construct, structure, and manipulate up to five different types of queries simultaneously. It includes a separate graphical interface that opens within your current workspace. You can run Query Builder against all Embarcadero Technologies supported database platforms.

### Change Category

DBArtisan lets you change the category of a target Stored Outline.

**NOTE:** This functionality is available for Oracle only.

**To change the category of a stored outline:**

1. Connect to the datasource that contains the stored outline. For more information, see Datasource Management.
2. On the **Datasource Explorer**, expand the datasource nodes until the **Stored Outlines** node is visible and then select the **Stored Outlines** node.
   
   DBArtisan displays the stored outlines in the right pane of the Explorer window.
3. In the right pane of the datasource explorer, select the stored outline for which you want to change categories.
4. On the **Datasource Explorer** tool bar, click **Command**, and then select **Change Category**.
   
   **OR**
   
   In the right pane of the application, right-click the outline, and then select **Change Category**.
   
   DBArtisan opens the Change Category dialog box.
5. From the **Category** dropdown, choose a new, target category.
6. Review the **Dependencies** and **Preview** panels and when ready, use the **Schedule** or **Execute** button to complete the task.

### Change Container Size

**NOTE:** This functionality is available for all platforms.

The Change Container Size lets you increase and resize a container, the allocation of space to a tablespace. Depending on the tablespace type, the container can be a directory, device, or file.

The table below describes the options and functionality on the Change Container Size dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase Container Size by (Extent)</td>
<td>Select to increase the container size.</td>
</tr>
<tr>
<td>Change Container Size to (Resize)</td>
<td>Select to resize the container.</td>
</tr>
</tbody>
</table>
Change Database Comment

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows.

The Change Database Comment dialog box lets you change the comment for a database.

The table below describes the options and functionality on the Change Database Comment dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Comment</td>
<td>Displays the current comment.</td>
</tr>
<tr>
<td>New Comment</td>
<td>Lets you type a new comment.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

Change Password

**NOTE:** This functionality is available for Microsoft SQL Server, Oracle, and Sybase ASE only.

The Change Password dialog box lets you change user passwords, which you should do on a regular basis to guard against security leaks.

The table below describes the options and functionality on the Change Password dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Password</td>
<td>Lets you type the new password.</td>
</tr>
<tr>
<td>Confirm Password</td>
<td>Lets you retype the new password.</td>
</tr>
</tbody>
</table>

**Completing the Change Password Dialog Box**

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Users** node.
   
   DBArtisan displays the Users in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Change Password**.
   
   OR

   In the right pane of the application, right-click the user, and then select **Change Password**.

   DBArtisan opens the Change Password dialog box.

3. **OPTIONAL:** In the **Old Password** box, type the old password.

4. In the **New Password** box, type the new password.

5. In the **Confirm Password** box, type the new password.
6  Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the Impact Analysis dialog box. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see:

Change Password  
Completing a Dialog Box

Change Status

**NOTE:** This functionality is available for Oracle only.

The Change Status dialog box lets you change the status of a tablespace from online to offline and vice-versa. A tablespace is a storage structure that acts as a partition for the database. You can also place the tablespace in read only mode or read/write mode.

The table below describes the options and functionality on the Change Status dialog box.

<table>
<thead>
<tr>
<th>New Status Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online</td>
<td>Select to take tablespace online.</td>
</tr>
<tr>
<td>Read Only</td>
<td>Select to make the tablespace read only and prevent further write operations on the tablespace.</td>
</tr>
<tr>
<td>Offline</td>
<td>Select to take the tablespace offline and prevent further access to its segments.</td>
</tr>
</tbody>
</table>

Completing the Change Status Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Tablespaces** node.
   DBArtisan displays the Tablespaces in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Change Status**.
   OR
   In the right pane of the application, right-click the tablespace, and then select **Change Status**.
   DBArtisan opens the Change Status dialog box.

3. To bring the tablespace online, click the **Online** option button.
4 To make the tablespace read only and prevent further write operations on the tablespace, select the Read Only check box.

5 To take the tablespace offline and prevent further access to its segments, click the Offline option button, click the list and then click Normal, Temporary, or Immediate to indicate the method for taking the tablespace offline.

6 Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the Impact Analysis dialog box. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see:

- Change Status
- Completing a Dialog Box

Check Data

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Check Data Wizard checks tablespaces for violations of referential and check constraints. The wizard also checks for consistency between a base tablespace and a LOB tablespace.

Upon completion, the wizard reports information about violations that are detected and the status of tablespaces that violate any constraints is set to CHECK-PENDING. Also, the CHECK-PENDING status of all tablespaces that do not violate any constraints is reset. Optionally, the utility deletes rows that violate constraints.

**Important Notes**

- Run the Check Data Wizard after a conditional restart or a point-in-time recovery on all tablespaces where parent and dependent tables might not be synchronized or where base tables and auxiliary tables might not be synchronized.

- Run the Check Data Wizard against a base tablespace only, not a LOB table space.

For more information, see Check Data Wizard - Panel 1.

Check Data Wizard - Panel 1

The table below describes the options and functionality on this panel of the Check Data Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select tablespaces:</td>
<td>If Wizard is opened from the Tablespace Editor, the target tablespaces are initially displayed. To check additional tablespaces, click the Add button.</td>
</tr>
</tbody>
</table>
Check Data Wizard - Panel 2
The table below describes the options and functionality on this panel of the Check Data Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want the utility to be restartable?</td>
<td>Lets you specify to execute Wizard in restartable mode. You can restart it from the Datasource Explorer.</td>
</tr>
<tr>
<td>Would you like to specify the scope?</td>
<td>Lets you specify scope options: Pending (default) - DBArtisan keeps track of the data rows that must be checked to ensure referential integrity. Auxonly - Select if you only want to check tables with LOB columns. All - Select whenever the scope information is in doubt. The scope information is recorded in the DB2 catalog. The scope information can become in doubt whenever you start the target table space with ACCESS(FORCE), or the catalog is recovered to a point in time. RefOnly - Select if you want to check all dependent tables in the specified table spaces except with LOB columns.</td>
</tr>
<tr>
<td>Would you like to specify the auxiliary error?</td>
<td>Lets you specify the auxiliary error options: Report (default) - DBArtisan sets the base table space to the auxiliary CHECK-pending (ACHKP) status. If CHECK DATA encounters only invalid LOB columns and no other LOB column errors, the base table space is set to the auxiliary warning status. Invalidate - DBArtisan sets the base table column to an invalid status, and sets the base table space to the auxiliary warning (AUXW) status. You can use SQL to update a LOB column in the AUXW status, however, any other attempt to access the column will result in a -904 SQL return code.</td>
</tr>
<tr>
<td>Exceptions</td>
<td>Lets you specify the number or exceptions, from 0 to 2147483647.</td>
</tr>
<tr>
<td>Error ddn</td>
<td>Lets you specify the name of a error data definition (DD) statement that corresponds to a data control block that contains the same name. Lets you select a previously saved dataset template or click Edit to open the Dataset Template Editor.</td>
</tr>
<tr>
<td>Work ddn</td>
<td>Lets you specify the name of a work data definition (DD) statement that corresponds to a data control block that contains the same name. Lets you select a previously saved dataset template or click Edit to open the Dataset Template Editor.</td>
</tr>
</tbody>
</table>

For more information, see Check Data Wizard.

Check Data Wizard - Panel 3
The table below describes the options and functionality on this panel of the Check Data Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exceptions</td>
<td>Displays the Source and Destination tables for each exception. You can select one or more rows from the grid.</td>
</tr>
<tr>
<td>Add Button</td>
<td>Opens the Exceptions dialog box.</td>
</tr>
</tbody>
</table>
The Check Index Wizard lets you test whether indexes are consistent with the data they index. The Wizard issues warning messages when an inconsistency is found.

The Check Index Wizard lets you test whether indexes are consistent with the data they index. The Wizard issues warning messages when an inconsistency is found.

Run the Check Index Wizard after a conditional restart or a point-in-time recovery on all tablespaces where you suspect that the indexes may not be consistent with the data. Also run the wizard before executing the Check Data Wizard to ensure that the indexes used by the Check Data Wizard are valid. This is especially important before using the Check Data with DELETE YES.

When checking an auxiliary table index, the Check Index Wizard verifies that each LOB is represented by an index entry, and that an index entry exists for every LOB.

For more information, see Check Index Wizard - Panel 1.
Check Index Wizard - Panel 1
The table below describes the options and functionality on this panel of the Check Index Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Index</td>
<td>If Wizard is opened from the Indexes Editor, the target tablespaces are initially displayed. To check additional Indexes, click the Add button.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Indexes Selector dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes selected index.</td>
</tr>
</tbody>
</table>

For more information, see Check Index Wizard.

Check Index Wizard - Panel 2
The table below describes the options and functionality on this panel of the Check Index Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select tablespaces</td>
<td>If Wizard is opened from the Tablespaces Editor, the target tablespaces are initially displayed.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Tablespace Selector dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes selected tablespace.</td>
</tr>
</tbody>
</table>

For more information, see Check Index Wizard.

Check Index Wizard - Panel 3
The table below describes the options and functionality on this panel of the Check Index Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want the utility to be restartable?</td>
<td>Lets you specify to execute Wizard in restartable mode. You can restart it from the Datasource Explorer.</td>
</tr>
<tr>
<td>Do you want to specify the work ddn?</td>
<td>Lets you specify the name of a work data definition (DD) statement that corresponds to a data control block that contains the same name. Lets you select a previously saved dataset template or click Edit to open the Dataset Template Editor.</td>
</tr>
<tr>
<td>Do you want to specify the device type for temporary data sets by DFSORT?</td>
<td>Lets you specify the device type for temporary data sets to be dynamically allocated by DFSORT.</td>
</tr>
<tr>
<td>Do you want to specify the sort number?</td>
<td>Lets you type the number of datasets to be dynamically allocated by the sort program. The range of the text field is 0 to 2147483647.</td>
</tr>
</tbody>
</table>

For more information, see Check Index Wizard.
Checkpoint
Running a Checkpoint lets you force all dirty pages for the current database to be written to disk. A dirty page is any data or log page which, after being read into the buffer cache, is modified but not yet written to disk. The Checkpoint statement saves time in a subsequent recovery by creating a point at which all modifications to data and log pages are guaranteed to have been written to disk. If the current database is in log-truncate mode, CHECKPOINT also truncates the inactive portion of the log.

NOTE: This functionality is available for Microsoft SQL Server, and Sybase ASE only.

Important Notes
The default permission for executing a checkpoint is the db_owner fixed database role.

To run a Checkpoint against one or more databases:
1. Connect to the datasource that contains the database. For more information, see Datasource Management.
2. On the Datasource Explorer, select the Databases node.
   DBArtisan displays the Databases in the right pane of the Explorer window.
3. In the right pane of the Explorer window, select one or more target databases.
4. On the Datasource Explorer tool bar, click Command, and then select Checkpoint.
   DBArtisan opens the Checkpoint dialog box.
5. Use the following table as a guide to working through the panels of the dialog box

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action options</td>
<td>Displays the names of the database(s) you chose.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Lets you review any dependencies before you proceed.</td>
</tr>
<tr>
<td>Preview</td>
<td>Displays the DDL generated to execute the Checkpoint operation.</td>
</tr>
</tbody>
</table>

6. Use one of the Schedule or Execute buttons to execute the Checkpoint.

Clone Partition
NOTE: This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Clone Partition dialog box lets you clone a definition of a partition.

The table below describes the options and functionality on the dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select 1 or More Partitions</td>
<td>Lets you select partitions to clone.</td>
</tr>
<tr>
<td>Select All</td>
<td>Click select all partitions to clone.</td>
</tr>
<tr>
<td>Unselect All</td>
<td>Click to unselect all partitions to clone.</td>
</tr>
</tbody>
</table>

For more information, see Completing a Dialog Box.
Clone Table

**NOTE:** This functionality is available for Oracle.

The Clone Table dialog box lets you clone a definition of a table.

The table below describes the options and functionality on the dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select 1 or More Tables</td>
<td>Lets you select tables to clone.</td>
</tr>
<tr>
<td>Select All</td>
<td>Click select all tables to clone.</td>
</tr>
<tr>
<td>Unselect All</td>
<td>Click to unselect all tables to clone.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

Coalesce

**NOTE:** The Coalesce dialog box is available for Oracle 7 or later.

The Coalesce Fragments dialog box helps you maximize the size of free space chunks in tablespaces to avoid the situation in which an object cannot acquire enough contiguous free space to accommodate its next extent size. Towards this goal, look for opportunities to coalesce adjacent blocks of free space into a single, larger block.

Starting with Version 7, Oracle automatically coalesces adjacent free space chunks with a background process. However, it still supports the commands for coalescing free space manually.

Depending on the size of the tablespace, coalescing its free space can take a long time. So determine when to perform this operation. If you coalesce immediately, DBArtisan locks the tablespace.

**Important Notes**

- You cannot coalesce on an UNDO tablespace.

For more information, see [Completing a Dialog Box](#).

Compile

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows and Oracle only.

DBArtisan lets you recompile specific objects by issuing the proper ALTER statement. The explicit recompilation of invalid objects eliminates the need for implicit run-time recompilation which, in turn, can cause run-time compilation errors and performance overhead. Recompile objects after you make changes to that object or dependent objects.

The objects you can compile and the steps in compiling them differ according to the type of datasource you are working against. See the following topics:

- [Compiling IBM DB2 for Windows, Unix, and Linux procedures](#)
- [Compiling Oracle objects](#)
Compiling IBM DB2 for Windows, Unix, and Linux procedures

The table below describes the options and functionality on the Confirm Compile dialog box for IBM DB2 for Linux, Unix, and Windows:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare with release option</td>
<td>Compiles the stored procedure into a state that will run optimally in a production environment.</td>
</tr>
<tr>
<td>Compile with debug option</td>
<td>Compiles the stored procedure with the debugging information and symbols required to service step, pause, and breakpoint requests.</td>
</tr>
<tr>
<td>Keep current specific name</td>
<td>Notifies DB2 that the stored procedure will keep its current system assigned name. With this option, DB2 is not required to reset internal references to the target object.</td>
</tr>
</tbody>
</table>

Completing the Compile Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the target object node.
   
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Compile**.
   
   OR
   
   In the right pane of the application, right-click the target object, and then select **Compile**.
   
   DBArtisan opens the Compile dialog box.

3. In the **Compile Options** section, select the appropriate options for the compilation.

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <strong>Preview dialog box</strong>.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <strong>Scheduling</strong>.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the <strong>Impact Analysis dialog box</strong>. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

Compiling Oracle objects

To recompile an object it must belong to your schema or you need ALTER ANY privileges on that object. You must also have appropriate privileges for any associated objects. Prior to compiling objects of a particular type, see the relevant topic below:

- Notes on compiling Functions
- Notes on compiling Java Sources
- Notes on compiling Materialized Views
- Notes on compiling Packages and Package Bodies
Notes on compiling Procedures

Notes on compiling Types and Type Bodies

Notes on compiling Views

To compile an Oracle database object:
1. Connect to the datasource that contains the database object. For more information, see Datasource Management.
2. On the left pane of the Datasource Explorer, select the node corresponding to the object type you want to compile.
   DBArtisan displays objects of that type in the right pane of the Explorer window.
3. In the right pane of the Explorer window, select a specific database object.
4. On the Datasource Explorer toolbar, click Command, and then select Compile.
   DBArtisan opens the Compile dialog box.
5. Use the following table as a guide to working through the panels of the dialog box

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action options</td>
<td>Compile dependents</td>
</tr>
<tr>
<td></td>
<td>Compile only invalid dependents</td>
</tr>
<tr>
<td></td>
<td>Compile system dependents</td>
</tr>
<tr>
<td></td>
<td>Compile with debug</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Lets you review any dependencies before you proceed.</td>
</tr>
<tr>
<td>Preview</td>
<td>Displays the DDL generated to execute the Checkpoint operation.</td>
</tr>
</tbody>
</table>

6. Use one of the Schedule or Execute buttons to execute the Checkpoint.

Notes on compiling Functions
DBArtisan lets you recompile a function. Oracle first recompiles any invalid objects on which the function depends. In addition, it marks any objects that depend on the function as invalid.

To recompile a function that is part of a package, compile the package itself. DBArtisan uses the ALTER FUNCTION statement to compile a stand-alone function. However, you should not use the ALTER FUNCTION statement to individually recompile a function that is part of a package.

For more information, see:
 Compile
Notes on compiling Java Sources
Oracle lets you compile a Java source. Oracle resolves references to other Java sources.

Notes on compiling Materialized Views
DBArtisan lets you compile materialized views. If a materialized view fails to revalidate after you recompile, that materialized view cannot be fast refreshed ON DEMAND or used for query rewrite.

For more information, see:
Compile
Completing a Dialog Box

Notes on compiling Packages and Package Bodies
DBArtisan lets you recompile a package, and recompiles all package objects together.

Recompiling a package in DBArtisan compiles both the package specification and the package body by issuing two consecutive ALTER statements. However, DBArtisan shows only the ALTER statement for the specification in the Preview: Confirm Compile dialog box. You can recompile only the package body by explicitly compiling the package body itself.

When recompiling the entire package, Oracle recompiles the package even if it is invalid. However, if there are compilation errors, the package remains invalid and Oracle invalidates all dependent objects.

Recompiling only a package body does not invalidate objects that depend upon the package specification regardless of whether or not the package body has compilation errors.

For more information, see:
Compile
Completing a Dialog Box

Notes on compiling Procedures
DBArtisan lets you compile a procedure that is part of a package, by compiling the package itself. DBArtisan uses the ALTER PROCEDURE statement to compile a stand-alone procedure. However, you should not use the ALTER PROCEDURE statement to individually recompile a procedure that is part of a package.

The table below describes the options and functionality on the Confirm Compile dialog box for IBM DB2 for Linux, Unix, and Windows:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare with release option</td>
<td>Compiles the stored procedure into a state that will run optimally in a production environment.</td>
</tr>
<tr>
<td>Compile with debug option</td>
<td>Compiles the stored procedure with the debugging information and symbols required to service step, pause, and breakpoint requests.</td>
</tr>
<tr>
<td>Keep current specific name</td>
<td>Notifies DB2 that the stored procedure will keep its current system assigned name. With this option, DB2 is not required to reset internal references to the target object.</td>
</tr>
</tbody>
</table>
The table below describes the options and functionality on the Confirm Compile dialog box for Oracle:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not compile dependent objects</td>
<td>Compiles only the current object and to ignore the object’s dependencies. This is the default setting.</td>
</tr>
<tr>
<td>Compile dependent objects</td>
<td>Compiles statements for all objects referenced by the object being compiled. For example, if you compile a function that references a specific procedure and you select to compile the dependent objects, an ALTER COMPILE statement is created for that referenced procedure.</td>
</tr>
<tr>
<td>Compile only invalid dependent objects</td>
<td>Creates ALTER COMPILE statements for only those objects that are currently invalid.</td>
</tr>
<tr>
<td>Compile dependent system objects</td>
<td>Compiles all of the referenced objects with the debug option.</td>
</tr>
<tr>
<td>Compile with debug option</td>
<td>Selecting the Compile with debug option check box instructs the Oracle PL/SQL compiler to generate and store the code for use in debugging sessions.</td>
</tr>
</tbody>
</table>

For more information, see:

- [Compile](#)
- [Completing the Compile Dialog Box](#)

**Notes on compiling Types and Type Bodies**

DBArtisan lets you recompile a type. DBArtisan recompiles both the type specification and the type body.

For more information, see:

- [Compile](#)
- [Completing the Compile Dialog Box](#)

**Notes on compiling Views**

DBArtisan lets you recompile a view when you have altered an object referenced by that view.

When you recompile a view, Oracle invalidates all dependant objects.

For more information, see:

- [Compile](#)
- [Completing the Compile Dialog Box](#)

**Connection Editor**

**NOTE:** This functionality is available for all platforms.

The Connection Editor lets you modify plan and package connections.

The table below describes the options and functionality of the Connection Editor:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connections</td>
<td>Displays the connections for the plan or package.</td>
</tr>
</tbody>
</table>
Convert to Partitioned

**NOTE:** This functionality is available for Oracle only.

The Convert to Partitioned Wizard lets you select partitioning methods, columns, and add partitions. Index-organized tables take up less storage space and quickly access table rows. Index-organized tables store rows in primary key order reducing the amount of storage space needed.

**Completing the Convert to Partitioned Wizard**

To complete this wizard, do the following:

1. On the **Datasource Explorer**, select the target object node.
   
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Convert to Partitioned**.
   
   OR

   In the right pane of the application, right-click the target object, and then select **Convert to Partitioned**.

   DBArtisan opens the Convert to Partitioned wizard.

3. Complete the wizard panels and then click **Finish**.

   DBArtisan opens the Preview:Create dialog box.

**Convert to Partitioned Wizard - Panel 1**

The table below describes the options and functionality on this panel of the Convert to Partitioned Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Displays the table owner.</td>
</tr>
<tr>
<td>Table</td>
<td>Displays the table name.</td>
</tr>
<tr>
<td>Tablespace</td>
<td>Lets you select the tablespace.</td>
</tr>
<tr>
<td>Partitioning Method</td>
<td>Lets you select a partitioning method.</td>
</tr>
<tr>
<td>Do you want to enable row movement?</td>
<td>Lets you enable row movement.</td>
</tr>
</tbody>
</table>

For more information, see [Convert to Partitioned Wizard](#).

**Convert to Partitioned Wizard - Panel 2**

The table below describes the options and functionality on this panel of the Convert to Partitioned Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the partitioning columns</td>
<td>Available Columns - Lets you select at least one available column, and then click the right arrow to move the column to the Selected Columns.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Click to add the connection.</td>
</tr>
</tbody>
</table>
Copy Object Names

**NOTE:** This functionality is available for Microsoft SQL Server and Oracle only.

The Copy Object Names functionality lets you copy and then paste object name(s) into other applications.

For more information, see [Completing a Dialog Box](#).

Copy Schema

For IBM DB2 for Windows, Unix, and Linux (version 9), the Copy Schema function lets you quickly make copies of a database schema and its corresponding objects.

**To copy a schema:**

1. In the datasources pane, right-click a datasource and select **Copy Schema** from the context menu.
2. Use the following table as a guide to understanding and choosing options the options in the Copying Schema dialog:

<table>
<thead>
<tr>
<th>Pane</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action Options</strong></td>
<td>Lets you specify Source Schema, Target Schema, Copy Mode, Log filename, Object Owner, Error table schema, Error table name, and Drop error table after action execution options.</td>
</tr>
<tr>
<td><strong>Tablespace mapping</strong></td>
<td>Lets you specify source and target tablespace options.</td>
</tr>
</tbody>
</table>

3. **Execute** or **Schedule** the action.

Copy to Copy

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.
The Copy to Copy dialog box lets you make image copies. Starting with either the local primary or recovery site primary copy, the Copy to Copy dialog box can make up to three copies of the following types of copies:

- Local primary
- Local backup
- Recovery site primary
- Recovery site backup

The copies are used by the RECOVER utility when recovering a table space or index space to the most recent time or to a previous time. These copies can also be used by MERGECOPY, UNLOAD, and possibly a subsequent Copy to Copy execution.

The table below describes the options and functionality on the Copy to Copy dialog box:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>Object Grid</td>
<td>Displays the target tablespaces or indexes. You can select multiple items in the list. The Partition column is an editable. You can add the partition numbers in this column, separated by commas.</td>
</tr>
<tr>
<td></td>
<td>Add</td>
<td>Opens the Object Selector dialog box.</td>
</tr>
<tr>
<td></td>
<td>Remove</td>
<td>Deleted the selected objects.</td>
</tr>
<tr>
<td>Utility ID</td>
<td></td>
<td>Lets you type the utility id for the job. DBArtisan identifies a single job by its utility identifier, the value given for the UID parameter when the job was created. Max size is 16 characters.</td>
</tr>
<tr>
<td>Options</td>
<td>From Last Image Copy</td>
<td>The most recent image copy that was taken for the tablespace or index space to be the input to COPYTOCOPY job. This could be a full or incremental copy.</td>
</tr>
<tr>
<td></td>
<td>From Last Full Image</td>
<td>The most recent full image copy that was taken for the object to be the input to COPYTOCOPY job.</td>
</tr>
<tr>
<td></td>
<td>Copy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>From Last Incremental</td>
<td>The most recent incremental image copy that was taken for the object to be the input to COPYTOCOPY job.</td>
</tr>
<tr>
<td></td>
<td>Image Copy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>From Specific Image</td>
<td>The particular image copy data set (dsn) as the input to COPYTOCOPY job.</td>
</tr>
<tr>
<td></td>
<td>Copy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Volume</td>
<td>Identifies the image copy data set if it is not a GDG and there is more than one copy with the same name.</td>
</tr>
<tr>
<td></td>
<td>Catalog</td>
<td>Select for an image copy that was created as a cataloged data set.</td>
</tr>
<tr>
<td></td>
<td>Volume serial number</td>
<td>Identifies the data set by an alphanumeric volume serial identifier of its first volume. Use this option only for an image copy that was created as a noncataloged data set.</td>
</tr>
<tr>
<td></td>
<td>From sequence number</td>
<td>Identifies the image copy data set by its file sequence number.</td>
</tr>
<tr>
<td>Data Sets</td>
<td>Local Site Primary Copy</td>
<td>Lets you specify a dataset name for the local site primary image copy. This relates to the copydsn1 parameter in the stored procedure.</td>
</tr>
<tr>
<td></td>
<td>Local Site Backup Copy</td>
<td>Lets you specify the device for the image copy. This relates to the copydev1 parameter in the stored procedure.</td>
</tr>
<tr>
<td></td>
<td>Recovery Site Primary</td>
<td>Lets you specify the primary space allocation amount for the image copy. This relates to the copyspaces1 parameter in the stored procedure.</td>
</tr>
<tr>
<td></td>
<td>Copy</td>
<td>Lets you specify a dataset name for the local site backup image copy. This relates to the copydsn2 parameter in the stored procedure.</td>
</tr>
</tbody>
</table>
Object Selector Dialog Box
The table below describes the options and functionality on the panel of the Object Selector dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Like</td>
<td>Lets you type the pattern for querying the tablespaces. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Tablespace Like</td>
<td>Lets you type the pattern for querying the tablespaces. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Creator Like</td>
<td>Lets you type the pattern for querying the creator. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Index Like</td>
<td>Lets you type the pattern for querying the index. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Match Case</td>
<td>Lets you specify match case.</td>
</tr>
<tr>
<td>Object Grid</td>
<td>Lets you select one or more rows in this list.</td>
</tr>
<tr>
<td>Query</td>
<td>Repopulates grid based on the object query.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all rows in Object Grid.</td>
</tr>
<tr>
<td>Unselect All</td>
<td>Unselects all rows in Object Grid.</td>
</tr>
</tbody>
</table>

For more information, see Copy to Copy.
Create Alias

The Create Alias dialog box lets you create an alternate name for a table or view without requiring you to know any of the underlying SQL commands. The dialog box constructs the necessary CREATE ALIAS statement based on the information you supply.

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows and IBM DB2 for OS/390 and z/OS.

**To create an alias for a table or view:**

1. Connect to the datasource that contains the table or view. For more information, see Datasource Management.
2. On the left pane of the Datasource Explorer, select the table or view node.
   DBArtisan displays objects of that type in the right pane of the Explorer window.
3. In the right pane of the Explorer window, select a specific table or view.
4. On the Datasource Explorer tool bar, click Command, and then select Create Alias.
   DBArtisan opens a Create Alias wizard.

For information on working with Create Alias wizards, see the following topics:
- Alias Wizard for IBM DB2 for Linux, Unix, and Windows
- Alias Wizard for IBM DB2 for OS/390 and z/OS

Create Insert Statements

**NOTE:** This functionality is available for all platforms.

The Create Insert Statements dialog box lets you create Insert Statements for selected columns and data.

The table below describes the options and functionality on the Create Insert Statements dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>Lets you select columns for a Select statement.</td>
</tr>
<tr>
<td>Where</td>
<td><strong>OPTIONAL:</strong> Let you enter a WHERE statement.</td>
</tr>
<tr>
<td>Include owner information in insert statements</td>
<td>Select to include the owner information.</td>
</tr>
<tr>
<td>Set row count</td>
<td>Lets you specify a row count, the number of rows in a table that were affected by the Insert statement executed against the table, or a view based on the table.</td>
</tr>
</tbody>
</table>

**Completing the Create Insert Statements Dialog Box**

To complete this dialog box, do the following:

1. In the right pane of the application, select the **Tables** node.
   DBArtisan displays the Tables in the right pane of the Explorer window.
2 On the **Datasource Explorer** tool bar, click **Command**, and then select **Create Insert Statements**.

   OR

   In the right pane of the application, right-click the table, and then select **Create Insert Statements**.

   DBArtisan opens the Create Insert Statements dialog box.

3 In **Columns**, select columns for a SELECT statement.

4 **OPTIONAL:** In **Where**, lets you enter a WHERE statement.

5 Click **OK**.

### Create or Edit Java Source

**NOTE:**  This functionality is available for Oracle only.

The Java Editor lets you enter Java code. The table below describes the Java Editor toolbar options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock/Unlock Connection</td>
<td>Click to lock or unlock connection.</td>
</tr>
<tr>
<td>Create</td>
<td>Click to open the Create Options dialog box, which lets you select the options for creating the java source.</td>
</tr>
<tr>
<td>Errors</td>
<td>Click to split the workspace in half, displaying the error messages in the lower half of the workspace.</td>
</tr>
</tbody>
</table>

When you add a script in a new file, you not only choose a name for the file, but choose among create options. Finally, when you complete your script, you can **Preview**, **Schedule** or immediately **Execute** it.

### Create Like

The Create Like Editor lets you create a new object based on an existing object. The following table shows availability of the Create Like function for particular object types by DBMS:

<table>
<thead>
<tr>
<th>IBM DB2 for Windows, Unix, and Linux</th>
<th>IBM DB2 for OS/390 and Z/OS</th>
<th>Microsoft SQL Server</th>
<th>Oracle</th>
<th>Sybase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Materialized Query Tables</td>
<td>Tables</td>
<td>Logins</td>
<td>Tables</td>
<td>Logins</td>
</tr>
<tr>
<td>Tables</td>
<td></td>
<td>Tables</td>
<td>Tables</td>
<td>Tables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Users</td>
<td></td>
<td>Users</td>
</tr>
</tbody>
</table>

**To create an object based on another object:**

1 Connect to the datasource that contains the object you want to copy. For more information, see **Datasource Management**.

2 On the left pane of the **Datasource Explorer**, expand nodes until the target object type is visible and then click the taget object type.

   DBArtisan displays objects of that type in the right pane of the Explorer window.

3 In the right pane of the Explorer window, select the specific object you want to copy.
4 On the **Datasource Explorer** tool bar, click **Command**, and then select **Create Like**.

DBArtisan opens a Create Like wizard.

5 Provide a name for the new object.

6 Modify settings on each panel of the wizard as required.

For particular object types, tasks and settings on the panels of the Create Like wizard are similar to those for the object editors for that object type. For information, see:

- **Using the Logins Editor (Microsoft SQL Server, Sybase)**
- **Using the Materialized Query Tables Editor (IBM DB2 for Linux, Unix, and Windows)**
- **Using the Tables editor**
- **Using the Users editor**

7 Click **Execute** to create the new object.

### Create or Edit User Accounts

**NOTE:** This functionality is available for all platforms.

The Create or Edit User Accounts dialog box lets you manage user accounts.

The table below describes the options and functionality on the Create or Edit User Accounts dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>Lets you type the user name.</td>
</tr>
<tr>
<td>Database roles</td>
<td>Lets you select the roles for the user account.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

### Create or Modify Constraint

**NOTE:** This functionality is available for all platforms.

The Create/Modify Constraint Dialog Box lets you create or edit a constraint without knowing the underlying commands.

For more information, see [Completing a Dialog Box](#).

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td><strong>Table Owner or Table Schema</strong> Select the owner of the table for which the check constraint is to be created.</td>
</tr>
<tr>
<td>Table</td>
<td>Lets you select where you want to place the check constraint.</td>
</tr>
<tr>
<td>Name</td>
<td>Lets you type the name of the constraint, which can be up to 30 characters long and must be unique across the entire database.</td>
</tr>
</tbody>
</table>
Create or Modify User Message Text

**NOTE:** This functionality is available for all platforms.

The Create or Modify User Message Text dialog box lets you manage user messages. You can open this dialog box in the following wizards:

- Microsoft SQL Server User Message Wizard
- Sybase ASE User Message Wizard

The table below describes the options and functionality on the Create or Modify User Message Text dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Lets you specify the message language.</td>
</tr>
<tr>
<td>Message Text</td>
<td>Lets you type a message.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

Create Synonym

**NOTE:** The Create Synonym functionality is available for IBM DB2 for Linux, Unix, and Windows, OS/390, and Oracle.

The Create Synonym wizard lets you create synonyms for objects.

A synonym is an alternate name for an object in the database. You can define synonyms on tables, views, sequences, procedures, functions, packages, package bodies, materialized views, and Java class objects.

There are two types of synonyms:

- Public
- Private
Public synonyms can be used by everyone in the database. Private synonyms belong to the synonym owner and the owner can control the synonym’s availability to others.

Synonyms simplify object names and provide added security. Synonyms let you:

- Simplify SQL statements.
- Reference an object without needing to specify its owner.
- Reference a remote object without needing to specify its database.
- Alias an object so its purpose becomes more understandable.
- Mask the name, owner, and location of an object.

**NOTE:** The schema object for which you define a synonym cannot be contained in a package.

DBArtisan lets you specify options for creating synonyms with the Create Synonym and Create Synonyms wizards. The wizard that DBArtisan returns depends on whether you are creating a synonym for a single object or are creating synonyms for multiple objects at the same time.

To create a synonym for one or more objects:

1. Connect to the datasource where you want to create a synonym. For more information, see Datasource Management.
2. On the Datasource Explorer, expand the target datasource.
3. Continue to expand folders under the datasource until the type of object you want to create is visible.
4. On the Datasource Explorer, click the specific object type. The right-hand pane is populated with all objects of that type.
5. In the right-hand pane, select one or more objects.
6. On the Datasource Explorer tool bar, click Command, and then select Create Synonym.
7. If you selected a single object, use the table below as a guide to completing the steps in this dialog:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Owner</td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Referenced Object Owner, Type, and Name</td>
<td></td>
</tr>
<tr>
<td>Database Link</td>
<td></td>
</tr>
<tr>
<td>DDL View</td>
<td></td>
</tr>
</tbody>
</table>
8 If you selected multiple objects, use the table below as a guide to completing the steps in this dialog:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Options</td>
<td>For each object in the Synonyms... list, select the object and enable or disable Public scope for the synonym, and if you leave Public disabled, choose the Schema in which to create the synonym.</td>
</tr>
<tr>
<td>Preview</td>
<td>Preview the DDL and if necessary navigate backward through the steps to make corrections. Finally, use the Execute button to Create the new object.</td>
</tr>
</tbody>
</table>

### DataLink Options

**NOTE:** This functionality is available for IBM DB2 UBB for OS/390 and z/OS only.

The DataLink Options dialog box lets you manage datalinks, datatypes that enable logical references from the database to a file that is stored outside the database.

The table below describes the options and functionality on the DataLink Options dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datalink Options</td>
<td>Lets you select datalink options:</td>
</tr>
<tr>
<td></td>
<td>No Link Control</td>
</tr>
<tr>
<td></td>
<td>File Link Control</td>
</tr>
<tr>
<td></td>
<td>File Link Options</td>
</tr>
<tr>
<td>File-Link Options</td>
<td>Lets you select file-link options:</td>
</tr>
<tr>
<td></td>
<td>Read</td>
</tr>
<tr>
<td></td>
<td>On Unlink</td>
</tr>
<tr>
<td></td>
<td>Write Permission</td>
</tr>
<tr>
<td></td>
<td>Recovery</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

### DBCC

**NOTE:** This functionality is available for Microsoft SQL Server and Sybase ASE only.

DBArtisan’s DBCC (Database Consistency Check) function box lets you:

- Specify single or multiple tables or indexes for validation.
- Perform database-wide validations.
- Perform object-level validations of databases.

DBCC options vary by the type of object and DBMS. Before performing a Database Consistency Check, see the relevant topic:

- [DBCC for Microsoft SQL Server Databases](#)
- [DBCC for Microsoft SQL Server Tables](#)
- [DBCC for Microsoft SQL Server Indexes](#)
• **DBCC for Sybase ASE Databases**
• **DBCC for Sybase ASE Tables**
• **DBCC for Sybase ASE Indexes**

**To perform a Database Consistency Check:**

1. Connect to the datasource that contains the object. For more information, see [Datasource Management](#).
2. On the left pane of the **Datasource Explorer**, expand the datasource nodes until the target object type is displayed.
3. Select the target object type. 

   DBArtisan displays objects of that type in the right pane of the Explorer window.
4. In the right pane of the Explorer window, select the specific object.
5. If you are checking consistency of a table or index, on the **Datasource Explorer** tool bar, click **Command**, and then select **DBCC**.

   OR

6. If you are checking consistency of a database, on the **Datasource Explorer** tool bar, click **Command**, then select **DBCC**, and then select one of the DBCC options listed in [DBCC for Microsoft SQL Server Databases](#) or [DBCC for Sybase ASE Databases](#).

   DBArtisan opens a dialog that lets you specify DBCC options.
7. Select options for the operation and when complete, use the **Schedule** or **Execute** button to complete the operation.

**DBCC for Microsoft SQL Server**

The DBCC (Database Consistency Check) dialog box lets you specify single or multiple tables or indexes for validation in Microsoft SQL Server. Use this dialog box to perform table-level or index-level validations of databases which are too large to undergo database-level DBCC operations in a time-efficient manner.

The DBCC dialog box includes the following elements:

- A window displaying the target database objects
- A drop-down list of DBCC Operations
- Buttons for previewing the operation’s SQL code, scheduling the operation, and executing the operation

For more information, see:

- [DBCC for Databases](#)
- [DBCC for Tables](#)
- [DBCC for Indexes](#)

**DBCC for Microsoft SQL Server Databases**

The DBCC dialog box for databases lets you perform database-wide validations. You should validate your databases as part of regular database maintenance to guard against corruption and failure. Microsoft SQL Server offers a set of DBCC commands to validate the integrity of your databases. Generally, you should perform these DBCC commands prior to dumping your databases to ensure that you are capturing clean backups of your databases.
The fundamental difference between the DBCC dialog box for databases, tables and indexes is the content of the DBCC Operation drop-down list.

The table below describes the options and functionality on the DBCC dialog box.

<table>
<thead>
<tr>
<th>DBCC Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Allocation</td>
<td>Executes a DBCC CHECKALLOC command. Makes sure that all data and index panels are correctly allocated and used. It reports on the amount of space allocated and used in the database. When checking allocation, you have the option to skip non-clustered indexes by selecting the Skip non-clustered indexes check box.</td>
</tr>
<tr>
<td>Check Catalog</td>
<td>Executes a DBCC CHECKCATALOG command. Checks for consistency in and between system tables.</td>
</tr>
<tr>
<td>Check Database</td>
<td>Executes a DBCC CHECKDB command. Verifies that all tables and indexes are properly linked, that indexes are in proper sorted order, that all pointers are consistent, that the data on each panel is reasonable, and that panel offsets are reasonable. When checking a database, you have the option to skip non-clustered indexes by selecting the Skip non-clustered indexes check box.</td>
</tr>
<tr>
<td>Check FileGroup</td>
<td>Executes a DBCC CHECKFILEGROUP command. Verifies that all tables and indexes for the specified filegroup are properly linked, that indexes are in proper sorted order, that all pointers are consistent, that the data on each panel is reasonable, and that panel offsets are reasonable. When checking filegroups, you have the option to skip non-clustered indexes by selecting the Skip non-clustered indexes check box.</td>
</tr>
<tr>
<td>Show Oldest Transaction</td>
<td>Executes a DBCC OPENTRAN command. Displays information on the oldest active transaction and the oldest distributed and non distributed replicated transactions, if any, within the specified database.</td>
</tr>
<tr>
<td>Update Usage</td>
<td>Executes a DBCC UPDATEUSAGE command. Reports and corrects the rows, used, reserved, and dpanels columns of the sysindexes table for any clustered indexes on objects of the type U (user-defined table) or S (system table).</td>
</tr>
</tbody>
</table>

For more information, see:

DBCC

**DBCC for Microsoft SQL Server Tables**

The DBCC dialog box for tables lets you perform table-level validations of databases. The fundamental difference between the DBCC dialog box for tables and indexes is the content of the DBCC Operation drop-down list.

The table below describes the options and functionality on the DBCC dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Current Identity Value</td>
<td>Checks the current identity value for the target objects, correcting values if needed depending on parameter specifications. Identity columns created with a NOT FOR REPLICATION clause in either the CREATE TABLE or ALTER TABLE statement are not corrected by this operation.</td>
</tr>
<tr>
<td>Check Fragmentation</td>
<td>Displays the target table's data and index fragmentation information, determining whether the table is heavily fragmented. When a table is heavily fragmented, you can reduce fragmentation and improve read performance by dropping and recreating a clustered index (without using the SORTED_DATA option). Doing so reorganizes the data, resulting in full data pages. To adjust the level of fullness, use the Rebuild Index operation's FILLFACTOR option. When INSERT, UPDATE, and DELETE statements fragment tables, they usually do so with unequal distribution across the entire database so that each page varies in fullness over time, forcing additional page reads for queries that scan part or all of a table.</td>
</tr>
</tbody>
</table>
For more information, see:

**DBCC**

**DBCC for Microsoft SQL Server Indexes**

The DBCC dialog box for indexes lets you perform index-level validations of databases. The fundamental difference between the DBCC dialog box for tables and indexes is the content of the DBCC Operation drop-down list.
The table below describes the options and functionality on the DBCC dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Fragmentation</td>
<td>Displays the target index’s table data and fragmentation information, determining whether the table is heavily fragmented. For more information, see tip below. When a table is heavily fragmented, you can reduce fragmentation and improve read performance by dropping and recreating a clustered index (without using the SORTED_DATA option). Doing so reorganizes the data, resulting in full data pages. To adjust the level of fullness, use the Rebuild Index operation’s FILLFACTOR option. When INSERT, UPDATE, and DELETE statements fragment tables, they usually do so with unequal distribution across the entire database so that each page varies in fullness over time, forcing additional page reads for queries that scan part or all of a table.</td>
</tr>
<tr>
<td>Check Index</td>
<td>Checks the linkages and sizes of text and image pages for selected indexes. DBCC CHECKTABLE requires a shared lock on all tables and indexes in the database for the duration of the operation.</td>
</tr>
<tr>
<td>Rebuild Index</td>
<td>Dynamically rebuilds all target indexes, allowing those which enforce either primary key or unique constraints to be rebuilt without need for dropping and recreating. This operation is not supported for use on system tables.</td>
</tr>
<tr>
<td>Show Statistics</td>
<td>Displays the current distribution statistics for the target indexes. The results returned indicate the selectivity of each target index (a lower density equals a higher selectivity) and provide the basis for determining the usefulness of target indexes to the optimizer.</td>
</tr>
<tr>
<td>Update Usage</td>
<td>Reports and corrects inaccuracies in the sysindexes table (which can result in incorrect space usage reports by the sp_spaceused system stored procedure) and corrects the rows, used, reserved, and dpages columns of the sysindexes table for tables and clustered indexes. If there are no inaccuracies in sysindexes, DBCC UPDATEUSAGE returns no data. Use this operation to synchronize space-usage counters. Executing this operation on large tables or databases can require some time, so it should typically be used only when you suspect incorrect values returned by sp_spaceused. Additional Options: Update Index Option, and DBCC General Option.</td>
</tr>
</tbody>
</table>

For more information, see:

DBCC

**DBCC for Sybase ASE**

The DBCC (Database Consistency Check) dialog box lets you specify single or multiple databases, tables or indexes for validation in Sybase ASE. Use this dialog box to perform table-level or index-level validations of databases which are too large to undergo database-level DBCC operations in a time-efficient manner.

The DBCC dialog box includes the following elements:

- A window displaying the target database objects
- A drop-down list of DBCC Operations
- Buttons for previewing the operation’s SQL code, scheduling the operation, and executing the operation

For more information, see:

DBCC for Tables
DBCC for Indexes
DBCC for Databases
DBCC for Sybase ASE Databases

The DBCC dialog box for databases lets you perform database-wide validations. The fundamental difference between the DBCC dialog box for databases, tables and indexes is the content of the DBCC Operation drop-down list.

The table below describes the options and functionality on the DBCC dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Allocation</td>
<td>Checks the allocation and use of all pages in the target database.</td>
</tr>
<tr>
<td>Check Catalog</td>
<td>Checks for consistency in and between system tables in the target database.</td>
</tr>
<tr>
<td>Check Database</td>
<td>Checks the allocation and structural integrity of all the objects in the target database.</td>
</tr>
<tr>
<td>Check Storage</td>
<td>Checks the target database for allocation, OAM page entries, page consistency, text valued columns, allocation of text valued columns, and text column chains. The results of this operation are stored in the dbccdb database.</td>
</tr>
<tr>
<td>Database Repair</td>
<td>Drops a damaged database.</td>
</tr>
</tbody>
</table>

For more information, see: [DBCC](#)

DBCC for Sybase ASE Tables

The DBCC dialog box for tables lets you perform table-level validations of databases. The fundamental difference between the DBCC dialog box for tables and indexes is the content of the DBCC Operation drop-down list.

The table below describes the options and functionality on the DBCC dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Allocation</td>
<td>Checks the database to see that every page is correctly allocated, and that no allocated page is unused. Use TABLEALLOC frequently (daily) to check page linkages in the Adaptive Server before performing a database dump to ensure the integrity of the dumped data.</td>
</tr>
<tr>
<td>Check Table</td>
<td>Checks the linkages and sizes of text, ntext and image pages for selected tables. For the data, index, text, ntext, and image pages of the target tables, this operation also checks that index and data pages are correctly linked, indexes are in their proper sorted order, pointers are consistent, the data on each page is reasonable, and the page offsets are reasonable. DBCC CHECKTABLE requires a shared lock on all tables and indexes in the database for the duration of the operation. However, DBCC CHECKTABLE does not check the allocations of pages in the specified table (for this, use DBCC CHECKALLOC). To perform DBCC CHECKTABLE on every table in the database, use DBCC CHECKDB.</td>
</tr>
<tr>
<td>Check Text</td>
<td>Upgrades text values after you have changed an Adaptive Server's character set to a multibyte character set.</td>
</tr>
<tr>
<td>Rebuild Index</td>
<td>Dynamically rebuilds one, multiple, or all indexes for a table in the target database, allowing indexes which enforce either primary key or unique constraints to be rebuilt without need for dropping and recreating. This operation is not supported for use on system tables.</td>
</tr>
</tbody>
</table>

DBCC Operation Options

DBArtisan offers additional options for selected operations which you can specify to further customize a database consistency check. The table below describes each option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Option</td>
<td>Click Fix Error to instruct DBArtisan to fix any allocation errors it finds. You must put your database in single-user mode to fix errors, so specify this option during times of low usage.</td>
</tr>
</tbody>
</table>
For more information, see:

**DBCC**

**DBCC for Sybase ASE Indexes**

The DBCC dialog box for indexes lets you perform index-level validations of databases. Unlike the DBCC dialog box for tables, this DBCC dialog box offers only one option on the DBCC Operation drop-down list: Check Allocation. This option checks the specified database to see that all pages are correctly allocated and that no allocated page is unused.

The table below describes the options and functionality on the DBCC dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCC Option</td>
<td>Checks the specified database to see that all pages are correctly allocated and that no page that is allocated is not used.</td>
</tr>
<tr>
<td>Error Option</td>
<td>DBArtisan to fixes any allocation errors it finds. You must put your database in single-user mode to fix errors, so specify this option during times of low usage</td>
</tr>
<tr>
<td>Job Scope</td>
<td>Produces a report based on the allocation pages listed in the object allocation map (OAM) pages for the table. It does not report and cannot fix unreferenced extents on allocation pages that are not listed in the OAM pages. The optimized option is the default. A full job is the equivalent to a table-level CHECKALLOC, reporting all types of allocation errors. A fast job does not produce an allocation report, but produces an exception report of pages that are referenced but not allocated in the extent.</td>
</tr>
</tbody>
</table>

For more information, see:

**DBCC**

**Deallocate Unused Space**

**NOTE:** This functionality is available for Oracle only.

The Deallocate Unused Space dialog box lets you deallocate space from clusters, indexes, and tables. You can also deallocate unused space from table partitions and subpartitions. When you find that allocated space is not being used, you can free that space for use by other objects by explicitly deallocating space. Oracle releases the freed space to the user quota for the tablespace in which the deallocation occurs.

Oracle deallocates unused space from the end of the object toward the high water mark. In other words, Oracle frees space starting with the space that would have been used last. If an extent is completely contained in the space to be deallocated, then the whole extent is freed. If an extent is only partially contained in the space to be deallocated, then Oracle shrinks that extent to the size of the used space up to the high water mark, and frees the unused space in the extent.
If you are deallocating unused space from an index and the index is range-partitioned or hash-partitioned, Oracle deallocates unused space from each partition in the index. If an index is a local index on a composite-partitioned table, Oracle deallocates unused space from each of the subpartitions in the index.

The table below describes the options and functionality on the Deallocate Unused Space dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the number of bytes above the high-water mark that the objects will have after deallocation. If no value is specified, all unused space will be freed.</td>
<td>If you do not specify an amount of unused space and the high water mark is above the size of INITIAL and MINEXTENTS, then all of the unused space is freed. If the high water mark is less than the size of INITIAL or MINEXTENTS, then all unused space above MINEXTENTS is freed. If you specify an amount of unused space and the remaining number of extents is smaller than MINEXTENTS, then the MINEXTENTS value changes to reflect the new number. If the initial extent becomes smaller as a result of the deallocation, the INITIAL value changes to reflect the new size of the initial extent.</td>
</tr>
</tbody>
</table>

**TIP:** You can verify that the deallocated space is freed by going to the Space Tab in the appropriate DBArtisan editor.

**Completing the Deallocate Unused Space Dialog Box**

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the Tables node.
   
   DBArtisan displays the tables in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Deallocate Unused Space**.
   
   OR

   In the right pane of the application, right-click the table, and then select **Deallocate Unused Space**.

   DBArtisan opens the Deallocate Unused Space dialog box.

3. Specify the number of bytes above the high-water mark that the objects will have after deallocation. If no value is specified, all unused space will be freed.

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the Impact Analysis dialog box. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see **Completing a Dialog Box**.
Detach/Attach

NOTE: This functionality is available for Microsoft SQL Server 2000 or later only.

The Detach/Attach dialog box lets you detach and reattach data and transaction log files. Detaching a database removes the database from the server but leaves the database intact within the data and transaction log files that compose the database. These data and transaction log files can then be used to attach the database to any instance of Microsoft SQL Server, including the server from which the database was detached. This makes the database available in exactly the same state it was in when it was detached. The DBArtisan commands are:

- Detach
- Attach

Important Notes
For more information, see Completing the Detach Dialog Box.

Detaching a Database
The table below describes the options and functionality on the Detach dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip Checks</td>
<td>Select to skip the UPDATE STATISTICS operation when detaching the database. This option is useful for databases that are to be moved to read-only media.</td>
</tr>
</tbody>
</table>

Completing the Detach Dialog Box
To complete this dialog box, do the following:

1. On theDatasource Explorer, select the Databases node.
   DBArtisan displays the Databases in the right pane of the Explorer window.

2. On theDatasource Explorer tool bar, click Command, and then select Detach.
   OR
   In the right pane of the application, right-click the database, and then select Detach.
   DBArtisan opens the Detach dialog box.

3. To skip the UPDATE STATISTICS operation when detaching the database, select the Skip Checks check box for the target database(s).
   TIP: This option is useful for databases that are to be moved to read-only media.

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
</tbody>
</table>
Attaching a Database
The table below describes the options and functionality on the Attach dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name to be Attached</td>
<td>Lets you type the name of the target database to attach the file(s) to.</td>
</tr>
<tr>
<td>Add Button</td>
<td>Click to add database file(s), and then type the name of the MDF (master data file) of the database to attach.</td>
</tr>
</tbody>
</table>

Completing the Attach Dialog Box
To complete this dialog box, do the following:

1. On the Datasource Explorer, select the Databases node.
   DBArtisan displays the Databases in the right pane of the Explorer window.
2. On the Datasource Explorer tool bar, click Command, and then select Attach.
   OR
   In the right pane of the application, right-click the database, and then select Attach.
   DBArtisan opens the Attach dialog box.
3. In the Database Name to be Attached box, type the database name.
4. In the grid, do one of the following:
   Microsoft SQL Server 2000 or 2005
   - Select the target database file(s).
   - To add database file(s), click Add and then enter the name of the MDF (master data file) of the database to attach.
   DBArtisan automatically adds the appropriate *.ldf file.
5. To drop database file(s), click Drop and then select the target file(s).
6. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Preview]</td>
<td>Opens the Preview dialog box.</td>
</tr>
</tbody>
</table>
The Disable/Enable dialog box lets you enable or disable any job queue. Job Queues are built-in mechanisms that let you schedule a variety of SQL-based or command-line driven tasks.

Completing the Enable/Disable Dialog Box for Job Queues
To complete this dialog box, do the following:

1. On the Datasource Explorer, select the Job Queues node.
   DBArtisan displays the job queues in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select Enable or Disable.
   OR
   In the right pane of the application, right-click the target job queue, and then select Enable or Disable.
   DBArtisan opens the Enable/Disable dialog box.

3. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the Impact Analysis dialog box. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see Completing a Dialog Box.
Enable/Disable Automatic Stored Outline Creation/Use Dialog Box

The table below describes the options and functionality on the Enable/Disable Automatic Stored Outline Creation/Use dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable automatic stored outline creation</td>
<td>If selected, Oracle does not automatically create a stored outline for every query submitted.</td>
</tr>
<tr>
<td>Enable automatic stored outline creation</td>
<td>If selected, Oracle automatically creates a stored outline for every query submitted on the system. Category - Lets you select an outline category. No Override - If selected, outlines will not override.</td>
</tr>
</tbody>
</table>

Completing the Enable/Disable Automatic Stored Outline Creation/Use Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Outlines** node.
   
   DBArtisan displays the Outlines in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Disable/Enable**.
   
   OR

3. In the right pane of the application, right-click the outline, and then select **Disable/Enable**.
   
   DBArtisan opens the Enable/Disable Automatic Stored Outline Creation dialog box.

4. Select **Disable automatic stored outline creation** to not automatically create a stored outline for every query submitted on the system.

5. Select **Enable automatic stored outline creation** to automatically create a stored outline for every query submitted on the system.

6. In **Category** select an outline category.

7. In **No Override** select to not allow outline overrides.

8. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <a href="#">Preview dialog box</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <a href="#">Scheduling</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the <a href="#">Impact Analysis dialog box</a>. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).
Disk Resize

**NOTE:** This functionality is available for Sybase ASE 12.5.01 only.

The Disk Resize dialog box lets you dynamically increase the size of database devices without having to add and configure additional database devices.

The table below describes the options and functionality on the Disk Resize dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Increase current size by:  | Lets you type the amount in server pages to increase the device(s). 512 pages equals 1MB.  
**NOTE:** Whether the Sybase database is configured for 2K, 4K, or 8K page sizes does not impact the sizing. |

### Completing the Disk Resize Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Database Devices** node.
   
   DBArtisan displays the Database Devices in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Disk Resize**.

   **OR**

   In the right pane of the application, right-click the database device, and then select **Disk Resize**.

   DBArtisan opens the Disk Resize dialog box.

3. In **Increase current size by:** type the amount in server pages to increase the device(s). 512 pages equals 1MB.

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Preview.png" alt="Preview" /></td>
<td>Opens the <strong>Preview dialog box</strong>.</td>
</tr>
<tr>
<td><img src="Schedule.png" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <a href="#">Scheduling</a>.</td>
</tr>
<tr>
<td><img src="Execute.png" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="Analysis.png" alt="Analysis" /></td>
<td>Opens the <strong>Impact Analysis dialog box</strong>. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

### Drop

**NOTE:** This functionality is available for all platforms.

The Confirm Drop dialog box lets you drop one or more database objects and remove their definition from the system catalog. The Confirm Drop dialog box lets you:

- Drop the object(s) immediately
• Schedule dropping the object(s)
• Analyze the impact of dropping the object(s)
• Preview the SQL to drop the object(s)
• Save the SQL to drop the object(s)
• Print the SQL to drop the object(s)
• Send an e-mail attachment with the SQL to drop the object(s)

Completing the Confirm Drop Dialog Box
To complete this dialog box, do the following:

1. On the Datasource Explorer, select the target object node.
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select Drop.
   OR
   In the right pane of the application, right-click the target object, and then select Drop.
   DBArtisan opens the Confirm Drop dialog box on the Action Options pane.

3. Navigate to the Dependencies pane to view database objects that are dependent on the object you are dropping.
   NOTE: For Microsoft SQL Server 2005, when dropping Logins, you also have the option to delete corresponding user objects.
   NOTE: For Microsoft SQL Server, when dropping Databases, you also have the option to delete the backup and restore history for the database.

4. Navigate to the Preview pane to verify that the correct DDL was created, and if necessary navigate backward and modify your choices.

5. Click Execute to drop the object.

Edit Data
NOTE: This functionality is available for all platforms.

The Edit Data function opens the Data Editor. You can use the Data Editor to edit your tables in real-time. The Data Editor supports all editable datatypes and is an alternative way to add, edit, or delete data from your tables.

NOTE: You can use Data Editor within Query Builder to edit data in tables while you create SELECT statements. You can open multiple Data Editor sessions so that you can continue to change your data until you find the best match query.

The Data Editor includes a Data Editor Filter that lets you select the columns in your table that you want to edit. You must select at least one column to use the Data Editor. The Data Editor Filter is not available for the Query Builder.

TIP: You can customize Data Editor options in the Options Editor.

For more information, see:
Data Editor Design
Using Data Editor

Error
The Error message displays a warning. After reading the warning, click Close to continue.

Estimate Size

**NOTE:** This functionality is available for Oracle and Sybase ASE only.

The Estimate Size dialog box for tables and indexes lets you estimate how large a table or index will become given a row growth projection. The results let you proactively plan your space-related object needs.

Estimate Table Sizes
The table below describes the options and functionality on the Estimate Size dialog box for tables:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Lets you select the table database.</td>
</tr>
<tr>
<td>Owner</td>
<td>Lets you select the table owner.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Lets you select the table by name.</td>
</tr>
<tr>
<td>Number of Rows</td>
<td>Lets you type number of rows to estimate size.</td>
</tr>
<tr>
<td></td>
<td>Click Calculator to open a calculator.</td>
</tr>
<tr>
<td>Estimate Table Size (KB)</td>
<td>Lets you type table size to estimate size.</td>
</tr>
<tr>
<td>Add Table</td>
<td>Click to add a table to the grid.</td>
</tr>
<tr>
<td>Remove Table</td>
<td>Click to remove a table from the grid.</td>
</tr>
<tr>
<td>Estimate Size</td>
<td>Click to start the estimate size functionality. DBArtisan then calculates the information, and displays it in a browser-ready HTML report that you can print or save.</td>
</tr>
<tr>
<td>Update Statistics</td>
<td>Opens the Update Statistics dialog box.</td>
</tr>
<tr>
<td>Save As</td>
<td>Lets you save as a result grid (.rsl) file.</td>
</tr>
</tbody>
</table>

**TIP:** To generate an HTML report, click the Report button.

For more information, see Completing a Dialog Box.
Estimate Index Sizes

The table below describes the options and functionality on the Estimate Size dialog box for indexes:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Lets you select the table database.</td>
</tr>
<tr>
<td>Owner</td>
<td>Lets you select the table owner.</td>
</tr>
<tr>
<td>Table Name</td>
<td>Lets you select the table.</td>
</tr>
<tr>
<td>Index Name</td>
<td>Lets you select the index.</td>
</tr>
<tr>
<td>Fill Factor</td>
<td>Lets you specify a percentage of how full each index page can become.</td>
</tr>
<tr>
<td>Number of Rows</td>
<td>Lets you type number of rows to estimate size.</td>
</tr>
<tr>
<td>Estimate Index Size (KB)</td>
<td>Lets you type index size to estimate size.</td>
</tr>
<tr>
<td>Add Index</td>
<td>Click to add an index to the grid.</td>
</tr>
<tr>
<td>Remove Index</td>
<td>Click to remove an index from the grid.</td>
</tr>
<tr>
<td>Estimate Size</td>
<td>Click to start the estimate size functionality. DBArtisan then calculates the information, and displays it in a browser-ready HTML report that you can print or save.</td>
</tr>
<tr>
<td>Update Statistics</td>
<td>Opens the Update Statistics dialog box.</td>
</tr>
<tr>
<td>Save As</td>
<td>Lets you save as a result grid (.rsl) file.</td>
</tr>
</tbody>
</table>

**TIP:** To generate an HTML report, click the Report button.

For more information, see [Completing a Dialog Box](#).

Execute

**NOTE:** This functionality is available for Oracle and Sybase ASE only.

The Execution dialog box lets you execute functions and procedures. The Execution dialog box displays the necessary parameters. Depending on the parameter datatype, you can use the built-in calculator and calendar to select the entered data.

DBArtisan lets you enter arguments and set execution options in the Execution dialog box.

Completing the Execution Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the target object node.
   
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Execute**.

   OR

3. In the right pane of the application, right-click the target object, and then select **Execute**.

   DBArtisan opens the Execute dialog box.
4 To use the calendar, click the **Down** arrow.
5 To use the calculator, click the **Down** arrow.
   **NOTE:** These built-in functions are context-sensitive to the type of parameter required.
6 To execute the function with show plan, click **Query Plan**.
7 To modify your results options, click **Results**.
8 Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <a href="#">Preview dialog box</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <a href="#">Scheduling</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the <a href="#">Impact Analysis dialog box</a>. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see:
- [Executing Functions](#)
- [Executing Procedures](#)
- [Execute](#)
- [Completing a Dialog Box](#)

### Executing Functions

The table below describes the options and functionality on the Execution dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Plan</td>
<td>Click the Query Plan button to activate the Oracle EXPLAIN PLAN functionality. For information on where DBArtisan stores the EXPLAIN PLAN output, see <a href="#">Query Options</a>. DBArtisan's Query Plan window now displays data for the estimated costs, number of rows, and bytes returned by each plan step.</td>
</tr>
<tr>
<td>Result Options</td>
<td>Click the Result Options button lets you set options for the current execution. For more information on Result Options, see the Options Editor.</td>
</tr>
<tr>
<td>Arguments</td>
<td>Enter arguments in the blank text box. The datatype to the left of the blank field lets you know what type of argument is expected. For numeric arguments, you can use the calculator to enter information. For date arguments, you can use the built-in calendar or you can override the calendar by selecting the check box and entering your argument in field that follows. NULL is the default argument in that field.</td>
</tr>
<tr>
<td>Default</td>
<td>Select to gather default information from the data dictionary. <strong>ORACLE ONLY:</strong> You can not specify non-default parameters after specifying a default parameter.</td>
</tr>
</tbody>
</table>

For more information, see:
Completing the Execution Dialog Box

Completing a Dialog Box

Executing Procedures

The Procedure Execution dialog box lets you:

- Save input parameters as *.prm files to preserve specific input parameter configurations.
- Open *.prm files to save the effort of reentering specific input parameters.
- Reset parameters to their default setting.

The table below describes the options and functionality of the Procedure Execution dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Displays the current procedure’s owner.</td>
</tr>
<tr>
<td>Procedure</td>
<td>Displays the name of the current procedure.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Specify the required input parameters in this window. If input parameters are not required for the execution of the target procedure, a message displays in this window, stating that the procedure “has no input parameters. Press execute to run it.”</td>
</tr>
<tr>
<td>Open</td>
<td>Click to open an Open dialog box, from which you can open an existing *.prm file. The saved parameters immediately populate the dialog box upon opening.</td>
</tr>
<tr>
<td>Save</td>
<td>Click to save the values of your input parameters as a *.prm file. You can reopen a saved *.prm file from this dialog box at any time.</td>
</tr>
<tr>
<td>Reset</td>
<td>Click to reset the parameters in the Parameter window to their default values.</td>
</tr>
<tr>
<td>Default</td>
<td>Select to gather default information from the data dictionary. <strong>ORACLE ONLY:</strong> You can not specify non-default parameters after specifying a default parameter.</td>
</tr>
<tr>
<td>Continue</td>
<td>Click to execute the procedure once you have entered values for all required parameters in the Parameter window.</td>
</tr>
</tbody>
</table>

For more information, see:

Completing the Execution Dialog Box

Completing a Dialog Box

Extract

**NOTE:** This functionality is available for all platforms.

DBArtisan lets you extract data from one database to another database and extract the statements required to create objects into an Interactive SQL window. You can extract an object from the Datasource Explorer or the Objects Editor.
Important Notes
With respect to functions and stored procedures, it can be necessary to have statements executed before and after creation of the procedure or function. This can be useful for example, if you need to create or drop temporary tables used by the function or procedure. DBArtisan supports the use of two tag pairs, ETStart and ETEnd, that let you embed statements in the first comment block of a stored procedure or function. The following shows the expected syntax:

```sql
create procedure dbo.procname(@a numeric) as
/*
<ETStart>SQL Statement</ETStart>
<ETEnd>SQL Statement</ETEnd>
*/
begin
...
```

For more information, see:

Extracting Data

Extracting Data

1. To extract data, do the following: Datasource Explorer, click the target object node.
   DBArtisan displays the selected object in the right pane of the Explorer window.

2. In the right pane of the Explorer window, click the target object.

3. On the Datasource Explorer tool bar, click Extract.
   OR
   On the Datasource Explorer tool bar, click Command, and then click Extract.
   OR
   In the right pane of the Explorer window, right-click the target object, and then click Extract.
   DBArtisan opens the DDL Editor.

For more information, see:

Completing a Dialog Box

Extract Data as XML

This function allows you to take SQL data, extract it, and make it available as XML data. The XML Editor Filter allows you to pick and choose among columns and designate rows and then creates the For XML statement that enables the operation. The resulting XML document is created and presented in an active XML Editor. At this point the document can be saved in XML format.

**NOTE:** This functionality is available for Oracle 9i, SQL Server 8.0, and Sybase 12.5.1 or later.

**CAUTION:** To use the Extract Data as XML feature in Sybase, you must first purchase and install an XML Services license from Sybase.

To Open the XML Editor Filter

1. On the Datasource Explorer, expand the target datasource.

2. Expand the Tables node.
3. In the right pane of the Explorer window, right-click any table listed, and then select **Extract Data as XML**. The table below describes the fields you may encounter as you complete the **XML Editor Filter** dialog box.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>Check the columns with data you want to extract.</td>
</tr>
<tr>
<td>Where Clause</td>
<td>Write a Where clause to identify how rows should be connected in the result. For example, WHERE doctors.city = hospitals.city.</td>
</tr>
<tr>
<td>Select</td>
<td></td>
</tr>
<tr>
<td>SYBASE ONLY: XML Options</td>
<td>Refer to the <a href="#">XML Mappings chapter</a> (Chapter 5) in Sybase documents for detailed information</td>
</tr>
<tr>
<td>ORACLE ONLY: Row Set Tag</td>
<td>Identify the XML element names you want to use to replace the Row Set and the Row tags. Max rows is the maximum number of rows to fetch.</td>
</tr>
<tr>
<td>Row Tag</td>
<td></td>
</tr>
<tr>
<td>Max Rows</td>
<td></td>
</tr>
<tr>
<td>SQL SERVER ONLY: Optional Settings</td>
<td></td>
</tr>
<tr>
<td>AUTO</td>
<td>AUTO mode returns query results as nested XML elements. For any table in the From clause with a column in the Select clause, it is represented as an XML element. When you select RAW mode, each row in the query result set is transformed into an XML element with the generic row identifier. The XML attribute name will be the same as the column name for non-null columns. XML DATA specifies that an XML data schema will be returned. ELEMENTS specifies that columns will be returned as subelements--otherwise they are mapped to XML attributes. This is an option only if AUTO is selected. When BINARY BASE64 is selected, any binary data is encoded in base-64 format. You must specify this option in RAW mode. It is the default for AUTO.</td>
</tr>
<tr>
<td>RAW</td>
<td></td>
</tr>
<tr>
<td>XMLDATA</td>
<td></td>
</tr>
<tr>
<td>ELEMENTS</td>
<td></td>
</tr>
<tr>
<td>BINARY BASE64</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** As you select options, inspect the **Select** box to verify the query you are creating.

### Filter

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Filter dialog box lets you filter the results of an object node to names with one or more patterns. This lets you display and work with a target object without waiting for all the objects in a large list to display.

The table below describes the options and functionality on the Filter dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td>Lets you type either a complete object name or a partial object name with wild cards. Optionally, you can type a comma / space separated list of patterns that the filter should apply.</td>
</tr>
<tr>
<td>Match Case</td>
<td>Lets you specify if the filter is case sensitive.</td>
</tr>
</tbody>
</table>

### Completing the Object Filter Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, right-click the target object node.
2. Select **Filter**.

DBArtisan opens the Filter dialog box.
3 In **Pattern**, type either a complete object name or a partial object name with wild cards.

4 **Select Match Case** if the filter is case sensitive.

5 Click **OK**.

DBArtisan displays the target objects in the right pane of the application.

---

**Flush Cache**

*NOTE:* This functionality is available for IBM DB2 for Linux, Unix, and Windows 8.1 only.

The Flush Cache dialog box lets you clear all dynamic SQL in the cache and forces IBM DB2 for Linux, Unix, and Windows to recompile the SQL the next time it is called.

**Completing the Flush Cache Dialog Box**

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the target object node.
   
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Flush Cache**.

   DBArtisan opens the Flush Cache dialog box.

3. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <a href="#">Preview dialog box</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <a href="#">Scheduling</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

**Free Packages**

*NOTE:* This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Free Packages dialog box lets you delete a specific version of a package, all versions of a package, or whole collections of packages.

*CAUTION:* Confirmation is required because this is a destructive action.

To complete the Free Plans dialog box, do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <a href="#">Preview dialog box</a>.</td>
</tr>
</tbody>
</table>
For more information, see Completing a Dialog Box.

Free Plans

NOTE: This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Free Plans dialog box lets you delete application plans. A Plan is an executable application created in the bind process. It can include one or more packages or debris.

Important Notes

Confirmation is required because this is a destructive action.

Completing the Free Plans Dialog Box

To complete the Free Plans dialog box, do one of the following:

1. On the Datasource Explorer, select the Plans node.

   DBArtisan displays the Plans in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select Free.

   OR

   In the right pane of the application, right-click the target object, and then select Free.

   DBArtisan opens the Free Plans dialog box.

3. Do one of the following:

   For more information, see Completing a Dialog Box.

   Important Note

Grant Roles

NOTE: This functionality is available for all platforms.

The Grant Role Dialog Box lets you select the roles to grant to a user. Roles are sets of user privileges you associate with access to objects within a database. Roles streamline the process of granting permissions. You can use roles to grant sets of permissions and privileges to users and groups.
Completing the Grant Roles Dialog Box
To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the target user.
2. In the right pane of the application, right-click the target user, and select **Open**.
   
   DBArtisan opens the Users Editor.
3. On the Definition Tab of the Users Editor, select the **Role**, and then click **Add**.
   
   DBArtisan opens the Grant Roles dialog box.
4. Select the role(s) to grant.
5. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <a href="#">Preview dialog box</a></td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <a href="#">Scheduling</a></td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.

Copying Images
The Image Copy Wizard lets you create up to four image copies, exact reproductions, of any of the following objects:

- Tablespace
- Tablespace partition
- Data set of a linear table space
- Index space defined with COPY YES
- Index space partition

The Image Copy Wizard lets you create two types of image copies:

- A full image copy of all pages in a tables pace, partition, data set, or index space.
- An incremental image copy of only pages that have been modified since the last use of the Image Copy Wizard.

For more information, see [Image Copy Wizard - Panel 1](#).
Image Copy Wizard - Panel 1
The table below describes the options and functionality on this panel of the Image Copy Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want the utility to be restartable?</td>
<td>Lets you specify to execute Wizard in restartable mode. You can restart it from the Datasource Explorer.</td>
</tr>
<tr>
<td>Tape Stacking</td>
<td>Stacks similar output data sets as successive files on one logical tape volume, where a logical tape volume can consist of a multi-volume aggregate, if any of the resulting data sets on the tapes span volumes. Within one utility execution, output data sets are stacked on a logical tape volume of the same usage type. For example, local primary image copies are stacked separately from local backup image copies. Also lets you specify the Copy and Recovery dataset information once as variable names appended with generated values.</td>
</tr>
<tr>
<td>Set Properties</td>
<td>Opens the <a href="#">Dataset Properties dialog box</a>.</td>
</tr>
<tr>
<td>Do you want to execute DFSMS concurrent copy to make the full image copy?</td>
<td>Lets you specify the Concurrent Copy function of Data Facility Storage Management Subsystem (DFSMS). The wizard records the resulting DFSMS concurrent copies in the catalog table SYSIBM.SYSCOPY with ICTYPE=F and STYPE=C. If you select No, to not to execute DFSMS concurrent copy to make a full image copy, you can select Check Page.</td>
</tr>
<tr>
<td>Do you want the objects to be processed in parallel?</td>
<td>DBArtisan initiates multiple concurrent requests for a single user query and performs I/O processing concurrently (in parallel) on multiple data partitions. If you select Yes, to process the objects in parallel, you can type the number of objects. If you select to process the tape units in parallel, you can type the number of tape units.</td>
</tr>
<tr>
<td>Do you want other programs to access or update the selected objects when the COPY utility is running?</td>
<td>If you select Yes, you can select the Level of Access.</td>
</tr>
</tbody>
</table>

For more information, see [Image Copy](#).

Dataset Properties Dialog Box
The table below describes the options and functionality on the Dataset Properties Dialog Box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Site Primary Copy</td>
<td>Lets you specify the Local Site Primary Copy.</td>
</tr>
<tr>
<td>Local Site Backup Copy</td>
<td>Lets you specify the Local Site Backup Copy.</td>
</tr>
<tr>
<td>Recovery Site Primary Copy</td>
<td>Lets you specify the Recovery Site Primary Copy.</td>
</tr>
<tr>
<td>Recovery Site Backup Copy</td>
<td>Lets you specify the Recovery Site Backup Copy.</td>
</tr>
</tbody>
</table>

For more information, see [Image Copy](#).

Image Copy Wizard - Panel 2
The table below describes the options and functionality on this panel of the Image Copy Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select tablespaces</td>
<td>Displays the tablespaces to be copied.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the <a href="#">Tablespace Selector dialog box</a>.</td>
</tr>
</tbody>
</table>
Tablespace Selector Dialog Box
The table below describes the options and functionality on the Tablespace Selector dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Like</td>
<td>Lets you type the pattern for querying the tablespaces. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Tablespace Like</td>
<td>Lets you type the pattern for querying the tablespaces. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Match Case</td>
<td>Lets you specify match case.</td>
</tr>
<tr>
<td>Tablespace Selector</td>
<td>Lets you select one or more rows in this list.</td>
</tr>
<tr>
<td>Query</td>
<td>Repopulates Tablespace Selector based on the tablespace query.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all rows in Tablespace Selector.</td>
</tr>
<tr>
<td>Unselect All</td>
<td>Unselects all rows in Tablespace Selector.</td>
</tr>
<tr>
<td>Add</td>
<td>Click to open the Select Partition(s) for Tablespace dialog box.</td>
</tr>
</tbody>
</table>

For more information, see Image Copy.

Select Partition(s) for Tablespace Dialog Box
The table below describes the options and functionality on the Select Partition(s) for Tablespace Selector dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add all partition</td>
<td>Select to add all partitions for the tablespace.</td>
</tr>
<tr>
<td>Select Partitions</td>
<td>Lets you select partition(s) or click Select All or Unselect All.</td>
</tr>
</tbody>
</table>

Image Copy Wizard - Panel 3
The table below describes the options and functionality on this panel of the Image Copy Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Indexspaces</td>
<td>Displays the indexspaces to be copied.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Indexspace Selector dialog box.</td>
</tr>
<tr>
<td>Set Properties</td>
<td>Opens the Object Properties dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected indexspace.</td>
</tr>
</tbody>
</table>

For more information, see Image Copy.
Indexspace Selector Dialog Box
The table below describes the options and functionality on the panel of the Indexspace Selector dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Like</td>
<td>Lets you type the pattern for querying the indexspaces. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>IndexSpace Like</td>
<td>Lets you type the pattern for querying the indexspaces. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Match Case</td>
<td>Lets you specify match case.</td>
</tr>
<tr>
<td>Indexspace Selector</td>
<td>Lets you select one or more rows in this list.</td>
</tr>
<tr>
<td>Query</td>
<td>Repopulates Indexspace Selector based on the tablespace query.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all rows in Indexspace Selector.</td>
</tr>
<tr>
<td>Unselect All</td>
<td>Unselects all rows in Indexspace Selector.</td>
</tr>
</tbody>
</table>

For more information, see Image Copy.

Image Copy Wizard - Panel 4
The table below describes the options and functionality on this panel of the Image Copy Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select index.</td>
<td>Displays the indexes to be copied.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Index Selector dialog box.</td>
</tr>
<tr>
<td>Set Properties</td>
<td>Opens the Object Properties dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected indexspace.</td>
</tr>
</tbody>
</table>

For more information, see Image Copy.

Index Selector Dialog Box
The table below describes the options and functionality on the panel of the Index Selector dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Like</td>
<td>Lets you type the pattern for querying the Indexes. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Index Creator</td>
<td>Lets you type the pattern for querying the Indexes. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Index Like</td>
<td>Lets you type the pattern for querying the Indexes. You can type multiple words, separated by spaces.</td>
</tr>
<tr>
<td>Match Case</td>
<td>Lets you specify match case.</td>
</tr>
<tr>
<td>Index Selector</td>
<td>Lets you select one or more rows in this list.</td>
</tr>
<tr>
<td>Query</td>
<td>Repopulates Index Selector based on the tablespace query.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all rows in Index Selector.</td>
</tr>
<tr>
<td>Unselect All</td>
<td>Unselects all rows in Index Selector.</td>
</tr>
</tbody>
</table>

For more information, see Image Copy.
Object Properties Dialog Box
The table below describes the options and functionality on the Object Properties dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Copy?</td>
<td>Select to copy the entire tablespace.</td>
</tr>
<tr>
<td>Change Limit</td>
<td>Lets you specify the Change Limit option, the changed pages in the tablespace. If the high Change Limit value is met, then a full image copy is taken.</td>
</tr>
<tr>
<td>Range</td>
<td>Lets you specify a change limit range between 0.0 and 100.0, the changed pages in the tablespace. If the high Change Limit value is met, then a full image copy is taken.</td>
</tr>
<tr>
<td>Report Only</td>
<td>Lets you specify the Report Only option.</td>
</tr>
<tr>
<td>Local Site Primary Copy</td>
<td>Lets you specify the Local Site Primary Copy.</td>
</tr>
<tr>
<td>Local Site Backup Copy</td>
<td>Lets you specify the Local Site Backup Copy.</td>
</tr>
<tr>
<td>Recovery Site Primary Copy</td>
<td>Lets you specify the Recovery Site Primary Copy.</td>
</tr>
<tr>
<td>Recovery Site Backup Copy</td>
<td>Lets you specify the Recovery Site Backup Copy.</td>
</tr>
</tbody>
</table>

For more information, see Image Copy.

Impact Analysis

**NOTE:** This functionality is available for all platforms.

The Impact Analysis dialog box displays the objects dependent on the object you are changing. You can use the Impact Analysis dialog box to review the objects that could be impacted should you proceed with the action in question.

For more information, see Completing a Dialog Box.

Load Java

**NOTE:** This functionality is available for Oracle 8i or later.

Before you can call Java stored procedures, you must load them into the Oracle database and publish them to SQL. The Java Load Wizard lets you:

- Select the owner of the Java object and the files to load.
- Select options for the loading of the files.
- Select Resolver options.

For more information, see:

Completing a Dialog Box
Java Load Wizard - Panel 1
Java Load Wizard - Panel 1
The table below describes the options of the first panel of the Java Load Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who owns the Java Object?</td>
<td>Lets you select the owner of the Java object.</td>
</tr>
<tr>
<td>Select files to be loaded</td>
<td>Select a file, and then click Add.</td>
</tr>
</tbody>
</table>

For more information, see:
- [Completing a Dialog Box](#)
- [Java Load Wizard](#)

Java Load Wizard - Panel 2
The table below describes the options of the second panel of the Java Load Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>When do you want the Java files to be resolved?</td>
<td>Lets you specify when the source file is loaded as a source schema object, the source file is compiled, class schema objects are created for each class defined in the compiled .java file, and the compiled code is stored in the class schema objects.</td>
</tr>
<tr>
<td>Select the Encoding Options</td>
<td>Lets you specify the encoding of the .java file.</td>
</tr>
<tr>
<td>Grant Access to the following users</td>
<td>Lets you select one or more users.</td>
</tr>
</tbody>
</table>

For more information, see:
- [Completing a Dialog Box](#)
- [Java Load Wizard](#)

Java Load Wizard - Panel 3
The table below describes the options of the third panel of the Java Load Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other Load Options</td>
<td><strong>OPTIONAL:</strong> Lets you select options.</td>
</tr>
<tr>
<td>Add Resolver Options</td>
<td>Lets you specify the objects to search within the schemas defined.</td>
</tr>
<tr>
<td></td>
<td>Add - Click to open the Select a Resolver Option dialog box to add a new resolver option in the list.</td>
</tr>
<tr>
<td></td>
<td>Edit - Click to open the Resolver Edit dialog box to modify a resolver option.</td>
</tr>
<tr>
<td></td>
<td>Remove - Select one or more resolver option and click to delete.</td>
</tr>
</tbody>
</table>

For more information, see:
- [Completing a Dialog Box](#)
- [Java Load Wizard](#)
Lob Storage Definition

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows, Microsoft SQL Server, Oracle, and Sybase ASE only.

The Lob Storage Definition dialog box lets you edit LOB storage parameters.

The table below describes the options and functionality on the Lob Storage Definition dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Definition</td>
<td>Lets you edit the column name.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Lets you edit the column configuration.</td>
</tr>
<tr>
<td>Storage</td>
<td>Lets you edit the storage parameters.</td>
</tr>
</tbody>
</table>

For more information, see Completing a Dialog Box.

Lock

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows only.

The Lock Table dialog box lets you lock tables to prevent other users from reading or updating the table data. Locking a table saves the time of locking each row of the table that needs to be updated. DBArtisan releases locks at the end of a transaction.

The table below describes the options and functionality on the Lock Table dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock Mode</td>
<td>Share - Lets other users view but not modify the table data.</td>
</tr>
<tr>
<td></td>
<td>Exclusive - Prevents other users from viewing or modifying the table data.</td>
</tr>
</tbody>
</table>

Completing the Lock Table Dialog Box

To complete this dialog box, do the following:

1. On the Datasource Explorer, select the Tables node.
   DBArtisan displays the Tables in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select Lock.
   OR
   In the right pane of the application, right-click the table, and then select Lock.
   DBArtisan opens the Lock dialog box.

3. Select a Lock Mode option:
   - Share - Lets other users view but not modify the table data.
   - Exclusive - Prevents other users from viewing or modifying the table data.
4  Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <a href="#">Preview dialog box</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <a href="#">Scheduling</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the <a href="#">Impact Analysis dialog box</a>. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

**Merge Copy**

*NOTE:* This functionality is available for IBM DB2 for OS/390 and z/OS.

The Merge Copy function makes a merged copy of the data sets within a table space. You can merge incremental copies of a tablespace to make one incremental copy, or you can merge incremental copies with a full image copy to make a new full image copy.

**To Create a Merge Copy**

1. Connect to a DB2 datasource and expand the Explorer tree.
2. Click the **Utility Execution** node and double-click **MERGECOPY**.

The tables below displays all the fields you’ll find in the merge copy wizard:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want the utility to be restartable?</td>
<td>Yes or No</td>
</tr>
<tr>
<td>Tape Stacking Properties</td>
<td>Yes or no. When you select Yes, you then set the Dataset properties.</td>
</tr>
<tr>
<td>Do you want to merge all incremental copies with the full image copy to make a new full image copy?</td>
<td>Self-evident</td>
</tr>
<tr>
<td>WorkDDN</td>
<td>Here you are setting the dataset information for the working data definition name.</td>
</tr>
<tr>
<td>Select tablespaces</td>
<td>Self-evident.</td>
</tr>
</tbody>
</table>

**Migrate**

*NOTE:* This functionality is available for all platforms.
The Migration dialog box lets you move object schema between DBMS or servers by copying the DDL between the source and target databases. DBArtisan automatically resolves naming and database object differences between platforms and generate the proper schema to build the object. To migrate objects between servers, DBArtisan establishes separate connections to both the source and target servers and copying the DDL between the source and target databases.

Some objects have additional requirements, exceptions, or options. Before migrating a database object, check your database platform for additional information:

- Migrating IBM DB2 for Windows/Unix Objects
- Migrating IBM DB2 for OS/390 and z/OS Objects
- Migrating Microsoft SQL Server Objects
- Migrating Oracle Objects
- Migrating Sybase ASE Objects

To migrate an object:
1. Connect to the datasource that contains the object. For more information, see Datasource Management.
2. On the left pane of the Datasource Explorer, expand the datasource nodes until the target object type is displayed.
3. Select the target object type.
   DBArtisan displays objects of that type in the right pane of the Explorer window.
4. In the right pane of the Explorer window, select the object to be migrated.
5. On the Datasource Explorer tool bar, click Command, and then select Migrate.
   DBArtisan opens the Migration Wizard dialog.
6. Use the following table as a guide to understanding and modifying the settings on this dialog:

<table>
<thead>
<tr>
<th>Step</th>
<th>Tasks and settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Select a target datasource and depending on the type of DBMS you are migrating to, choose a database.</td>
</tr>
<tr>
<td>Options</td>
<td>Enable or disable the migration option and features for this operation. Note that different object types and DBMS offer different options.</td>
</tr>
<tr>
<td>Summary</td>
<td>Review the summary of the objects that will be migrated and the migration options you chose.</td>
</tr>
<tr>
<td>Execute</td>
<td>Click this panel to execute the operation and view an HTML migration report displaying the details of the operation.</td>
</tr>
</tbody>
</table>

Migrating IBM DB2 for Windows/Unix Objects
The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>The login must already exist on the target server. The user must already exist in the target database.</td>
</tr>
<tr>
<td>Indexes</td>
<td>The base tables must already exist in the target database.</td>
</tr>
<tr>
<td>Nodegroups</td>
<td>The base tables must already exist in the target database.</td>
</tr>
</tbody>
</table>
For more information, see:

Migrate

Completing the Migration Dialog Box

### Migrating IBM DB2 for OS/390 and z/OS Objects

The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>The login must already exist on the target server. The user must already exist in the target database.</td>
</tr>
<tr>
<td>Indexes</td>
<td>The base tables must already exist in the target database.</td>
</tr>
<tr>
<td>Procedures</td>
<td>Any objects referenced by the stored procedure, including tables, views and other stored procedures, must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>The owner of the stored procedure must already exist in the target database.</td>
</tr>
<tr>
<td>Tables</td>
<td>Any referenced user datatypes must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any defaults referenced by the table must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any rules referenced by the table must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>If the tables have foreign key references to tables not included in the migration, these tables must already exist in the target database.</td>
</tr>
<tr>
<td>Tablespaces</td>
<td>The container mappings and directories must already exist on the target server.</td>
</tr>
<tr>
<td>Triggers</td>
<td>The base tables and their owners must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any objects referenced by the trigger, including other tables, views and other stored procedures, must already exist in the target database.</td>
</tr>
<tr>
<td>User Datatypes</td>
<td>The owner of the user datatype must exist in the target database.</td>
</tr>
<tr>
<td>Users</td>
<td>The database objects must already exist in the target database.</td>
</tr>
</tbody>
</table>
For more information, see:

Migrate

Completing the Migration Dialog Box

### Migrating Microsoft SQL Server Objects

The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Views</td>
<td>Any objects referenced by the view, including tables and other views, must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>The owner of the view must already exist in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>The login must already exist on the target server.</td>
</tr>
<tr>
<td></td>
<td>The user must already exist in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defaults</td>
<td>The owner of the default must exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any bound user datatypes must exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any bound table columns must exist in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups</td>
<td>To migrate any object privileges granted to groups, the database objects must already exist in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linked Servers</td>
<td>Links do not migrate. So, after you migrate a linked server, open the Linked Server Editor and create new linked server logins.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedures</td>
<td>The user who owns the procedure(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any objects referenced by the procedure(s) must already exist in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Servers</td>
<td>DBArtisan lets you migrate the schema for one or more remote servers between cross-platform DBMSs. Specifically, you can migrate remote servers between different versions of Microsoft SQL Servers and Sybase ASE. To migrate remote servers, DBArtisan establishes separate connections to both the source and target servers and copies the DDL between the source and target databases.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roles</td>
<td>To migrate any object privileges granted to roles, the database objects must already exist in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rules</td>
<td>The owner of the rule must exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any bound user datatypes must exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any bound table columns must exist in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segments</td>
<td>To migrate segments successfully, all referenced database fragments must already exist in the target database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables</td>
<td>Any referenced user datatypes must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>If you are copying segment placement also, then the segments must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any defaults referenced by the table must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any rules referenced by the table must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>If the tables have foreign key references to tables not included in the migration, these tables must already exist in the target database.</td>
</tr>
</tbody>
</table>
For more information, see:

Migrate

Completing the Migration Dialog Box

Migrating Oracle Objects
The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clusters</td>
<td>The user who owns the cluster(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>The tablespace on which the cluster(s) are placed must already exist in the target database.</td>
</tr>
<tr>
<td>Directories</td>
<td>To migrate directories successfully, the object must exist in the target database.</td>
</tr>
<tr>
<td>Functions</td>
<td>The user who owns the function(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any objects referenced by the function(s) must already exist in the target database.</td>
</tr>
<tr>
<td>Indexes</td>
<td>The tables on which the indexes are being built must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>The user who owns the indexes must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>The tablespace on which the indexes are being placed must already exist in the target database.</td>
</tr>
<tr>
<td>Libraries</td>
<td>The object type must exist on the target server.</td>
</tr>
<tr>
<td>Materialized Views</td>
<td>The user who owns the materialized view(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any objects referenced by the materialized view(s) must already exist in the target database.</td>
</tr>
<tr>
<td>Materialized View Logs</td>
<td>The user who owns the materialized view log(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any objects referenced by the materialized view log(s) must already exist in the target database.</td>
</tr>
<tr>
<td>Packages</td>
<td>The user who owns the package(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any objects referenced by the package(s) must already exist in the target database.</td>
</tr>
<tr>
<td>Procedures</td>
<td>The user who owns the procedure(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any objects referenced by the procedure(s) must already exist in the target database.</td>
</tr>
<tr>
<td>Rollback Segments</td>
<td>The user who owns the rollback segment(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>The tablespaces where the rollback segment(s) are to be placed must already exist in the target database.</td>
</tr>
<tr>
<td>Sequences</td>
<td>The user who owns the sequence(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any objects referenced by the sequence(s) must already exist in the target database.</td>
</tr>
</tbody>
</table>
For more information, see:

Migrate

Completing the Migration Dialog Box

## Migrating Sybase ASE Objects

The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

<table>
<thead>
<tr>
<th>Object</th>
<th>Condition(s) on Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aliases</td>
<td>The login must already exist on the target server.</td>
</tr>
<tr>
<td></td>
<td>The user must already exist in the target database.</td>
</tr>
<tr>
<td>Defaults</td>
<td>The owner of the default must exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any bound user datatypes must exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any bound table columns must exist in the target database.</td>
</tr>
<tr>
<td>Groups</td>
<td>To migrate any object privileges granted to groups, the database objects must already exist in the target database.</td>
</tr>
<tr>
<td>Procedures</td>
<td>The user who owns the procedure(s) must already exist in the target database.</td>
</tr>
<tr>
<td></td>
<td>Any objects referenced by the procedure(s) must already exist in the target database.</td>
</tr>
</tbody>
</table>
Remote Servers

DBArtisan lets you migrate the schema for one or more remote servers between cross-platform DBMSs. Specifically, you can migrate remote servers between different versions of Sybase ASE and Microsoft SQL Servers.

Roles

To migrate any object privileges granted to roles, the database objects must already exist in the target database.

Rules

The owner of the rule must exist in the target database.
Any bound user datatypes must exist in the target database.
Any bound table columns must exist in the target database.

Segments

To migrate segments successfully, all referenced database fragments must already exist in the target database.

Tables

Any referenced user datatypes must already exist in the target database.
If you are copying segment placement also, then the segments must already exist in the target database.
Any defaults referenced by the table must already exist in the target database.
Any rules referenced by the table must already exist in the target database.
If the tables have foreign key references to tables not included in the migration, these tables must already exist in the target database.

NOTE: DBArtisan no longer changes the table lock scheme from ALLPAGES back to DATAROWS on the target server when using BCP.exe version dated after December 1997.

Triggers

The base tables and their owners must already exist in the target database.
Any objects referenced by the trigger, including other tables, views and other stored procedures, must already exist in the target database.

User Datatypes

The owner of the user datatype must exist in the target database.
Any bound defaults and rules must exist in the target database.

Users

The logins must already exist in the target server.
The database objects must already exist in the target database.

Views

Any objects referenced by the view, including tables and other views, must already exist in the target database.
The owner of the view must already exist in the target database.

For more information, see:

Migrate

Completing the Migration Dialog Box

Modify Constraint

The table below describes the options and functionality on the Modify Constraint dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Displays owner.</td>
</tr>
<tr>
<td>Table</td>
<td>Displays table.</td>
</tr>
<tr>
<td>Specify Columns in Constraint</td>
<td>Lets you select the columns and click the right arrow to include in constraint.</td>
</tr>
</tbody>
</table>
Modify Recovery

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Modify Recovery Wizard lets you delete records from SYSIBM.SYSCOPY, related log records from SYSIBM.SYSLGRNX, and entries from the DBD. You can delete records based on a specific age or based on the date the record was written. You can also delete records for an entire table space, partition or data set.

You should run the wizard regularly to clear outdated information from SYSIBM.SYSCOPY and SYSIBM.SYSLGRNX. These tables, and particularly SYSIBM.SYSLGRNX, can become very large and take up considerable amounts of space. By deleting outdated information from these tables, performance for processes that access data from these tables can be improved.

For more information, see [Modify Recovery Wizard - Panel 1](#).

### Modify Recovery Wizard - Panel 1

The table below describes the options and functionality on this panel of the Modify Recovery Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select tablespaces</td>
<td>Displays the available tablespaces.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the <a href="#">Tablespace Selector dialog box</a>.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected items.</td>
</tr>
</tbody>
</table>

For information, see [Modify Recovery](#).

### Modify Recovery Wizard - Panel 2

The table below describes the options and functionality on this panel of the Modify Recovery Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete history record(s) by</td>
<td>Lets you select Age or Date.</td>
</tr>
<tr>
<td>Number of Days</td>
<td>Available if you specified to delete history records by age. Lets you type a number between 1 and 32767.</td>
</tr>
<tr>
<td>Date</td>
<td>Available if you specified to delete history records by date. Lets you type a date in eight digit format yyyymmd (e.g. 20000218)</td>
</tr>
<tr>
<td>Delete Options</td>
<td>Lets you select delete options.</td>
</tr>
</tbody>
</table>

For information, see [Modify Recovery](#).

Modify Statistics

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Modify Statistics Wizard lets you delete unwanted statistics history records from the statistics catalog tables. You can delete statistics history records that were written before a specific date, or records of a specific age. You can delete the records for an entire table space, index space, or index.
You should run the wizard regularly to clear outdated information from the statistics history catalog tables. Deleting outdated information from the statistics history catalog tables, improves the performance of processes that access data from those tables.

For more information, see Modify Statistics Wizard - Panel 1.

**Modify Statistics Wizard - Panel 1**
The table below describes the options and functionality on this panel of the Modify Statistics Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete history record(s) by</td>
<td>Lets you select Age or Date.</td>
</tr>
<tr>
<td>Number of Days</td>
<td>Available if you specified to delete history records by age. Lets you type a number between 1 and 32767.</td>
</tr>
<tr>
<td>Date</td>
<td>Available if you specified to delete history records by date. Lets you type a date in eight digit format yyyymmmdd (e.g. 20000218)</td>
</tr>
<tr>
<td>Delete Options</td>
<td>Lets you select delete options.</td>
</tr>
</tbody>
</table>

For information, see Modify Statistics.

**Modify Statistics Wizard - Panel 2**
The table below describes the options and functionality on this panel of the Modify Statistics Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select tablespaces</td>
<td>Displays the available tablespaces.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Tablespace Selector dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected items.</td>
</tr>
</tbody>
</table>

For information, see Modify Statistics.

**Modify Statistics Wizard - Panel 3**
The table below describes the options and functionality on this panel of the Modify Statistics Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select indexes</td>
<td>Displays the available indexes.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Index Selector dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected items.</td>
</tr>
</tbody>
</table>

For information, see Modify Statistics.
Modify Statistics Wizard - Panel 4
The table below describes the options and functionality on this panel of the Modify Statistics Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select indexspaces</td>
<td>Displays the available indexspaces.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Indexspaces Selector dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected items.</td>
</tr>
</tbody>
</table>

For information, see Modify Statistics.

Move Log

**NOTE:** This functionality is available for Microsoft SQL Server and Sybase ASE only.

The Move Transaction Log dialog box lets you move a transaction log from one device to another.

The table below describes the options and functionality on the Move Transaction Log dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Device</td>
<td>Let you select the device to which you want to move the transaction log.</td>
</tr>
</tbody>
</table>

Completing the Move Log Dialog Box
To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Databases** node.
   DBArtisan displays the Databases in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Move Log**.
   **OR**
   In the right pane of the application, right-click the **Database**, and then select **Move Log**.
   DBArtisan opens the Move Log dialog box.

3. In **New Device**, select the device to which you want to move the transaction log.

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <strong>Preview dialog box</strong>.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <strong>Scheduling</strong>.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the <strong>Impact Analysis dialog box</strong>. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>
Open

NOTE: This functionality is available for all platforms.

DBArtisan stores information about server and object types in object editors. An editor is a tabbed dialog box that groups related information together.

Each editor contains a context-sensitive Commands menu with pertinent functionality for the object. Many of the object editors contain a DDL Tab. This tab displays the underlying target object's SQL.

TIP: The Object Editor tool bar has a refresh button, that lets you refresh Object Editor contents, clear the Object Editors, and log SQL if SQL Logging is turned on.

If an objects has dependent objects, such as tables, triggers, procedures and views, you can view and access their dependent objects in the editor.

For more information, see Object Editors.

Open Control Files

NOTE: This functionality is available for Oracle 8 or later only.

The control file is a small binary file containing a record of the database schema. It is one of the most essential files in the database because it is necessary for the database to start and operate successfully.

The control file contains information about the associated database that is required for the database to be accessed by an instance, both at startup and during normal operation.

NOTE: Only the Oracle database server can modify a control file's information; no user can edit a database's control file directly.

The Control File Information dialog box displays control file information, including:

- Type
- Record Size
- Records Total
- Records Used
- First Index
- Last Index
- Last Recid

For more information, see Control Files.

Package Editor

NOTE: This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Package Editor lets you modify a package.
The table below describes the options and functionality on the Package Editor:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan Name</td>
<td>Displays the plan name.</td>
</tr>
<tr>
<td>Location</td>
<td>Lets you select the name of the location to connect to.</td>
</tr>
<tr>
<td>Collection</td>
<td>Lets you select the location of the DBMS where the plan binds and where the description of the plan resides.</td>
</tr>
<tr>
<td>Package</td>
<td>Lets you select a package.</td>
</tr>
<tr>
<td>Add</td>
<td>Click to add the plan.</td>
</tr>
<tr>
<td>Close</td>
<td>Click to close the editor and return to the wizard.</td>
</tr>
</tbody>
</table>

**Partitioned Columns**

**NOTE:** This functionality is available for Oracle only.

The Partitioned Columns dialog box lets you partition columns. Partitioning lets you break large columns into smaller pieces, which are called partitions. Partitions make the data in your table easier to manage and analyze. Your SQL statements can access the partitions rather than the entire table. Partitions are most useful in data warehouse applications, which store large amounts of data.

The table below describes the options and functionality on the Partitioned Columns dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Columns</td>
<td>Select column and click the right arrow button to move to Partitioning Columns.</td>
</tr>
<tr>
<td>Partitioning Columns</td>
<td>Select column and click the left arrow to move to Available Columns.</td>
</tr>
<tr>
<td>Apply</td>
<td>Click to apply changes.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

**Partition Editor**

The Partition Editor lets you specify partition management and partition options.

The table below describes the options and functionality on the Partition Editor.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition number</td>
<td>Identifies a partition of the object.</td>
</tr>
<tr>
<td>DB2 manages the data sets on a volume of the specified storage group</td>
<td>A DB2-managed data set that resides on a volume of the identified storage group.</td>
</tr>
<tr>
<td>Stogroup Name - Lets you specify the stogroup.</td>
<td></td>
</tr>
<tr>
<td>Primary Space Allocation - Lets you specify the minimum primary space allocation for a DB2-managed data set of the partition.</td>
<td></td>
</tr>
<tr>
<td>Secondary Space Allocation - Lets you specify the minimum secondary space allocation for a DB2-managed data set of the partition.</td>
<td></td>
</tr>
<tr>
<td>User manages the data sets on a specified VCAT catalog-name</td>
<td>A user-managed data set with a name that starts with catalog-name.</td>
</tr>
</tbody>
</table>
The Partition Upper Bound dialog box lets you specify the non-inclusive upper bound for the partitioning columns. The table below describes the options and functionality on the Partition Upper Bound dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitioning Column</td>
<td>Lets you select the partitioning column.</td>
</tr>
<tr>
<td>Upper Bound</td>
<td>Lets you select the default or type the value of the upper bound. All rows in the partition have partitioning keys that compare to less than the partition bound and greater than or equal to the partition bound for the preceding partition. <strong>NOTE:</strong> The set button is not available if your selection is higher than the upper bound for the column in the original partition. If the upper bound is MAXVALUE, you need to type a value in the list.</td>
</tr>
</tbody>
</table>

The table below describes when you should enable editing of the upper bound:

<table>
<thead>
<tr>
<th>Partition Type</th>
<th>Index</th>
<th>Table</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash</td>
<td>N/A</td>
<td>N/A</td>
<td>Does not have upper bound.</td>
</tr>
<tr>
<td>Range Local</td>
<td>Disable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>Range Global</td>
<td>Enable all but the last partition</td>
<td>Enable</td>
<td>The upper bound button is disabled if there is only one partition.</td>
</tr>
<tr>
<td>Composite Global</td>
<td>Enable all but the last partition</td>
<td>Enable</td>
<td>The upper bound button is disabled if there is only one partition.</td>
</tr>
</tbody>
</table>

**Completing the Partition Upper Bound Dialog Box**

To complete this dialog box, do the following:

1. Click the **Partitioning Column** list and then click the partitioning column.
2. In the **Upper Bound** list, click the default or type the value of the upper bound. All rows in the partition will have partitioning keys that compare to less than the partition bound and greater than or equal to the partition bound for the preceding partition.
3. Click **Set**.
4. Click **OK**.

For more information, see:
Partition Upper Bound

Completing a Dialog Box

Place

NOTE: Place functionality is available for Sybase ASE.

The Placement dialog box lets you place tables and indexes on different segments. From a performance standpoint it is not recommended to have a table and its supporting indexes on the same device or disk segment. It is also good to have more frequently accessed indexes and tables grouped together on higher speed devices, if possible.

The table below describes the options and functionality on the Placement dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Segment Location</td>
<td>Lets you specify the segment on which you can place objects, the default, logsegment or system.</td>
</tr>
</tbody>
</table>

Completing the Placement Dialog Box

To complete this dialog box, do the following:

1. On the Datasource Explorer, select the target object node.
   
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select Place.
   
   OR

   In the right pane of the application, right-click the target object, and then select Place.

   DBArtisan opens the Placement dialog box.

3. In New Segment Location, select the segments to place the object.

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview</td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td>Execute</td>
<td>Executes the task.</td>
</tr>
<tr>
<td>Analysis</td>
<td>Opens the Impact Analysis dialog box. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see Completing a Dialog Box.

Preview: Create

NOTE: This functionality is available for all platforms.
The Preview:Create dialog box lets you preview any SQL related command you want to perform on an object.

The table below describes the options and functionality of the Preview:Create dialog box:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Saves SQL]</td>
<td>Saves SQL.</td>
</tr>
<tr>
<td>![Prints SQL]</td>
<td>Prints SQL.</td>
</tr>
<tr>
<td>![Sends SQL via e-mail]</td>
<td>Sends SQL via e-mail.</td>
</tr>
<tr>
<td>![Batch]</td>
<td>IBM DB2 FOR WINDOWS/UNIX UTILITIES ONLY: Opens the Job Cards for Batch Execution dialog box. You can check the status of this utility on the Utilities Tab of the Database Monitor.</td>
</tr>
<tr>
<td>![Schedule]</td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td>![Execute]</td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

For more information, see Completing a Dialog Box.

Preview

**NOTE:** This functionality is available for all platforms.

The Preview dialog box lets you preview any SQL related command you want to perform on an object.

The table below describes the options and functionality on the Preview dialog box:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Saves SQL]</td>
<td>Saves SQL.</td>
</tr>
<tr>
<td>![Prints SQL]</td>
<td>Prints SQL.</td>
</tr>
<tr>
<td>![Sends SQL via e-mail]</td>
<td>Sends SQL via e-mail.</td>
</tr>
<tr>
<td>![Batch]</td>
<td>IBM DB2 for OS/390 and z/OS ONLY: Opens the Job Cards for Batch Execution dialog box. You can check the status of this utility on the Utilities Tab of the Database Monitor.</td>
</tr>
<tr>
<td>![Schedule]</td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td>![Execute]</td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

For more information, see Completing a Dialog Box.
Job Cards for Batch Execution
The Job Cards for Batch Execution dialog box lets you specify the job cards that should be used to execute the utility in batch.
For more information, see Completing a Dialog Box.

Quiesce Database
The Quiesce Database dialog box lets you restrict user access to a database.

**NOTE:** This functionality is available for IBM DB2 for Windows/Unix 8.1 only. DBArtisan does not support Quiesce Database for an IBM DB2 8 server from a IBM DB2 for Windows/Unix 7 client or for an IBM DB2 for Windows/Unix 7 server from an IBM DB2 for Windows/Unix 7 client.

The table below describes the options and functionality on the Quiesce Database dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Lets you select a connection option.</td>
</tr>
</tbody>
</table>

Completing the Quiesce Database Dialog Box
To complete this dialog box, do the following:

1. On the Datasource Explorer, select the Datasources node.
   DBArtisan displays the Databases in the right pane of the Explorer window.
2. On the Datasource Explorer tool bar, click Command, and then select Quiesce.
   OR
   In the right pane of the application, right-click the target object, and then select Quiesce.
   DBArtisan opens the Quiesce dialog box.
3. Select a Connection option.
4. Set the For user or For group radio box and specify a user or group name as appropriate.
5. Click the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

Quiesce Instance
**NOTE:** This functionality is available for IBM DB2 for Windows/Unix 8.1 only.

The Quiesce Instance dialog box lets you restrict user access to an instance.
The table below describes the options and functionality on the Quiesce Instance dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>For user</td>
<td>Let you select and type user name to restrict.</td>
</tr>
<tr>
<td>For group</td>
<td>Let you select and type group name to restrict.</td>
</tr>
<tr>
<td>Connections</td>
<td>Let you select a connection option to restrict.</td>
</tr>
</tbody>
</table>

### Completing the Quiesce Instance Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Instance** node.
2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Quiesce**.
   
   **OR**
   
   In the right pane of the application, right-click the target object, and then select Quiesce.
   
   DBArtisan opens the Quiesce dialog box.
3. To specify a user, select **For user** option, and then type user name.
4. To specify a group, select **For group** option, and then type group name.
5. Select a **Connections** option.
6. Click the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Execute]</td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

### Quiesce Tablespaces

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows and IBM DB2 for OS/390 and z/OS only.

The Quiesce Tablespaces dialog box lets you temporarily prevent other users from updating or reading selected tables on a tablespace. For example, you may not want the table to be updated during a backup procedure to capture a complete backup of the table as it exists.

The table below describes the options and functionality on the Quiesce Tablespaces dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiesce Mode</td>
<td>Exclusive - Prevents other users from viewing or modifying the table data.</td>
</tr>
<tr>
<td></td>
<td>Intent to Update - Lets other users view but not update the table data.</td>
</tr>
<tr>
<td></td>
<td>Reset - Lets you reset a table’s quiesce mode.</td>
</tr>
<tr>
<td></td>
<td>Share - Lets all users view (including you) but not modify the table data.</td>
</tr>
</tbody>
</table>
Completing the Quiesce Tablespaces Dialog Box
To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Tables** node.

   DBArtisan displays the Tables in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Quiesce Tablespaces**.

   OR

   In the right pane of the application, right-click the target object, and then select **Quiesce Tablespaces**.

   DBArtisan opens the Quiesce Tablespaces dialog box.

3. Select a **Quiesce Mode**:
   - **Exclusive** - Prevents other users from viewing or modifying the table data.
   - **Intent to Update** - Lets other users view but not update the table data.
   - **Reset** - Lets you reset a table’s quiesce mode.
   - **Share** - Lets all users view (including you) but not modify the table data.

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <strong>Preview dialog box</strong>.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <strong>Scheduling</strong>.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the <strong>Impact Analysis dialog box</strong>. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see **Completing a Dialog Box**.

Reassign by Category
DBArtisan lets you reassign the category of stored outlines in Oracle.

**NOTE:** This functionality is available for Oracle only.

Outlines are a set of results for the execution plan generation of a particular SQL statement. When you create an outline, plan stability examines the optimization results using the same data used to generate the execution plan. That is, Oracle uses the input to the execution plan to generate an outline, and not the execution plan itself.

**To reassign a stored outline to a different category:**

1. Connect to the datasource that contains the stored outline. For more information, see **Datasource Management**.

2. On the **Datasource Explorer**, expand the datasource nodes until the **Stored Outlines** node is visible and then select the **Stored Outlines** node.

   DBArtisan displays the stored outlines in the right pane of the Explorer window.
3  In the right pane of the datasource explorer, select the stored outline for which you want to reassign categories.

4  On the **Datasource Explorer** tool bar, click **Command**, and then select **Reassign**.

   DBArtisan opens the **Reassign by Category** dialog box.

5  From the **Target Category** dropdown, choose a new, target category.

6  Review the **Preview** panel and when ready, use the **Schedule** or **Execute** button to complete the task.

### Rebind Packages

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows only.

The Rebind Package dialog box lets you update the best access path for SQL statements when the contents of a package changes.

**TIP:** If the physical storage of a package is changed or dropped, rebinding updates the path of the SQL statements.

#### Completing the Rebind Packages Dialog Box

To complete this dialog box, do the following:

1  On the **Datasource Explorer**, select the **Packages** node.

2  To filter the Owner and Object lists, click the Filter button next to each list.

   DBArtisan opens the **Filter** dialog box.

   DBArtisan displays the Packages in the right pane of the Explorer window.

3  On the **Datasource Explorer** tool bar, click **Command**, and then select **Rebind**.

   **OR**

   In the right pane of the application, right-click the target object, and then select **Rebind**.

   DBArtisan opens the Rebind dialog box.

4  Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <strong>Preview</strong> dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="image" alt="Analysis" /></td>
<td>Opens the <strong>Impact Analysis</strong> dialog box. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

### Rebind Plans

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows and OS/390 only.
The Rebind Plans dialog box lets you update the best access path for SQL statements when the contents of a plan change.

**TIP:** If the physical storage of a plan is changed or dropped, rebinding updates the path of the SQL statements.

**TIP:** To filter the Owner and Object lists, click the Filter button next to each list. DBArtisan opens the Filter dialog box.

### Completing the Rebind Plans Dialog Box

To complete the Rebind Plans dialog box, do one of the following:

1. On the **Datasource Explorer**, select the **Plans** node.
   
   DBArtisan displays the Plans in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Rebind**.
   
   OR

   In the right pane of the application, right-click the target object, and then select **Rebind**.

   DBArtisan opens the Rebind Plans dialog box.

3. Do one of the following:

   - **Rebuild Index**

   **NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS.

   The Rebuild Index Wizard reconstructs indexes from the table that they reference.

   The table below describes the options and functionality on the Rebuild Index Wizard.

<table>
<thead>
<tr>
<th>Panel</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel 1</td>
<td>Select Indexes</td>
<td>Lets you select an index to rebuild.</td>
</tr>
<tr>
<td></td>
<td>Add</td>
<td>Opens the <strong>Index Selector dialog box</strong>.</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td>Deletes the selected index(es).</td>
</tr>
<tr>
<td>Panel 2</td>
<td>Select tablespaces</td>
<td>Lets you select the target tablespace.</td>
</tr>
<tr>
<td></td>
<td>Add</td>
<td>Opens the <strong>Tablespace Selector dialog box</strong>.</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td>Deletes the selected items.</td>
</tr>
<tr>
<td>Panel 3</td>
<td>Do you want the utility to be restartable?</td>
<td>Select to make the utility restartable.</td>
</tr>
</tbody>
</table>
Rebuild Indexes

**NOTE:** This functionality is available for Oracle only.

The Rebuild Indexes dialog box lets you rebuild an index that has become fragmented. Rebuilding an index is a good alternative to coalescing an index because you can move the index to a different tablespace and change both tablespace and storage parameters while eliminating fragmentation. However, rebuilding an index has a higher cost than coalescing an index. These same qualities also make rebuilding an index a viable alternative to dropping an index then re-creating it.

As a rule of thumb, check indexes for rebuilds when their level (or tree depth) reaches four or greater, or many deleted leaf rows are found. The Rebuild Indexes dialog box can also be used to easily move an index from one tablespace to another.

**Important Notes**
- If you are rebuilding a function-based index, the index is enabled when the rebuild is finished.
- You cannot rebuild a partitioned index. You must rebuild each partition or subpartition individually.

The table below describes the options and functionality on the Rebuild dialog box.

<table>
<thead>
<tr>
<th>Panel</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Do you want to reuse the db2 managed dataset?</td>
<td>Default is No.</td>
</tr>
<tr>
<td></td>
<td>WorkDDN</td>
<td>Lets you select a work ddn template from the list or click Edit to specify the dataset information.</td>
</tr>
<tr>
<td></td>
<td>Do you want to specify to sort device type?</td>
<td>Default is No.</td>
</tr>
<tr>
<td></td>
<td>Do you want to specify the sort number?</td>
<td>Default is No.</td>
</tr>
<tr>
<td></td>
<td>Do you want to specify the sort keys?</td>
<td>Default is No.</td>
</tr>
<tr>
<td>Panel 4</td>
<td>Do you want to specify the stats online?</td>
<td>Lets you specify the stats online.</td>
</tr>
<tr>
<td></td>
<td>Do you want to output message to SYSPrint?</td>
<td>The default is no.</td>
</tr>
<tr>
<td></td>
<td>Do you want to specify the correlation stats option?</td>
<td>Lets you specify correlation stats on the next panel of the wizard.</td>
</tr>
<tr>
<td></td>
<td>Do you want to force aggregation or rollup processing to be done even though some parts do not contain data?</td>
<td>The default is Default.</td>
</tr>
<tr>
<td>Panel 5</td>
<td>KeyCard</td>
<td>Collects all of the distinct values in all of the 1 to n key column combinations for the specified indexes. n is the number of columns in the index.</td>
</tr>
<tr>
<td></td>
<td>Frequent Item</td>
<td>Lets you add items to the grid.</td>
</tr>
<tr>
<td></td>
<td>Add</td>
<td>Opens the Add Frequent Item dialog box</td>
</tr>
<tr>
<td>Panel 6</td>
<td>Update Statistics</td>
<td>Lets you specify update statistics.</td>
</tr>
<tr>
<td></td>
<td>History Statistics</td>
<td>Lets you specify history statistics.</td>
</tr>
</tbody>
</table>

Rebuild Indexes

**NOTE:** This functionality is available for Oracle only.

The Rebuild Indexes dialog box lets you rebuild an index that has become fragmented. Rebuilding an index is a good alternative to coalescing an index because you can move the index to a different tablespace and change both tablespace and storage parameters while eliminating fragmentation. However, rebuilding an index has a higher cost than coalescing an index. These same qualities also make rebuilding an index a viable alternative to dropping an index then re-creating it.

As a rule of thumb, check indexes for rebuilds when their level (or tree depth) reaches four or greater, or many deleted leaf rows are found. The Rebuild Indexes dialog box can also be used to easily move an index from one tablespace to another.

**Important Notes**
- If you are rebuilding a function-based index, the index is enabled when the rebuild is finished.
- You cannot rebuild a partitioned index. You must rebuild each partition or subpartition individually.

The table below describes the options and functionality on the Rebuild dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Tablespace</td>
<td>Defaults to the tablespace which currently includes the index. To change the tablespace containing the index, choose a new tablespace from the list.</td>
</tr>
</tbody>
</table>
Completing the Rebuild Indexes Dialog Box

To complete this dialog box, do the following:

1. On the Datasource Explorer, select the Indexes node.
   DBArtisan displays the Indexes in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select Rebuild.
   OR
   In the right pane of the application, right-click the index, and then select Rebuild.
   DBArtisan opens the Rebuild Indexes dialog box.

3. To move the index to a new tablespace, click the New Tablespace list and then click the new tablespace.

4. In the Logging box, click:
   • The Recoverable option button to make the operation log in the redo file.
   • The Non-Recoverable option button if you do not want the operation logged in the redo file.

5. If you are using Parallel Server, select the Parallel Server check box and:
   • Type a value indicating the number of query server processes that should be used in the operation in the Degree box.
   • Type a value indicating how you want the parallel query partitioned between the Parallel Servers in the Instances box.

6. In the Order box:
   • Click the Reverse option button to rebuild the index to store the bytes of the index block in reverse order.
   • Click the No Reverse option button to rebuild the index to store the bytes of the index block in order.
   NOTE: This option is only available for Oracle8.

7. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Preview]</td>
<td>Opens the Preview dialog box.</td>
</tr>
</tbody>
</table>
For more information, see:  
Rebuild Indexes  
Completing a Dialog Box

Rebuild Outlines

**NOTE:** This functionality is available for Oracle only.

The Rebuild Outlines dialog box lets you rebuild an outline.

Outlines are a set of results for the execution plan generation of a particular SQL statement. When you create an outline, plan stability examines the optimization results using the same data used to generate the execution plan. That is, Oracle uses the input to the execution plan to generate an outline, and not the execution plan itself.

The table below describes the options and functionality on the Rebuild Outlines dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline(s) to be rebuilt box</td>
<td>Displays the outline(s) available to rebuild.</td>
</tr>
</tbody>
</table>

For more information, see Completing a Dialog Box.

Recompile

**NOTE:** This functionality is available for Microsoft SQL Server and Sybase ASE only.

The Recompile dialog box lets you specify single or multiple objects for recompilation. Recompilation causes each procedure and trigger that uses the target table to be recompiled the next time it runs.

The queries used by procedures and triggers are optimized only once, when they are compiled. As you add indexes or make other changes to your database that affect its statistics, your compiled procedures and triggers may lose efficiency. By recompiling the procedures and triggers that act on a table, you can optimize the queries for maximum efficiency.

Redistribute

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows only.

The Redistribute dialog box lets you redistribute the data in a nodegroup.
The table below describes the options and functionality on the Redistribute dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redistribute Options</td>
<td>Lets you click the appropriate option button:</td>
</tr>
<tr>
<td></td>
<td>Uniform</td>
</tr>
<tr>
<td></td>
<td>Using Dist. File</td>
</tr>
<tr>
<td></td>
<td>Using Target Map</td>
</tr>
<tr>
<td></td>
<td>Continue</td>
</tr>
<tr>
<td></td>
<td>Rollback</td>
</tr>
<tr>
<td>Using Dist. File</td>
<td>Lets you select and type the distinct file name in the corresponding box.</td>
</tr>
<tr>
<td>Using target File</td>
<td>Lets you select and type the target file name in the corresponding box.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

**Refit**

*NOTE:* This functionality is available for Sybase ASE only.

The Refit dialog box lets you refit a database device. If you are restoring a damaged master database, or have added devices since the last database or transaction log dump, then refit a device after reinitializing. Refitting a device ensures that the information in sysusages and sysdatabases is consistent with the information in sysdevices.

The table below describes the options and functionality on the Refit dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redistribute Options</td>
<td>Lets you click the appropriate option button:</td>
</tr>
<tr>
<td></td>
<td>Uniform</td>
</tr>
<tr>
<td></td>
<td>Using Dist. File</td>
</tr>
<tr>
<td></td>
<td>Using Target Map</td>
</tr>
<tr>
<td></td>
<td>Continue</td>
</tr>
<tr>
<td></td>
<td>Rollback</td>
</tr>
<tr>
<td>Using Dist. File</td>
<td>Lets you select and type the distinct file name in the corresponding box.</td>
</tr>
<tr>
<td>Using target File</td>
<td>Lets you select and type the target file name in the corresponding box.</td>
</tr>
</tbody>
</table>

**Completing the Refit Dialog Box**

To complete this dialog box, do the following:

1. On the [Datasource Explorer](#), select the [Database Devices](#) node.
   
   DBArtisan displays the Database Devices in the right pane of the Explorer window.

2. On the [Datasource Explorer](#) tool bar, click [Command](#), and then select [Refit](#).

   OR

   In the right pane of the application, right-click the target object, and then select [Refit](#).

   DBArtisan opens the Refit dialog box.
3 Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="button.png" alt="Preview" /></td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td><img src="button.png" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td><img src="button.png" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
<tr>
<td><img src="button.png" alt="Analysis" /></td>
<td>Opens the Impact Analysis dialog box. This option is not available for every dialog box.</td>
</tr>
</tbody>
</table>

**Refresh Table**

The Refresh Summary Table Dialog box lets you reload materialized query tables that have been defined with refresh options.

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows 8.1 and later.

**To reload a materialized query table:**

1. Connect to the datasource that contains the materialized query table. For more information, see Datasource Management.

2. On the Datasource Explorer, expand the datasource nodes until the Materialized Query Tables node is visible and then select the Materialized Query Tables node.

   DBArtisan displays the materialized query tables in the right pane of the Explorer window.

3. In the right pane of the datasource explorer, select the materialized query table you want to reload.

4. On the Datasource Explorer tool bar, click Command, and then select Refresh.

   DBArtisan opens the Refresh Materialized Query Table dialog box.

5. Use the following table as a guide to understanding and modifying settings on this dialog:

<table>
<thead>
<tr>
<th>Pane</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Options</td>
<td>Select an Online Option of ALLOW NO ACCESS, ALLOW READ ACCESS or ALLOW WRITE ACCESS. Enable or disable Query Optimization. Select an INCREMENTAL or NOT INCREMENTAL Refresh Option.</td>
</tr>
<tr>
<td>Preview</td>
<td>Preview the DDL generated by the options you chose.</td>
</tr>
</tbody>
</table>

6. Review the Preview panel and when ready, use the Schedule or Execute button to complete the task.

**Important Notes**

For procedures on restoring a damaged master database, consult the Commands Reference Manual.
Reinitialize

**NOTE:** This functionality is available for Sybase ASE only.

The reinitialize function allows you to keep a database device on the Sybase server open while resetting the device to its state immediately after it was opened.

**Completing the Reinitialize Dialog Box**

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, expand the **Storage** node, and click **Database Devices**.
2. In the right pane, click the device you want to reinitialize.
3. Select **Reinit...** from the Command menu or right-click and select it.
4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

**Rename**

**NOTE:** This functionality is available for all platforms.

DBArtisan lets you rename an object. In general, all referenced or referring objects are updated to reflect the new name. The following table notes the exceptions and provides prerequisite tasks to be performed before renaming an object.

<table>
<thead>
<tr>
<th>DBMS</th>
<th>Notes and restrictions on renaming</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Microsoft SQL Server</strong></td>
<td>Microsoft SQL Server lets you rename a database if you own it. Before renaming a database, set it to single-user mode. Microsoft SQL Server will not rename a table if it is referenced within the body of other objects that call it, such as tables, triggers or views. As a result, renaming a table can result in broken dependencies with other objects. Also, Microsoft SQL Server does not let you rename System Tables.</td>
</tr>
<tr>
<td><strong>IBM DB2 for OS/390 and z/OS</strong></td>
<td>DBArtisan lets you rename a primary key if the underlying table has only one owner. The rename operation does not rename the table if it is referenced within the body of other objects, such as tables, triggers or views, that call it. As a result, renaming a table can result in broken dependencies with other objects.</td>
</tr>
</tbody>
</table>
To rename an object:
1. Connect to the datasource that contains the object. For more information, see Datasource Management.
2. On the left pane of the Datasource Explorer, expand the datasource nodes until the target object type is displayed.
3. Select the target object type.
   DBArtisan displays objects of that type in the right pane of the Explorer window.
4. In the right pane of the Explorer window, select the object to be renamed.
5. On the Datasource Explorer tool bar, click Command, and then select Rename.
   DBArtisan opens the Rename dialog.
6. Use the following table as a guide to understanding and modifying settings in the dialog:

<table>
<thead>
<tr>
<th>DBMS</th>
<th>Notes and restrictions on renaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybase ASE</td>
<td>Before renaming a database, set it to single-user mode. System indexes can not be renamed. The rename operation does not rename the stored procedure if it is referenced within the body of other objects, such as another stored procedure, that call it. As a result, renaming a stored procedure can result in broken dependencies with other objects. The rename operation does not rename the table if it is referenced within the body of other objects, such as tables, triggers or views, that call it. As a result, renaming a table can result in broken dependencies with other objects. The rename operation does not rename the view if it is referenced within the body of other objects, such as stored procedures, triggers or other views, that call it. As a result, renaming a view can result in broken dependencies with other objects.</td>
</tr>
</tbody>
</table>

### Step  Settings and tasks
<table>
<thead>
<tr>
<th>Rename</th>
<th>Name</th>
<th>Provide the new name for the object.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependencies</td>
<td>Review the referring and referred objects for which naming will be automatically resolved when you execute the renaming operation.</td>
<td></td>
</tr>
<tr>
<td>Preview</td>
<td>Preview the DDL generated for the operation and when ready, use the Schedule or Execute button to perform this action.</td>
<td></td>
</tr>
</tbody>
</table>

**Reorganize**

**NOTE:** Reorganize functionality is available for IBM DB2 for Linux, Unix, and Windows, Oracle, and Sybase ASE only.

The Reorganize dialog box lets you alter a table’s physical storage to ensure that it is stored efficiently and to eliminate fragmentation.
Completing the Reorganize Dialog Box

To complete this dialog box, do the following:

1. **On the Datasource Explorer**, select the target object node.
   
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. **On the Datasource Explorer** tool bar, click **Command**, and then select **Reorganize**.
   
   OR
   
   In the right pane of the application, right-click the target object, and then select **Reorganize**.
   
   DBArtisan opens the Reorganize dialog box.

3. Use the information in the following topics to provide reorganization details, depending on the DBMS that DBArtisan is working with:
   
   - [Reorganizing IBM DB2 for Linux, Unix, and Windows Objects](#)
   - [Reorganizing Oracle Objects](#)
   - [Reorganizing Sybase ASE Objects](#)

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the <strong>Preview dialog box</strong>.</td>
</tr>
<tr>
<td><img src="image" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <a href="#">Scheduling</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

For more information, see:

- [Reorganize](#)
- [Completing a Dialog Box](#)
Reorganizing IBM DB2 for Linux, Unix, and Windows Objects

Reorganize Dialog Box (One Table)
The table below describes the options and functionality on the Reorganize dialog box.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Options</td>
<td>• Choose a <strong>Reorg Type</strong> of table or all indexes of the table. &lt;br&gt; • Optionally, choose an <strong>Index Schema</strong> and specify an <strong>Index name</strong>. &lt;br&gt; • Enable or disable <strong>Inplace Reorganization</strong> and if you enable, choose an <strong>Inplace reorganization mode</strong> of Start, Stop, Pause, or Resume. &lt;br&gt; • Select an <strong>Access mode</strong> to control read and write access. &lt;br&gt; • Select an <strong>Index reorganization mode</strong> to clean up empty pages, delete after cleaning up empty pages, or convert to a type 2 index. &lt;br&gt; • If you did not enable <strong>Inplace Reorganization</strong>, select a <strong>Tablespace</strong>. &lt;br&gt; • Select a <strong>Long Tablespace</strong>. &lt;br&gt; • Enable or disable the following options: <strong>Index scan</strong>, <strong>Reorganize long field and LOB data</strong>, <strong>Reset Dictionary</strong>, <strong>No truncate table</strong>, and <strong>Reorganize all partitions</strong>.</td>
</tr>
<tr>
<td>Partitions</td>
<td>If you did not enable <strong>Reorganize all partitions</strong>, select the partitions to reorganize.</td>
</tr>
<tr>
<td>Preview</td>
<td>Preview the DDL code generated from your choices.</td>
</tr>
</tbody>
</table>

For more information, see [Completing the Reorganize Dialog Box](#).

Reorganizing Oracle Objects
The Reorganize dialog box lets you reduce query processing time against tables.

The table below describes the options and functionality on the Reorganize dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporary Tablespace</td>
<td>Associates a temporary tablespace with the table’s tablespace. You can select another tablespace from the list.</td>
</tr>
</tbody>
</table>

For more information, see [Completing the Reorganize Dialog Box](#).

Reorganizing Oracle Objects
The Reorganize dialog box lets you reduce query processing time against tables.

The table below describes the options and functionality on the Reorganize dialog box:

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If you want to move the table(s) to a new tablespace, click the <strong>New Tablespace</strong> list and select the target tablespace.</td>
</tr>
</tbody>
</table>
For more information, see Completing the Reorganize Dialog Box.

Reorganizing Sybase ASE Objects
The Reorganize dialog box lets you reduce query processing time against tables. This functionality is available for both tables and indexes.

For more information, see Completing the Reorganize Dialog Box.

Reorganizing Sybase ASE Tables
The Reorganize Table dialog box lets you reduce the query processing time against a table by reorganizing the table to ensure that space is properly allocated to it. For lengthy reorganization processes, this dialog box also lets you execute a process in increments, lets you resume an incomplete process, and lets you specify the duration of each increment.

**TIP:** Frequent update activity on a table can cause data rows to migrate and to chain over multiple data pages. Chained or forwarded rows can degrade performance because more physical reads are required to access a row of data. Consequently, you should monitor chained rows regularly to spot performance bottlenecks before they become severe. In addition, altering physical storage parameters can lead to fragmentation of space on your data pages, which also results in reduced performance levels.

You should consider reorganizing a table if you are experiencing slow performance due to:

- A large number of chained or forwarded rows on your data pages

---

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
</table>
| Data Block Storage | - In the **Percent Free** box, type the appropriate percent free value for the table.  
- In the **Percent Used** box, type the appropriate percent used value for the table.  
- In the **Initial Transactions** box, type the appropriate initial transactions value for the table.  
- In the **Max Transactions** box, type the appropriate maximum transactions value for the table. |
| Extents            | - In the **Initial Extent** box, type the appropriate initial extent KB value for the table.  
- In the **Next Extent** box, type the appropriate next extent KB value for the table.  
- In the **Percent Increase** box, type the appropriate percent increase value for the table.  
- In the **Minimum Extents** box, type the appropriate minimum extents value for the table.  
- In the **Maximum Extents** box, type the appropriate maximum extents value for the table. |
| Freelists          | - In the **Freelists** box, type the new value.  
- In the **Freelist Group** box, type the new value.  
- To modify the bufferpool, click the **Bufferpool**, click the list and select **DEFAULT** to use the default bufferpool, **KEEP** to retain the object in memory to avoid I/O conflicts, or **RECYCLE** to save cache space by ridding data blocks from memory as soon as they are no longer in use. This option is only available for Oracle8. |
- A large amount of fragmentation in your data pages

**NOTE:** You can reorganize tables in Sybase ASE versions 12 and 12.5.

The table below describes the options and functionality on the Reorganize Table dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compact</td>
<td>Lets you reclaim space and undo row forwarding. Minimizes interference with other activities by using multiple small transactions of brief duration. Each transaction is limited to eight pages of reorg processing. These three commands also provide <code>resume</code> and time options that allow you to set a time limit on how long a reorg runs and to resume a reorg from the point at which the previous reorg stopped, making it possible to use a series of partial reorganizations at off-peak times to reorg a large table.</td>
</tr>
<tr>
<td>Reclaim Space</td>
<td>Lets you reclaim unused space resulting from deletions and row-shortening updates on a page. Minimizes interference with other activities by using multiple small transactions of brief duration. Each transaction is limited to eight pages of reorg processing. These three commands also provide <code>resume</code> and time options that allow you to set a time limit on how long a reorg runs and to resume a reorg from the point at which the previous reorg stopped, making it possible to use a series of partial reorganizations at off-peak times to reorg a large table.</td>
</tr>
<tr>
<td>Rebuild</td>
<td>Lets you undo row forwarding and reclaim unused page space. It also rewrites all rows to comply with the target table’s clustered index, writes rows to data pages to comply with space management setting changes (via <code>sp_chgattribute</code>), and drops and re-creates all the target table’s (or tables’) indexes. Reorg rebuild holds an exclusive table lock for its entire duration. On a large table this can be a significant amount of time. However, reorg rebuild accomplishes everything that dropping and re-creating a clustered index does and takes less time. In addition, reorg rebuild rebuilds the table using all of the table’s current space management settings. Dropping and re-creating an index does not use the space management setting for <code>reservepagegap</code>. In most cases, reorg rebuild requires additional disk space equal to the size of the table it is rebuilding and its indexes.</td>
</tr>
<tr>
<td>Undo Row Forwarding</td>
<td>Lets you undo row forwarding, a process that occurs when an update increases a row’s length in a data-only-locked table such that the row is too large to fit on its original page.</td>
</tr>
<tr>
<td>Options</td>
<td>Start at the point where a previous reorg left off - Select to resume a previously initiated but incomplete partial reorganization. Then specify the duration for which you want the resumed reorganization to continue before stopping again. This box is disabled for the rebuild command.</td>
</tr>
</tbody>
</table>

For more information, see [Completing the Reorganize Dialog Box](#).

**Incremental Reorganizations**

If target tables are too long to reorganize in one session, DBArtisan lets you reorganize them in increments over multiple sessions by specifying a maximum duration for each session. After DBArtisan reorganizes tables for the specified duration, the operation stops until you resume it again from the Options box of the ReOrganize Table dialog box. The Options box lets you specify to resume a previously initiated but incomplete partial reorganization. It also lets you specify the duration for which you want a resumed reorganization to continue before stopping again. The Option box is disabled for the rebuild command.

**NOTE:** The duration you specify refers to elapsed time, not CPU time.
In the option box, if you select the check box without specifying a duration, DBArtisan executes the reorg at the point where the previous reorg stopped and continues to the end of the target tables. If you clear the check box and specify a duration, the reorg starts at the beginning of the target tables and continues for the specified number of minutes. If you select the check box and specify a duration, DBArtisan runs the reorg from the point where it last left off, and continues for the specified number of minutes.

**NOTE:** If you reorganize a table using one command (Compact, Reclaim Space, or Undo Forwarding) for a specified duration, you cannot resume the process from its resume point using a different command. For example, you cannot compact a table for an hour, and then reclaim space on the remainder of the table. A resumed reorganization process must utilize the same command from start to finish. Selecting a different command begins a new reorganization process.

**CAUTION:** While this option lets you reorganize a large table in multiple manageable pieces, any updates to the table between reorganization runs can cause pages to be skipped or processed more than once.

For more information, see [Completing a Dialog Box](#).

**Reorganize Sybase ASE Indexes**

The Reorganize Index dialog box lets you reduce the query processing time against a table by running a reorg rebuild command on the target index.

This operation:

- Undoes row forwarding and reclaim unused page space
- Rewrites all rows in the table to comply with the table’s clustered index
- Writes rows to data pages to comply with space management setting changes (via sp_chgattribute)
- Drops and re-creates the table’s indexes

Reorg rebuild holds an exclusive table lock for its entire duration. On a large table this can be a significant amount of time. However, reorg rebuild accomplishes everything that dropping and re-creating a clustered index does and takes less time. In addition, reorg rebuild rebuilds the table using all of the table’s current space management settings. Dropping and re-creating an index does not use the space management setting for reservepagegap. In most cases, reorg rebuild requires additional disk space equal to the size of the table it is rebuilding and its indexes.

For more information, see [Completing the Reorganize Dialog Box](#).

**Reorganize Tablespace**

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Reorganize Tablespace Wizard reorganizes a tablespace to improve access performance and reclaim fragmented space. The wizard can also reorganize a single partition or range of partitions of a partitioned table space. You can specify the degree of access to the data during reorganization, and optionally collect inline statistics.

**Completing the Reorganize Tablespace Wizard**

To complete the Reorganize Tablespace Wizard, do the following:

1. On the **Datasource Explorer**, select the tablespaces node.

   DBArtisan displays the tablespaces in the right pane of the Explorer window.
2 On the **Datasource Explorer** tool bar, click **Command**, and then select **Reorganize**.

DBArtisan opens the Reorganize Tablespace Wizard.

3 Complete the wizard panels, and then click **Finish**.

DBArtisan opens the **Preview: Create dialog box**.

For more information, see **Reorganize Tablespace**.

### Reorganize Tablespace Wizard - Panel 1

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want the utility to be</td>
<td>Select to make the utility restartable.</td>
</tr>
<tr>
<td>restartable?</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>Displays the list of databases in the system. You can select the tablespace</td>
</tr>
<tr>
<td>Database</td>
<td>database or click the filter button to limit the list of databases to a</td>
</tr>
<tr>
<td>Database</td>
<td>string pattern.</td>
</tr>
<tr>
<td>Tablespace</td>
<td>Displays the list of tablespaces.</td>
</tr>
<tr>
<td>Partition</td>
<td>Lets you specify a partition or a range of partitions.</td>
</tr>
<tr>
<td>Reuse</td>
<td>Select to reuse this tablespace.</td>
</tr>
</tbody>
</table>

For more information, see **Reorganize Tablespace**.

### Reorganize Tablespace Wizard - Panel 2

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td>List of the sequence of events after you execute the wizard.</td>
</tr>
<tr>
<td>Sort Data</td>
<td>Sorts columns alphanumerically in single result sets.</td>
</tr>
<tr>
<td>No Sysrec</td>
<td>Select Yes for no sysrec.</td>
</tr>
<tr>
<td>Sortkeys</td>
<td>Select Yes for sortkeys. Specifies that index keys are to be sorted and</td>
</tr>
<tr>
<td></td>
<td>built in parallel during the SORTBLD phase to improve performance.</td>
</tr>
</tbody>
</table>

For more information, see **Reorganize Tablespace**.

### Reorganize Tablespace Wizard - Panel 3

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Share level</td>
<td>Lets you select share level that allows users to view but not modify the table data: None, Reference, or Change.</td>
</tr>
</tbody>
</table>
Deadline Expression Builder
The table below describes the options and functionality on the Deadline Expression Builder dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Date</td>
<td>Lets you select today as the basis of the deadline, click + or -, type the number of years, months, days, hours, minutes, seconds and microseconds.</td>
</tr>
<tr>
<td>Current Timestamp</td>
<td>Lets you select the current timestamp as the basis of the deadline, click + or -, type the number of years, months, days, hours, minutes, seconds and microseconds.</td>
</tr>
</tbody>
</table>

For more information, see Reorganize Tablespace.

Reorganize Tablespace Wizard - Panel 4
The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Site Primary</td>
<td>Lets you specify the Local Site Primary.</td>
</tr>
<tr>
<td>Local Site Backup</td>
<td>Lets you specify the Local Site Backup.</td>
</tr>
<tr>
<td>Recovery Site Primary</td>
<td>Lets you specify the Recovery Site Primary.</td>
</tr>
<tr>
<td>Recovery Site Backup</td>
<td>Lets you specify the Recovery Site Backup.</td>
</tr>
</tbody>
</table>

For more information, see Reorganize Tablespace.
Reorganize Tablespace Wizard - Panel 5
The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off Pos Limit</td>
<td>The specified value is compared to the result of the calculation ((\text{NEAROFFPOS} + \text{FAROFFPOS}) \times 100 / \text{CARDF}) for the specified partitions in SYSIBM.SYSINDEXPART for the explicit clustering indexes for every table in the specified table space. Alternatively, the values in SYSINDEXPART are checked for a single non-partitioned table space, or for each partition if you specified an entire partitioned table space as the target object. If at least one calculated value exceeds the OFFPOSLIMIT value, REORG is performed or recommended. <strong>NOTE:</strong> This option is valid for non-LOB table spaces only.</td>
</tr>
<tr>
<td>Ind Ref Limit</td>
<td>The specified value is compared to the result of the calculation ((\text{NEARINDREF} + \text{FARINDREF}) \times 100 / \text{CARDF}) for the specified partitions in SYSIBM.SYSTABLEPART for the specified table space. Alternatively, the values in SYSTABLEPART are checked for a single non-partitioned table space, or for each partition if you specified an entire partitioned table space as the target object. If at least one calculated value exceeds the INDREFLIMIT value, REORG is performed or recommended. <strong>NOTE:</strong> This option is valid for non-LOB table spaces only.</td>
</tr>
<tr>
<td>Report Only</td>
<td>The reorganization is only recommended, not performed.</td>
</tr>
<tr>
<td>Unload</td>
<td>Specifies whether the utility job is to continue processing or end after the data is unloaded.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Continue</strong> - Specifies that, after the data has been unloaded, the utility continues processing.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Pause</strong> - Specifies that after the data has been unloaded, processing ends.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Only</strong> - Specifies that after the data has been unloaded, the utility job ends and the status in SYSIBM.SYSUTIL corresponding to this utility ID is removed.</td>
</tr>
<tr>
<td></td>
<td>- <strong>External</strong> - Specifies that after the data has been unloaded, the utility job is to end and the status in SYSIBM.SYSUTIL corresponding to this utility ID is removed.</td>
</tr>
</tbody>
</table>

For more information, see [Reorganize Tablespace](#).

Reorganize Tablespace Wizard - Panel 5
**NOTE:** This panel is displays only if the share level is Change.

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>Displays the schema names in the system.</td>
</tr>
<tr>
<td>Table</td>
<td>Displays all tables for the selected schema.</td>
</tr>
<tr>
<td>Maxrow</td>
<td>Lets you specify the maximum amount of time for the last iteration of log processing. During that iteration, applications have read-only access. Type an integer value or select DEFER. DEFER Specifies that the iterations of log processing with read/write access can continue indefinitely.</td>
</tr>
<tr>
<td>Drain</td>
<td>Lets you select Writers or All. Specifies that DBArtisan drains the write claim class after the delay. The number of log records, and thus the estimated time, for a future iteration of log processing will be 0.</td>
</tr>
</tbody>
</table>
Reorganize Tablespace Wizard - Panel 6

**NOTE:** This panel displays only if the share level is Reference or Change.

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Site Primary Copy</td>
<td>Lets you specify the Local Site Primary Copy.</td>
</tr>
<tr>
<td>Local Site Backup Copy</td>
<td>Lets you specify the Local Site Backup Copy.</td>
</tr>
<tr>
<td>Recovery Site Primary Copy</td>
<td>Lets you specify the Recovery Site Primary Copy.</td>
</tr>
<tr>
<td>Recovery Site Backup Copy</td>
<td>Lets you specify the Recovery Site Backup Copy.</td>
</tr>
</tbody>
</table>

For more information, see [Reorganize Tablespace](#).

Reorganize Tablespace Wizard - Panel 7

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Off Pos Limit     | The specified value is compared to the result of the calculation ($\text{NEAROFFPOS} + \text{FAROFFPOS}$) $\times 100 / \text{CARDF}$ for the specified partitions in SYSIBM.SYSINDEXPART for the explicit clustering indexes for every table. Alternatively, the values in SYSINDEXPART are checked for a single non-partitioned table space, or for each partition if you specified an entire partitioned table space as the target object. If at least one calculated value exceeds the OFFPOSLIMIT value, REORG is performed or recommended.  
**NOTE:** This option is valid for non-LOB table spaces only. |

For more information, see [Reorganize Tablespace](#).
For more information, see Reorganize Tablespace.

Reorganize Tablespace Wizard - Panel 6
The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ind Ref Limit</td>
<td>The specified value is compared to the result of the calculation (NEARINDREF + FARINDREF) × 100 / CARDF for the specified partitions in SYSIBM.SYSTABLEPART for the specified table space. Alternatively, the values in SYSTABLEPART are checked for a single non-partitioned table space, or for each partition if you specified an entire partitioned table space as the target object. If at least one calculated value exceeds the INDREFLIMIT value, REORG is performed or recommended. <strong>NOTE:</strong> This option is valid for non-LOB table spaces only.</td>
</tr>
<tr>
<td>Report Only</td>
<td>The reorganization is only recommended, not performed.</td>
</tr>
<tr>
<td>Unload</td>
<td>Specifies whether the utility job is to continue processing or end after the data is unloaded.</td>
</tr>
<tr>
<td></td>
<td>Continue - Specifies that, after the data has been unloaded, the utility continues processing.</td>
</tr>
<tr>
<td></td>
<td>Pause - Specifies that after the data has been unloaded, processing ends.</td>
</tr>
<tr>
<td></td>
<td>Only - Specifies that after the data has been unloaded, the utility job ends and the status in SYSIBM.SYSUTIL corresponding to this utility ID is removed.</td>
</tr>
<tr>
<td></td>
<td>External - Specifies that after the data has been unloaded, the utility job is to end and the status in SYSIBM.SYSUTIL corresponding to this utility ID is removed.</td>
</tr>
</tbody>
</table>

For more information, see Reorganize Tablespace.

Reorganize Tablespace Wizard - Statistics Tables
The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Tables</td>
<td>Lets you select tables (and columns) to collect statistics for. The sample column is editable and you can type an integer value into this column.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Table Selector dialog box.</td>
</tr>
</tbody>
</table>
Reorganize Tablespace Wizard - Statistics Indexes

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Index</td>
<td>Lets you specify the indexes and the correlation values for collecting the statistics. The sample column is editable and you can type an integer value into this column.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Index Selector dialog box.</td>
</tr>
<tr>
<td>Set Properties</td>
<td>Lets you specify the correlation information for each index.</td>
</tr>
</tbody>
</table>

For more information, see [Reorganize Tablespace](#).

Reorganize Tablespace Wizard - Panel 7

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Punch DDN</td>
<td>Lets you select punch DDN from the list. Contains generated LOAD statements that let you reload the discarded data. Required if Unload External option is chosen on Panel 5.</td>
</tr>
<tr>
<td>Discard DDN</td>
<td>Lets you select discard DDN from the list.</td>
</tr>
<tr>
<td>Unload DDN</td>
<td>Lets you select DD name of the unload data set. Required unless NOSYSREC on Panel 2 or Share level CHANGE Panel 3 is selected.</td>
</tr>
<tr>
<td>Work DDN</td>
<td>Lets you select the DD statement for a temporary data set used for intermediate output.</td>
</tr>
</tbody>
</table>

For more information, see [Reorganize Tablespace](#).

Reorganize Tablespace Wizard - Panel 7

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discard</td>
<td>Lets you select discard DDN from the list.</td>
</tr>
<tr>
<td>No Pad</td>
<td>DBArtisan does not add a character to fill empty space.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Table Selector dialog box.</td>
</tr>
<tr>
<td>Set Condition</td>
<td>Lets you customize the selection of columns in the selected tables.</td>
</tr>
</tbody>
</table>

For more information, see [Reorganize Tablespace](#).
Reorganize Tablespace Wizard - Panel 8
The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to specify the device type?</td>
<td>Lets you specify the device type.</td>
</tr>
<tr>
<td>Do you want to specify the sort number?</td>
<td>Lets you type the number of datasets to be dynamically allocated by the sort program. The range of the text field is 0 to 2147483647.</td>
</tr>
<tr>
<td>Preformat</td>
<td>Lets you specify that the remaining pages are preformatted up to the high allocated RBA in the table space and index spaces associated with the table specified in table-name. The preformatting occurs after the data has been loaded and the indexes are built.</td>
</tr>
</tbody>
</table>

For more information, see Reorganize Tablespace.

Reorganize Tablespace Wizard - Panel 9
The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Punch DDN</td>
<td>Lets you select punch DDN from the list. Contains generated LOAD statements that let you reload the discarded data. Required if Unload External option is chosen on Panel 5.</td>
</tr>
<tr>
<td>Discard DDN</td>
<td>Lets you select discard DDN from the list.</td>
</tr>
<tr>
<td>Unload DDN</td>
<td>Lets you select DD name of the unload data set. Required unless NOSYSREC on Panel 2 or Share level CHANGE Panel 3 is selected.</td>
</tr>
<tr>
<td>Work DDN</td>
<td>Lets you select the DD statement for a temporary data set used for intermediate output.</td>
</tr>
</tbody>
</table>

For more information, see Reorganize Tablespace.

Condition Dialog Box
The Condition Dialog Box lets you type free form condition text.

**NOTE:** DBArtisan does not test the correctness of the condition text.

Report

**NOTE:** This functionality is available for all platforms.

The Generate Report dialog box lets you generate detailed reports about all the objects in your database. The reports give you the complete information about the properties of your objects. They derive their contents from the information displayed in the object editors. To facilitate the online publication of these reports, DBArtisan generates these reports in HTML. You can view these reports directly in DBArtisan’s built-in HTML browser.

Completing the Generate Report Dialog Box
To complete the Generate Report dialog box, do the following:

1. On the Datasource Explorer, select the target object node.

DBArtisan displays the target objects in the right pane of the Explorer window.
On the **Datasource Explorer** tool bar, click **Command**, and then select **Report**.

OR

In the right pane of the application, right-click the target object, and then select **Report**.

**DBArtisan** opens the Generate Report dialog box.

### Generating a Summary Report

1. Select the **Summary Report (Listing all items)** option.

   **DBArtisan** displays the report. You can find data in the report, save, send, and print the report.

2. To exit the report, click **Close**.

### Generating a Detail Report

1. Select the **Detail Report (Report on each supported item)** option.

   **DBArtisan** opens the Report dialog box.

2. In **Report Home Page File Name**, type the report name or click **Browse** to locate the report.


   **DBArtisan** displays Object Name and Object Type.

4. Click **Execute**.

   **DBArtisan** displays the report. You can find data in the report, save, send, and print the report.

5. To exit the report, click **Close**.

For more information, see:

*Report*

*Completing a Dialog Box*

---

**Restart**

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows only.

The Restart Sequence dialog box lets you restart a sequence.

A sequence is a programmable database objects that generates a definable sequence of values. A sequence can be made available to many users.

For more information, see *Completing a Dialog Box*.

---

**Revoke Role**

**NOTE:** This functionality is available for all platforms.

The Revoke Role dialog box lets you revoke a role to restrict user access to objects.
Completing the Revoke Roles Dialog Box

To complete this dialog box, do the following:

1. On the Datasource Explorer, select the target user.
2. In the right pane of the application, right-click the target user, and select Open.
   
   DBArtisan opens the Users Editor.
3. On the Definition Tab of the Users Editor, select the Role, and then click Drop.
   
   DBArtisan opens the Revoke Roles dialog box.
4. Select the role(s) to revoke.
5. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Preview]</td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td>![Schedule]</td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td>![Execute]</td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

For more information, see Completing a Dialog Box.

Runstats Index

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS.

The Runstats Index Wizard gathers statistics on indexes. If no statistics are available, DB2 makes fixed default assumptions which may result in inefficiencies. To ensure the effectiveness of the paths selected, you can run the wizard:

- After a table is loaded.
- After an index is physically created.
- After running the Rebuild Index Wizard, or Reorganize Index Wizard without collecting inline statistics.
- Before running Reorganize Index Wizard with the OFFPOSLIMIT, INDREFLIMIT, or LEAFDISTLIMIT options.

For more information, see Runstats Index Wizard - Panel 1.

Runstats Index Wizard - Panel 1

The table below describes the options and functionality on this panel of the Runstats Index Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select indexes</td>
<td>Displays the available indexes. The Partition Column is editable and can contain one or more numbers separated by commas or, a range of numbers separated by: (e.g. 1,2,3 or 1:3 or 1 are valid entries). The maximum value for the partition column is 254 and the minimum value is 0.</td>
</tr>
</tbody>
</table>
Runstats Index Wizard - Panel 2
The table below describes the options and functionality on this panel of the Runstats Index Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select tablespaces</td>
<td>Displays the available tablespaces. The Partition Column is editable and can contain one or more numbers separated by commas or, a range of numbers separated by: (e.g. 1,2,3 or 1:3 or 1 are valid entries). The maximum value for the partition column is 254 and the minimum value is 0.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Tablespace Selector dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected items.</td>
</tr>
<tr>
<td>Set Properties</td>
<td>Opens the Correlation Statistics dialog box for the selected row.</td>
</tr>
</tbody>
</table>

For more information, see Runstats Index.

Runstats Index Wizard - Panel 3
The table below describes the options and functionality on this panel of the Runstats Index Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access level while RUNSTATS is running</td>
<td>Lets you select an access level, either Readonly or Change.</td>
</tr>
<tr>
<td>Do you want to output message to SYSPRINT?</td>
<td>Lets you specify to output message to SYSPRINT.</td>
</tr>
<tr>
<td>Do you want to force aggregation or rollup processing to be done even though some parts do not contain data?</td>
<td>Lets you specify to process with forced aggregation, gathered into a mass, or rollup.</td>
</tr>
</tbody>
</table>

For more information, see Runstats Index.

Runstats Index Wizard - Panel 4
The table below describes the options and functionality on this panel of the Runstats Index Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Statistics</td>
<td>Lets you update the catalog statistics.</td>
</tr>
<tr>
<td>History Statistics</td>
<td>Lets you update the history statistics.</td>
</tr>
</tbody>
</table>
For more information, see Runstats Index.

Set Correlation Option Dialog Box
The table below describes the options and functionality on the Set Correlation Option Dialog Box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyCard</td>
<td>Collects all of the distinct values in all of the 1 to n key column</td>
</tr>
<tr>
<td></td>
<td>combinations for the specified indexes. n is the number of columns in the</td>
</tr>
<tr>
<td></td>
<td>index.</td>
</tr>
<tr>
<td>Frequent Item</td>
<td>Lets you add items to the grid.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Add Frequent Item dialog box</td>
</tr>
</tbody>
</table>

For more information, see Runstats Index.

Add Frequent Item Dialog Box
The table below describes the options and functionality on the Add Frequent Item Dialog Box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Columns</td>
<td>Lets you type the number of columns. Default is 1.</td>
</tr>
<tr>
<td>Count</td>
<td>Lets you type the count. Default is 10.</td>
</tr>
</tbody>
</table>

For more information, see Runstats Index.

Runstats Tablespace
**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS.

The Runstats Tablespace Wizard gathers statistics on tablespaces, indexes, and columns. DBArtisan records this information in the DB2 catalog and uses it to select access paths to data during the bind process. The wizard lets you evaluate database design and helps you decide to reorganize tablespaces or indexes.

For more information, see Runstats Tablespace Wizard - Panel 1.

Runstats Tablespace Wizard - Panel 1
The table below describes the options and functionality on this panel of the Runstats Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>Lets you select a database.</td>
</tr>
<tr>
<td>Tablespace Name</td>
<td>Lets you select a tablespace.</td>
</tr>
<tr>
<td>Partition Number</td>
<td>Lets you specify on which the command needs to be run.</td>
</tr>
<tr>
<td>Do you want to runstats on all tables in this tablespace?</td>
<td>Default is Yes.</td>
</tr>
<tr>
<td>Do you want to runstats on all indexes in this tablespace?</td>
<td>Default is Yes.</td>
</tr>
</tbody>
</table>
For more information, see Runstats Tablespace.

Runstats Tablespace Wizard - Panel 2

**NOTE:** This panel displays if you selected No for either option on Panel 1.

The table below describes the options and functionality on this panel of the Runstats Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Sample</td>
<td>Lets you type a table sample. Default is 25. You can enter any value between 1 and 100.</td>
</tr>
<tr>
<td>KeyCard</td>
<td>Collects all of the distinct values in all of the 1 to n key column combinations for the specified indexes. n is the number of columns in the index.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Add Frequent Item dialog box</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes selected item(s.)</td>
</tr>
</tbody>
</table>

For more information, see Runstats Tablespace.

Runstats Tablespace Wizard - Panel 3

**NOTE:** This panel displays if you selected No for the Runstats on All Tables option on panel 1.

The table below describes the options and functionality on this panel of the Runstats Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Tables</td>
<td>Lets you select the tables in the table that runstats will be run on. The Sample column is blank initially for each row. It is editable with values between 1 - 100. The Indicator column is blank initially for each new row that is added. However if you set the custom columns for a row (via set properties button), this column shows an asterisk (*).</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Table Selector dialog box</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected table(s.)</td>
</tr>
<tr>
<td>Set Properties</td>
<td>Lets you customize the selection of columns in the selected table(s.)</td>
</tr>
</tbody>
</table>

For more information, see Runstats Tablespace.

Runstats Tablespace Wizard - Panel 3

The table below describes the options and functionality on this panel of the Runstats Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Indexes</td>
<td>Lets you select index(es) for the runstats operation.</td>
</tr>
<tr>
<td>Add</td>
<td>Opens the Select Column for Table dialog box</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes selected index(es).</td>
</tr>
<tr>
<td>Set Properties</td>
<td>Opens the Set Correlation Options dialog box</td>
</tr>
</tbody>
</table>
For more information, see Runstats Tablespace.

Select Column for Table Dialog Box
The table below describes the options and functionality on the Select Column for Table Dialog Box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Name</td>
<td>Displays the column name.</td>
</tr>
<tr>
<td>Included</td>
<td>Select to include column.</td>
</tr>
<tr>
<td>Select All</td>
<td>Click to select all columns.</td>
</tr>
<tr>
<td>Unselect All</td>
<td>Click to deselect all columns.</td>
</tr>
</tbody>
</table>

Runstats Tablespace Wizard - Options
The table below describes the options and functionality on this panel of the Runstats Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access level while RUNSTATS is running</td>
<td>Lets you select an access level, either Readonly or Change.</td>
</tr>
<tr>
<td>Do you want to output message to SYSPRINT?</td>
<td>Lets you specify to output message to SYSPRINT.</td>
</tr>
<tr>
<td>Do you want to force aggregation or rollup processing to be done even though some parts do not contain data?</td>
<td>Lets you specify to process with forced aggregation, gathered into a mass, or rollup.</td>
</tr>
</tbody>
</table>

For more information, see Runstats Tablespace.

Runstats Tablespace Wizard - Statistics
The table below describes the options and functionality on this panel of the Runstats Tablespace Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Statistics</td>
<td>Lets you update the catalog statistics.</td>
</tr>
<tr>
<td>History Statistics</td>
<td>Lets you update the history statistics.</td>
</tr>
</tbody>
</table>

For more information, see Runstats Tablespace.

Table Selector Dialog Box
The table below describes the options and functionality on the panel of the Table Selector dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace Like</td>
<td>Displays the tablespace.</td>
</tr>
<tr>
<td>Table Creator</td>
<td>Displays the table(s) creator of the tablespace that was selected on Panel 2.</td>
</tr>
</tbody>
</table>
For more information, see Runstats Tablespace.

Index Selector Dialog Box
The table below describes the options and functionality on the panel of the Index Selector dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Creator</td>
<td>Displays the index(es) creator of the tablespace that was selected on Panel 2.</td>
</tr>
<tr>
<td>Index</td>
<td>Displays the index(es) of the tablespace that was selected on Panel 2.</td>
</tr>
<tr>
<td>Index Selector</td>
<td>Lets you select one or more rows in this list.</td>
</tr>
<tr>
<td>Query</td>
<td>Repopulates Index Selector based on the tablespace query.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all rows in Index Selector.</td>
</tr>
<tr>
<td>Unselect All</td>
<td>Unselects all rows in Index Selector.</td>
</tr>
<tr>
<td>Add</td>
<td>Click to add the index to the wizard.</td>
</tr>
</tbody>
</table>

For more information, see Runstats Tablespace.

Schema
NOTE: This functionality is available for all platforms.

The Schema dialog box lets you view everything that makes up a selected table.
For more information, see Viewing Schema.

Viewing Schema
To view schema, do the following:

1. On the Datasource Explorer, select the target object node.

   DBArtisan displays the target objects in the right pane of the Explorer window.
2 On the **Datasource Explorer** tool bar, click **Command**, and then select **Schema**.

**OR**

In the right pane of the application, right-click the target object, and then select **Schema**.

DBArtisan opens Schema window.

For more information, see [Completing a Dialog Box](#).

**Select * From**

**NOTE:** This functionality is available for all platforms.

The Select * From dialog box lets you retrieve all data from a selected table or view.

**Completing the Select * From Dialog Box**

To complete this dialog box, do the following:

1 On the **Datasource Explorer**, select the target object node.

DBArtisan displays the target objects in the right pane of the Explorer window.

2 On the **Datasource Explorer** tool bar, click **Command**, and then select **Select * From**.

**OR**

In the right pane of the application, right-click the target object, and then select **Select * From**.

3 DBArtisan opens the Select * From dialog box.

For more information, see [Completing a Dialog Box](#).

**Set Integrity**

**NOTE:** This functionality is available for IBM DB2 only.


**To Open the Set Integrity Wizard**

1 On the **Explorer**, expand a DB2 database, and click the **Table** node.

2 Select the appropriate table(s), right-click, and choose **Set Integrity** from the menu. You can select more than one table by clicking CTRL + the tables you want.

**OR**

Select the appropriate table, click **Command** on the toolbar, and then choose **Set Integrity** from the drop-down menu.
The table below lists all fields you can see in the Set Integrity Wizard. Depending on the options you choose, you may not see them all.

<table>
<thead>
<tr>
<th>Required Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tables</strong></td>
<td>The list of one or more tables you selected appear.</td>
</tr>
<tr>
<td><strong>Integrity Option</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Integrity Option | OFF--When selected, tables have their foreign key and check constraints, and column generation disabled and so are put in a pending state. This also means materialized query or staging tables are not immediately refreshed and are put in a pending state.  
TO DATALINK RECONCILE PENDING--When selected, DATALINK integrity is disabled and tables are put in a check pending no access state. Dependent and descendant tables are not affected.  
IMMEDIATE CHECKED--This turns on a table’s integrity checking turned on. Any checking that was deferred is carried out.  
FULL ACCESS--Tables become fully accessible as soon as the SET INTEGRITY statement executes.  
PRUNE--This is appropriate only for staging tables. The content of the staging table is pruned and set to an inconsistent state. If any table in the table-name list is not a staging table, an error is returned.  
UNCHECKED--Allows you turn on some or all integrity checking but the table will not be checked for integrity violations. This can affect data integrity. |
| **Table Readability/Cascade/Descendent Types** | |
| Specifies the readability of the table while it is in check pending state: | NO ACCESS--The table(s) are put in a check pending no access state so read/write access to the table is prohibited.  
READ ACCESS--The table(s) are put in a check pending read state. This allows read access to the non-appended portions of any tables. |
| Specifies whether to be immediately cascaded to all descendents | CASCADE IMMEDIATE--The check pending state for foreign key constraints is immediately extended to descendant foreign key constraints or to materialized query or staging tables.  
CASCADE DEFERRED--Only the selected tables are put in the check pending state. Descendant foreign key, materialized query, or staging tables remain unchanged. |
| Descendent Types | Materialized Query Tables--When selected, the check pending state is immediately cascaded to only descendant materialized query tables.  
Foreign Key Tables--When selected, the check pending state is cascaded immediately only to descendant foreign key tables.  
Staging Tables--When selected, the check pending state is cascaded immediately only to descendant staging tables. |
| Do you want to check on the appended portion (if any) of the table? | Default/Yes/No  
Force Generated--If you do not specify this generated column current values will be compared to the computed value of the expression as if an equality check constraint existed. If this is specified, generated columns are computed only for the appended portion.  
Prune--Possible only for staging tables. When you check this, the contents of the staging table are pruned and the staging table is set to an inconsistent state.  
Full Access--When selected, tables will become accessible after the SET INTEGRITY statement executes. |
| Specify Exception Tables | |
| List of Base Tables | Any row that is in violation of a foreign key or check constraint is copied to the exception table you select. |
The Set Database(s) Online/Offline dialog box lets you disable your databases to prevent access, and enable your databases to grant access through the Datasource menu.

NOTE: This functionality is available for Microsoft SQL Server, Oracle, and Sybase ASE only.

Important Notes
For Sybase, DBArtisan only lets you set databases online.

To set one or more databases online or offline:

1. Connect to the datasource that contains the database. For more information, see Datasource Management.
2. On the Datasource Explorer, select the Databases node.
   
   DBArtisan displays the Databases in the right pane of the Explorer window.
3. In the right pane of the Explorer window, select one or more target databases.
4. On the Datasource Explorer tool bar, click Command, and then select Set Online/Offline.
   
   DBArtisan opens the Checkpoint dialog box.
5. Use the following table as a guide to working through the panels of the dialog box.

<table>
<thead>
<tr>
<th>Required Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMMEDIATE UNCHECKED options</td>
<td>Foreign Key--These constraints are turned on when the table is removed from check pending status. Check--Check constraints are turned on when the table is removed from check pending status. Datalink Reconcile Pending--DATALINK integrity constraints are turned on when the table is removed from check pending status. Materialized Query--Immediate refreshing is turned on for a materialized query table when it is removed from a check pending state. Generated Column--When the table is removed from check pending status, generated columns are turned on. Staging--Immediate propagation is turned on for a staging table.</td>
</tr>
<tr>
<td>Do you want tables to become fully accessible after the SET INTEGRITY statement executes?</td>
<td>Yes/No</td>
</tr>
</tbody>
</table>

### Preview DDL

**DDL View**

The DDL window displays the command code you created by completing the prior panels of the wizard. You have the ability to save the code to disk, to print it, to email it, or to schedule the execution of the task. The code itself is read-only and can only be changed by returning to prior panels of the wizard.

### Set Online/Offline

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action options</td>
<td>Set offline</td>
</tr>
</tbody>
</table>
Set Tablespace Quota

**NOTE:** This functionality is available for Oracle only.

The Set Tablespace Quota dialog box lets you set a quota for a tablespace. When you assign a quota:

- Users with privileges to create certain types of objects can create those objects in the specified tablespace.
- Oracle limits the amount of space that can be allocated for storage of a user's objects within the specified tablespace to the amount of the quota.

The table below describes the options and functionality on the Set Tablespace Quota dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quota</td>
<td>Lets you set a quota on the selected tablespace. You can select an unlimited, or specified size. Unlimited - Lets you place an unlimited quota on the tablespace. Other - Lets you place a specified quota in KB or MB on the tablespace.</td>
</tr>
</tbody>
</table>

### Completing the Set Tablespace Quota Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Tablespaces** node.
   
   DBArtisan displays the Tablespaces in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Quota**.
   
   OR
   
   In the right pane of the application, right-click the tablespace, and then select **Quota**.
   
   DBArtisan opens the Set Tablespace Quota dialog box.

3. Click the **Unlimited** or **Other** option button.

4. If you click **Other**, type the value of the quota and then click list to indicate **KB** or **MB**.

5. Click **OK**.

6. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="preview.png" alt="Preview" /></td>
<td>Opens the <strong>Preview dialog box</strong>.</td>
</tr>
<tr>
<td><img src="schedule.png" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <strong>Scheduling</strong>.</td>
</tr>
</tbody>
</table>
For more information, see:

Set Tablespaces Quota
Completing a Dialog Box

Set UNDO Tablespace

**NOTE:** This functionality is available for Oracle 9 or later.

Set UNDO Tablespace dialog box lets you dynamically set an UNDO tablespace if the tablespace is running in AUTO UNDO mode.

The table below describes the options and functionality on the Set UNDO Tablespace dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Undo Tablespace</td>
<td>Display the possible UNDO tablespace choices. Select NULL to un-assign all UNDO tablespaces</td>
</tr>
</tbody>
</table>

**Completing the Set UNDO Tablespace Dialog Box**

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Tablespaces** node.
   
   DBArtisan displays the Tablespaces in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Set UNDO Tablespace**.

   OR

   In the right pane of the application, right-click the tablespace, and then select **Set UNDO Tablespace**.

   DBArtisan opens the Set UNDO Tablespace dialog box.

3. In **New Undo Tablespace** select from the possible UNDO tablespace choices. Select NULL to un-assign all UNDO tablespaces.

4. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute</td>
<td>Executes the task.</td>
</tr>
<tr>
<td>Analysis</td>
<td>Opens the Impact Analysis dialog box. This option is not available for every dialog box.</td>
</tr>
<tr>
<td>Preview</td>
<td>Opens the <strong>Preview dialog box</strong>.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td>Execute</td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>
For more information, see:

Set Tablespace Quota

Completing a Dialog Box

**Shrink**

**NOTE:** The Shrink functionality is available for Microsoft SQL Server and Oracle.

**Microsoft SQL Server**
The Shrink Database dialog box lets you reclaim space from a database that is too large.

The table below describes the options and functionality on the Shrink Database dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move data pages to beginning of file before shrink</td>
<td>Select to move data pages to the beginning of the file before the shrink.</td>
</tr>
<tr>
<td>Release All Unused Space</td>
<td>Deselect to set the target free space to retain, and then in the Target Free Space to Retain (percent) box, type the new value of free space to retain. The new size for the database must be at least as large as the Minimum Allowable Size displayed in the Current File Size box.</td>
</tr>
<tr>
<td>Target free space percent after shrink</td>
<td>Lets you specify the target free space percent after the shrink.</td>
</tr>
</tbody>
</table>

**Oracle: Shrink Rollback Segments**
The Shrink Rollback Segments dialog box lets you shrink the size of rollback segments. The proper sizing of rollback segments is critical to their overall performance. Performance degrades whenever a rollback segment must extend, wrap or shrink in response to transaction loads. Ideally, you want to make the extents of rollback segments as small as possible while still ensuring that each transaction can fit into a single extent.

After an abnormally large transaction load, you might consider shrinking a rollback segment to eliminate unnecessary space. Oracle lets you shrink a rollback segment manually by a specific amount or back to its Optimal Size.

**Important Notes**
For Oracle 9 or later, Shrink is not available if auto-UNDO management is enabled.

The table below describes the options and functionality on the Shrink Rollback Segments dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the size...</td>
<td>If you do not provide a specific number the Rollback Segment uses the OPTIMAL value specified in the Storage clause. If an OPTIMAL value is not specified, the size defaults to the MINEXTENTS value of the Storage clause.</td>
</tr>
</tbody>
</table>

**Oracle: Shrink Tables or Indexes**
The Shrink dialog box lets you shrink the size of tables or indexes.

**Important Notes**
Shrink is only available for tables in which the Row Movement property is set to TRUE.
The table below describes the options and functionality on the Shrink dialog box:

<table>
<thead>
<tr>
<th>Step</th>
<th>Functionality</th>
</tr>
</thead>
</table>
| Action Options    | Enable **Compact** to restrict shrinking to defragmenting the segment space and compact rows.  
|                   | Enable **Cascade** to simultaneously shrink all dependent objects.            |
| Dependencies      | View the dependencies on the table or index that you are shrinking.           |
| Preview           | Preview the DDL generated from your choices.                                  |

For more information, see [Completing a Dialog Box](#).

**Start Database**

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Start Database dialog box lets you start a database:

- When the database that has been stopped with a Stop Database dialog box.
- After a tablespace, partition, or index has been placed in group buffer pool RECOVER pending status (GRECP) or if pages have been put on the logical page list (LPL) for that object.

Depending on the specified options, the database can be made available for read-only processing, read-write processing, or utility-only processing. In a data sharing environment, the command can be issued from any DB2 on the group that has access to the database.

**Important Notes**

The table below describes the options and functionality on the Start Database dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Grid</td>
<td>Displays currently stopped databases. The Partition column is editable and can include one or more unique numeric values separated by commas. Initially the Partition column is blank. No validation is made for the correctness of partition numbers so make sure that the partitions exist.</td>
</tr>
<tr>
<td>Access</td>
<td>Lets you select access options.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

**Status**

**NOTE:** The Status functionality is available for Microsoft SQL Server, Oracle, and Sybase ASE.

The Set Constraint(s) Status dialog box lets you change the status of check constraints, foreign key constraints, primary key constraints, and unique key constraints.

For detailed instructions, see:

- [Setting Constraint Status for Microsoft SQL Server Objects](#)
- [Setting Constraint Status for Oracle](#)
Setting Constraint Status for Microsoft SQL Server Objects
The Set Constraint Status dialog box lets you specify the ability of a group of constraints to be replicated, and (for Microsoft SQL Server version 7 or later) enable or disable check constraints, foreign key constraints, primary key constraints, and unique key constraints.

The table below describes the options and functionality on the Set Constraint Status dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Deselect to temporarily override listed check constraints. Useful when you need to execute special processes that would ordinarily incur constraint-related errors.</td>
</tr>
<tr>
<td>Not for Replication</td>
<td>When you duplicate the table schema and data of a source database containing constraints marked “Not for Replication”, these objects are not carried over to the duplicate of the schema.</td>
</tr>
</tbody>
</table>

Completing the Set Constraint(s) Status Dialog Box
To complete this dialog box, do the following:

1. On the Datasource Explorer, select the target object node.

   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select Status.

   OR

   In the right pane of the application, right-click the target object, and then select Status.

   DBArtisan opens the Set Constraint(s) Status dialog box.

3. Use the table above to select dialog box options on each panel.

4. Use the Execute or Schedule button to complete the operation.

Setting Constraint Status for Oracle
The Set Constraint(s) Status dialog box lets you change the status of check constraints, foreign key constraints, primary key constraints, and unique key constraints. DBArtisan lets you enable or disable selected constraints and, in the case of primary key and unique key constraints, lets you enable with or without validation and disable with or without the changes cascading.

When enabled, the rule defined by the constraint is enforced on the data values in the columns on which the constraint is placed. When disabled, the constraint rule is not enforced but the constraint continues to be stored in the data dictionary.

Temporarily disabling constraints can improve performance when you are loading large amounts of data or when you are making massive changes to a table. Disabling constraints also can be useful if you are importing or exporting one table at a time.

NOTE: Primary keys for index-organized tables cannot be disabled.

NOTE: You cannot drop a unique or primary key constraint that is part of a referential integrity constraint without also dropping the foreign key. To drop the referenced key and the foreign key together, select the Cascade check box in the Set Constraint(s) Status dialog box.
The table below describes the options and functionality on the Set Constraint(s) Status dialog box.

**NOTE:** The options differ by object.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable</td>
<td>Enabling the constraint and not selecting the Validate check box automatically uses Oracle ENABLE NOVALIDATE clause which enables a constraint so that it does not validate the existing data. A table using constraints in enable novalidate mode can contain invalid data but you cannot add new invalid data to that table. The enable novalidate mode is useful as an intermediate state or when you do not want the constraint to check for possible exceptions (e.g., after a data warehouse load).</td>
</tr>
<tr>
<td>Validate</td>
<td>Enabling the constraint and selecting the Validate check box causes Oracle to validate all existing data in the key columns of the table with the constraint. If an exception exists, Oracle returns an error and the constraint remains disabled.</td>
</tr>
<tr>
<td>Cascade</td>
<td>Selecting the Cascade check box when disabling a primary key or foreign key constraint instructs Oracle to simultaneously disable any constraints that depend on the primary or unique key. Selecting the Delete Cascade check box instructs Oracle to delete data in the child table (on which the foreign key is defined) if the referenced data is the parent table is deleted.</td>
</tr>
</tbody>
</table>

**Completing the Set Constraint(s) Status Dialog Box**

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the target object node.
   - DBArtisan displays the target objects in the right pane of the Explorer window.
2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Status**.
   - OR
   - In the right pane of the application, right-click the target object, and then select **Status**.
   - DBArtisan opens the Set Constraint(s) Status dialog box.
3. Use the table above to select dialog box options.
4. Use the **Execute** or **Schedule** button to complete the operation.

**Stop Database**

**NOTE:** This functionality is available for IBM DB2 for OS/390 and z/OS only.

The Start Database dialog box lets you stop a database:

**Important Notes**

The table below describes the options and functionality on the Start Database dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Grid</td>
<td>Displays currently running databases. The Partition column is editable and can include one or more unique numeric values separated by commas. Initially the Partition column is blank. No validation is made for the correctness of partition numbers so make sure that the partitions exist.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).
Summary Definition

**NOTE:** This functionality is available for IBM DB2 for Linux, Unix, and Windows only.

The Summary Definition dialog box lets you enter the query for your table. If you are creating a table as a result of a query, you need to include the query in the AS FullSelect box.

**TIP:** You can open the query in the ISQL Editor and use the copy/paste function to add the query to the AS FullSelect box.

**Important Notes**

You must complete the Summary Definition dialog box to proceed with the Table Wizard.

The table below describes the options and functionality on the Summary Definition dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name</td>
<td>Read-only field that displays the table name.</td>
</tr>
<tr>
<td>AS FullSelect</td>
<td>Enter the query in box.</td>
</tr>
<tr>
<td>Summary table Options</td>
<td>Lets you set able options for the table you are creating.</td>
</tr>
<tr>
<td></td>
<td>Definition Only - Lets you set copy options for the table definition.</td>
</tr>
<tr>
<td></td>
<td>If you select this check box, you can set the copy options to include</td>
</tr>
<tr>
<td></td>
<td>Column Defaults and Identity Column Attr. The default selection is Column</td>
</tr>
<tr>
<td></td>
<td>Defaults.</td>
</tr>
<tr>
<td></td>
<td>Refreshable Options - Lets you set table refresh options for data and query.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

Switch Online

**NOTE:** The Switch Online functionality is available for IBM DB2 for Linux, Unix, and Windows only.

The Switch Online dialog box lets you access a tablespace by switching it online after the parent container(s) have been recovered or restored.

**Completing the Switch Online Dialog Box**

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the target object node.
   
   **DBArtisan** displays the target objects in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Switch Online**.
   
   **DBArtisan** opens the Switch Online dialog box.

3. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Preview" /></td>
<td>Opens the <strong>Preview dialog box</strong>.</td>
</tr>
<tr>
<td><img src="#" alt="Schedule" /></td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see <a href="#">Scheduling</a>.</td>
</tr>
</tbody>
</table>
Truncate

NOTE: The Truncate functionality is available for IBM DB2 for Linux, Unix, and Windows, Microsoft SQL Server, Oracle, IBM DB2 for OS/390 and z/OS and Sybase ASE.

The Truncate dialog box lets you quickly delete the rows of a table.

Important Notes
If you truncate a table, DBArtisan deletes all the rows. These rows are not logged as individual drops and cannot be recovered from a transaction log or other type of log.

Completing the Truncate Dialog Box
To complete this dialog box, do the following:

1. On the Datasource Explorer, select the target object node.
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the Datasource Explorer tool bar, click Command, and then select Truncate.
   OR
   In the right pane of the application, right-click the target object, and then select Truncate.
   DBArtisan opens the Truncate dialog box.

3. Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Preview]</td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td>![Schedule]</td>
<td>Opens Embarcadero Job Scheduler, or (if unavailable) another scheduling application. For more, see Scheduling.</td>
</tr>
<tr>
<td>![Execute]</td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

For more information, see:

- Completing a Dialog Box
- Truncating IBM DB2 for Linux, Unix, and Windows Objects
- Truncating Microsoft SQL Server Objects
- Truncating Oracle Objects
- Truncating IBM DB2 for OS/390 and z/OS Objects
- Truncating Sybase ASE Objects
Truncating IBM DB2 Objects
The Truncate dialog box lets you quickly delete the rows of a table.

**NOTE:** If you truncate a table, DBArtisan deletes all the rows. These rows are not logged as individual drops and cannot be recovered from a transaction log or other type of log.

For more information, see:

Truncate
Completing the Truncate Dialog Box

Truncating Microsoft SQL Server Objects
The Truncate dialog box lets you quickly delete the rows of a table.

**NOTE:** If you truncate a table, DBArtisan deletes all the rows. These rows are not logged as individual drops and cannot be recovered from a transaction log or other type of log.

For more information, see:

Truncate
Completing the Truncate Dialog Box

Truncating Oracle Objects
The Truncate dialog box lets you truncate tables and clusters. Truncating a table or cluster is a quick and efficient way to delete all of a table’s or a cluster’s existing rows. You can truncate any table or cluster in their schema or, if you have the DROP ANY TABLE system privilege, you can truncate any table in any schema.

When you truncate a table or cluster, you can specify whether space currently allocated for the table is returned to the containing tablespace or if it is returned to the system. The table below includes notes on the objects you can truncate:

<table>
<thead>
<tr>
<th>Object</th>
<th>Notes</th>
</tr>
</thead>
</table>
| Cluster | Before truncating a cluster containing a parent key, disable all referencing foreign keys existing in different tables.  
Truncating a cluster does not generate any rollback information and commits immediately.  
Oracle alters the storage parameter NEXT to the size of the last extent deleted from the segment. |
| Table | Before truncating a table containing a parent key, disable all referencing foreign keys existing in different tables.  
Truncating a table does not generate any rollback information and commits immediately.  
Oracle alters the storage parameter NEXT to the size of the last extent deleted from the segment.  
Oracle automatically deletes all data in the table's indexes and any materialized view direct-load INSERT information associated with a truncated table.  
If the table is not empty, all associated nonpartitioned indexes and all partitions of associated global partitioned indexes are marked unusable.  
You cannot truncate a hash cluster nor can you truncate individual tables in a hash cluster or an index cluster. |
The table below describes the options and functionality on the Truncate dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Handling</td>
<td>Drop Storage - Select if you want the freed extents returned to the system</td>
</tr>
<tr>
<td></td>
<td>where they can be used by other objects.</td>
</tr>
<tr>
<td></td>
<td>Reuse Storage - Select if you want the space to remain allocated to the</td>
</tr>
<tr>
<td></td>
<td>table or cluster you have just truncated.</td>
</tr>
</tbody>
</table>

For more information, see:

- Truncate
- Completing the Truncate Dialog Box

### Truncating IBM DB2 for OS/390 and z/OS Objects

The Truncate dialog box lets you quickly delete the rows of a table.

**CAUTION:** If you truncate a table, DBArtisan deletes all the rows. These rows are not logged as individual drops and cannot be recovered from a transaction log or other type of log.

Truncating a table is a faster alternative to deleting all of its rows. You cannot recover a truncated table because TRUNCATE TABLE is an unlogged command.

For more information, see:

- Truncate
- Completing the Truncate Dialog Box

### Truncating Sybase ASE Objects

The Truncate dialog box lets you truncate single or multiple tables. Truncating a table is a faster alternative to deleting all its rows. You cannot recover a truncated table because TRUNCATE TABLE is an unlogged command.

**NOTE:** You cannot truncate a table referenced by a foreign key constraint. Instead, use a DELETE statement without a WHERE clause.

**TIP:** When you truncate a table, Sybase ASE removes all rows from the target table, but retains the table structure (its indexes, columns, constraints, etc.). The counter used by an identity for new rows is reset to the seed for the column. To retain the identity counter, use a DELETE statement instead of TRUNCATE. To remove the target table definition and its data, use a DROP TABLE statement.

For more information, see:

- Truncate
- Completing the Truncate Dialog Box

### Unload Data Utility Wizard

The Unload Data Utility Wizard lets you export table data. After defining the data export operation, you have the opportunity to save the job definition for future reuse.
The table below describes the options and functionality on the panels of the Unload Data Utility Wizard:

<table>
<thead>
<tr>
<th>Panel</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel 1</td>
<td>Database</td>
<td>Lets you specify the database for the unload operation.</td>
</tr>
<tr>
<td></td>
<td>Tablespace</td>
<td>Lets you specify the tablespace for the unload operation.</td>
</tr>
<tr>
<td></td>
<td>Partition</td>
<td>Lets you specify the partition for the unload operation.</td>
</tr>
<tr>
<td></td>
<td>Specify the from copy data set name</td>
<td>Lets you specify the from copy data set.</td>
</tr>
<tr>
<td></td>
<td>Would you like to specify volume information?</td>
<td>Lets you specify volume information.</td>
</tr>
<tr>
<td>Panel 2</td>
<td>Select tables</td>
<td>Lets you specify the tables for the unload operation.</td>
</tr>
<tr>
<td>Panel 3</td>
<td>Specify the PUNCH DDN</td>
<td>Lets you specify the punch to unload the object list to.</td>
</tr>
<tr>
<td></td>
<td>Specify the UNLOAD DDN</td>
<td>Lets you specify the unload data set templates to unload the object list to.</td>
</tr>
<tr>
<td></td>
<td>Encoding type</td>
<td>Lets you specify the encoding scheme for the output data set and select the appropriate radio button. If you do not specify either EBCDIC, ASCII, UNICODE, or CCSID, the encoding scheme of the source data is preserved.</td>
</tr>
<tr>
<td></td>
<td>CCSID(s)</td>
<td>Lets you specify the up to three coded character set identifiers (CCSIDs) to be used for the data in the output data set, including data unloaded in the external character formats. The CCSID identifies the way in which the characters in the string are encoded.</td>
</tr>
<tr>
<td>Panel 4</td>
<td>Perform CCSID code substitution during unload</td>
<td>When a string is converted from one CCSID to another (including EBCDIC, ASCII, and UNICODE), a substitution character is sometimes placed in the output string, for example, when a character that exists in the source CCSID does not exist in the target CCSID.</td>
</tr>
<tr>
<td></td>
<td>Use padding with variable length data type</td>
<td>Select No to ensure that variable length columns in the unloaded records occupy the actual data length without additional padding. As a result, the unloaded or discarded records may have varying lengths.</td>
</tr>
<tr>
<td></td>
<td>Float</td>
<td>Lets you specify the output format for numeric floating point data. You may select either Hexadecimal (s390) or Binary (IEEE). This option applies to the binary output format only.</td>
</tr>
<tr>
<td></td>
<td>Max number or records of error allowed</td>
<td>The unloading process terminates when this value is reached.</td>
</tr>
<tr>
<td></td>
<td>Sharelevel</td>
<td>Lets you specify the type of application access allowed during the unloading process.</td>
</tr>
</tbody>
</table>

**Unquiesce Database**

**NOTE:** This functionality is available for IBM DB2 for Windows/Unix 8.1 only. DBArtisan does not support Unquiesce Database for an IBM DB2 for Windows/Unix 8 server from a IBM DB2 for Windows/Unix 7 client or for an IBM DB2 for Windows/Unix 7 server from an IBM DB2 for Windows/Unix 7 client.

The Unquiesce Database dialog box lets you restore user access to a database.
Completing the Unquiesce Database Dialog Box
To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Datasources** node.
   DBArtisan displays the Databases in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Unquiesce**.
   OR
   In the right pane of the application, right-click the target object, and then select **Unquiesce**.
   DBArtisan opens the Unquiesce Database dialog box.

3. Click the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).

**Unquiesce Instance**

**NOTE:** This functionality is available for IBM DB2 for Windows/Unix 8.1 only.

The Unquiesce Instance dialog box lets you restore user access to an instance.

Completing the Unquiesce Instance Dialog Box
To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the **Datasources** node.
   DBArtisan displays the Instances in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **Unquiesce**.
   OR
   In the right pane of the application, right-click the target object, and then select **Unquiesce**.
   DBArtisan opens the Unquiesce Instance dialog box.

3. Click the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

For more information, see [Completing a Dialog Box](#).
Update Statistics
The Update Statistics dialog box lets you update the statistics for an active table or index. As indexes grow and shrink in response to data modification, the accuracy of their statistics can deteriorate.

The following topics provide details on updating statistics by supported DBMS and object type:

- Updating statistics for tables or indexes (IBM DB2 for Linux, Unix, and Windows)
- Updating statistics for views (IBM DB2 for Windows, Unix, and Linux)
- Updating statistics for databases, indexes, or tables (Microsoft SQL Server)
- Updating statistics for indexes or tables (Sybase)

Updating statistics for tables or indexes (IBM DB2 for Linux, Unix, and Windows)

To update statistics for an index or table:
1. Connect to the datasource that contains the index or table. For more information, see Datasource Management.
2. On the Datasource Explorer, expand datasource nodes until the Tables or Indexes node is visible and then select the Indexes or Tables node.
   DBArtisan displays the objects of that type in the right pane of the Explorer window.
3. In the right pane of the Explorer window, select the specific, target table or index.
4. On the Datasource Explorer tool bar, click Command, and then select Update Statistics.
   DBArtisan opens the Update Statistics dialog box.
5. Use the following table as a guide to understanding and modifying the settings in this dialog.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Options</td>
<td>Update table statistics</td>
<td>Updates table statistics.</td>
</tr>
<tr>
<td></td>
<td>Do not collect column statistics</td>
<td>Column statistics provide information that the optimizer uses to choose the best access plans for queries.</td>
</tr>
<tr>
<td></td>
<td>Collect column statistics on key columns only</td>
<td>Collects column statistics on columns that make up all the indexes defined on the table. Column statistics provide information that the optimizer uses to choose the best access plans for queries.</td>
</tr>
<tr>
<td></td>
<td>Collect column statistics on all columns</td>
<td>Collects column statistics for all columns. Column statistics provide information that the optimizer uses to choose the best access plans for queries.</td>
</tr>
<tr>
<td></td>
<td>Do not collect distribution statistics</td>
<td>Does not collect basic statistics or distribution statistics on the columns. For efficiency both of RUNSTATS and subsequent query-plan analysis, you might collect distribution statistics on only the table columns that queries use in WHERE, GROUP BY, and similar clauses. You might also collect cardinality statistics on combined groups of columns. The optimizer uses such information to detect column correlation when it estimates selectivity for queries that reference the columns in the group.</td>
</tr>
</tbody>
</table>
6  Do one of the following:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Preview" /></td>
<td>Opens the Preview dialog box.</td>
</tr>
<tr>
<td><img src="image" alt="Execute" /></td>
<td>Executes the task.</td>
</tr>
</tbody>
</table>

**Updating statistics for views (IBM DB2 for Windows, Unix, and Linux)**

For IBM DB2 for Windows, Unix, and Linux (version 9), DBArtisan lets you update the statistics for a view.
To update statistics for a view:

1. On the Datasource Explorer, select a Views node.
   DBArtisan displays the views in the right pane of the Explorer window.

2. In the right pane, right-click a view and then select Update Statistics from the context menu.
   The Update Statistics dialog opens.

3. Use the following table as a guide to choosing options:

<table>
<thead>
<tr>
<th>Pane</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Options</td>
<td>Lets you enable or disable column statistics update, distribution statistics update as well as frequency and quantiles options, and enable or disable write access.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Lets you view referencing or referenced objects.</td>
</tr>
<tr>
<td>Preview</td>
<td>Lets you view the DDL generated for the operation.</td>
</tr>
</tbody>
</table>

4. Schedule or Execute the statistics update.

Updating statistics for databases, indexes, or tables (Microsoft SQL Server)

You can update statistics so that Microsoft SQL Server performs the most efficient query possible. This feature updates statistical information on your database so that the query processor can determine the optimal strategy for evaluating a query. These statistics record the key values used for distribution in an database.

You can use the Update Statistics dialog box if there is significant change in the key values in the database, if a large amount of data in an indexed column has been added, changed, or removed, or if a table has been truncated causing significant changes in the distribution of key values.

The Update Statistics dialog box lets you specify tables and indexes for validation. This dialog box offers different update options depending on your version of Microsoft SQL Server.

   TIP: Avoid updating statistics on your target tables during busy access periods. Microsoft SQL Server locks remote tables and indexes while reading data for update statistics.

For Microsoft SQL Server version 7 or later, the Update Statistics dialog box lets you specify a full or a percentage of a full scan to be used for updating table or index statistics. It also lets you enable or disable future automatic recomputations of statistics. These recomputations are made at Microsoft SQL Server’s discretion. When updating statistics for tables, this dialog box also lets you specify the type of statistics you require.

To update statistics for a database, index, or table:

1. Connect to the datasource that contains the database, index, or table. For more information, see Datasource Management.

2. On the Datasource Explorer, expand datasource nodes until the Databases, Tables, or Indexes node is visible and then select the Databases, Indexes, or Tables node.
   DBArtisan displays the objects of that type in the right pane of the Explorer window.

3. In the right pane of the Explorer window, select the specific, target database, table, or index.

4. On the Datasource Explorer tool bar, click Command, and then select Update Statistics.
   DBArtisan opens the Update Statistics dialog box.
Use the following table as a guide to understanding and modifying the settings in this dialog:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action Options</strong></td>
<td><strong>Scan Range</strong> <em>(tables and indexes only)</em>&lt;br&gt;Full Scan - Select when you want index statistics on every available row.&lt;br&gt;Sample Scan - Select when database size prohibits a full scan and you can afford to rely on statistics extrapolated from a sample of all available rows.</td>
</tr>
<tr>
<td></td>
<td><strong>Sample Count</strong> <em>(tables and indexes only)</em>&lt;br&gt;If you specified a Scan Range of Sample Scan, provide a count.</td>
</tr>
<tr>
<td></td>
<td><strong>Sample Unit</strong> <em>(tables and indexes only)</em>&lt;br&gt;If you specified a Scan Range of Sample Scan, specify either % or Rows.</td>
</tr>
<tr>
<td></td>
<td><strong>Statistics Type</strong> <em>(tables only)</em>&lt;br&gt;Index - Select if you only require statistics on the target tables’ indexed columns.&lt;br&gt;Columns - Select if you require statistics on the target tables in their entirety.&lt;br&gt;All existing statistics - Select if you require statistics on the whole database.</td>
</tr>
<tr>
<td></td>
<td><strong>Statistics Recompute</strong> <em>(tables and indexes only)</em>&lt;br&gt;Select if you want Microsoft SQL Server to recompute and update the statistics for the index as part of its normal internal maintenance.&lt;br&gt;Deselect if you want the scheduling of future recomputations to be solely your responsibility.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Lets you view referencing or referenced objects.</td>
</tr>
<tr>
<td>Preview</td>
<td>Lets you view the DDL generated for the operation.</td>
</tr>
</tbody>
</table>

Use the **Execute** or **Schedule** button to perform the operation.

**Updating statistics for indexes or tables (Sybase)**

The Update Statistics dialog box lets you specify tables and indexes for validation. This dialog box offers different update options depending on your version of Sybase ASE.

**TIP:** Avoid updating statistics on your target tables during busy access periods. Sybase ASE locks remote tables and indexes while reading data for update statistics.

To update statistics for a database, index, or table:

1. Connect to the datasource that contains the index or table. For more information, see Datasource Management.
2. On the **Datasource Explorer**, expand datasource nodes until the **Tables** or **Indexes** node is visible and then select the **Indexes** or **Tables** node.
3. DBArtisan displays the objects of that type in the right pane of the Explorer window.
4. In the right pane of the Explorer window, select the specific, target table or index.
5. On the **Datasource Explorer** tool bar, click **Command**, and then select **Update Statistics**.
6. DBArtisan opens the **Update Statistics** dialog box.
7. Use the following table as a guide to understanding and modifying the settings in this dialog:

<table>
<thead>
<tr>
<th>Step</th>
<th>Settings and tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action Options</strong></td>
<td><strong>Index</strong> <em>(tables only)</em>&lt;br&gt;Enabling this option updates statistics for indexes of the table.</td>
</tr>
</tbody>
</table>
Use the **Execute** or **Schedule** button to perform the operation.

### View Contents

**NOTE:** This functionality is available for Microsoft SQL Server only.

The Backup Device Information dialog box lets you view the contents of a Dump or Backup Device which stores backups of databases and transaction logs.

The table below describes the options and functionality on the Backup Device Information dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Device</td>
<td>Select the tape or disk used in a backup or restore operation.</td>
</tr>
</tbody>
</table>

### Completing the Backup Device Information Dialog Box

To complete this dialog box, do the following:

1. On the **Datasource Explorer**, select the target object node.
   
   DBArtisan displays the target objects in the right pane of the Explorer window.

2. On the **Datasource Explorer** tool bar, click **Command**, and then select **View Contents**.
   
   OR
   
   In the right pane of the application, right-click the target object, and then select **View Contents**.

   DBArtisan opens the Backup Device Information dialog box.

3. **Close** the dialog.
SQL Scripting

DBArtisan incorporates a powerful SQL scripting environment, the ISQL Editor. The ISQL Editor lets you write, debug, test and deploy solid SQL code for your database applications. The scripting environment lets you:

- Open multiple interactive script windows.
- Execute multiple scripts on the same desktop.
- Capture multiple result sets on the same desktop.

DBArtisan's scripting environment is comprised of different windows:

- ISQL Editor
- DDL Editor
- Results Editor

These windows are context sensitive to the type of script you are opening or extracting. For example, if you extract the schema for a table, DBArtisan opens a DDL Window containing the script. If you execute a script, a result window displays containing the results of your execution statement.

TIP: Since you must drop a database object before you can recreate it, you can set the DDL Editor to automatically include DROP statements for specified objects.

TIP: You can have multiple ISQL windows open at the same time, with each running separate queries.

NOTE: When you modify an ISQL window without performing a SAVE or RENAME operation, you will see an * in the tab, such as SQL 2*. The asterisk disappears as soon as you create a unique name for the script.

ISQL Editor

The ISQL Editor includes the ISQL Window and DDL Editor.

The ISQL Window lets you:

- Insert files and open files.
- Rename and configure query tabs.
- Find and replace with regular expressions.
- Mail your script files.

TIP: For Oracle, DBArtisan displays REF CURSOR contents in the ISQL Window and Results Tab.

TIP: To toggle to the next SQL window, press CTRL +T.

TIP: For IBM DB2 for Linux, Unix, and Windows and Oracle, you can access the Code Completion functionality with the CTRL+K shortcut.

Related Topics

- Toolbar Options
- Opening ISQL Windows
- Opening DDL Editors
• Opening Script Files
• Inserting Files into an ISQL Window
• Splitter Windows
• Find and Replace in an ISQL Window
• Regular Expressions
• Navigating in an ISQL Window
• Scheduling
• Sending SQL Scripts
• Renaming and Closing Query Window Tabs
• Printing a Script
• Saving and Closing Scripts

Toolbar Options
The table below describes the options of the ISQL Editor toolbar:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock</td>
<td>Lets you lock an ISQL Window to a specific datasource connection. When locked, the ISQL Window does not respond to datasource change events in other windows, and only uses the current connection for subsequent execution.</td>
</tr>
<tr>
<td>Format</td>
<td>Transforms spaghetti-style written SQL code into an easier read format.</td>
</tr>
<tr>
<td>Syntax Check</td>
<td>Checks any set of SQL to ensure it is valid before it is submitted to the database.</td>
</tr>
<tr>
<td>Analyze Code</td>
<td>Reports the number and type of statements contained within the SQL code.</td>
</tr>
<tr>
<td>Debug</td>
<td>Opens the Embarcadero SQL Debugger.</td>
</tr>
</tbody>
</table>

ISQL Windows
A script is a collection of SQL statements used to perform a task. Scripts, stored as files, form the foundation of most application projects. Definitions of tables, indexes, stored procedures, as well as reports and batch jobs, are usually stored in script files and shared by groups of developers using version control software. You can manage SQL scripts in SQL windows. DBArtisan lets you open multiple SQL windows in one or more workspaces.

Opening ISQL Windows
To open the ISQL Window, do the following:

1. On the File menu, click New ISQL.

   DBArtisan opens an SQL window in your current workspace.

For more information, see
ISQL Window Status Bar
ISQL Editor.
ISQL Window Status Bar

The ISQL window Status bar lets you view:

- Auto commit status (Oracle) - Automatically commits SQL statements as soon as the statements are run.
- Begin Transaction ON/OFF (SQL Server and Sybase)
- Keyboard Emulation Mode - Lets you customize your editing environment to an emulation mode.

**TIP:** For Microsoft SQL Server and Sybase, to set Begin Transaction status to “Yes”, on the ISQL Window toolbar, click the SQL Begin Tran button.

**TIP:** For Oracle, you can apply auto commit status changes to all open ISQL windows. You can modify emulation node on the Editor Tab of the Options Editor. You can modify the Oracle Auto Commit status and ISQL Tab of the Options Editor.

DDL Editors

**NOTE:** This functionality is available for all platforms.

DBArtisan lets you open a new DDL Editor when you want to create a script that is tied to an object type in your database. The DDL Editor opens containing a template script for the selected object type. Because the DDL Editor is directly tied to a database, database warning messages can be issued. For example, if you have a create table script which includes a DROP TABLE statement, the DDL Editor warns you about the existence of this statement and that you could lose existing table data.

The table below describes the options and functionality on the Create New Database Object dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Type</td>
<td>Lets you select the object type to which you want to attach the script.</td>
</tr>
<tr>
<td>Owner</td>
<td>Lets you type the name of the object owner for the object. The name of the owner connected to the current datasource is used as the default.</td>
</tr>
<tr>
<td>Object Name</td>
<td>Lets you type the name of the object type.</td>
</tr>
</tbody>
</table>

DBArtisan opens a DDL Editor containing a template for the object type you selected.

For more information, see [Opening DDL Editors](#).

Opening DDL Editors

To open DDL Editors, do the following.

1. On the **File** menu, click **New DDL Editor**.
2. In **Object Type** select the object type to which you want to attach the script.
3. In **Owner** type the name of the object owner for the object. The name of the owner connected to the current datasource is used as the default.
4. In **Object Name** type the name of the object type.
5. Click **OK**.

DBArtisan pastes the DDL into the ISQL Editor.
For more information, see ISQL Editor.

Open Files

NOTE: This functionality is available for all platforms.

The Open Files dialog box lets you open existing files.

The table below describes the options and functionality on the Open Files dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>Lets you type a script file name. You can open multiple script files simultaneously by selecting multiple file names in the file list.</td>
</tr>
<tr>
<td>Files of Type</td>
<td>Lets you control the files displayed in the file list by changing the selection. DBArtisan displays multiple default file extensions.</td>
</tr>
<tr>
<td>Open Button</td>
<td>Click to open one or more files into the current workspace.</td>
</tr>
</tbody>
</table>

Completing the Open Files Dialog Box

To complete the Open Files dialog box, do the following:

1. On the File menu, click Open.
2. Select a script file.

OR

3. In File Name, type a script file name.
4. In Files of type, select types of files to display.
5. Click Open to open one or more files into the current workspace.

DBArtisan pastes the script into the ISQL Editor.

For more information, see ISQL Editor.

What Type of File

NOTE: This functionality is available for all platforms.

The What Type of File Is dialog box lets you select options for unknown file types.

The table below describes the options and functionality on the What Type of File Is dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The file is a general purpose SQL script</td>
<td>Select if the unknown file is a SQL script.</td>
</tr>
<tr>
<td>The file contains the DDL to create a database object of the file contains Oracle Anonymous PL/SQL.</td>
<td>Lets you select object type, type the owner, and object name.</td>
</tr>
<tr>
<td>Always open unknown files into a SQL window without prompting.</td>
<td>Select to hide What Type of File Is Dialog Box for future unknown file types.</td>
</tr>
</tbody>
</table>
Completing the What Type of File Is Dialog Box
To complete the What Type of File Is dialog box, do the following:

1. On the File menu, click Open.
   DBArtisan opens the Open File(s) dialog box.
2. In the Open File(s) dialog box, select the target script, and then click Open.
   DBArtisan opens the What type of file dialog box.
3. Select options, and then click OK.
   DBArtisan opens the target script in an SQL Editor.

For more information, see ISQL Editor.

Insert File into Current File

NOTE: This functionality is available for all platforms.

The ISQL Editor facilitates the reuse of SQL scripts by letting you insert an existing file into another script.

The table below describes the options and functionality on the Insert File into Current File dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>Lets you type a script file name. You can open multiple script files simultaneously by selecting multiple file names in the file list.</td>
</tr>
<tr>
<td>Files of Type</td>
<td>Lets you control the files displayed in the file list by changing the selection. DBArtisan displays multiple default file extensions.</td>
</tr>
<tr>
<td>Open Button</td>
<td>Click to insert one or more files into the current workspace.</td>
</tr>
</tbody>
</table>

Completing the Insert File into Current File Dialog Box
To complete the Insert File into Current File dialog box, do the following:

1. On the Edit menu, click Edit Insert File.
   DBArtisan opens the Insert File into Current File dialog box.
2. Select a script file.
   OR
3. In File Name, type a script file name.
4. In Files of type, select types of files to display.
5. Click Open to insert one or more files into the current workspace.
   DBArtisan inserts the file.

For more information, see ISQL Editor.

Splitting Windows
You can split an SQL window into four different screens so that you can view different sections of a document simultaneously. You have the option to split the ISQL Window horizontally, vertically or into quadrants.
Splitting the ISQL Window Horizontally
To split the ISQL Window horizontally, do the following:

1. Point to the split box at the top of the vertical scroll bar on the right side of the SQL window.
2. When the pointer changes, drag it to the desired position.

Splitting the ISQL Window Vertically
To split the ISQL Window vertically, do the following:

1. Point to the split box at the top of the horizontal scroll bar on the bottom left side of the SQL window.
2. When the pointer changes, drag it to the desired position.

Removing Splits from an ISQL Window
Double-click the appropriate split bar to return the SQL window to its normal state.
For more information, see ISQL Editor.

Find
NOTE: This functionality is available for all platforms.

The Find dialog box lets you search text in your SQL scripts.
The table below describes the options and functionality on the Find dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find What</td>
<td>Lets you type your search string.</td>
</tr>
<tr>
<td>Match whole word only</td>
<td>Select to search for only the complete word.</td>
</tr>
<tr>
<td>Match Case</td>
<td>Select to make the search case-sensitive.</td>
</tr>
<tr>
<td>Regular expression</td>
<td>Select if you are looking for a regular text expression.</td>
</tr>
<tr>
<td>Wrap around search</td>
<td>Lets you search from the end of the script and back to the insertion point.</td>
</tr>
<tr>
<td>Direction</td>
<td>Lets you specify the direction you want to search. Click the Up or Down option button.</td>
</tr>
<tr>
<td>Find Next Button</td>
<td>Click to find the next occurrence of your search string.</td>
</tr>
<tr>
<td>Mark All Button</td>
<td>Click to place a small blue dot next to every line number in the script which meets the required search string criteria.</td>
</tr>
</tbody>
</table>

Completing the Find Dialog Box
To complete the Find dialog box, do the following

1. On the Edit menu, click Find.
2. In Find What, type your search string.
3. Select Match whole word only to search for only the complete word.
4. Select Match Case to make the search case-sensitive.
5. Select Regular expression to search for a regular text expression.
6. Select Wrap around search to search from the end of the script and back to the insertion point.
7 In **Direction**, click **Up** or **Down**.

8 Click **Find Next** to find the next occurrence of your search string.

9 Click **Mark All** to place a small blue dot next to every line number in the script which meets the required search string criteria.

   The ISQL Editor highlights the object name if it matches the search criteria.

For more information, see **Find**.

### Replace

**NOTE:** This functionality is available for all platforms.

The Replace dialog box lets you search and replace text in your SQL scripts.

The table below describes the options and functionality on the Replace dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find What</td>
<td>Lets you type your search string.</td>
</tr>
<tr>
<td>Replace With</td>
<td>Lets you type the replacement text.</td>
</tr>
<tr>
<td>Match Case</td>
<td>Select to make the search case-sensitive.</td>
</tr>
<tr>
<td>Regular expression</td>
<td>Select if you are looking for a <a href="#">regular text expression</a>.</td>
</tr>
<tr>
<td>Wrap around search</td>
<td>Lets you search from the end of the script and back to the insertion point.</td>
</tr>
<tr>
<td>Direction</td>
<td>Lets you specify the direction you want to search. Click the Up or Down option button.</td>
</tr>
<tr>
<td>Find Next Button</td>
<td>Click to find the next occurrence of your search string.</td>
</tr>
<tr>
<td>Replace Button</td>
<td>Click replace the current selection.</td>
</tr>
<tr>
<td>Replace All Button</td>
<td>Click to automatically find and replace all occurrences of your search string within the current window.</td>
</tr>
</tbody>
</table>

### Completing the Replace Dialog Box

To complete the Replace dialog box, do the following:

1. On the **Edit** menu, click **Replace**.
2. In **Find What**, type your search string.
3. In **Replace With**, type the replacement text.
4. Select **Match whole word only** to search for only the complete word.
5. Select **Match Case** to make the search case-sensitive.
6. Select **Regular expression** to search for a regular text expression.
7. Select **Wrap around search** to search from the end of the script and back to the insertion point.
8. In **Direction**, click **Up** or **Down**.
9. Click **Find Next** to find the next occurrence of your search string.
10. Click **Replace** to replace the current selection.
11 Click **Replace All** to automatically find and replace all occurrences of your search string within the current window.

For more information, see ISQL Editor.

### Regular Expressions

Regular Expressions are offered as an optional search criteria in the SQL windows search facility. Regular Expressions serve as powerful notation for describing string matching patterns. Special characters are used to denote certain match criteria on which the ISQL Editor should conduct its search. The table below describes the special characters and their meanings:

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
</table>
| ^         | Circumflex - Constrains the search to the start of a line | ^Rap -- Match lines beginning with Rap  
^Emb -- Match lines beginning with Emb |
| $         | Dollar - A dollar as the last character of the string constrains matches to the end of lines. | if$ -- Match lines ending with if  
^end$ -- Match lines consisting of the single word end |
| .         | Period - A period anywhere in the string matches any single character. | T..l -- Matches Tool, Till, Tail etc.  
H.w -- Matches Huw, How, Haw etc.  
^Sin.ers -- Matches lines beginning with Sinders, Sinners etc. |
| *         | Asterisk - An expression followed by an asterisk matches zero or more occurrences of that expression. | to* -- Matches t, to, too etc.  
00* -- matches 0, 00, 000, 0000 etc. |
| +         | Plus - An expression followed by a plus sign matches one or more occurrences of that expression. | to+ -- Matches to, too etc.  
10+ -- Matches 10, 100, 1000, 10000 etc.  
l/(cd+)/ -- Matches (0), (12464), (12) etc. |
| ?         | Question mark - An expression followed by a question mark optionally matches that expression. | for? -- Matches f and for  
10? -- Matches 1 and 10 |
| ()        | Brackets - Brackets can be used to group characters together prior to using a * + or?. | Rap(id)? -- Matches Rap and Rapid  
B(an)*a -- Matches Ba, Bana and Banana |
| []        | Square brackets - A string enclosed in square brackets matches any character in that string, but no others. If the first character of the string is a circumflex, the expression matches any character except the characters in the string. A range of characters can be specified by two characters separated by a -. These should be given in ASCII order (A-Z, a-z, 0-9 etc.). | ([0-9]) -- Matches (0), (4), (5) etc.  
/([0-9]+/) -- Matches (100), (342), (4), (23456) etc.  
H[uo]w -- Matches Huw and How  
Gre[^py] -- Matches Green, Great etc. but not Grep, Grey etc.  
z-a] -- Matches nothing  
^[A-Z] -- Match lines beginning with an upper-case letter |
| \        | Backslash - A backslash quotes any character. This allows a search for a character that is usually a regular expression specifier. | \$ -- Matches a dollar sign $  
\+ -- Matches a + |
Goto

NOTE: This functionality is available for all platforms.

The Goto dialog box lets you move to a specific line or column in your script.

The table below describes the options on the Goto dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Number</td>
<td>Lets you type or select the target line number.</td>
</tr>
</tbody>
</table>

Completing the Goto Dialog Box

To complete the Goto dialog box, do the following:

1. On the Edit menu, click Goto.
   DBArtisan opens the Go To dialog box.
2. In Line Number, type or select the target line number.
3. Click OK.
   DBArtisan moves the cursor to the target line.

For more information, see ISQL Editor.

Column Look-up

Column Look-up allows users to quickly reference and select table column names while coding or editing queries in the ISQL window.

Important Notes

You can set the options for column look-up in the Code Workbench.

Using Column Look-up

To use this feature, do the following:

1. In the ISQL Window, type the name of the table, view, or alias and then a period. You must construct a clause containing the table, view, or alias name before the column look-up feature will work. For example:
   
   ```sql
   SELECT employee.
   ```

   DBArtisan opens the list of available columns.
2. Select the target column name.
   DBArtisan adds the column name to the SQL statement.

Sending SQL Scripts

If you have MAPI-compliant E-mail software installed on your computer, then you can send SQL scripts to other users.
To send a SQL script, do the following:

1. On the **File** menu, click **Send**.
   
   DBArtisan opens your E-mail application.

2. In the **Address** box, type the name of the addressee(s) and any other options.
   
   **NOTE:** The ISQL Editor automatically attaches a file containing your SQL script to the e-mail message.

3. Click **Send**.
   
   DBArtisan sends the result set to the specified addressee(s).

For more information, see *ISQL Editor*.

---

### Renaming and Closing Query Window Tabs

SQL windows are tabbed windows that can be closed or renamed.

To rename a Query Tab, you need an open SQL window that includes an executed script. For information on how to execute scripts, see:

- [Executing Scripts](#)
- [Script Execution Facility](#)

#### Renaming a Query Window Tab

To rename a Query Window Tab, do the following:

1. Right-click the **Query Tab** on the **SQL window**, and then click **Rename**.
   
   DBArtisan opens the Rename Tab dialog box.

   **NOTE:** The Query Tab can be located on the top or bottom of the SQL window. You can set the location of the tab when configuring Datasource options.

2. In the **New Name** box, type the name of the new **Query** window.

3. Click **OK**.
   
   DBArtisan changes the name and closes the Rename Tab dialog box. The new name of the tab displays directly on the Query Window Tab at the top of the window.

#### Closing a Query Window Tab

To close a Query Window Tab, do the following:

1. At the top of the **ISQL window**, right-click the **Query Tab**, and then click **Close** or **Close All**.
   
   DBArtisan closes the Query.

For more information, see *ISQL Editor*.

---

### Print

**NOTE:** This functionality is available for all platforms.

The Print dialog box lets you specify a range of panels to print, or print the contents of a script window to a file.
The table below describes the options and functionality on the Print dialog box.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope</td>
<td>Printer</td>
<td>Displays default printer.</td>
</tr>
<tr>
<td></td>
<td>Print Range</td>
<td>Lets you select the appropriate print range.</td>
</tr>
<tr>
<td></td>
<td>Number of Copies</td>
<td>Lets you click the Up or Down arrow or type the number of copies you want.</td>
</tr>
<tr>
<td>Page Setup</td>
<td>Header</td>
<td>Lets you type header type to display at the top of the page.</td>
</tr>
<tr>
<td></td>
<td>Footer</td>
<td>Lets you type header type to display at the bottom of the page.</td>
</tr>
<tr>
<td></td>
<td>Header/Footer not within Margins</td>
<td>Select to position header and footer outside the margins.</td>
</tr>
<tr>
<td></td>
<td>Margins</td>
<td>Lets you specify margins in either inches or centimeters.</td>
</tr>
<tr>
<td>Options</td>
<td>Chromacoding</td>
<td>Lets you select Use Color if you have a color printer.</td>
</tr>
<tr>
<td></td>
<td>Line Numbering</td>
<td>Lets you specify the interval between numbered lines.</td>
</tr>
<tr>
<td>Other Options</td>
<td></td>
<td>Lets you select other options.</td>
</tr>
<tr>
<td>Documents</td>
<td>Document Box</td>
<td>Lets you select documents to print.</td>
</tr>
<tr>
<td></td>
<td>Clear</td>
<td>Click to clear list.</td>
</tr>
<tr>
<td></td>
<td>Invert</td>
<td>Click to switch printing order.</td>
</tr>
<tr>
<td>Configurations</td>
<td>New Configuration Name</td>
<td>Lets you type a new configuration which saves your current settings, and then click Create.</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td>Lets you delete an existing configuration.</td>
</tr>
<tr>
<td></td>
<td>Load</td>
<td>Lets you load an existing configuration.</td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>Lets you update an existing configuration.</td>
</tr>
</tbody>
</table>

The table below describes the buttons on the Print dialog box:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save Settings</td>
<td>Lets you save settings.</td>
</tr>
<tr>
<td>Font</td>
<td>Lets you open the Font dialog box and select fonts.</td>
</tr>
<tr>
<td>Setup</td>
<td>Lets you open the Print Setup dialog box and select printer options.</td>
</tr>
<tr>
<td>Preview</td>
<td>Lets you open the Print Preview dialog box that lets you preview the document before you print it.</td>
</tr>
<tr>
<td>Print</td>
<td>Lets you print the document.</td>
</tr>
</tbody>
</table>

Completing the Print Dialog Box

To complete the Print dialog box, do the following:

1. On the File menu, click Print.
2. On the tabs, select options.
3. Click Print.

For more information, see ISQL Editor.
Saving and Closing Scripts

Untitled scripts are named SQL1...SQLn by default, and all scripts are saved with the SQL extension. You can save your files with any valid name. If you have set the Auto-Save feature in the Options Editor, a temporary copy of your scripts is automatically saved periodically as you work.

For more information, see:
- Saving a Script
- Closing a Script
- Closing the Error List
- ISQL Editor

Saving a Script

To complete the Save dialog box, do the following:

1. On the **File** menu, click **Save**.
   
   DBArtisan opens the Save As dialog box.

2. If this is a new file, in the **File Name** box, type the name of the file.

3. If this is an existing file and you want to use save as, on the **Main** menu, click **Save As**, and in the **File Name** box, type the name of the file.

4. Click **Save**.
   
   DBArtisan closes the Save As dialog box.

For more information, see Saving and Closing Scripts.

Saving Modified Documents

The table below describes the options and functionality on the Save Modified Documents dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save All</td>
<td>Click to save documents in all open ISQL windows.</td>
</tr>
<tr>
<td>Save Selected</td>
<td>Click to save selected documents.</td>
</tr>
<tr>
<td>Invert</td>
<td>Click to clear selection.</td>
</tr>
<tr>
<td>Save None</td>
<td>Click to not save documents and close the application.</td>
</tr>
</tbody>
</table>

For more information, see Saving and Closing Scripts.

Closing a Script

To close a script, do the following:

1. On the **Main** menu, click **Close**.
   
   DBArtisan starts closing the script.

2. If you have not saved your script, DBArtisan prompts you to save the file. Click **Yes** to save and **No** to close without saving.

For more information, see Saving and Closing Scripts.
Closing the Error List
To close the error list, do the following:

1. On the Query menu, click Show Errors.

   DBArtisan closes the error list.

For more information, see ISQL Editor.

Editing Scripts
The ISQL Window incorporates a powerful editor that lets you edit your SQL scripts. The ISQL Window includes editing features such as:

- Uppercase or lowercase character conversion.
- Commenting and uncommenting blocks of text.
- Selecting text.
- Inserting PL/SQL tags.
- Setting and navigating with bookmarks.

The ISQL Editor provides Paste SQL Syntax and Paste SQL Statements utilities from which you can paste syntax for SQL commands and functions directly into an ISQL Window.

Editing a Script
To edit a script, do the following:

1. In the SQL window, type your changes or additions.

2. When you are finished with your changes, on the ISQL window tool bar, click Execute to compile the script.

For more information, see ISQL Editor.

Paste SQL Syntax

   NOTE: This functionality is available for all platforms.

The Paste SQL Syntax facility lets you paste SQL syntax without having to refer to documentation to find syntax for SQL commands. You can also paste the SQL directly into an ISQL window. The Paste SQL Syntax facility includes SQL syntax for:

- Commands
- Functions
- XML
- Other object syntax
The table below describes the options and functionality on the SQL Syntax dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Statements</td>
<td>Lets you select the target command, functions, or optimizer hint.</td>
</tr>
<tr>
<td>Syntax</td>
<td>Displays the syntax.</td>
</tr>
<tr>
<td>Paste Button</td>
<td>Click to paste the SQL statement into your ISQL Window.</td>
</tr>
</tbody>
</table>

**NOTE:** You must change the placeholders (e.g., expression) in the statements to reflect the specific objects in your database. For assistance with basic SQL statements, such as Select, Insert, Update, or Delete, use the Paste SQL Statement facility instead.

The table below describes options for each RDBMS platform:

<table>
<thead>
<tr>
<th>Oracle</th>
<th>Sybase ASE</th>
<th>Microsoft</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Commands</td>
<td>SQL Commands</td>
<td>SQL Commands</td>
</tr>
<tr>
<td>Number Functions</td>
<td>Aggregate Functions</td>
<td>Aggregate Functions</td>
</tr>
<tr>
<td>Character Functions</td>
<td>Datatype Conversion Functions</td>
<td>Datatype Conversion Functions</td>
</tr>
<tr>
<td>Date Functions</td>
<td>Date Functions</td>
<td>Date Functions</td>
</tr>
<tr>
<td>Conversion Functions</td>
<td>Mathematical Functions</td>
<td>Mathematical Functions</td>
</tr>
<tr>
<td>Group Functions</td>
<td>String Functions</td>
<td>String Functions</td>
</tr>
<tr>
<td>Other Functions (User, NVL, etc.)</td>
<td>System Functions</td>
<td>System Functions</td>
</tr>
<tr>
<td>Optimizer Hints</td>
<td>System Diagnostics</td>
<td>Text/Image Functions</td>
</tr>
</tbody>
</table>

**Completing the SQL Syntax Dialog Box**

To complete the SQL Syntax dialog box, do the following:

1. Place your insertion point in an open SQL window.
2. On the Edit menu, click Paste SQL Syntax.
3. In SQL Statements, select the target command, functions, or optimizer hint.
4. To paste the SQL statement into your ISQL Window, click Paste.

For more information, see Editing Scripts.

**Paste SQL Statements**

**NOTE:** The functionality is available for all platforms.

The Paste SQL Statement facility lets you create Select, Insert, Update, and Delete SQL statements. The Paste SQL Statement window displays all available database object names from which you can choose to construct SQL statements. These statements are dependent on the object type you choose.

The table below describes the options and functionality on the Paste SQL dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasource</td>
<td>Lets you select the target datasource.</td>
</tr>
</tbody>
</table>
Completing the Paste SQL Dialog Box

To complete the Paste dialog box, do the following

1. Place your insertion point in an open Editor window.
2. On the Edit menu, click Paste SQL.
3. In Datasource, select the target datasource.
4. In Database, select the target database.
5. In Owner, select the owner.
6. In Object Type, select the target object type.
7. Click Select if you want a SELECT SQL statement.
8. Click Update if you want an UPDATE SQL statement.
9. Click Insert if you want an INSERT SQL statement.
10. Click Delete if you want a DELETE SQL statement.
11. In the Object Type Box, select the check box next to the target object type, or click All to select all.
12. Click Paste to paste this object into your Editor window.
13. In the Middle Box, click the target object type properties or click All to select all.
14. Click **Paste** to paste the object properties into your Editor window, under the Object Type box.

15. In the **Right Box**, click **Paste** when you are satisfied with the entire SQL statement.

   DBArtisan pastes the SQL statement into your Editor window.

   For more information, see **Editing Scripts**.

**Selecting Text**
The ISQL Editor lets you select a single word, a single line, or a block of text.

**Selecting a Single Word**
To select a single word, do the following:

1. In the **ISQL Editor** window, position the pointer in the word and double-click.

**Selecting a Line of Text**
DBArtisan offers two ways to select a line of text:

1. In the **ISQL Editor** window, click the line number listed on the left side of the window.

   OR

2. Position the pointer at the beginning of the line of text, hold down the SHIFT key, and then click the end of the line of text.

   **NOTE:** For more keyboard commands, see **Keyboard Commands**.

**Selecting a Block of Text**
To select a block of text, do the following:

1. In the **ISQL Editor** window, drag until the block of text is selected.

   For more information, see **Editing Scripts**.

**Moving and Copying Text in an ISQL Window**
You can move or copy information anywhere in an Editor window or into a different Editor window by dragging it.

**Moving Text**
To move text, do the following:

1. In the **ISQL Editor** window, select the text you want to move.

2. Drag to the target location.

**Copying Text**
To copy text, do the following:

1. In the **ISQL Editor** window, select the text you want to copy.

2. On the **Edit** menu, click **Copy**.
3 Drag to the target location.

**NOTE:** If you are dragging between different Editor windows, arrange the Editor windows so that the source and destination windows are open and visible. You must be able to see both the original and target locations.

For more information, see *Editing Scripts*.

**Commenting and Uncommenting Scripts**
For most developers and development teams, documenting code is a fundamental part of the coding process. Besides contributing explanatory material for others who need to analyze or maintain your code later, the ability to comment and uncomment code can be very useful for bypassing statements in a procedure during compilation.

**Commenting Code**
To commenting out code, do the following:

1. On the **Edit** menu, click **Comment Out**.

   DBArtisan comments code.

**Uncommenting Code**
To uncomment code, do the following:

1. On the **Edit** menu, click **Undo Comment Out**.

   DBArtisan uncomments code.

For more information, see *Editing Scripts*.

**Changing Case**
When writing SQL scripts, you can change the letter casing of certain statements or lines of code. You can change case from lowercase to uppercase, or from uppercase to lowercase, using the case functions.

**Changing Case**
To change case, do the following:

1. Select one or more letters in your script.

2. On the **Edit** menu, click **Upper Case**.

   OR

   On the **Edit** menu, click **Lower Case**.

   DBArtisan changes the case.

For more information, see *Editing Scripts*.

**Cutting, Copying and Pasting Text and Columns in an ISQL Window**
The Editor window incorporates Cut, Copy and Paste text, and whole columns functions. You can move the selected text or columns to and from the Windows clipboard.

For more information, see:
Copying and Pasting Text

To copy and paste text, do the following:

1. In the Editor window, select the target text.
2. On the Edit menu, click Copy.
   DBArtisan copies the text.
3. On the Edit menu, click Paste.
   DBArtisan pastes the text.

For more information, see Cutting, Copying and Pasting Text and Columns in an ISQL Window.

Cutting and Pasting Text

To cut and paste text, do the following:

1. In the Editor window, select the target text.
2. On the Edit menu, click Cut.
   DBArtisan cuts the text.
3. On the Edit menu, click Paste.
   DBArtisan pastes the text.

For more information, see Cutting, Copying and Pasting Text and Columns in an ISQL Window.

Copying and Pasting a Column

To copy and paste a column, do the following:

1. In the Editor window, position the pointer in front of the column of the target text.
2. Press ALT and simultaneously drag the pointer over the target column.
3. On the Edit menu, click Copy.
   DBArtisan copies the column.
4. In the Editor window, position the pointer where you want to paste the column.
5. On the Edit menu, click Paste.
   DBArtisan pastes the column.

For more information, see Cutting, Copying and Pasting Text and Columns in an ISQL Window.
Cutting and Pasting a Column
To cut and paste a column, do the following:

1. In the Editor window, position the pointer in front of the column of the target text.
2. Press **ALT** and simultaneously drag the pointer over the target.
3. On the **Edit** menu, click **Cut**. DBArtisan cuts the column.
4. In the Editor window, position the pointer where you want to paste the column.
5. On the **Edit** menu, click **Paste**. DBArtisan pastes the column.

For more information, see [*Cutting, Copying and Pasting Text and Columns in an ISQL Window*](#).

Setting Bookmarks
Bookmarks are useful tools for navigating throughout an Editor window. You can jump back and forth between bookmarks easily, and there is no practical limit to the number of bookmarks you can set.

The table below describes the options for setting bookmarks:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookmarks:</td>
<td>Bookmarks are valuable navigation aids for jumping from one portion of a script to another. You can add bookmarks in important areas of your scripts, then jump back and forth between bookmarks.</td>
</tr>
</tbody>
</table>

**Setting a Bookmark**
To set a bookmark, do the following:

1. In the Editor window, position the pointer in front of the line you want to bookmark.
2. On the Editor window tool bar, click **Bookmark**. DBArtisan inserts a blue dot in the gutter next to the line you have book marked.

For more information, see:
- [*Moving Between Bookmarks*](#)
- [*Clearing Bookmarks*](#)

**Moving Between Bookmarks**
You use the Next Bookmark and the Previous Bookmark buttons to move back and forth between bookmarks.

**Goto the Next Bookmark**
To goto the next bookmark, do the following:

1. In the Editor window, position the pointer where you have set bookmarks, and then click **Bookmark**. DBArtisan jumps to the next bookmark.
Goto the Previous Bookmark
To goto the previous bookmark, do the following:

1. In the Editor window, position the pointer where you have set bookmarks, and then click Goto Bookmark. DBArtisan jumps to the previous bookmark.

For more information, see Setting Bookmarks.

Clearing Bookmarks
To clear bookmarks, do the following:

1. On the Editor tool bar, click Clear Bookmark. DBArtisan clears all bookmarks.

   **NOTE:** This does not clear any error indicators (red dots) that might be showing in the gutter.

For more information, see Setting Bookmarks.

Executing Scripts
The ISQL Editor lets you execute all or part of your SQL scripts. Unless you are executing large scripts that have multiple statements within them, or you need to view optimizer statistics, you execute most of your SQL scripts without options.

**NOTE:** For Oracle Client 8.0.5, if you execute a multi-line SQL statement with an error in the syntax, the cursor does not jump to the line of the error.

The ISQL Editor incorporates multiple features to refine and ease script execution. The table below describes these features:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Script Execution Facility</strong></td>
<td>The Script Execution Facility lets you execute scripts in parallel against multiple datasources. The facility also lets you schedule a job to perform the script execution at the appointed time, which saves development time and ensures accuracy and integrity across your databases.</td>
</tr>
<tr>
<td><strong>Step Execution Facility</strong></td>
<td>The Step Execution Facility processes batches from one delimiter to the next.</td>
</tr>
<tr>
<td><strong>Query Plan</strong></td>
<td>The Query Plan Facility provides a graphical display that lets you identify the execution path that your SQL follows. DBArtisan's Query Plan window now displays data for the estimated costs, number of rows, and bytes returned by each plan step.</td>
</tr>
<tr>
<td><strong>Query Options Dialog Box</strong></td>
<td>The Query Options dialog box lets you customize what you see when you execute your query.</td>
</tr>
</tbody>
</table>

Executing a Script
To execute a script, do the following:

1. On the Editor window tool bar, click **Execute**.

   OR

   In the Editor window, right-click, and then click **Execute**.

   DBArtisan starts executing the script.
Executing Partial Scripts
To execute a partial script, select the portion (see Setting ISQL Options) of the script to be executed and follow the steps outlined above. This feature is helpful while debugging large scripts and stored procedures.

Script Execution Facility
DBArtisan has a Script Execution Facility that lets you run parallel queries against multiple datasources. This facility is also a stand-alone utility that is available from the utilities tool bar. If used as a stand-alone utility, you can directly type your script or copy and paste a script in an additional tab. Once you have selected the desired options, DBArtisan establishes multiple threads and database connections to run the script simultaneously against the selected target datasources. Additionally, ANSI SQL scripts can run against multiple datasources from different DBMS vendors. For example, you can simultaneously execute an ordinary SELECT statement against Oracle, Sybase ASE, Microsoft SQL Server and IBM DB2 for Linux, Unix, and Windows servers.

The Script Execution Facility runs with two output modes:
- Graphical
- File

Graphical output mode opens multiple result windows, one for each target datasource.

File output mode sends all output to a specified file or files. This feature allows for the execution of scripts against a large number of target datasources in a single operation and enables scheduling capabilities.

Once the scripts have finished executing, you have the option to send a notification message indicating that the script execution has completed via e-mail or Net Send. This message includes attachments of all the output files generated.

For more information, see Executing Scripts Against Multiple Datasources.

Executing Scripts Against Multiple Datasources
To execute scripts against multiple datasources, do the following:

1. On the Utilities menu, click Script Execution Facility.

The table below describes the options and functionality on the Script Execution Facility dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Script</td>
<td>Lets you type, copy and paste, or cut and paste a script.</td>
</tr>
</tbody>
</table>
| Target   | Only Connected Datasources - Select to see only your currently connected datasources.  
|          | All DBMS Types - Select to see all DBMS types.                             |
|          | In the Datasource grid, select the check boxes next to the datasources against which you want to run your query, or click Select All to select all datasources.  
|          | Database - Lets you type the name of the target database.                  |

NOTE: This option is not available for a DDL Editor or PL/ISQL Editor. To execute a script from a DDL or a PL/ISQL Editor, use the stand-alone Script Execution Facility.
Compilation Error Tab
DBArtisan displays any errors encountered when a package, function or procedure was last compiled by a user in a separate tab of those object editors. The Errors Tab provides the line number, column position and error message for each compilation error.

For more information, see Executing Scripts.

Command History
The ISQL Editor is equipped with a command history box, which lets you keep a history of previously used SQL commands. The Command History is a list that is available in the Results window. Command history lets you go back and run previously used commands from the list rather than opening or inserting a script. You can set options on how you want to utilize this feature in the Editor Tab of the Options Editor.

Executing from Command History
To execute from Command History, do the following:

1. In the Results window, click the Command History list, and then click the command you want to execute.
   - The Query window is automatically populated with the selected command.

2. To execute the command, click Execute.

For more information, see Executing Scripts.

Canceling a Query
The ISQL Editor lets you cancel a query while the rows are still being returned.
Canceling a Query
To cancel a query, do the following:

1. On the Editor window tool bar, click Cancel.

   **NOTE:** This button is only enabled after a script has begun executing.

For more information, see Executing Scripts.

Step Execution Facility
Step execution of scripts is an invaluable method to debug your scripts. The Step Execution facility lets you step through each batch in your script. While some batches can include many lines of code, some batches can consist of one line. DBArtisan parses the statements and moves from batch to batch during the step execution process, each step controlled by you clicking the step execution button.

The table below describes how DBArtisan provides a number of useful tools for working with step execution of your scripts:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Window Gutter</td>
<td>The ISQL Window Gutter is a vertical gray display bar located to the left of the ISQL window. It includes location indicators such as line numbers, error indicators, and bookmarks. The gutter is a quick visual cue to your current position in the script.</td>
</tr>
<tr>
<td>Script Line Numbers</td>
<td>Line numbers are included to let you navigate large scripts. Error messages in the output window indicate the line number where the error occurred.</td>
</tr>
<tr>
<td>Automatic Error Flagging</td>
<td>When using the Step Execution mode, DBArtisan flags errors with a red dot in the ISQL window gutter. The errors are flagged sequentially as they are stepped into.</td>
</tr>
<tr>
<td>Point and Click Error Navigation</td>
<td>DBArtisan displays errors in the output window at the bottom of the screen and selects the errors as they occur. You can click each error and DBArtisan takes you directly to the line where that error occurred.</td>
</tr>
<tr>
<td>Step</td>
<td>Click the Step button to step into the next batch of code.</td>
</tr>
<tr>
<td>Step Back</td>
<td>Click the Step Back icon to step back to the most previous batch of code.</td>
</tr>
<tr>
<td>Step Over</td>
<td>Click the Step Over icon to jump over a batch to the next batch.imity.</td>
</tr>
<tr>
<td>Run to Cursor</td>
<td>Click the Run to Cursor icon to execute all of the code between the beginning of the script to wherever you have inserted the pointer.</td>
</tr>
<tr>
<td>Cancel Step Execution</td>
<td>Click the Cancel Step Execution icon to change to regular execution mode.</td>
</tr>
</tbody>
</table>

For more information, see Using the Step Execution Facility.

Using the Step Execution Facility
DBArtisan offers three ways to use the step execution facility:

1. Open a script.

2. On the Query menu, click Step-Execute.

   DBArtisan starts the procedure and displays errors at the bottom of the Editor window.

   **NOTE:** DBArtisan indicates each executing line with a yellow arrow in the gutter of the Editor window. The gutter is that gray area between the line numbers on the left and the code window. As you step through your batches, DBArtisan moves the arrow to indicate your current location.
3 To continue stepping through the script, on the Editor window tool bar, click Step, which displays in the Editor window after you have started the Step Execute procedure.

4 To step back, on the Editor window tool bar, click Step Back, which displays in the Editor window after you have started the Step Execute procedure.

5 To step over a batch and execute the next batch, on the Editor window tool bar, click Step Over, which displays in the Editor window after you have started the Step Execute procedure.

6 To stop Step Execution mode, on the Editor window tool bar, click Stop Step Execute, which displays in the Editor window after you have started the Step Execute procedure.

7 To Run to Cursor, on the Editor window tool bar, click Run to Cursor, which is available when the pointer is in the Editor window.

For more information, see Executing Scripts.

Using the Query Plan Facility
Each RDBMS platform provides a graphical display that lets you identify the execution path that your SQL follows. For Microsoft SQL Server, DBArtisan provides the Query Plan facility. The Query Plan button is a toggle. Set it to enable the Show Plan mode.

DBArtisan's Query Plan window displays data for the estimated costs, number of rows, and bytes returned by each plan step.

**NOTE:** For IBM DB2 for Linux, Unix, and Windows, DBArtisan includes a tree view of statements and associated costs.

Using the Query Plan Facility
To use the Query Plan Facility, do the following:

1 Open a script.

2 On the Query menu, click Query Plan.

DBArtisan starts the Show Plan mode.

3 To generate the Show Plan in a separate result window, click Execute.

For more information, see Using the Query Options Dialog Box.

Query Options
DBArtisan lets you customize your query environment, with options differing by platform.

**NOTE:** The options you select only apply to the current window.

To Open the Query Options Dialog Box

1 In the ISQL Editor window, right-click, and then click Query Options.

OR

When the ISQL Editor is open, select Query Options from the Query menu.

The Query Options dialog opens.
2. Use the following table, which provides an alphabetical listing of options, option descriptions, and the platform to which the option applies, as a guide to setting the options for your query environment:

<table>
<thead>
<tr>
<th>Option/Variable</th>
<th>Description</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abort On Overflow</td>
<td>If set to TRUE, queries will be aborted on encountering a value greater than the specified size.</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>Abort On Truncation</td>
<td>Specifies behavior following a loss of scale by an exact numeric datatype during an implicit datatype conversion. When set to TRUE, a statement that causes the error is aborted but other statements in the transaction or batch continue to be processed. When set to FALSE, DBArtisan truncates the query results and continues processing.</td>
<td>Sybase</td>
</tr>
<tr>
<td>ANSI NULL</td>
<td>When set to TRUE, controls results of logical operations with NULL values.</td>
<td>Sybase</td>
</tr>
<tr>
<td>BatchDelimiter</td>
<td>The batch separator must be a viewable character and not a space, new line, or tab. The defaults have been selected to ensure compatibility with the features of DBArtisan and the respective platform, but can be customized. Note: A custom delimiter works only from within an ISQL window and can’t be used for extraction operations. <strong>Oracle</strong>: “/” (forward slash) <strong>DB2</strong>: “;” (semicolon) <strong>Sybase</strong>: “go” <strong>SQL Server</strong>: “go” <strong>MySQL</strong>: “;” (semicolon)</td>
<td>DB2 (W/U/L), MySQL, Oracle, SQL Server, Sybase</td>
</tr>
<tr>
<td>Big Tables</td>
<td>When set to TRUE, allows big result sets by saving all temporary sets to file. This can slow queries.</td>
<td>MySQL</td>
</tr>
<tr>
<td>Chained</td>
<td>Invokes a begin transaction before the following statements: delete, insert, open, fetch, select, and update. You must still explicitly close the transaction with a commit.</td>
<td>Sybase</td>
</tr>
<tr>
<td>Check syntax when executing</td>
<td>TRUE/FALSE</td>
<td>DB2 (W/U/L), Oracle</td>
</tr>
<tr>
<td>Client Character</td>
<td>Default character set.</td>
<td>MySQL</td>
</tr>
<tr>
<td>Create Explain plan tables if required</td>
<td>If set to TRUE, Explain Plan tables are created, as necessary. If set to FALSE and you don’t manually create tables, Explain Plan operations will fail.</td>
<td>DB2 (W/U/L)</td>
</tr>
<tr>
<td>Create explain plan tables on the SYSTOOLS schema</td>
<td>If set to TRUE, Explain Plan tables are created on the SYSTOOLS schema. If the tables already exist in the user’s default schema, DBArtisan continues to use those tables. Refer to DB/2 documentation for a listing of Explain Plan tables that must be deleted in order to use the SYSTOOLS option. If set to FALSE, Explain Plan tables are created under the user’s default schema.</td>
<td>DB2 (W/U/L)</td>
</tr>
<tr>
<td>Execution Information</td>
<td>True/False</td>
<td>Oracle</td>
</tr>
<tr>
<td>Force Plan</td>
<td>When set to TRUE, processes a join in the same order as tables appear in the FROM clause of a SELECT statement only.</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>I/O Activity</td>
<td>True/False</td>
<td>Oracle</td>
</tr>
<tr>
<td>Ignore Overflow</td>
<td>When set to TRUE, DBArtisan will ignore any overflow caused by a resulting value that is larger than a column’s specified size.</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>Option/Variable</td>
<td>Description</td>
<td>Platform</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Index Activity</td>
<td>True/False</td>
<td>Oracle</td>
</tr>
<tr>
<td>Index Selection</td>
<td>Valuable when tuning query performance.</td>
<td>Sybase</td>
</tr>
<tr>
<td>Insert ID</td>
<td>Choose a value to be used a following INSERT or ALTER TABLE statement when you supply an AUTO_INCREMENT value.</td>
<td>MySQL</td>
</tr>
<tr>
<td>Interactive Timeout</td>
<td>28800 default</td>
<td>MySQL</td>
</tr>
<tr>
<td>Isolation Level</td>
<td>Lets you set DB/2 Isolation Levels of UNCOMMITED READ, RESET, CURSOR STABILITY, REPEATABLE READ, or READ STABILITY.</td>
<td>DB2 (W/U/L)</td>
</tr>
</tbody>
</table>
| Isolation Level     | **Read Committed**: Microsoft SQL Server default transaction isolation level. Specifies that shared locks are held while data is read to avoid dirty reads. You can change the data before the end of the transaction, resulting in nonrepeatable reads or phantom data.  
**Read Uncommitted**: The lowest level of transaction isolation. Transactions are isolated to ensure that physically corrupt data is not read. Applies dirty read, or isolation level 0 locking, which ensures that no shared locks are issued and no exclusive locks are honored.  
If set, it is possible to read uncommitted or dirty data values in the data can be changed and rows can appear or disappear in the data set before the end of the transaction.  
**Repeatable Read**: Places locks on all data used in a query, preventing other users from updating the data. Other users can insert new phantom rows into the data and are included in later reads in the current transaction.  
Concurrency is lower than Read Committed. Use this option only when necessary.  
**Serializable**: The highest level of transaction isolation. Transactions are completely isolated from one another. Places a range lock on the data set, preventing other users from updating or inserting rows into the data set until the transaction is complete.  
Concurrency is lower than Repeatable Read. Use this option only when necessary. | SQL Server         |
<p>| Isolation Level     | 0                                                                           | Sybase           |
|                     | 1: Sybase default isolation level. Prevents dirty reads.                    |                  |
|                     | 2: Prevents dirty and non-repeatable reads.                                 |                  |
|                     | 3: Prevents dirty and non-repeatable reads and phantoms. This level is equivalent to performing all selects with holdlock. |                  |
| Join Selection      | Valuable when tuning query performance.                                    | Sybase           |
| Last Insert ID      | Set the value to be stored in the binary log when you use LAST_INSERT_ID() in a statement that updates a table. | MySQL            |
| LONG Size Bytes     | 8,192 is the default                                                        | Oracle           |
| Max Errors Before Aborting | Select the maximum number of errors encountered before DBArtisan aborts a script. Setting this value to zero disables the feature. | DB2 (W/U/L), MySQL, Oracle, SQL Server, Sybase |
| Network Activity    | True/False                                                                  | Oracle           |</p>
<table>
<thead>
<tr>
<th>Option/Variable</th>
<th>Description</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Count</td>
<td>Terminates the message indicating the number of rows affected by a Transact-SQL statement from being returned as part of the results.</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>No Exec</td>
<td>When set to TRUE, compiles each query without executing it.</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>Output to Execution Window</td>
<td>TRUE/FALSE</td>
<td>Sybase</td>
</tr>
<tr>
<td>Output to Server Error Log</td>
<td>TRUE/FALSE</td>
<td>Sybase</td>
</tr>
<tr>
<td>Parse Activity</td>
<td>TRUE/FALSE</td>
<td>Oracle</td>
</tr>
<tr>
<td>Parse Only</td>
<td>When set to TRUE, checks the syntax of each Transact-SQL statement and returns any error messages without compiling or executing the statement. When TRUE, makes Microsoft SQL Server only parse the statement. When FALSE, makes Microsoft SQL Server compile and execute the statement. Do not use Parse Only in a stored procedure or a trigger.</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>Prefetch</td>
<td>When set to TRUE, enables large I/Os to the data cache. When set to FALSE, disables large I/Os to the data cache.</td>
<td>Sybase</td>
</tr>
<tr>
<td>Query Cache Type</td>
<td>The query is cached for ON or DEMAND.</td>
<td>MySQL</td>
</tr>
<tr>
<td>Row Count</td>
<td>When set to TRUE, DBArtisan terminates the query after returning the specified number of rows.</td>
<td>DB2 (W/U/L), MySQL, Oracle, SQL Server, Sybase</td>
</tr>
<tr>
<td>Run Script with batch execution</td>
<td>TRUE/FALSE</td>
<td>DB2 (W/U/L), Oracle, SQL Server, Sybase</td>
</tr>
<tr>
<td>SET ... Options</td>
<td>The <strong>Send Set Options</strong> setting dictates whether the remaining ANSI SQL Options in this category are sent to the server. The default for this option is set using the <strong>Enable Set Query Options</strong> setting on the ISQL tab of the Options editor. For details, see ISQL options.</td>
<td>SQL Server</td>
</tr>
<tr>
<td>SET Other Variables</td>
<td>Variables to be set at runtime.</td>
<td>MySQL</td>
</tr>
<tr>
<td>Set Quoted Identifier</td>
<td>TRUE/FALSE</td>
<td>Sybase</td>
</tr>
<tr>
<td>Show Plan</td>
<td>When set to TRUE, reports data retrieval methods chosen by the Microsoft SQL Server query optimizer.</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>Sort Activity</td>
<td>TRUE/FALSE</td>
<td>Oracle</td>
</tr>
<tr>
<td>SQL Auto IS NULL</td>
<td>When set to TRUE, enables you to find the last inserted row for a table.</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Big Selects</td>
<td>When set to TRUE, SELECT statements likely to take a very long time to execute will be aborted (i.e., where the number of rows examined exceeds the max join size)</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Big Tables</td>
<td>TRUE/FALSE</td>
<td>MySQL</td>
</tr>
</tbody>
</table>
### Result Options

The Result Options dialog box lets you set the SQL Results Window options. The table below describes Results options:

<table>
<thead>
<tr>
<th>Option/Variable</th>
<th>Description</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Buffer Result</td>
<td>When set to TRUE, forces results from SELECT statements into temporary tables.</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Log Bin</td>
<td>When set to TRUE, allows logging to the binary log.</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Log Off</td>
<td>When set to TRUE, no logging is done to the general query log.</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Log Update</td>
<td>When set to TRUE, allows logging to the binary log.</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Low Priority Updates</td>
<td>When set to TRUE, gives table modifying operations lower priority than SELECT operations</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Max Join Size</td>
<td>When set to TRUE, you can catch SELECT statements where keys are not used properly and that would probably take a long time. Set it if your users tend to perform joins that lack a WHERE clause, that take a long time, or that return millions of rows.</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Quote Show Create</td>
<td>When set to TRUE, table and column names will be quoted.</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Safe Updates</td>
<td>When set to TRUE, the query aborts UPDATE or DELETE statements that do not use a key in the WHERE clause or a LIMIT clause. This makes it possible to catch UPDATE or DELETE statements where keys are not used properly and that would probably change or delete a large number of rows</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Select</td>
<td>The maximum number of records that should be returned from SELECT statements.</td>
<td>MySQL</td>
</tr>
<tr>
<td>SQL Warnings</td>
<td>Defines whether or not single row insert statements generate an information string in the event of a warning.</td>
<td>MySQL</td>
</tr>
<tr>
<td>Statistics I/O</td>
<td>Lets you display information regarding the amount of disk activity generated by Transact-SQL statements.</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>Statistics Subquery Cache</td>
<td>Displays the number of cache hits, misses, and the number of rows in the subquery cache for each subquery.</td>
<td>Sybase</td>
</tr>
<tr>
<td>Statistics Time</td>
<td>Displays the number of milliseconds required to parse, compile, and execute each statement.</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>Table Activity</td>
<td>True/False</td>
<td>Oracle</td>
</tr>
<tr>
<td>Table Count</td>
<td>Sets the number of tables that Sybase ASE considers at one time while optimizing a join.</td>
<td>Sybase</td>
</tr>
<tr>
<td>Text Size</td>
<td>8,192 is the default</td>
<td>SQL Server, Sybase</td>
</tr>
<tr>
<td>Transaction Isolation</td>
<td>Repeatable Read is the default. Read Committed, Read Uncommitted, and Serializable are the other options. Refer to MS SQL Query Options dialog box for an explanation.</td>
<td>MySQL</td>
</tr>
<tr>
<td>Unique Checks</td>
<td>Performs uniqueness checks for secondary indexes of MyISAM tables.</td>
<td>MySQL</td>
</tr>
</tbody>
</table>

3. When finished, click **OK**.

### Result Options

The Result Options dialog box lets you set the SQL Results Window options. The table below describes Results options:

**NOTE:** The options you select only apply to the current window.
Using the Token Matching Capability
When you are working with a large script with multiple levels of embedded steps, compare left and right parentheses, curly braces, square brackets and BEGIN/END pairs to make sure that you have delimited your code properly.

The Token Matching capability of DBArtisan helps you achieve clean code.

Finding Matching Tokens
To find matching tokens, do the following:

1. Position the insertion pointer just to the left of the first token in a token pair you want to locate.
2. Click Match Token.
   
   DBArtisan jumps the pointer to the next available token.

For more information, see Query Plan Facility.
Results Editor
The results of your SQL queries are displayed in the Results Tab of each Editor Window, which captures result sets in a virtual data grid that accommodates large result sets. The data grid offers many of the features of a basic Windows spreadsheet, giving you a great deal of flexibility in editing the worksheet and formatting its contents.

TIP: For Oracle, DBArtisan displays REF CURSOR contents in the ISQL Window and Results Tab.

You have many choices for navigating and viewing your SQL query results. The Results window includes functionality to set result window options, find and replace, export data to other products such as Microsoft Excel, and mail your results files.

For more information, see Configuring Result Set Windows.

Results window toolbar options
The table below describes the options of the Results window toolbar:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock</td>
<td>Lets you lock an ISQL Window to a specific datasource connection. When locked, the ISQL Window does not respond to datasource change events in other windows, and only uses the current connection for subsequent execution.</td>
</tr>
<tr>
<td>Options</td>
<td>Open the Options editor, letting you specify results window preferences. For more information, see Results (ISQL) options.</td>
</tr>
<tr>
<td>ReExecute</td>
<td>Lets you execute the script again without switching to the Query tab.</td>
</tr>
<tr>
<td>Close</td>
<td>Lets you close the current results window.</td>
</tr>
</tbody>
</table>

Configuring Result Set Windows
Result set windows can be configured in a variety of ways. You can configure your result set windows to present result sets in multiple or single panels, attached or detached from the corresponding ISQL window. These options can be set globally to save you the hassle of setting options for each result window. Additionally, Result windows can be torn off and dropped onto an open area of the workspace to create separate windows. These are known as Tear Off Tab Windows.

You can set the Result Window to display result sets in any of the following modes:

- Single result sets in one attached tab window.
- Multiple result sets in one attached tab window.
- Single result sets in one separate unattached tab windows.
- Multiple result sets in one separate unattached tab windows.

To configure result set windows:
1. On the File menu, click Options.
2. In the Options Editor, click the list, and then click Results or click the Results tab.
3. Refer to Results (ISQL) options for details on how to set result windows options.
Configuring results window grid properties
You can alter the physical appearance in a result set window. Effects include items such as 3-D buttons, highlighted headings, and enabled or disabled row and column gridlines.

To configure the appearance of result set grids:
1. On the File menu, click Options.
2. In the Options Editor, click the list, and then click Grid Properties or click the Grid Properties tab.
3. Refer to Grid properties (Results window) options for details on how to set result windows options.

Exporting Data to Other Products
You can export data from a result set to traditional spreadsheet products, such as Microsoft Excel. You can copy part or all of a result set and paste it into your traditional spreadsheet product by way of the Microsoft Windows Clipboard function. You can also save your result sets as:
- Tab delimited files,
- Comma separated files
- Pipe delimited ("|") files
- HTML
- Excel 2000 or higher
- XML
- User-specified delimited files

For more information, see Results Editor.

Setting Result Windows to Read Only Mode
To set your result windows to read only mode to keep anyone from accidentally editing or altering a result set, do the following:
1. Select a Result window that you want to make read only.
2. On the Edit menu, click Read Only.
   DBArtisan places a check-mark next to read only and sets the command.

   NOTE: The Read Only command is a toggle. When it is set, the formatting buttons on the Edit menu are not available.

For more information, see Results Editor.

Result Window Status Bar
The Result Window Status Bar displays information about each Result window at the bottom of each window. You can display or hide the status bar by setting the Status Bar preference. This preference applies to all result windows.

For more information, see Main Menu.
Mailing Result Sets
If you have MAPI-compliant electronic mail software installed on your computer, then you can mail result sets to other users.

Mailing a Result Set
To mail a result set, do the following:

1. Open the Message dialog box:
2. On the File menu, click Send.

DBArtisan opens the open Message dialog box.
3. In the Address box, type the name of the addressee(s) and any other options.

The ISQL Editor automatically attaches a file containing your result set to the mail message.
4. Click Send.

DBArtisan sends the result set to the specified addressee(s).

For more information, see Results Editor.

Closing Result Window Tabs
DBArtisan lets you close tabbed Result set windows.

Closing a Result Window Tab
To close a Result Window Tab, do the following:

1. On the Result Window Tab tool bar, click Close.

OR

Right-click the Results Tab, and then click Close.

DBArtisan closes the Result Window Tab.

For more information, see Results Editor.

Saving and Closing Result Sets
You can save your result sets using the standard Save and Save As functions. You can save multiple copies of the same result set and modify each copy to specific formatting requirements.

Saving Results
To save results, do the following:

1. On the File menu, click Save.

DBArtisan opens the Save Results dialog box.
2. In the File name box, type the name of the result set.
3. In Save as type, select the file type.

TIP: You can save data in Excel 2000 or later .xls, tab-delimited, comma-delimited, pipe-delimited, user-specified delimited .txt, HTML, and XML formats.
4  To include column titles, select Include column titles when saving.
5  If you chose User Specified Delimiter, in User Specified Delimiter type the delimiter.
6  Click Save.

   DBArtisan saves the file and closes the Save As dialog box.

Closing a Result Set
To close a result set, do the following:
1  On the Main menu, click Close.

   DBArtisan closes the Result Set.
2  If you have not saved your result set, DBArtisan prompts you to save the file. Click Yes to save and No to close without saving.

For more information, see Results Editor.

Editing Result Sets
The Results Editor provides many ways to edit and customize your result windows. The Data Grid offers a host of features for manipulating, sorting and formatting data.

Topics
- Cutting, Copying, and Pasting Cell Contents
- Cutting, Copying, and Pasting Rows
- Cutting, Copying, and Pasting Columns
- Adding and Inserting Rows
- Adding and Inserting Columns
- Deleting Rows and Column
- Resizing Rows and Columns
- Sorting Data

Cutting, Copying, and Pasting Cell Contents
The Result window supports standard cut, copy and paste functionality.

Cutting Cell Contents
To cut cell contents, do the following:
1  In the Results window, double click or tab to the target cell. A double line bounds the selected cell. You can also select text using standard text selection techniques.
2  On the Edit menu, click Cut.

   DBArtisan cuts the cell.
Copying Cell Contents
To copy cell contents, do the following:

1. In the Results window, double click or tab to the target cell. A double line bounds the selected cell. You can also select text using standard text selection techniques.

2. On the Edit menu, click Copy.

DBArtisan copies the cell.

Pasting Cell Contents
To paste cell contents, do the following:

1. In the Results window, double click or tab to the target cell. A double line bounds the selected cell.

2. On the Edit menu, click Paste.

DBArtisan pastes the cell.

For more information, see Editing Result Sets.

Cutting, Copying, and Pasting Rows
You can perform standard cut, copy, and paste functions on rows, just as you can on individual cells.

Cutting Rows
To cut a row, do the following:

1. In the Results window, click the numbered row heading on the left side of the row.

2. On the Edit menu, click Cut.

DBArtisan cuts the row.

Copying Rows
To copy rows, do the following:

1. In the Results window, double click or tab to the target row. A double line bounds the selected row. You can also select text using standard text selection techniques.

2. On the Edit menu, click Copy.

DBArtisan copies the row.

Pasting Rows
To paste rows, do the following:

1. In the Results window, double click or tab to the target row. A double line bounds the selected row.

2. On the Edit menu, click Paste.

DBArtisan pastes the row.

For more information, see Editing Result Sets.

Cutting, Copying, and Pasting Columns
You can perform standard cut, copy, and paste functions on columns, just as you can on rows.
Cutting Columns
To cut columns, do the following:
1. In the Results window, click the column heading above the first row.
2. On the Edit menu, click Cut.
   DBArtisan cuts the column.

Copying Columns
To copy columns, do the following:
1. In the Results window, click the column heading.
2. On the Edit menu, click Copy.
   DBArtisan copies the column.

Pasting Columns
To paste columns, do the following:
1. In the Results window, click the column heading above the first row to select the target column.
2. On the Edit menu, click Paste.
   DBArtisan pastes the column.

For more information, see Editing Result Sets.

Adding and Inserting Rows
You can add or insert rows to expand or rearrange your result sets.

Adding a Row
To add a row, do the following:
1. To add a row as the last row of the result set, position the pointer inside the result set.
2. On the Edit menu, click Add Row.
3. To add a row inside the result set, click the numbered row heading where you want to add a row.
4. On the Edit menu, click Add Row.

Inserting a Row
To insert a row, do the following:
1. To insert a row as the last row of the result set, position the pointer inside the result set.
2. On the Edit menu, click Insert Row.
3. To insert a row inside the result set, click the numbered row heading where you want to insert a row.
4. On the Edit menu, click Insert Row.

For more information, see Editing Result Sets.
Adding and Inserting Columns
You can add or insert columns to expand or rearrange your result sets.

Adding a Column
To add a column, do the following:
1. Position the pointer inside the result set.
2. Select Edit, Add Column from the main menu. The new column is added as the last column of the result set.

Inserting a Column
To insert a column, do the following:
1. Select the column where you want to insert a column.
2. Select Edit, Insert Column from the main menu. The new column is inserted to the left of the column that you selected.

For more information, see Editing Result Sets.

Deleting Rows and Columns
You can delete entire rows and columns to edit your result sets.

Deleting a Row
To delete a row, do the following:
1. Select the target row of data to delete.
2. On the Edit menu, click Delete Row.
   DBArtisan deletes the row.

Deleting a Column
To delete a column, do the following:
1. Select the target column of data to delete.
2. On the Edit menu, click Delete Column.
   DBArtisan deletes the column.

For more information, see Editing Result Sets.

Resizing Rows and Columns
Resizing rows and columns can aid readability of the spreadsheet contents or condense space for editing and formatting purposes.

Resizing Rows to Their Default Height
To resize rows to their default height, do the following:
1. Select one or more rows by clicking on the numbered row headings to the left of the rows.
2. Right-click the selected rows, and then click Resize Rows.
Resizing Rows to a New Height
To resize rows to a new height, do the following:

1. Select one or more rows by clicking the numbered row headings to the left of the rows.
2. Change the pointer to a solid horizontal bar with arrows on top and bottom by moving it to one of the upper or lower borders of the row heading.
3. Click and grab the row border and drag the pointer to enlarge or shrink the height of the row.

Resizing Columns to Their Default Widths
To resize columns to their default widths, do the following:

1. Select one or more columns by clicking the column headings.
2. Right-click the selected columns, and then click Resize Columns.

Resizing Columns to a New Width
To resize columns to a new width, do the following:

1. Select one or more columns by clicking the column headings.
2. Change the pointer to a solid horizontal bar with arrows on top and bottom by moving it to one of the upper or lower borders of the column heading.
3. Click and grab the column border and drag the pointer to enlarge or shrink the height of the column.

For more information, see Editing Result Sets.

Sorting Data
To order and organize data in a coherent manner, you can sort columns alphanumerically in single result sets.

Sorting Data
To sort data, do the following:

1. Double click the column header to sort the data in the column in ascending, alphanumerical order. To sort the column in descending order, double click the column header again.

   NOTE: This option is not valid for multiple result sets.

For more information, see Editing Result Sets.

Formatting Result Sets
The ISQL Editor provides flexibility in formatting result sets, for analysis and reporting, from within a Result Window. Result sets can be formatted to best suit your purpose, whether it be sending via e-mail, printing, or exporting a file to other programs, such as Microsoft Excel. Some of these features change only the screen display of the results, while others allow you to format for printing.

Topics
- Changing the displayed view
- Setting Border Properties
- Setting Fonts
**Setting Colors**

For information on general appearance of results, see Configuring results window grid properties.

**Changing the displayed view**

You can view results in a standard grid, in HTML format, or as flat ASCII. By default, when a result window is generated, the results are displayed according to the view option currently selected on the Results (ISQL) options. You can change the display in the results window.

To change the result view on the fly:

1. Right-click in the results window.
2. On the shortcut menu, choose one of the View as Grid, View as HTML, or View as ASCII options.

**Format Border Styles**

You can use the shortcut menu to alter border properties.

**Completing the Format Styles Dialog Box**

To complete the Format Styles dialog box, do the following:

1. Right-click the Result data grid, and then click Border.
2. On the Border box, you can indicate whether or not a border should appear on the top, bottom, right, left, or for a range of cells by clicking the corresponding boxes.
3. To set a range of cells apart by setting a particular border style around that range, select the range in the result set before opening the Format Styles dialog box. To select the Range property, click the range box.
4. In the Type box, you can select the type of line you want to border the cell or cells by clicking the corresponding boxes.
5. To select a color, click the Color list, and then click the border color.
6. Click OK.

DBArtisan saves your changes and closes the Format Styles dialog box.

For more information, see Formatting Result Sets.

**Format Font Styles**

You can use the shortcut menu to alter font properties. Selecting the Font command displays the Format Styles tabbed dialog box.

**Completing the Format Styles Dialog Box**

To complete the Format Styles dialog box, do the following:

1. Right-click the Result data grid, and then click Font.

DBArtisan opens the Format Styles dialog box.

2. In the Font box, type or click the font you want to use.
3. In the Style box, type or click the font style you want to use.
4. In the Size box, type or click the size you want to use.
5 To make a line cross through the length of the text, in the Effects box, select the Strikeout check box.

6 To underline the text, in the Effects box, select the Underline check box.

7 To change the script style, click the Script list, and then click the new script style.

The Sample box displays the sample text of your selections.

For more information, see Formatting Result Sets.

Format Color Styles
You can use the shortcut menu to alter color properties. Selecting the Color command displays the Cell Properties dialog box.

Completing the Cell Properties Dialog Box
To complete the Format Styles dialog box, do the following:

1 Right-click the Result data grid, and then click Color.

   DBArtisan opens the Cell Properties dialog box.

2 Change the Text Color and Background Color options.

For more information, see Formatting Result Sets.
MySQL Datasources

As easy as MySQL is to use, DBArtisan makes it even easier. The DBArtisan interface saves you having to write commands for many operations. For example, as you register MySQL datasources, DBArtisan makes it simple to perform multiple flush tasks simultaneously, set up or edit tables, and generate a variety of indexes.

What follows are descriptions of the tools you need to create and manage MySQL datasources. This section includes detailed instructions for creating and editing:

• Flush Options,
• Foreign Keys,
• Functions,
• Indexes/Primary Keys/Unique Keys,
• Users/Privileges,
• MySQL Databases,
• and Tables.

Additionally, from the Utilities menu you can extract and migrate schema, configure your servers, load or unload data, and backup or restore databases:

• Using the schema extraction wizard
• Using the schema migration wizard
• Using the Schema Publication wizard
• Server Configuration
• Database Backups
• Database Restore/Import

Flush Options

In addition to the basic datasource-level menu choices, MySQL gives you the option to flush some internal caches. The Flush options you select do not carry over from one datasource to another, and in fact must be reset every time you want to flush the caches.

A FLUSH command empties internal caches used by MySQL. Only a user with RELOAD privileges can execute FLUSH operations.

To Open the Flush Options Dialog Box:
1. On the Explorer Tree, right-click the datasource you want to flush, and then select Flush from the menu.

The MySQL Flush options are as follows:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES_KEY_FILE</td>
<td>Reloads the key files for the DES_ENCRYPT and DES_DECRYPT functions</td>
</tr>
<tr>
<td>HOSTS</td>
<td>Empties the host cache tables. It's wise to flush the host tables if any hosts changed IP numbers or you get a &quot;host is blocked&quot; error message.</td>
</tr>
</tbody>
</table>
If you want to carry out a flush operation not included among the DBArtisan options, you can write your own flush commands using SQL Scripting.

### Foreign Keys

MySQL recognizes two types of tables, MyISAM and InnoDB. MyISAM tables access data records using an index, where InnoDB tables allow transactions and foreign keys. Therefore, the discussion of foreign keys is limited to your InnoDB tables.

After you create a foreign key using the Create Foreign Key Wizard, further actions are available to you from the toolbar or by right-clicking an individual foreign key.

- **Open** - Launches Foreign Key editor for selected index(s).
- **Drop** - Drops selected Foreign Key (s).
- **Extract** - Reverse engineers selected Foreign Key (s) into DDL and loads into working DDL editor.
- **Migrate**
- **Copy Object Name(s)**
- **Find Object**
- **Select All**
- **UnSelect All**

### Creating or Editing a Foreign Key

Embarcadero DBArtisan’s Create Foreign Key Wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data. By using the Create Foreign Key Wizard you obviate the need for remembering the code underlying the creation process.
To Open the Create Foreign Key Wizard
1  On the Explorer, find the database where you want to add the new foreign key.
2  On the Foreign Keys branch, right-click and select New.
   OR
   Click the Foreign Keys branch and then select New from toolbar.

To Open the Foreign Key Editor
1  On the Explorer, find the database where you want to edit the foreign key.
2  Click the Foreign Keys branch and select the specific foreign key in the right pane.
3  Right-click the Foreign Key, and select Open.
   OR
   Choose Open from the toolbar.
   For the changes you make to take effect, click Alter on the toolbar.

   **NOTE:** Once the Foreign Key Editor is open, you can navigate to any foreign key in the same database.

The tables below displays all the fields you’ll find creating or editing a foreign key:

<table>
<thead>
<tr>
<th>Panel/Tab</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreign Key</td>
<td><em>Constraint Name:</em> System</td>
<td>The constraint name must be unique. You can either rely on MySQL to assign a unique name or you can specify one.</td>
</tr>
<tr>
<td>Properties</td>
<td>Generated or User Specified</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Constraint State:</em> On Delete</td>
<td>The default constraint state is NO ACTION. This ensures that referenced values cannot be updated or deleted if to do so would violate referential integrity.</td>
</tr>
<tr>
<td></td>
<td>and On Update</td>
<td>CASCADE permits a referenced row in a child table to be deleted/updated if it is deleted/updated in the parent table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A row in the child table is SET NULL when rows in the parent table are deleted/updated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RESTRICT prevents a command from executing if changes to referential data prompts an error.</td>
</tr>
<tr>
<td>Column Mapping</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Because MySQL requires matching indexes on the child and parent tables involved in a foreign key relationship, the foreign key columns you select must be found within an already created index in the main table. They must also be selected in the same order.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Main Table (child table)</td>
<td>Choose the table to which you want to add the foreign key from the drop-down list. Then select the relevant column(s).</td>
</tr>
<tr>
<td></td>
<td>Referenced Table (parent table)</td>
<td>Select the database and table from the drop down lists that you want this foreign key to refer to. You must also select the appropriate columns.</td>
</tr>
<tr>
<td></td>
<td>Enforce constraints</td>
<td>When checked, it ensures that the constraints you have selected will be enacted when values change in the parent table.</td>
</tr>
<tr>
<td></td>
<td>New Index</td>
<td>The column(s) on the child table must be a part of the same index as the column(s) on the parent table. If the index(es) need to be created, you can open the Create Index Wizard, create an index, and be returned to the Foreign Key Wizard.</td>
</tr>
<tr>
<td>DDL Vew</td>
<td>You can see the SQL generated by the selections you made. Click <strong>Execute</strong> to create the new foreign key. Note that this is a read-only panel.</td>
<td></td>
</tr>
</tbody>
</table>
Functions

As for the other platforms, users can create subroutines that can be reused so you can control, access, and manipulate the data that underlies an object. DBArtisan will construct the CREATE FUNCTION statement based on the information you supply when you generate a new function. Of course you are also able to edit any function after it has been created. It is important to keep in mind that to create a user-defined function, functions must be written in C or C++, your operating system must support dynamic loading, and you must have compiled mysqld dynamically (not statically).

- New - Launches UDF creation wizard.
- Open - Launches UDF editor for selected index(s).
- Drop - Drops selected UDF(s).
- Report - TBD.
- Extract - Reverse engineers selected UDF(s) into DDL and loads into working DDL editor.
- Migrate - Disabled.

Creating or Editing a Function

Embarcadero DBArtisan’s Create Function Wizard lets you write a function you without knowing the underlying commands. The wizard gives you the opportunity to bring a function that’s stored in an external library into MySQL.

To Open the Create Function Wizard
1. On the Explorer, find the MySQL database on the server where you want to add the new function.
2. On the Functions branch of the MySQL database, right-click and select New.

OR

Click the Functions branch of the MySQL database and then select New from the toolbar.

NOTE: No functions are installed by default.

To Open the Functions Editor
1. On the Explorer, find the MySQL database where you want to edit the function.
2. Click the Functions branch and select the specific function in the right pane.
3. Right-click the function, and select Open.

OR

Choose Open from the toolbar.

For the changes you make to take effect, click Alter on the toolbar.

The table that follows lists all the fields you’ll find as you create or edit a function:
Indexes/Primary Keys/Unique Keys

Indexes for tables operate in much the same way as an index for a book works. Indexes help you find what you are looking for without altering the data itself. Specialized indexes include primary keys and unique keys.

A Primary Key uniquely identifies a row in a table and is used to relate a table to foreign keys in other tables.

A Unique Key ensures no duplicate values are entered in table columns where you place a unique constraint.

A sound indexing strategy is critical to overall system performance. One pitfall to avoid is placing many indexes on a table without regard for their cumulative cost. Indexes can improve read but can slow write performance because more information in the system catalog has to be updated. Consequently, extra indexes can actually slow overall performance if data modification occurs frequently on the table.

DBArtisan separates system indexes from user-defined indexes in the Datasource Explorer to ensure that system indexes are not accidentally altered or dropped.

- New - Launches Index creation wizard.
- Open - Launches Index editor for selected index(s).
- Drop - Drops selected index(s).
- Extract - Reverse engineers selected table(s) into DDL and loads into working DDL editor.
- Migrate
- Copy Object Name(s)
- Find Object
- Select All
- UnSelect All

**Function Properties**

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Name:</td>
<td>In 64 characters or less, you can name your function.</td>
</tr>
<tr>
<td>Return Value:</td>
<td>STRING, REAL (also known as Double), INTEGER</td>
</tr>
<tr>
<td>Shared Object Library:</td>
<td>Identify the file that holds the library where your functions are stored.</td>
</tr>
<tr>
<td>Aggregate</td>
<td>Check the box if the function you are creating will collapse a large amount of data into a single output, or aggregate the data.</td>
</tr>
</tbody>
</table>

**DDL View**

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDL View</td>
<td>You can see the SQL generated by the selections you made. Click <strong>Execute</strong> to create the new foreign key. Note that this is a read-only panel.</td>
</tr>
</tbody>
</table>
Creating or Editing an Index/Primary Key/Unique Key

Whether you are creating an index, a primary key, or a unique key, the same creation wizard opens with defaults in place appropriate to what you want to generate.

To Open the Create Index Wizard:

1. On the Explorer, find the database where you want to add the new index/primary key/unique key.

2. On the Indexes/Primary Key/Unique Key branch, right-click and select New.

OR

Click the Indexes/Primary Key/Unique Key branch and then select New from toolbar.

The Create Index Wizard can also be opened from within the Create Foreign Key Wizard.

To Open the Index/Primary Key/Unique Key Editor

1. On the Explorer, find the MySQL database where you want to edit the index/primary key/unique key.

2. Click the Index/Primary Key/Unique Key branch and select the specific index in the right pane.

3. Right-click the Index/Primary Key/Unique Key, and select Open.

OR

Choose Open from the toolbar.

For the changes you make to take effect, click Alter on the toolbar.

The table that follows describes all the fields you’ll encounter as you create or edit an Index/Primary Key/Unique Key:
Tables and Columns

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name</td>
<td>Note that the field is populated with a drop-down list of all the tables in the database. The tables appear in alphabetical order, and the field is filled with the first table from the list. Choose the table you want to index.</td>
</tr>
<tr>
<td>Index Name</td>
<td>The name you type can be up to 64 alphanumeric characters.</td>
</tr>
</tbody>
</table>
| Constraint Type      | **Primary**: Each record of a table is identified as uniquely and relates to a foreign key in another table.  
**Unique**: Ensures that there is no duplication between values in columns where you place this constraint.  
**Full Text**: Enables the search for several words in arbitrary order in a table.  
**Spatial**: Allows you to find records that are defined by location, or geometry types. |
| Index Storage Type   | **Hash**: Used for equality comparisons. Only whole keys can be used to search for a row.  
**BTree**: Tree data structure that keeps data sorted. BTrees grow from the bottom up as elements are inserted.  
**RTree**: Tree data structure used for spatial indexes and access to multidimensional information. |
| Specify Columns in Index | Columns are listed by name and datatype and whether or not they are nullable. As you check selections, the sort order is identified. |

DDL View

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDL View</td>
<td>You can see the SQL generated by the selections you made. Click Execute to create the new foreign key. Note that this is a read-only panel.</td>
</tr>
</tbody>
</table>

Users/Privileges

Adding users to a MySQL database is made simple when you use DBArtisan. Unlike adding new users to other platforms, MySQL doesn’t rely solely on a userid/password combination. The MySQL server must not only validate the user and password, but also the host from which the connection is being requested before the new user is admitted to the server’s inner sanctum. The Create User Wizard and the User Editor enable these validations.

After you create a foreign key using the Create User Wizard, further actions are available to you from the toolbar or by right-clicking an individual user or from the toolbar:

- Open
- Alter
- New -- Launches the Create New User Wizard
- Drop
- Extract
- Migrate
- Report - N/A
- Command
• Create Like
• Change Password
• Copy Object Name(s)
• Refresh
• Close

Creating or Editing a User
The Create User Wizard lets you give a new user the privileges he or she needs to function on the MySQL database of choice. Here is where you enter not only the User ID and password, but also validate the host from which that user will be connecting to the server. Users can be limited to access from a single host or be given access from multiple hosts.

To Open the Create User Wizard
1. On the Explorer, find the database where you want to add the new user and expand the Security node.
2. On the Users branch, right-click and select New.
   OR
   Click the Users branch and then select New from the toolbar.

To Drop a User
1. On the Explorer, find the database where you want to drop the user.
2. Click the Users branch and select the specific user in the right pane.
3. Right-click the user, and select Drop.
4. Confirm the action.

To Open the User Editor
1. On the Explorer, find the database where you want to edit the user.
2. Click the Users branch and select the specific user in the right pane.
3. Right-click the user, and select Open.
   OR
   Choose Open from the toolbar.
   For the changes you make to take effect, click Alter on the toolbar.
   NOTE: Once the User editor is open, you can navigate to any user in the same datasource.

The table below displays all the fields you’ll find when you create or edit a user:
### User Information

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Name</strong></td>
<td>The name of the new user. At this time, the name cannot exceed 16 characters, although the field accepts 34. DBArtisan can’t enforce the discrepancy, so we urge you to show a little restraint. This is the name that is displayed in the User branch of the Explorer tree. If you leave this field blank and complete the process, you create an &lt;anonymous&gt; user. Only one anonymous user per datasource is allowed.</td>
</tr>
<tr>
<td><strong>Full Name</strong></td>
<td><strong>OPTIONAL.</strong> At this time, the name cannot exceed 16 characters, although the field accepts 34. If you entered a nickname or an alias in the user name field, you can display the user’s true identity here.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td><strong>OPTIONAL.</strong> Write a brief description of the user if you want.</td>
</tr>
<tr>
<td><strong>Email</strong></td>
<td><strong>OPTIONAL.</strong> The new user’s email address.</td>
</tr>
<tr>
<td><strong>Contact Information</strong></td>
<td><strong>OPTIONAL.</strong> The new user’s title, address, phone number or whatever contact information you want to include.</td>
</tr>
<tr>
<td><strong>User Icon</strong></td>
<td><strong>OPTIONAL.</strong> If you want to identify your new user by color or pattern, you can use the icon editor to make a distinguishing mark. You can assign groups of users the same icon or assign each user his or her own icon. This tool helps you distinguish users at a glance. <strong>Note:</strong> If you don’t want to use the head that’s preloaded, simply click the red X and start designing your own with the other tools.</td>
</tr>
</tbody>
</table>

### User Hosts

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New Host</strong></td>
<td>Click the dotted square to open a text box where you can enter the ip address of the computer on which the MySQL server is running (e.g., 10.10.12.114). It’s possible to enter more than one address to give the user access to more than one server, but you need to enter them one at a time. % is the symbol you can use to mask user hosts.</td>
</tr>
<tr>
<td><strong>Password/Confirm</strong></td>
<td>Enter the password and confirmation the user needs to use to gain access to the MySQL datasource. If you want to apply a password to every datasource user, click Apply To All.</td>
</tr>
</tbody>
</table>

### System Privileges

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Each host you identified for the given user is listed here. You can set the same or different privileges for each host for a given user, but you must set them host by host.</td>
</tr>
<tr>
<td><strong>Privileges</strong></td>
<td>System privileges are Reload, Shutdown, Process, File, Show DBs, Execute, Replication Client, Replication Slave, Super (User). To grant system privileges, click the green check mark and the Grant Privileges Dialog Box opens. To revoke system privileges, click the red X and the Revoke Privileges dialog box opens.</td>
</tr>
</tbody>
</table>

### Object Privileges

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database/Table Name</strong></td>
<td>Each host you identified for the given user is listed here. You can set the same or different privileges for each host for a given user, but you must set them host by host.</td>
</tr>
</tbody>
</table>
Each database is a collection of tables. The goal of a database is to manage large amounts of data in a multi-user environment as well as protect data against unauthorized access.

The Create Database Wizard lets you add a new MySQL database using the Embarcadero DBArtisan interface so you don’t have to write all the code yourself in a scripting window or from a command prompt.

After you have created a new database, the following additional tasks are available:

- **New** – Launches the aforementioned Database creation wizard.
- **Alter** – Enabled when a property of the current database has been modified.
- **Drop**
- **Extract** – Reverse engineers DDL for current database and loads into working DDL editor.
- **Report** – TBD.
- **Command Menu**
• Refresh – standard repopulation feature.
• Close – Close editor with save validation.

Creating or Editing MySQL Databases
Databases are the shell into which you will pour the tables that contain the data records you want to manage. The Create Database Wizard makes it easy for you to add databases to your registered MySQL servers.

To open the Create Database Wizard
1. On the Explorer, find the Datasource where you want to add the new database and expand it.
2. On the Databases node, right-click and select New.
   OR
   Click the Databases node and then select New from toolbar.

To Open the Database Editor
1. On the Explorer, find the Datasource where you want to edit the database.
2. Click the Databases branch and select the specific database.
3. Choose Open from the toolbar.
   For the changes you make to take effect, click Alter on the toolbar.
   NOTE: Once the Database editor is open, you can navigate to any database in the same datasource.

The table below displays all the fields you’ll find as you create or edit a database:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name:</td>
<td>The database name you choose can be up to 64 characters long. Feel free to use all alphabetic characters or mix in some special or numeric characters. But a name cannot be exclusively special or numeric characters.</td>
</tr>
<tr>
<td>Default Character Set: (optional)</td>
<td>This is the character set for a language or alphabet. If you do not make a selection from the drop-down list (click the arrow to see the choices), the database will use the server's default character set.</td>
</tr>
<tr>
<td>Collation:</td>
<td>The collation encodes the rules governing character used for a language (like Greek) or an alphabet. The database will use the server's default collation unless you specify otherwise. To see the collation choices available to you, click the arrow and make a selection.</td>
</tr>
</tbody>
</table>

NOTE: It's possible to create databases with different character sets and collations on the same MySQL server.
Database Privileges

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
</table>
| Scope (in the creation process, database is the only option available) | **Database**: Select user privileges for the specific database you are creating or editing. Privileges are available for all datasource users.  
**Global**: Displays system-wide privileges for datasource users. To change global privileges you need to edit the user (see [Creating or Editing a User](http://dev.mysql.com/doc/mysql/en/storage-engines.html)).  
**All**: Displays both system-wide and database privileges. |
| Grantee | A list of all server users and their hosts is displayed. You can pick and choose to whom you want to grant database privileges and whether each user will have user only or user with grant privileges. |
| Privileges | Privileges include Alter, Create, Create Temporary Tables, Delete, Drop, Index, Insert, Lock Tables, References, Select, Update, with Grant.  
To grant privileges, click the green check mark and the [Grant/Revoke Privileges Dialog Box](http://dev.mysql.com/doc/mysql/en/storage-engines.html) opens where you can grant privileges.  
To revoke object privileges, click the red X and the [Grant/Revoke Privileges Dialog Box](http://dev.mysql.com/doc/mysql/en/storage-engines.html) opens where you can revoke privileges. |

DDL View

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDL View</td>
<td>The DDL window displays the command code you created by completing the prior panels of the wizard. You have the ability to save the code to disk, to print it, to email it, to open an ISQL window where you can tinker with the code, or to schedule the execution of the code.</td>
</tr>
<tr>
<td>Launch object editor after execute</td>
<td>When this option is checked, the script will execute immediately.</td>
</tr>
</tbody>
</table>

Tables

To state the obvious, tables are the building blocks of any database and are the basic unit of data storage. A table’s properties are determined by its columns and rows, or data records. MySQL servers can store tables in multiple formats, including MyISAM and InnoDB. MyISAM tables (ISAM is the acronym for indexed sequential access method) are used most often for read operations. The read operation is very fast, but you cannot include any referential integrity, such as a foreign key. Also, MyISAM tables only issue table-level locks. InnoDB tables, on the other hand, do permit transactions and foreign key constraints. InnoDB tables also lock data at the row level, which is appropriate for high transaction tables. Additional table types available are MERGE, MEMORY, FEDERATED, and ARCHIVE among others. For a complete discussion of table types, go to the MySQL documentation of table and engine types: [http://dev.mysql.com/doc/mysql/en/storage-engines.html](http://dev.mysql.com/doc/mysql/en/storage-engines.html)

- New - Launches table creation wizard.
- Open - Launches Table Editor for selected table(s).
- Drop - Drops selected table(s).
- Extract - Reverse engineers selected table(s) into DDL and loads into working DDL editor.
• Migrate
• Rename
• Analyze tables
• Create like
• Check Table
• Rebuild table
• Repair table
• Optimize table
• Disable Keys
• Enable keys
• Truncate table
• Flush table
• Convert Table
• Schema
• Select * From
• Copy Object Name
• Find Object
• Select All
• Unselect All

Creating or Editing Tables
DBArtisan lets you create many different table types using the Create Table Wizard. After you have successfully added a table to the database, you can reopen and edit it using the Table Editor.

To Open the Create Table Wizard
1  On the Explorer, find the database where you want to add the new table and expand the Databases node.

2  On the Tables branch, right-click and select New.

   OR
   Click the Tables branch and then select New from toolbar.

To Open the Table Editor
1  On the Explorer, find the database where you want to edit the table.

2  Click the Tables branch and select the specific table in the right pane.
3 Right-click the **table**, and select **Open**.

OR

Choose **Open** from the toolbar.

For the changes you make to take effect, click **Alter** on the toolbar.

**NOTE:** Once the Table editor is open, you can navigate to any table in the same datasource.

In the following you’ll find all the fields you’ll encounter as you create or edit a table:
### Properties

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name</td>
<td>Enter a meaningful and unique name that is no more than 64 characters. <strong>NOTE:</strong> You must name the table to enable the Create Table Wizard.</td>
</tr>
<tr>
<td>Storage Type</td>
<td>From the drop-down list, select how you want the data to be stored. Storage types are described in MySQL documentation. <a href="http://dev.mysql.com/doc/mysql/en/storage-engines.html">http://dev.mysql.com/doc/mysql/en/storage-engines.html</a></td>
</tr>
<tr>
<td>Row Format</td>
<td><strong>DEFAULT</strong> -- returns the default value if there is one \n<strong>FIXED</strong> -- each row is stored with a fixed number of bytes. \n<strong>DYNAMIC</strong> -- data records have variable length \n<strong>COMPRESSED</strong> -- each record is compressed separately.</td>
</tr>
<tr>
<td>Create via SELECT</td>
<td>When you create a table using a SELECT command, the individual columns take their data types from the SELECT command, and don’t have to be declared explicitly. Attributes, however, are not carried over.</td>
</tr>
<tr>
<td>Unique Key Violations</td>
<td>When you try to insert or update a row that causes a violation, you can choose to <strong>IGNORE</strong> the violation and continue processing the next row. Or, you can choose <strong>REPLACE</strong>, which is equivalent to an insert statement if a violation occurs. The row will be replaced with the new data.</td>
</tr>
<tr>
<td>Default Character Set</td>
<td>This is the character set for a language or alphabet.</td>
</tr>
<tr>
<td>Default Collation</td>
<td>The collation encodes the rules governing character used for a language (like Greek) or an alphabet.</td>
</tr>
<tr>
<td>Auto-increment</td>
<td>You can specify the initial value used for the auto-increment sequence. This is possible only for MyISAM tables and for InnoDB tables built in MySQL versions 5.0.3 or greater. For InnoDB tables built in versions prior to 5.0.3, you can manually insert a dummy row with a value one less than the desired value after creating the dummy table; you then delete the dummy row.</td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Min. Rows/Max Rows</td>
<td>Note that if you set the value for either parameter at 0, MySQL removes the setting. MySQL will take care of the row settings in this case.</td>
</tr>
<tr>
<td>Average Row Length</td>
<td></td>
</tr>
<tr>
<td>Pack Keys</td>
<td>0 \n1 \nDefault</td>
</tr>
<tr>
<td>Check Sum</td>
<td>True \nFalse</td>
</tr>
<tr>
<td>Delay Key Write</td>
<td>True \nFalse</td>
</tr>
</tbody>
</table>

### Columns

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Name(s)</td>
<td>Click the little square box above the left panel to open a text field where you name the column. You can create multiple columns by returning to the little square box as many times as you need.</td>
</tr>
<tr>
<td>Required Information</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>MySQL documentation explains the column types available to you. <a href="http://dev.mysql.com/doc/mysql/en/column-types.html">http://dev.mysql.com/doc/mysql/en/column-types.html</a></td>
</tr>
<tr>
<td>Enum/Set Values</td>
<td></td>
</tr>
<tr>
<td>Unsigned</td>
<td>True/False</td>
</tr>
<tr>
<td>Zerofill</td>
<td>True/False</td>
</tr>
<tr>
<td>Binary</td>
<td>True/False</td>
</tr>
<tr>
<td>Width</td>
<td></td>
</tr>
<tr>
<td>Scale</td>
<td></td>
</tr>
<tr>
<td>Auto-increment</td>
<td>True/False—True allows an integer column to behave like a sequence based column. If you insert NULL (recommended) or 0 in the column, the column value is automatically to the next sequence value.</td>
</tr>
<tr>
<td>Allow Nulls</td>
<td>True/False</td>
</tr>
<tr>
<td>Default Character</td>
<td></td>
</tr>
<tr>
<td>Default Collation</td>
<td></td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Use Default Value</td>
<td></td>
</tr>
</tbody>
</table>
Indexes

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>In the upper panel, specify the name of any index you want to use in the table you are creating. Click the little square box to enter the name. You can enter multiple indexes.</td>
</tr>
</tbody>
</table>

Privileges

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grantee</td>
<td>Select column by column which users you want to bestow with privileges for the target table. The Grant/Revoke Privileges Dialog Box opens.</td>
</tr>
</tbody>
</table>

Merge Tables

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Space

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is available only as you edit a table.</td>
<td>Table usage and its distribution of table space is graphically displayed.</td>
</tr>
</tbody>
</table>

DDL View

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDL View</td>
<td>The DDL window displays the command code you created by completing the prior panels of the wizard. You have the ability to save the code to disk, to print it, to email it, to open an ISQL window where you can tinker with the code, or to schedule the execution of the code.</td>
</tr>
<tr>
<td>Launch object editor after execute</td>
<td>When this option is checked, the script will execute immediately.</td>
</tr>
</tbody>
</table>

Schema Publication

DBArtisan features a Schema Publication Wizard that can generate HTML reports documenting the details of your databases, which you can share over an intranet or the Internet. DBArtisan uses HTML Templates to create the reports that you can manually edit to create customized reports.

The Schema Publication Wizard lets you:

- Generate reports for all or any level of your database.
- View, find, and print the HTML reports directly in DBArtisan with the built-in HTML browser.
• Create a new schema publication operation.
• Reuse a previously defined schema publication operation.
• Select a datasource for which you want to generate an HTML report.
• Select the server object type and database object type for which you want to publish a schema report and lets you customize the report.
• Select specific objects and to set certain script options for each object type.
• Specify the report directory, HTML file name, and report title.
• Save the schema publication operation for re-use.
• Preview the definition of your schema publication operation.

Publishing a Schema
Generating a report for any level of your MySQL database is made easy by DBArtisan.

To Open the Schema Publication Wizard:
1. On the Datasource Explorer, select the MySQL server.
2. On the Utilities menu, click Schema Publication.

The sections below describe the fields you may encounter as you create the publication operation:

Playback or New Operation

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Publication Operation</td>
<td>This initiates the process of creating a new publication operation.</td>
</tr>
<tr>
<td>Previous Publication Operation</td>
<td>If you want to replay a publication operation, when you select the radio button, the browse function is enabled so you can find the publication file you want to re-use.</td>
</tr>
</tbody>
</table>

Server Connection
Here you select the MySQL server and database(s) that you want to include in this particular publication instance.

Object Type Selection
Specify the server object(s) you want to include in the schema report. You can also select database objects organized by owner. To create a customized publication script based on individual objects, click the customize box.

Object Selection
Select the objects you had in mind when you opted to customize the object list.

Publication Option
Give the report a name, and decide if you want to create separate html files for each object encompassed by the report.
Summary
A summary of the publication operation as you created it in the preceding steps is displayed. If you want to make
changes, you need to go Back. You can also designate the file where you want the output script to live and decide
whether or not you want to save this operation for use again. Files are saved with the .sav extension.

Execute
Watch the progress of the publication as it is generated and see any errors if they occur. Then click Continue or Finish
to see the results.

Server Configuration
DBArtisan places Configuration under the Server node of the Datasource Explorer. When you select Configuration,
DBArtisan shows configuration information about your datasource in the right pane of the Explorer window. You can
use the Datasource Explorer or shortcut menu to open the System Configuration information panel and make changes
to the server’s configuration.

You can:
- Open
- Extract
- Report
- or Refresh

Configuring a Server Variable

To Open the System Configuration information panel:
1. On the Explorer, find the server where you want to make changes to the configuration.
2. Expand the Server node and double-click the Configuration branch.
3. Double-click any variable in the right-pane.

OR
- Right-click in the variable pane and select Open.

The System Configuration information panel displays the name, value, and dynamic status of each system variable.
Make changes to the value column and click Alter on the toolbar. When you do this, the Preview SQL box opens.

Database Backups
Implementing a plan to back up critical databases is perhaps the most important job for any database administrator.
Without a sound disaster recovery plan, your company has no safeguards against hardware and software failures.

The Database Backup Wizard lets you backup your databases without knowing the underlying syntax. It provides
support for backing up databases as well as transaction logs, including commands to stripe the backup across
multiple devices.
The Database Backup Wizard lets you:

- Create a new dump operation.
- Re-use a previously defined operation.
- Specify the target server, database, and type of backup operation.
- Specify backup device type.
- Specify how to report the progress of the backup operation.

You can back up the server while the database is online and in use. However, the backup will fail if you are performing one of the following operations while attempting the back up:

- Creating indexes.
- Performing non-logged operations.
- Shrinking the database.
- Creating or deleting database files.

Creating or Editing a Backup Operation

DBArtisan lets you create different backup operations using the Database Backup Wizard. After you have successfully created a backup operation, you can reopen and edit it.

To Open the Database Backup Wizard

1. On the Datasource Explorer, select the Databases node.
   
   DBArtisan displays the databases in the right pane of the Explorer window.

2. Select the target database.

3. On the Utilities menu, click Database Backup.

In the following you’ll find all the fields you’ll encounter as you create or edit a backup operation:

Playback or New Operation

Backup Selection

Select Databases

Select Tables

Options Editor for the Wizard

Output File Type

Normal Output File Options

Datafile Options

Summary
### Playback or New Operation

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Dump Operation</td>
<td>This initiates the process of creating a new backup operation.</td>
</tr>
<tr>
<td>Previous Dump Operation</td>
<td>If you want to replay a backup operation, when you select the radio button, the browse function is enabled so you can find the .DBDump file you want to re-use.</td>
</tr>
</tbody>
</table>

### Backup Selection

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Databases</td>
<td>This is the default operation.</td>
</tr>
<tr>
<td>Backup tables within a single database</td>
<td></td>
</tr>
</tbody>
</table>

### Select Databases

From the checklist, choose one or more databases you want to include in the new dump operation.

### Select Tables

From the drop-down list, choose the database you want to back-up tables from and then check one or more tables to include in the new dump operation.

### Options Editor for the Wizard

**NOTE:** By default, the options on this wizard panel are sorted by category. You can also sort them alphabetically if that makes life easier.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TABLES:</strong></td>
<td></td>
</tr>
<tr>
<td>No Data</td>
<td>True/False: Backup DDL only, without saving any data rows.</td>
</tr>
<tr>
<td>Continue if Errors Occur</td>
<td>True/False: Force the dump to continue even in the event there is a SQL error when dumping a table.</td>
</tr>
<tr>
<td>Flush Server Logs</td>
<td>True/False: Flush MySQL logs before the dump begins.</td>
</tr>
<tr>
<td>Drop Table Statements</td>
<td>True/False: DROP TABLE statements will be added.</td>
</tr>
<tr>
<td>Lock/Unlock Statements</td>
<td>True/False: Add LOCK/UNLOCK table statements around each table dump.</td>
</tr>
<tr>
<td>Lock All Tables</td>
<td>True/False: Lock all tables prior to beginning the dump.</td>
</tr>
<tr>
<td>Allow Keyword Column Names</td>
<td>True/False: Permits creation of column names that are keywords.</td>
</tr>
<tr>
<td>Complete Insert Statements</td>
<td>True/False: Complete insert statements will include column names.</td>
</tr>
<tr>
<td>Use MySQL Table Options</td>
<td>True/False: Include all MySQL-specific table options in the CREATE TABLE statements.</td>
</tr>
</tbody>
</table>
### Required Information

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Insert Delayed Statements</td>
<td>True/False: Insert rows using INSERT DELAYED statements.</td>
</tr>
<tr>
<td>Disable Keys</td>
<td>True/False</td>
</tr>
<tr>
<td></td>
<td>NOTE: The disable keys command is only valid for MyISAM tables.</td>
</tr>
<tr>
<td>Use Extended Inserts</td>
<td>True/False: Enables extended INSERT syntax that includes several VALUES lists.</td>
</tr>
<tr>
<td>Use Create Table Statements</td>
<td>True/False: Script CREATE TABLE statements for each table.</td>
</tr>
<tr>
<td><strong>GENERAL OPTIONS:</strong></td>
<td></td>
</tr>
<tr>
<td>Suppress Information</td>
<td>True/False: Suppress additional information like program version, host, and so forth.</td>
</tr>
<tr>
<td>Exclude Create Database Statements</td>
<td>True/False: Will not write CREATE DATABASE statements.</td>
</tr>
<tr>
<td>Retrieve Single Rows</td>
<td>True/False: Instead of retrieving the row set, retrieve one row at a time.</td>
</tr>
<tr>
<td>Use Quotes</td>
<td>True/False: Use quotation marks around database, table, and column names.</td>
</tr>
<tr>
<td>Use Begin Statement</td>
<td>True/False: Before dumping data from the server, issue a BEGIN statement.</td>
</tr>
<tr>
<td><strong>REPLICATION:</strong></td>
<td></td>
</tr>
<tr>
<td>Use First-Slave Option</td>
<td>True/False: Lock all tables across all databases.</td>
</tr>
<tr>
<td>Use Master-Data Option</td>
<td>True/False: Lock all tables and generate CHANGE MASTER TO statements.</td>
</tr>
<tr>
<td>Delete Binary Logs</td>
<td>True/False: Delete binary logs after the dump operation on the master replication server.</td>
</tr>
<tr>
<td><strong>CHARACTER SET:</strong></td>
<td></td>
</tr>
<tr>
<td>Suppress Set Names Statement</td>
<td>True/False: Suppress SET NAMES default_character_set statement.</td>
</tr>
<tr>
<td>Default Character Set</td>
<td>Identify the character set that you want to use as the default.</td>
</tr>
<tr>
<td><strong>NETWORK:</strong></td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td>Choose TCP, Socket, Pipe, or Memory as the connection protocol.</td>
</tr>
<tr>
<td>Max Allowed Packet</td>
<td>Enter the maximum size of the client/server communication buffer.</td>
</tr>
<tr>
<td>Net Buffer Length</td>
<td>Enter the maximum size of the client/server communication buffer.</td>
</tr>
<tr>
<td>Compress</td>
<td>True/False: Compresses all information sent between the client and the server if both support compression.</td>
</tr>
<tr>
<td><strong>COMPATIBILITY:</strong></td>
<td></td>
</tr>
<tr>
<td>ANSI</td>
<td>True/False: Produces ANSI-compatible content.</td>
</tr>
<tr>
<td>MySQL 3.23</td>
<td>True/False: Produce MySQL 3.23-compatible output.</td>
</tr>
<tr>
<td>MySQL 4.0</td>
<td>True/False: Produce MySQL 4.0-compatible output.</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>True/False: Produce PostgreSQL-compatible output.</td>
</tr>
<tr>
<td>Oracle</td>
<td>True/False: Produce Oracle-compatible output.</td>
</tr>
<tr>
<td>MSSQL</td>
<td>True/False: Produce MSSQL-compatible output.</td>
</tr>
</tbody>
</table>
Here you find a summary of all the choices you have made in creating this operation. If you want to make changes, you can make them in this grid or go Back. You also have the option of saving the operation. Saved files are identified with a .DBDump extension.
Database Restore/Import

To both validate your backups and practice the disaster recovery process, you should regularly restore your database dumps. If possible, every day, you should restore your databases on a separate server to enhance system availability. In this manner, you can maintain a warm backup of your database guarding against downtime from system failure.

The Database Restore/Import Wizard lets you:

- Create a new load operation.
- Reuse a previously defined operation.
- Specify the target server, database, and type of restore operation.
- Specify the recovery completion state.
- Specify how to report the progress of the load operation.
- Specify which file to load from the device.
- Specify whether to perform safety checks before replacing the database.

Creating or Editing a Restore/Import Operation

DBArtisan lets you create different restore/import operations using the Database Import Wizard. After you have successfully created an import operation, you can reopen and edit it.

To Open the Database Import Wizard

1. On the Datasource Explorer, select the Databases node.
   - DBArtisan displays the databases in the right pane of the Explorer window.

2. Select the target database.

3. On the Utilities menu, click Database Restore.

In the following you’ll find all the fields you’ll encounter as you create or edit a restore/import operation:

Playback or New Operation

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Import Operation</td>
<td>This initiates the process of creating a new import operation.</td>
</tr>
<tr>
<td>Previous Import Operation</td>
<td>If you want to replay an import operation, when you select the radio button,</td>
</tr>
<tr>
<td></td>
<td>the browse function is enabled so you can find the .DBDump file you want to</td>
</tr>
<tr>
<td></td>
<td>reuse.</td>
</tr>
</tbody>
</table>

Target Database

Select the target database from the drop-down list.

Select Data Files

Type or browse to the file(s) that has the data you want to import.
Database Import Wizard Options

**NOTE:** By default, the options on this wizard panel are sorted by category. You can also sort them alphabetically if that makes life easier.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NETWORK OPTIONS:</strong></td>
<td>Choose TCP, Socket, Pipe, or Memory as the connection protocol.</td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
</tr>
<tr>
<td>Compress</td>
<td>True/False: Compresses all information sent between the client and the server if both support compression.</td>
</tr>
<tr>
<td><strong>PRE-LOAD OPTIONS:</strong></td>
<td>True/False: Delete all table data before the data file is imported.</td>
</tr>
<tr>
<td>Delete Table Data</td>
<td></td>
</tr>
<tr>
<td>Force Continue</td>
<td>True/False: If there's an error, force the program to continue processing the load file.</td>
</tr>
<tr>
<td>Lock Tables</td>
<td>True/False: Keep tables locked to writing before processing text files.</td>
</tr>
<tr>
<td>Input File Location</td>
<td>Local/Server: Where the input files are stored.</td>
</tr>
<tr>
<td>Ignore Header Rows</td>
<td>Supply a value or leave blank. The default is zero. You are asking to ignore x number of rows from the data file.</td>
</tr>
<tr>
<td><strong>FORMATTING:</strong></td>
<td>Include the value by which the fields are terminated.</td>
</tr>
<tr>
<td>Fields Terminated By</td>
<td></td>
</tr>
<tr>
<td>Fields Enclosed By Display</td>
<td>Include the value by which the fields are enclosed.</td>
</tr>
<tr>
<td>Fields Optionally Enclosed By</td>
<td>Include the value by which the fields are optionally enclosed.</td>
</tr>
<tr>
<td>Fields Escaped By</td>
<td>Include the value by which the fields are escaped.</td>
</tr>
<tr>
<td>Lines Terminated By</td>
<td>Include the value by which the fields are terminated.</td>
</tr>
</tbody>
</table>

**Duplicate Key Options**

Choose among handling duplicate entries by enabling new rows to replace existing rows, ignore input rows that duplicate existing rows, or allowing an error to occur when a duplicate is found and ignore the remainder of the input file.

**Summary**

Here you find a summary of all the choices you have made in creating this operation. If you want to make changes, you can make them in this grid or go Back. You also have the option of saving the operation. Saved files are identified with a .DBImport extension.
Utilities

DBArtisan incorporates a number of powerful utilities that let you create, edit and manage your development environment. You can use utilities to:

- Conduct extensive database searches across multiple databases.
- Perform complex cross-platform database migrations.
- Extract schema for multiple database objects.
- Load and unload table data.
- Configure your server.
- Monitor and analyze all processes on the server.
- Schedule tasks.

Cross-Platform Utilities

The table below describes the utilities for IBM DB2 for Linux, Unix, and Windows, IBM DB2 for z/OS and OS/390, Microsoft SQL Server, MySQL, Oracle, and Sybase ASE:

<table>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the schema extraction wizard</td>
<td>The Schema Extraction Wizard lets you reverse-engineer an entire database or any portion of it.</td>
</tr>
<tr>
<td>Using the schema migration wizard</td>
<td>The Schema Migration Wizard offers robust cross-DBMS-platform object migration capabilities that can migrate schema and table data between databases on the same or different servers.</td>
</tr>
<tr>
<td>Schema Publication Wizard</td>
<td>The Schema Publication Wizard lets you generate HTML reports documenting the details of your databases.</td>
</tr>
<tr>
<td>Database Monitor</td>
<td>The Database Monitor displays all processes connected to the server and lets you scan your databases to determine potential performance problems.</td>
</tr>
<tr>
<td>SQL Logfile</td>
<td>DBArtisan lets you log all of its SQL statements to a file so you have an audit trail of application activity.</td>
</tr>
</tbody>
</table>

IBM DB2 for Linux, Unix, and Windows Utilities

The table below describes the utilities for IBM DB2 for Linux, Unix, and Windows:

<table>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Export Wizard</td>
<td>The Data Export Wizard lets you export table data.</td>
</tr>
<tr>
<td>Data Import Wizard</td>
<td>The Data Import Wizard lets you import table data.</td>
</tr>
<tr>
<td>Database Manager</td>
<td>The Database Manager lets you view and manage your databases.</td>
</tr>
</tbody>
</table>

Microsoft SQL Server Utilities

The table below describes the utilities for Microsoft SQL Server:

<table>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Unload Wizard</td>
<td>The Data Unload Wizard lets you export table data.</td>
</tr>
<tr>
<td>Data Load Wizard</td>
<td>The Data Load Wizard lets you load data.</td>
</tr>
</tbody>
</table>
Oracle Utilities

The table below describes the utilities for Oracle:

<table>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Manager</td>
<td>The Database Manager lets you view and manage your databases.</td>
</tr>
<tr>
<td>Database Backup Wizard</td>
<td>The Database Backup Wizard lets you backup your databases without having to know the underlying syntax.</td>
</tr>
<tr>
<td>Database Recovery Wizard</td>
<td>The Database Recovery Wizard lets you restore your databases on a separate server to enhance system availability.</td>
</tr>
<tr>
<td>Data Export Wizard</td>
<td>The Data Export Wizard lets you extract object definitions and table data from an Oracle database and create a binary export file.</td>
</tr>
<tr>
<td>Data Import Wizard</td>
<td>The Data Import Wizard lets you extract object definitions from an Oracle Export file and inputting them into the database.</td>
</tr>
<tr>
<td>Archive Activity</td>
<td>DBArtisan lets you operate an Oracle database in ARCHIVELOG or NOARCHIVELOG mode.</td>
</tr>
<tr>
<td>Backup Control Files</td>
<td>DBArtisan lets you use the graphical interface for constructing the ALTER DATABASE command needed to backup your control files.</td>
</tr>
<tr>
<td>Chained Row Analysis</td>
<td>DBArtisan lets you analyze chained rows.</td>
</tr>
<tr>
<td>Free Space Deficit</td>
<td>The Free Space Deficit Monitor identifies any objects not be able to acquire their next extent.</td>
</tr>
<tr>
<td>Instance Parameter</td>
<td>The Instance Parameter Configuration lets you view and edit the settings for each Oracle instance.</td>
</tr>
<tr>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td>Configuring Resource</td>
<td>DBArtisan lets you use the visual interface for constructing an ALTER RESOURCE COST command.</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
</tr>
<tr>
<td>Session Information</td>
<td>DBArtisan lets you view session information.</td>
</tr>
<tr>
<td>Capacity Analyst</td>
<td>Collects critical database metrics at scheduled intervals and provides visual and reporting mechanisms for historical trend analysis and forecasting.</td>
</tr>
<tr>
<td>Performance Analyst</td>
<td>A client-side only visual database monitor that provides insight into current database performance through the use of key performance metrics, bottleneck analysis, and user-configurable thresholds.</td>
</tr>
<tr>
<td>Space Analyst</td>
<td>A storage structure/object space diagnostic facility that aids in the proactive analysis of current and potential storage-related bottlenecks, and provides a Reorganization Wizard that handles full storage structure and object reorganization needs.</td>
</tr>
</tbody>
</table>

Schema extraction and migration utilities

DBArtisan lets you migrate schema and table data between databases on the same or different servers. Similarly, you can build and store a schema extraction script. Both features employ a wizard that walks you through the process and make use of similar options.

For detailed information, see the following topics:

- [Using the schema extraction wizard](#)
- [Using the schema migration wizard](#)
Using the schema extraction wizard

The Schema Extraction wizard walks you through the process of building an extraction script, executing it for storage of the database or object schema in a file. The basic process includes choosing the server and database objects for extractions and specifying options that dictate how the extraction script is built. This utility allows you to:

- Extract and archive entire databases or sets of database objects.
- Generate the complete definition of any database object type.
- Reverse-engineer schema, and also place the SQL statements in their proper sequence, regardless of the objects you select.
- Create a new schema extraction operation.
- Reuse a previously defined schema extraction operation.
- Specify the target object types to extract.
- Set the general script type and options.
- Preview the definition of your schema extraction operation.

To build and execute a schema extraction script:

1. Connect to the datasource where you want to extract schema. For details, see Datasource Management.
2. On the Datasource Explorer, select the target database or database object.
4. Use the following table as a guide to understanding and setting options in the wizard:

<table>
<thead>
<tr>
<th>Panel</th>
<th>Tasks and Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Welcome</td>
<td>Starts you on the schema extraction</td>
<td></td>
</tr>
<tr>
<td>2 - Playback</td>
<td>Perform new extraction and Playback an existing extraction</td>
<td>Lets you start a new extraction or select an existing extraction script you want to replay from the drop-down list. <strong>NOTE:</strong> When the script opens, you can make changes to the operation you are replaying. The changes you make will overwrite the current script. In Normal mode, objects are extracted in memory, which for larger extractions can result in the operation running out of memory. In Iterative mode, objects are extracted one at a time, and the operation in this mode will never run out of memory.</td>
</tr>
<tr>
<td>3 - Connection</td>
<td>Source Datasource</td>
<td>Lets you select the server and depending on the DBMS you are working with, one or more databases from which you want to extract the schema.</td>
</tr>
<tr>
<td>4 - Object Type Selection</td>
<td>Server Objects and Database Objects</td>
<td>Lets you select the server object types and database object types (optionally specifying database objects with a specified owner) that will be extracted. If you do not enable the Customize Object List option, all objects of the selected types will be extracted.</td>
</tr>
<tr>
<td>5 - Object Selection</td>
<td>Only displayed if you selected the Customize Object List option on the previous panel.</td>
<td>For the types you chose on the previous panel, lets you use the <strong>Type</strong> and <strong>Database</strong> dropdowns to display available objects. You can then select the specific objects for extraction.</td>
</tr>
</tbody>
</table>
Using the schema migration wizard
DBArtisan offers robust cross-DBMS-platform object migration capabilities that can migrate schema and table data between databases on the same or different servers. DBArtisan's migration capabilities automatically resolve differences between DBMS platforms, alleviating the responsibility of knowing the syntax and object variations inherent in database platforms. The Schema Migration Wizard sequences the creation of objects in the proper order to eliminate dependency errors. It also has the intelligence to use the fastest method available for copying table data.

The Schema Migration Wizard lets you:

- Create a new schema migration operation.
- Re-use a previously defined schema migration operation.
- Specify a source and a target datasource.
- Specify server object types, database object types, and general migration options.
- Select specific objects and to set certain migration options for each object type.
- Specify object ownership.
- Specify if the migration operation is to be saved for reuse.

**CAUTION:** Schema migration jobs created in earlier versions of DBArtisan that were saved with a .migrate file extension must be re-created in DBArtisan 8.1 or later. Only migration jobs with a .sav extension can be reused.
The table below describes the object types that can be migrated to different supported platforms and the utility used to accomplish the migration operation:

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Object Type</th>
<th>Data Out Method</th>
<th>Data In Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>DB2</td>
<td>All Object Types &amp; Data</td>
<td>DB2 Export API Call</td>
<td>DB2 Import API Call</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Tables, data, views, indexes, users</td>
<td>SQL SELECT</td>
<td>BCP Utility</td>
<td></td>
</tr>
<tr>
<td>Sybase ASE</td>
<td>Tables, data, views, indexes, users</td>
<td>SQL SELECT</td>
<td>BCP Utility</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>Tables, data, views, indexes, users</td>
<td>SQL SELECT</td>
<td>SQL * Loader Utility</td>
<td></td>
</tr>
<tr>
<td>SQL Server</td>
<td>SQL Server</td>
<td>All Object Types &amp; Data</td>
<td>BCP Utility</td>
<td>BCP Utility</td>
</tr>
<tr>
<td>Sybase ASE</td>
<td>All Object Types &amp; Data</td>
<td>SQL SELECT</td>
<td>SQL * Loader Utility</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>All Object Types and Data</td>
<td>SQL SELECT</td>
<td>SQL * Utility</td>
<td></td>
</tr>
<tr>
<td>DB2</td>
<td>Tables, data, views, indexes, logins</td>
<td>SQL SELECT</td>
<td>DB2 Import API Calls</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>SQL Server</td>
<td>All Object Types &amp; Data</td>
<td>BCP Utility</td>
<td>BCP Utility</td>
</tr>
<tr>
<td>Sybase ASE</td>
<td>Tables, data, views, indexes, logins</td>
<td>BCP Utility</td>
<td>BCP Utility</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>All Object Types and Data</td>
<td>SQL SELECT</td>
<td>SQL * Utility</td>
<td></td>
</tr>
<tr>
<td>DB2</td>
<td>Tables, data, views, indexes, logins</td>
<td>SQL SELECT</td>
<td>SQL * Utility</td>
<td></td>
</tr>
<tr>
<td>Sybase ASE</td>
<td>Sybase ASE</td>
<td>All Object Types &amp; Data</td>
<td>BCP Utility</td>
<td>BCP Utility</td>
</tr>
<tr>
<td>SQL Server</td>
<td>All Object Types &amp; Data</td>
<td>BCP Utility</td>
<td>BCP Utility</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>Tables, data, views, indexes, logins</td>
<td>SQL SELECT</td>
<td>SQL * Utility</td>
<td></td>
</tr>
<tr>
<td>DB2</td>
<td>Tables, data, views, indexes, logins</td>
<td>SQL SELECT</td>
<td>DB2 Import API Calls</td>
<td></td>
</tr>
</tbody>
</table>

To copy objects between servers, the Schema Migration Wizard establishes separate connections to both the source and target servers. Based on the copy options you choose, it then migrates schema from the source to the target database. If the bulk copy command is used when copying table data, a temporary BCP file is built in the temp directory of your computer.

To copy objects between servers, the Schema Migration Wizard establishes separate connections to both the source and target servers. Based on the copy options you choose, it then migrates schema from the source to the target database. If the bulk copy command is used when copying table data, a temporary BCP file is built in the temp directory of your computer.

To **Open the Schema Migration Wizard**:
1. Connect to the datasource from which you want to migrate schema. For details, see [Datasource Management](#).  
2. On the Datasource Explorer, select the target database or database object.  
3. On the *Utilities* menu, click *Schema Migration*.  
   The Migration Wizard opens.
4. Use the following table as a guide to understanding and setting options in the wizard:

<table>
<thead>
<tr>
<th>Panel</th>
<th>Tasks and Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Welcome</td>
<td>Starts you on the schema migration</td>
<td></td>
</tr>
<tr>
<td>2 - Playback</td>
<td><strong>Perform new migration and Playback an existing migration</strong></td>
<td>Lets you start a new migration or select an existing migration script you want to replay from the drop-down list. <strong>NOTE:</strong> When the wizard opens, you can make changes to the operation you are replaying..</td>
</tr>
<tr>
<td></td>
<td>Normal mode and Iterative mode</td>
<td>In Normal mode, objects are extracted in memory, which for larger extractions can result in the operation running out of memory. In Iterative mode, objects are extracted one at a time, and the operation in this mode will never run out of memory.</td>
</tr>
</tbody>
</table>
Executed statements before or after creation of functions or procedures

During migration, it can be necessary to have statements executed before and after creation of the procedure or function. This can be useful for example, if you need to create or drop temporary tables used by the function or procedure. DBArtsan supports the use of two tag pairs, ETStart and ETEnd, that let you embed statements in the first comment block of a stored procedure or function. The following shows the expected syntax:

```sql
create procedure dbo.procname(@a numeric) as
/*
<ETStart>SQL Statement</ETStart>
<ETEnd>SQL Statement</ETEnd>
*/
begin
...
```

### Executing statements before or after creation of functions or procedures

**Panel**

<table>
<thead>
<tr>
<th>Tasks and Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3 - Connection</strong></td>
<td>Source Datasource and Target Datasource</td>
</tr>
<tr>
<td><strong>4 - Object Type Selection</strong></td>
<td>Server Objects and Database Objects</td>
</tr>
<tr>
<td><strong>5 - Object Selection</strong></td>
<td>Only displayed if you selected the Customize Object List option on the previous panel.</td>
</tr>
<tr>
<td><strong>6 - Options</strong></td>
<td>For a detailed descriptions of the options present on this panel, see Common schema migration and extraction options.</td>
</tr>
<tr>
<td><strong>7 - Summary</strong></td>
<td>Displays a summary of the selections you made. The settings shown on this panel are read-only, and you must move back to previous panels in order to make changes. This panel presents the following options:</td>
</tr>
<tr>
<td><strong>Output script to file</strong></td>
<td>UNDER WHAT CIRCUMSTANCES IS THIS FEATURE, IN GENERAL ENABLED</td>
</tr>
<tr>
<td><strong>Save existing operation</strong></td>
<td>Lets you save a copy the choices you made on the wizard panels. On subsequent schema migrations, you can then playback the saved migration settings, optionally changing option settings. You specify the path and name of the file in which to save the migration operation.</td>
</tr>
<tr>
<td><strong>Click Finish</strong></td>
<td>when ready to proceed with the extraction.</td>
</tr>
<tr>
<td><strong>8 - Execute</strong></td>
<td>Displays the status of the migration with details on the number of objects retrieved and created, the number of errors detected, and the elapsed time. Filter options (Show All, Show Errors, and Show Warnings) let you display specific types of messages generated during the migration process. When the extraction completes, you can also use the following option:</td>
</tr>
<tr>
<td><strong>Continue</strong></td>
<td>Opens the extraction script in the ISQL Editor. For more information, see SQL Scripting.</td>
</tr>
<tr>
<td><strong>Report</strong></td>
<td>DOES THIS REALLY HAPPEN ON A SUCCESSFUL MIGRATION?</td>
</tr>
<tr>
<td><strong>Report</strong></td>
<td>Opens a detailed report on the results of the schema migration.</td>
</tr>
</tbody>
</table>
Common schema migration and extraction options

When extracting or migrating schema, the wizards present a set of options that let you specify how the extraction or migration script is created. The following table describes these options.

**NOTE:** Depending on the DBMS you are working with, the types of objects you choose to migrate/extract, and the specific migration or extraction operation, not all options are available.

<table>
<thead>
<tr>
<th>Tasks and Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Dependency Translators</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>Script Dependencies</td>
<td>Include dependent objects such as indexes, views, and triggers in the extraction/migration.</td>
</tr>
<tr>
<td>Group Dependencies</td>
<td>If set to <strong>True</strong>, objects that are dependent on each other are placed together in the script. Example: Table A would be followed by all objects dependent on Table A. If set to <strong>False</strong>, all objects of the same type are extracted simultaneously.</td>
</tr>
<tr>
<td>Iterative dependencies</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>Parse text to determine dependencies</td>
<td>For dependency information not stored in system catalogs, this method parses object text for dependencies, ensuring dependency ordering on extract.</td>
</tr>
<tr>
<td>Script... options</td>
<td>Depending on the DBMS you working with and the types of objects you selected, these options let you specify whether you want to include Create statements, Drop statements, System Privileges, Object Privileges, and If Not Exists statements in the script.</td>
</tr>
<tr>
<td>Use Alter Statements</td>
<td>True/False to script dependent objects as Alter statements.</td>
</tr>
<tr>
<td>Include Storage</td>
<td>Lets you include or exclude storage attributes of objects (tablespace, filegroup, and in Oracle, the STORAGE clause).</td>
</tr>
<tr>
<td>Generate DDL... options</td>
<td>If set to <strong>True</strong>, these options let you include system or non-system objects in the extraction/migration. If set to <strong>False</strong>, these objects are ignored.</td>
</tr>
<tr>
<td>Only allow explicitly included types</td>
<td>If set to TRUE, only objects specified using the Include... options are migrated/extracted. If set to FALSE, all dependent objects are migrated/extracted.</td>
</tr>
<tr>
<td>Script Use Statement</td>
<td>Optimizes extraction through use of Use statements in the generated script.</td>
</tr>
<tr>
<td>Script Owner</td>
<td>True/False to include the owner in the scripts or not.</td>
</tr>
<tr>
<td>Owner</td>
<td>Retain Owner/Change Owner</td>
</tr>
<tr>
<td>Change Owner To</td>
<td>If you opted to change the owner, choose the new owner to use when scripting options from the drop-down list.</td>
</tr>
<tr>
<td>Copy Owner</td>
<td>Lets you retain the owner in the extraction/migration.</td>
</tr>
<tr>
<td>Include Users and Roles</td>
<td>If set to TRUE, user and role definitions are included in the extraction/migration.</td>
</tr>
<tr>
<td>Tables: Include... options</td>
<td>Depending on the DBMS you are working with, these options let you include aliases, defaults, indexes, packages, views, rules, and table constraints in table scripts.</td>
</tr>
<tr>
<td>Truncate table</td>
<td>If set to <strong>True</strong>, a TRUNCATE TABLE SQL statement is generated. This option is enabled only if the <strong>Script Drop Statements</strong> option is set to FALSE.</td>
</tr>
<tr>
<td>Table triggers</td>
<td>Extract dependent triggers for any tables in the extraction. If false, no dependent triggers are extracted.</td>
</tr>
<tr>
<td>Data Movement</td>
<td>These choices determine how or if data should be copied from tables. Extraction offers No movement, Data Copy, and Insert Statements. Migration offers No Movement, Select Into, Data Copy, and Insert Statements.</td>
</tr>
<tr>
<td>Batch Size</td>
<td>Lets you specify the number of rows in one batch update for data copy.</td>
</tr>
<tr>
<td>Preserve Existing Indexes</td>
<td>If set to TRUE, existing indexes are captured for recreation.</td>
</tr>
<tr>
<td>Preserve Existing Triggers</td>
<td>If set to TRUE, existing triggers are captured for recreation.</td>
</tr>
</tbody>
</table>
For context information, see the following topics:

- [Using the schema extraction wizard](#)
- [Using the schema migration wizard](#)

## Using the Schema Publication wizard

The Schema Publication Wizard lets you generate HTML reports documenting the details of your schema, databases, and objects, which you can share over an intranet or the Internet. DBArtisan uses HTML Templates to create the reports which you can manually edit to create customized reports.

Schema publication produces a two-pane report. The left-hand pane lets you navigate through all objects contained in the report while the right-hand pane shows details for an individual object. The following example shows a report on three Microsoft SQL Server default objects, displaying details for a default named `UTDS.DefaultObject`. 

<table>
<thead>
<tr>
<th>Tasks and Settings</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Overwrite Existing** | Choose one of the following options:  
  **Overwrite Existing**: objects are overwritten with no prompt. Objects on the target datasource are dropped before new objects are created  
  **Prompt Whether To Overwrite**: The user will be prompted before the object is dropped and re-created. *(Note: this will be done on an object-by-object basis).*  
  **Do Not Overwrite Existing**: If the object exists on the target datasource, no attempts is made to recreate it on the target.  
  **Ignore Overwrite**: (default) Artisan will try to create the migrated objects, without checking the existence on the target datasource. |
| **Generate Script** | If set to FALSE, the generated DDL is executed on the target datasource. If set to TRUE, DBArtisan generates a script which could be used to migrate the schema at a later date. |
| **Update statistics** | If set to TRUE, the operation generates an UPDATE STATISTICS SQL statement to be included in the extraction/migration. |
The Schema Publication Wizard lets you:

- Generate reports for all or any level of your database.
- View, find, and print the HTML reports directly in DBArtisan with the built-in HTML browser.
- Create a new schema publication operation.
- Re-use a previously defined schema publication operation.
- Select a datasource for which you want to generate an HTML report.
- Select the server object type and database object type for which you want to publish a schema report and lets you customize the report.
- Select specific objects and to set certain script options for each object type.
- Specify the report directory, HTML file name, and report title.
- Save the schema publication operation for re-use.
- Preview the definition of your schema publication operation.

To create a schema report:

1. Connect to the datasource for which you want to produce a schema report. For details, see Datasource Management.

2. On the Utilities menu, click Schema Publication.
3 Use the following table as a guide to understanding and setting options in the wizard:

<table>
<thead>
<tr>
<th>Panel</th>
<th>Tasks and Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome</td>
<td>New Publication Operation</td>
<td>Select to define a new schema publication operation.</td>
</tr>
<tr>
<td></td>
<td>Previous Publication Operation and Publication Definition File Name</td>
<td>Lets you enable use of a previously defined publication operation and provide the path and file name of the operation.</td>
</tr>
<tr>
<td>Select a Datasource</td>
<td></td>
<td>Lets you select the datasource to be published.</td>
</tr>
<tr>
<td>Select Database</td>
<td>Only displayed for Microsoft SQL Server or Sybase datasources.</td>
<td>Lets you select the target database.</td>
</tr>
<tr>
<td>Select Object Types</td>
<td>Server Object Types and Database Object Types</td>
<td>Lets you select the server object types and database object types (optionally specifying database objects with a specified owner) that will be published.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you do not enable the Customize Object List option, all objects of the selected types will be published.</td>
</tr>
<tr>
<td>Select Objects</td>
<td>Only displayed if you selected the Customize Object List option on the previous panel.</td>
<td>For the types you chose on the previous panel, lets you use the ObjectType dropdown to display available objects. You can then select the specific objects for publication.</td>
</tr>
<tr>
<td>Customization</td>
<td>Report Directory and HTML File Name</td>
<td>Lets you provide the location and name of the file that is to contain the published schema.</td>
</tr>
<tr>
<td></td>
<td>Report Title</td>
<td>Lets you type the title of the report.</td>
</tr>
<tr>
<td></td>
<td>View report file when the operation is complete</td>
<td>Select if you want to view the report immediately after it is generated.</td>
</tr>
<tr>
<td></td>
<td>Create one HTM file for all objects</td>
<td>Select to create one HTM file for all objects.</td>
</tr>
<tr>
<td>Save this Operation?</td>
<td>If you want to reuse this schema publication operation, type the location and name of the file that is to store the schema publication operation. Use a .report file extension.</td>
<td></td>
</tr>
<tr>
<td>Preview</td>
<td>Let you review your selections. The settings shown on this panel are read-only, and you must move back to previous panels in order to make changes.</td>
<td></td>
</tr>
</tbody>
</table>

4 When ready to publish, click **Execute**.

DBArtisan starts the Schema Publication operation and generates an HTML publication report displaying the details of the operation.

**Database Monitor**

The Database Monitor displays all processes connected to the server. The Database Monitor:

- Lets you scan databases for potential performance problems.
- Includes tabs that provide a wealth of information about your servers.
• **Color-codes** each process to reflect its state and let you identify problems quickly.

  **TIP:** You can view some Database Monitor Information by clicking the Performance node of the Datasource Explorer.

**Important Notes**

• Processes and Locks are results of DB2 API calls and are not located in the system catalog tables.

• The Database Monitor may display incomplete SQL due to limitations set by the DBMS. Platforms set limitations on the values returned by the commands that gather the information.

**Opening the Database Monitor**

1. On the **Utilities** menu, click **Database Monitor**.

DBArtisan opens the Database Monitor.

For more information, see:

• Color-Coding
• Setting Scope Level
• Configuring the Database Monitor
• Processes Tab
• Locks Tab
• Utilities Tab
• Procedures Tab
• Functions Tab
• Traces Tab
• Logs Tab
• Bufferpools Tab
• All Locks Tab
• Blocking Locks Tab
• Active Rollback Segment Tab
• Data Dictionary Cache Tab
• Datafile I/O Tab
• General Statistics Tab
• Hit Ratio By User Tab
• Latches Tab
• Library Cache Tab
• MTS Dispatcher Contention Tab
• MTS Server Contention Tab
• Open Cursors Tab
• Parallel Queries Tab
• SGA Detail Tab
• SGA Hit Ratios Tab
• SGA Summary Tab
• Tablespace I/O Tab
• System I/O Summary Tab
• Database I/O Summary Tab
• User I/O Summary Tab
• Waits Tab
• Top SQL Cache Tab
• Last 50 Backups Tab
• File I/O Summary Tab
• Top Twenty SQL Tab
• Wait Events Tab

**Color-Coding**

The Database Monitor color-codes each process to reflect its state, to let you quickly spot problem conditions.

The table below describes the Database Monitor color-codes:

<table>
<thead>
<tr>
<th>Color</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellow</td>
<td>Sleeping or Inactive</td>
</tr>
<tr>
<td>Green</td>
<td>Running</td>
</tr>
<tr>
<td>Red</td>
<td>Blocked by another process</td>
</tr>
</tbody>
</table>

For more information, see [Database Monitor](#).

**Setting Scope Level in the Database Monitor**

**NOTE:** The functionality is available for IBM DB2 for z/OS and OS/390 only.

The Scope list lets you specify the scope of the element being monitored in a datasharing environment. Data Sharing lets one instance of the DB2 catalog to be shared equally among all members of the group.

The Scope list is available on the following tabs:

- Traces Tab
- Functions Tab
- Procedures Tab
- Utilities Tab
To set the scope level, do the following:

1. On the **Database Monitor**, click the target tab.

2. On the **Threads, Traces, Procedures** and **Functions** tabs:
   - In **Scope**, select **Group** for the monitoring results of all datasharing members in the set.
   - OR
   - In **Scope**, select **Local** for the monitoring results for the member that your machine is connected to.
     DBArtisan displays the results in the Database Monitor.

3. On the **Utility Tab**:
   - In **Scope**, select **Group** for the monitoring results of all datasharing members in the set.
   - OR
   - In **Scope**, select **Member** for data by the member name.
     DBArtisan displays the results in the Database Monitor.

### Refreshing the Database Monitor

1. On the **Database Monitor** tool bar, click the box to raise or lower the numbers, or type a new value in the box.

For more information, see [Database Monitor](#).

### Configuring the Database Monitor

The **Database Monitor** lets you:

- **Toggle** to split the screen and display the target process SQL
- **Open a SQL window and display the SQL**
- **View a Query Plan for SQL running for selected processes**

### SQL Toggle

DBArtisan lets you split the Database Monitor screen and display the SQL running for selected processes.

1. On the **Database Monitor** tool bar, click **Toggle**.

   **NOTE:** To open a SQL window or view a Query Plan for SQL running on selected processes, you must have your Database Monitor Screen split to display SQL on the bottom.

For more information, see [Configuring the Database Monitor](#).
ISQL
DBArtisan displays the SQL running for selected processes in a SQL Editor. You must turn on the SQL Toggle to display the target SQL in the split screen. To display the SQL in a separate window do the following:

1. On the Database Monitor, click the target process.  
   **NOTE:** To open a SQL Editor or view a Query Plan for processes, SQL Toggle must be on, displaying the target SQL in the split screen.

2. If SQL Toggle is not on, on the Database Monitor tool bar, click Toggle.

3. On the Database Monitor tool bar, click ISQL.
   
   DBArtisan opens a SQL window with the target process SQL.

4. To execute the SQL, click the SQL Window tool bar, click Execute.

For more information, see Configuring the Database Monitor.

Query Plan
The Database Monitor lets you view the Query Plan for target processes. To view the Query Plan, do the following:

1. On the Database Monitor, select the target process.  
   **NOTE:** To open a SQL Editor or view a Query Plan for processes, SQL Toggle must be on, displaying the target SQL in the split screen.

2. If SQL Toggle is not on, on the Database Monitor tool bar, click Toggle.

   
   DBArtisan executes the SQL with query plan and opens a Results Editor with the results.

For more information, see Configuring the Database Monitor.

Session Details
   **NOTE:** The functionality is available for Oracle only.

The Database Monitor lets you view user session details through the User Session Information dialog box. Once you open the User Session Information dialog box, you can view the SQL associated with the session.

The table below describes the tabs of the User Session Information dialog box:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical Details</td>
<td>Lets you can scroll up and down to see parameters and values.</td>
</tr>
<tr>
<td>Waits</td>
<td>Displays either historic or current wait statistics and lets you display or filter out idle wait events.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> Data returned for historic session waits is different than the data returned for current session waits.</td>
</tr>
<tr>
<td>Current SQL</td>
<td>Lets you view the SQL for this session.</td>
</tr>
</tbody>
</table>


Completing the Session Details Dialog Box
1 On the **Database Monitor** tool bar, click **Session**. DBArtisan opens the User Session Information dialog box.
2 On the **Statistical Details Tab**, in the **Session Parameters** grid, you can scroll up and down to see parameters and values.
3 On the **Waits Tab**, view select **Historical Waits** or **Current Waits**.
4 Select **Include Idle Waits** to display idle wait events.
5 On the **Current SQL Tab**, view the SQL for this session.
6 When you finish viewing SQL, click **OK**. DBArtisan returns to the Database Monitor dialog box.
For more information, see [Configuring the Database Monitor](#).

Processes Tab
The Database Monitor Processes Tab displays all processes connected to the server. The Processes Tab of the Database Monitor lets you:
- Scan databases for potential performance problems.
- View each process for potential performance problems.
- View login time and seconds idle since last call.
For more information, see:
[Killing Processes](#)
(Database Monitor)

Killing Processes
If you have system administrator privileges, DBArtisan lets you kill processes from the Database Monitor.
1 On the **Utilities** menu, click **Database Monitor**. DBArtisan opens the Database Monitor.
2 Click the target process to kill.
3 On the **Database Monitor** tool bar, click **Kill**.
4 Click **Yes**. DBArtisan confirms you want to kill the process and closes the DBArtisan message.
For more information, see [Configuring the Database Monitor](#).

Threads Tab
**NOTE:** This tab is available for IBM DB2 for z/OS and OS/390 only.
The Threads Tab displays the threads on the target datasource.

For more information, see:

Setting Scope Level

Database Monitor

Locks Tab
The Database Monitor Locks Tab displays all processes that are currently holding locks on an IBM DB2 database.

Locks include:

- Shared Lock

A Shared Lock lets other users view and read objects concurrently. The Shared Lock owner or other concurrent processes cannot alter data in table spaces, tables, or partitions.

- Exclusive Lock

DBArtisan promotes locks from Shared to Exclusive when a user decides to update an object. Once DBArtisan promotes the lock, the user must complete the update before the lock can return to shared mode; another user cannot demote a shared lock. Exclusive locks exclude other users from viewing updates before they are committed. There can only be one exclusive lock for a set of objects. The first thread gets the exclusive lock and holds it until the update is committed. Once the update is committed, the exclusive lock returns to shared mode.

**NOTE:** You cannot update a lock in DBArtisan.

For more information, see Database Monitor.

Utilities Tab

**NOTE:** This tab is available for IBM DB2 for z/OS and OS/390 only.

The Utilities Tab displays wizards that are typically run from the command line and do not apply to a particular type of schema object. Some of these wizards are also available via the command menus for objects. The wizards include:

- Check Data
- Check Index
- Copy
- Copy to Copy
- Load
- Merge Copy
- Modify Recovery
- Modify Statistics
- Rebuild Index
- Recover
- Reorganize Index
• Reorganize Tablespace
• Runstats Index
• Runstats Tablespace
• Unload
• Maintain Template

TIP: To run a wizard, select the command in the right pane of the application and click the Execute button on the Datasource Explorer toolbar.

For more information, see:
Setting Scope Level
Database Monitor

Procedures Tab

NOTE: This tab is available for IBM DB2 for z/OS and OS/390 only.

The Procedures Tab displays the procedures on the target datasource.

For more information, see:
Setting Scope Level
Database Monitor

Functions Tab

NOTE: This tab is available for IBM DB2 for z/OS and OS/390 only.

The Functions Tab displays the functions on the target datasource.

For more information, see:
Setting Scope Level
Database Monitor

Traces Tab

NOTE: This tab is available for IBM DB2 for z/OS and OS/390 only.

The Traces Tab displays the traces on the target datasource.

For more information, see:
Setting Scope Level
Database Monitor
Logs Tab

**NOTE:** This tab is available for IBM DB2 for z/OS and OS/390 only.

The Logs Tab displays the logs on the target datasource.

For more information, see Database Monitor.

Bufferpools Tab

**NOTE:** This tab is available for IBM DB2 for z/OS and OS/390 only.

The Threads Tab displays the bufferpools on the target datasource.

For more information, see Database Monitor.

All Locks Tab

The Database Monitor All Locks Tab displays all processes that are currently holding locks on a database.

For more information, see Database Monitor.

Blocking Locks Tab

The Blocking Locks Tab displays the locks that are blocked.

**TIP:** Detect blocking locks as they occur, as they can escalate into dead lock situations which can force you to shut down and restart your server to clear the traffic.

For more information, see Database Monitor.

Active Rollback Segment Tab

Rollback or undo segments are used to guarantee transactional integrity. When a transaction has not been committed or rolled back, a rollback segment will be in use with live transactional data. The Active Rollback Segment tab displays information concerning active rollback segments.

For more information, see Database Monitor.

Data Dictionary Cache Tab

The Data Dictionary Cache Tab displays key indicators in determining the performance of the shared pool. It shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much of this vital reference information be kept in RAM as possible.

For more information, see Database Monitor.
Datafile I/O Tab
The Datafile I/O Tab displays details concerning the physical I/O activity at the datafile level.

Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database against the physical datafile used to hold information. The checkpoint and archive processes (CKPT and ARCH), also perform I/O work in the database.

For more information, see Database Monitor.

General Statistics Tab
The General Statistics Tab displays general statistics data, including names and values.

For more information, see Database Monitor.

Hit Ratio By User Tab
The database server reads and updates all data from a bufferpool because memory access is much faster than disk access. Data is copied from disk to a bufferpool as needed by the applications using the database. When the server needs to read/write data and the data is already in the bufferpool, no disk access is required. However, if the data is not in the bufferpool, it needs to be read from the disk, which is significantly slower process.

The buffer pool hit ratio indicates the percentage of time that the database server did not need to load a page to service the request for a page.

For more information, see Database Monitor.

Latches Tab
Latches protect many memory structures in Oracle's SGA. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry.

For more information, see Database Monitor.

Library Cache Tab
The library cache holds commonly used SQL statements - basically database code objects. A method for improving performance in Oracle is to encourage the reuse of SQL statements so expensive parse operations may be avoided. The library cache assists this tuning effort.

For more information, see Database Monitor.

MTS Dispatcher Contention Tab
The MTS Dispatcher Contention Tab displays Microsoft Transaction Server data. DBArtisan lets you develop and deploy COM-based applications using Microsoft Transaction Server.

For more information, see Database Monitor.
MTS Server Contention Tab
The MTS Server Contention Tab displays Microsoft Transaction Server data. DBArtisan lets you develop and deploy COM-based applications using Microsoft Transaction Server.

For more information, see Database Monitor.

Open Cursors Tab
Open cursors consume Oracle resources and have the potential to degrade performance, especially if the SQL code being used in the cursor is inefficient. The Open Cursors Tab lets you quickly spot user accounts that have many cursors opened as well as the actual performance statistics for each opened cursor.

For more information, see Database Monitor.

Parallel Queries Tab
The Parallel Queries Tab displays the total number of queries eligible to be run in parallel.

For more information, see Database Monitor.

SGA Detail Tab
The Oracle System Global Area (SGA) is Oracle's memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database.

For more information, see Database Monitor.

SGA Hit Ratios Tab
The Oracle System Global Area (SGA) is Oracle's memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database.

For more information, see Database Monitor.

SGA Summary Tab
The Oracle System Global Area (SGA) is Oracle's memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database.

For more information, see Database Monitor.

Tablespace I/O Tab
The Tablespace I/O tab displays details concerning the physical I/O activity at the tablespace level.

Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. Other processes like the checkpoint and archive processes (CKPT and ARCH) may also be used.

For more information, see Database Monitor.
System I/O Summary Tab

**NOTE:** This tab is available for Microsoft SQL Server only.

The System I/O Summary Tab displays a summary of I/O activities that occur at the system level. For more information, see Database Monitor.

Database I/O Summary Tab

**NOTE:** This tab is available for Microsoft SQL Server only.

The Database I/O Summary Tab displays an overview of I/O operations at the database level. For more information, see Database Monitor.

User I/O Summary Tab

**NOTE:** This tab is available for Microsoft SQL Server only.

The User I/O Summary Tab displays an overview of I/O operations at the user operation level. For more information, see Database Monitor.

Waits Tab

**NOTE:** This tab is available for Microsoft SQL Server only.

The Waits Tab displays a detailed view of all wait activity occurring on the server. For more information, see Database Monitor.

Top SQL Cache Tab

**NOTE:** This tab is available for Microsoft SQL Server only.

The Top SQL Cache Tab displays the top 100 objects in the SQL cache, ranked by use counts. For more information, see Database Monitor.

Last 50 Backups Tab

**NOTE:** This tab is available for Microsoft SQL Server only.

The Last 50 Backups Tab displays the last fifty backups performed by the Microsoft SQL Server and lets you reference hot backup details at the server level within DBArtisan. For more information, see Database Monitor.
**File I/O Summary Tab**

**NOTE:** This tab is available for Microsoft SQL Server 8.0 only.

The File I/O Summary Tab displays an overview of I/O operations at the database file level. For more information, see Database Monitor.

**Top Twenty SQL Tab**

**NOTE:** This tab is available for Oracle only.

The Top Twenty SQL Tab lets you view the Top Twenty SQL, the top (worst) SQL running in the Oracle shared pool. For more information, see Database Monitor, SQL Toggle, Plan, ISQL.

**Wait Events Tab**

**NOTE:** This tab is available for Oracle only.

The Wait Events Tab lets you analyze bottlenecks and discover what a database is waiting on. For more information, see Database Monitor.

**Instance Configuration**

To let you configure and tune your database(s), DBArtisan offers a utility that lets you view and edit the configuration parameters for each server. These configuration parameters are stored in the system catalog and used when restarting the server. For detailed information on the meaning and purpose of each server configuration parameter, see the System Administration Guide and Performance and Tuning Guide.

**NOTE:** Many changes to instance configuration parameters require you to shutdown and to restart the server before the changes take effect.

1. On the Datasources menu, click Configure.
2. To modify a parameter, double-click the parameter or click Edit. DBArtisan opens the Edit Configuration dialog box.

**NOTE:** To modify a parameter, the parameter must have a Yes in the Session Modifiable or System Modifiable column of the Instance Parameter grid.
The table below describes the options and functionality on the Edit Configuration dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Value</td>
<td>Lets you type the value for the parameter.</td>
</tr>
</tbody>
</table>

**SQL Logfile**

You can instruct DBArtisan to log all of its SQL statements to a file so that you have an audit trail of application activity. The SQL Logfile captures all the SQL sent by the DBArtisan to populate its editors or to execute your commands. You can use this logfile not only to audit application activity, but also to build SQL scripts for querying system catalogs.

**Topics**
- Activating SQL Logging
- Opening the Logfile
- Options
- Truncating the Logfile

**Activating SQL Logging**

You can activate SQL Logging directly from the DBArtisan main menu or toolbar.

**Activating SQL Logging**

1. On the Logfile menu, click Log SQL.

**Opening the Logfile**

You can open the SQL Logfile directly from the main menu.

**Opening the Logfile**

1. On the Logfile menu, click Open.

   DBArtisan opens the SQL Logfile into an SQL window.

**Logfile Options**

You can open the Options Editor from the Logfile menu option and make any changes to your logging options.

**Opening the Options Editor**

1. On the Logfile menu, click Options.

   DBArtisan opens the Options Editor to the Logging Tab.

2. To make any changes to your logging options, make the appropriate changes and then click the OK button.

   DBArtisan closes the Options Editor.
Truncating the Logfile
If the SQL Logfile is growing too large, you can flush it directly from the main menu.

1. On the Logfile menu, click **Truncate**.

   DBArtisan deletes the contents of the SQL Logfile.

Data Export Wizard for IBM DB2 for Linux, Unix, and Windows
DBArtisan offers a simple, five-panel wizard that lets you export table data. After defining the data export operation, you have the opportunity to save the job definition for future reuse.

The Export Wizard lets you:

- Create a new Export operation.
- Re-use a previously defined Export operation.
- Specify source server, data export file format, and data export specification.
- Select the tables and view to be exported.
- Specify the Output Directory.
- Specify the Message Directory.
- Save the Job Definition for future reuse.
- Preview the definition of your data export operation.

Starting the Export Wizard
1. On the **Datasource Explorer**, select the database object.

   DBArtisan displays the database objects in the right pane of the Explorer Window.

2. On the **Utilities** menu, click **Data Unload**.

   For more information, see **Export Wizard - Panel 1**.

Export Wizard - Panel 1
The table below describes the options and functionality on the first panel of the Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you wish to initiate a new export operation or 'play back' a previously saved operation?</td>
<td>New Export Operation - Select to define a new data unload operation. Previous Export Operation - Select to use a previously defined schema extraction operation. Then in the Export Definition File Name box type the location and name of the job file or click the browse button.</td>
</tr>
</tbody>
</table>

For more information, see **Unloading Data**.
Export Wizard - Panel 2
The table below describes the options and functionality on the second panel of the Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What datasource has the data you wish to export?</td>
<td>Lets you click the list and select the datasource.</td>
</tr>
<tr>
<td>Specify the format for the data export file(s) to be generated.</td>
<td>Lets you click the list and select the file format.</td>
</tr>
<tr>
<td>Option Button</td>
<td>Available if you select a delimited option. Click to open the Delimited ASCII File Type Options dialog box.</td>
</tr>
<tr>
<td>How would you like to specify the data to export?</td>
<td>Enter a SELECT statement. Select tables/views from a list (one export file per table/view).</td>
</tr>
</tbody>
</table>

For more information, see Unloading Data.

Export Wizard - Panel 3

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the data to be extracted from the database by entering a SELECT statement</td>
<td>Enter the statement that will retrieve the row(s) or column(s) you want from one or more tables. Statements can be complicated. Go to <a href="http://msdn.microsoft.com/library">http://msdn.microsoft.com/library</a> and do a search on SELECT for help.</td>
</tr>
<tr>
<td>Would you like to store large objects (LOBs) in separate files?</td>
<td>When you choose to store LOBs in separate files, you will be given the chance to store them in files outside the database tablespaces. When you choose not to store your LOBs in separate files, they will exist in a table row in the database.</td>
</tr>
</tbody>
</table>

Export Wizard - LOB Paths

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List paths where the LOB files are to be stored</td>
<td>Add, Edit, or Remove files where you want LOBs to be stored</td>
</tr>
<tr>
<td>List LOB file names</td>
<td>Add, Edit, or Remove LOB file names.</td>
</tr>
</tbody>
</table>

File Type Options Dialog Boxes
The table below describes the options and functionality of the Integrated Exchange Format File Type Options dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept data despite code page mismatches and suppress translation</td>
<td></td>
</tr>
</tbody>
</table>


The table below describes the options and functionality of the Integrated Exchange Format File Type Options dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop existing indexes and create new ones</td>
<td></td>
</tr>
<tr>
<td>Load each row without checking target length</td>
<td></td>
</tr>
<tr>
<td>Index Schema</td>
<td></td>
</tr>
</tbody>
</table>

The table below describes the options and functionality of the Delimited ASCII File Type Options dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimiters</td>
<td>Lets you select character string and decimal point specifiers.</td>
</tr>
<tr>
<td>Prefix positive decimal values with a blank</td>
<td></td>
</tr>
<tr>
<td>Use ISO date format</td>
<td></td>
</tr>
</tbody>
</table>

For more information, see Unloading Data.

The table below describes the options and functionality of the Non-Delimited ASCII File Type Options dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not recognize x'1A' as the end of the file</td>
<td></td>
</tr>
<tr>
<td>Determine the location of an implied decimal point from the column definition</td>
<td></td>
</tr>
<tr>
<td>Record Length</td>
<td>Maximum is 32,767</td>
</tr>
<tr>
<td>Truncate trailing blanks OR Truncate trailing NULLs (0x00) characters</td>
<td>Select one or the other</td>
</tr>
</tbody>
</table>

For more information, see Unloading Data.

**Export Wizard - Panel 3**

The table below describes the options and functionality on the third panel of the Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Lets you select the tables and/or views you want to export or click the Check All button.</td>
</tr>
<tr>
<td>Count Rows</td>
<td>Lets you update the table row counts.</td>
</tr>
</tbody>
</table>
Export Wizard - Panel 4
The table below describes the options and functionality on the fourth panel of the Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check All</td>
<td>Lets you select all tables or views.</td>
</tr>
<tr>
<td>Uncheck All</td>
<td>Lets you deselect all tables or views.</td>
</tr>
</tbody>
</table>

For more information, see Unloading Data.

Export Wizard - Panel 5
Review the data export operation description to verify its accuracy.

1. If you need to make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections.

2. If you are satisfied with the data export operation, click **Execute**.

   For IBM DB2 for z/OS and OS/390, DBArtisan opens the **Preview:Create dialog box**.

Data Import Wizard for IBM DB2 for Linux, Unix, and Windows
DBArtisan offers a simple, six-panel wizard that lets you load data via DBArtisan’s import utility. After defining the data import operation, you have the opportunity to save the job definition for future reuse.

The Import Wizard lets you:

- Create a new Data Load operation.
- Re-use a previously defined Data Load operation.
- Specify data import file.
- Specify file format target datasource.
- Specify import mode.
- Specify target database.
- Specify target schema.
• Specify target table or view.
• Specify the method for selecting columns in a table and select the columns into which the table or view data should be imported.
• Specify import LOBs from separate files, and if so, then specify the location for the message file.
• Define the number of rows to skip in the data file import if the operation restarts.
• Define the number of rows imported before committing to the database.
• Specify the use of Compound SQL statements.
• Save the operation for re-use at a later time.
• Preview the definition of your data unload operation.

Starting the Import Wizard
1. On the **Datasource Explorer**, select the database object.
   DBArtisan displays the database objects in the right pane of the Explorer Window.
2. On the **Utilities** menu, click **Data Load**.
   DBArtisan opens the Import Wizard.

**Import Wizard - Panel 1**
The table below describes the options and functionality on the first panel of the Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you wish to initiate a new import operation or ‘play back’ a previously saved operation?</td>
<td>New Data Load Operation - Select to define a new data load operation. Previous Data Load Operation - Select to open a previously defined schema extraction operation. In the Data Load Definition File Name box, type the location and name of the job file or click the browse button.</td>
</tr>
</tbody>
</table>

For more information, see **Import Wizard**.

**Import Wizard - Panel 2**
The table below describes the options and functionality on the second panel of the Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What file has the data you wish to import?</td>
<td>Lets you type the location and name of the file or click the browse button.</td>
</tr>
<tr>
<td>Specify the format of the file to be imported</td>
<td>Lets you click the list and click the file format. Options Button - Click if you want to set the ASCII file type options for a delimited option. Opens the ASCII File Type Options dialog box.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>What mode would you like to use for this import operation?</td>
<td>INSERT - The target table must exist and imported rows are inserted into the table without affecting existing rows in the table. INSERT_UPDATE - The target table must exist and have primary key constraint defined on it. Imported rows that match the existing row of data on which the primary key is defined are updated. New rows that do not match the primary key are inserted into the table. REPLACE - The target table must exist and all data is replaced with the contents of the import data file. CREATE - The table defined in the import data file is created and rows of data are inserted into the newly created table. REPLACE_CREATE - If the target table exists, the data will be replaced with the data contained in the import data file and the table definition remains unchanged. If the target table does not exist, the table defined in the import data file is created and rows of data are inserted into the newly created table. You must use a *.IXF file to use this mode.</td>
</tr>
<tr>
<td>What is the target object into which the data will be imported?</td>
<td>Database - Lets you select the database. Schema - Lets you select the schema. Object - Lets you select the table or view.</td>
</tr>
</tbody>
</table>

For more information, see [Import Wizard](#).

**Import Wizard - Panel 3**

The table below describes the options and functionality on the third panel of the Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What method should be used for selecting columns for the export file?</td>
<td>Lets you specify a method: Default, Position, Location, or Names.</td>
</tr>
<tr>
<td>Indicate the columns into which data should be imported, specifying the relevant column selection attributes</td>
<td>Lets you select the columns. Click Edit to open the Column Position Attributes dialog box.</td>
</tr>
</tbody>
</table>

For more information, see [Import Wizard](#).

**Column Position Attributes**

The table below describes the options and functionality on the Column Position Attributes dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include</td>
<td>Select to include the column.</td>
</tr>
<tr>
<td>Position</td>
<td>Lets you type the position.</td>
</tr>
</tbody>
</table>

For more information, see [Import Wizard](#).
Import Wizard - Panel 4
The table below describes the options and functionality on the fourth panel of the Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Would you like to retrieve large objects (LOB's) from separate files? | Add Button - Click to open the Browse for Folder dialog box. Click the LOB path.  
Edit Button - Click to open the Browse for Folder dialog box.  
Remove - Click to remove the LOB path. |
| Specify a path and name for the message file     | Type the directory location of the message file or click the Browse button to find the file.                                                |

For more information, see Import Wizard.

Import Wizard - Panel 5
The table below describes the options and functionality on the fifth panel of the Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many of the file's records should be skipped before the import begins?</td>
<td>Restart - Lets you type the value of the number of rows that should be skipped before the import operation begins.</td>
</tr>
<tr>
<td>How many records should be imported before committing them to the database?</td>
<td>Commit Records - Lets you type the value of the number of records that should be imported before they are committed to the database.</td>
</tr>
<tr>
<td>Use of Compound SQL may improve performance.</td>
<td>Statements - Lets you type the value of the number, which cannot exceed 100, of the number of SQL statements per compound block.</td>
</tr>
</tbody>
</table>

For more information, see Import Wizard.

Import Wizard - Panel 6
The table below describes the options and functionality on the sixth panel of the Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you wish to re-use this data unload operation?</td>
<td>To re-use the operation, type the location and name of the data load definition file or click the browse button. DBArtisan gives the operation a default extension of *.LOAD.</td>
</tr>
</tbody>
</table>

For more information, see Import Wizard.

Import Wizard - Panel 7
Review the data load operation description to verify its accuracy.

1. To make any changes, click Back to scroll back to the appropriate panels of the wizard to make your corrections.
2. If you are satisfied with the data load operation, click Execute.

For IBM DB2 for z/OS and OS/390, DBArtisan opens the Preview:Create dialog box.

For more information, see Import Wizard.
Load Data Utility Wizard for DB2 OS390

The DBArtisan Load Data Utility for DB2 z/OS enables you to load records into one or more tables in a tablespace without creating an intermediate exported file. If the tablespace already contains data, you can opt to add the new data to the existing data or to replace it. The data that’s loaded is subject to the same editing or validation routines currently associated with the table and any field procedure associated with particular table columns. As output, you get a loaded tablespace or partition, a file of records that have been discarded, and an error report if you specify the Enforce Constraints option or if unique indexes were part of the load operation.

To use this utility, you need a sufficient set of privileges.

One of the following conditions must be met:

- You must own the table
- Have load privileges for the database
- DBAdmin or DBCTRL authority for the database.
- SYSCTRL or SYSADM authority.

For more complete information than can be presented here, refer to the IBM Utility Guide and Reference. You can find it here: [http://www-306.ibm.com/software/data/db2/zos/v8books.html](http://www-306.ibm.com/software/data/db2/zos/v8books.html)

**To Open the Load Data Utility Wizard**

1. Connect to a DB2 OS390 database on the Explorer tree.
2. Expand the database and click the Tablespace subnode.
3. In the right window, select a tablespace and click Load Data from the Command drop-down menu.

   OR

   On the Explorer node, click Utility Execution, and then click Load in the right pane.

**Creating a Load Data Utility Operation**

The DBArtisan Load Data Utility for DB2 z/OS enables you to load records into one or more tables in a tablespace without creating an intermediate exported file. If the tablespace already contains data, you can opt to add the new data to the existing data or to replace it.

The following sections describe fields you may encounter as you move through the Load Data Utility Wizard:

**Initial Settings**

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want the utility to be restartable?</td>
<td>The utility restart information will be included if a recover operation is required.</td>
</tr>
<tr>
<td>Tape Stacking</td>
<td>Select this if you are backing up this operation onto physical media. Click Properties to specify the particulars for the dataset as it’s recorded by opening the Load Utility - Tape Stacking Dialog Box.</td>
</tr>
<tr>
<td>Do you want to specify a cursor?</td>
<td>If you want to specify a cursor you need to type the name of the cursor for the input data set in the field below. No is the default. Any cursor name you specify cannot exceed eight characters.</td>
</tr>
</tbody>
</table>
### Resumption/ Replace/Statistics

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resume</td>
<td>If you say No, records load into an empty tablespace. This is the default. If you say Yes, records are loaded into a non-empty tablespace. Loading begins at the current end of data in the tablespace. If you specify the sharelevel as change, the before and after triggers for each row that's loaded are activated. The default sharelevel is none--this specifies that applications cannot concurrently access the tablespace during the load operation. When you select Yes, the remaining options on this panel are unavailable.</td>
</tr>
<tr>
<td>Replace</td>
<td>Here you indicate whether the tablespace and its indexes need to be resert to empty before records load. If you select Yes, you are telling the load operation to serialize at the tablespace level. No is the default.</td>
</tr>
<tr>
<td>Would you like to specify the statistics option?</td>
<td>Here you indicate whether you want to gather statistics for the tablespace. No is the default. If you select Yes, statistics are gathered on a base tablespace, but not an LOB tablespace. If you gather statistics on encrypted data, the results may not be useful.</td>
</tr>
<tr>
<td>Do you want to output message to SYSPRINT?</td>
<td>This is the output dataset for messages and printed output. Default is the default.</td>
</tr>
<tr>
<td>Do you want to force aggregation or rollup processing to be done even though some parts do not contain data?</td>
<td>Yes, No, or default. Default is the default. Here you specify whether you want statistics to aggregate or rollup when RUNSTATS is executed even in the event that some parts are empty.</td>
</tr>
</tbody>
</table>

### Select Tables (for running statistics)

Here you are presented with a dialog box where you identify the table you want to run statistics on. Click ADD to open the Table/Object Selector dialog box. Click Set Properties to open the Select Column for Table dialog box. Click Delete to remove a table you selected from the run statistics operation.

### Select Tables to Load Data Into

Here you are presented with the dialog box where you identify the tables you want to load the data into. Click ADD to select one or more tables. Click Set Properties to open the Into Table Specification Dialog Box.

### Select Indexes

Here you are presented with a dialog box where you identify the indexes you want to run statistics on. Click ADD to open the Index/Object Selector dialog box. Click Set Properties to open the Set Correlation option dialog box where you set the KeyCard and Frequent Item options. When you ADD Frequent Options, the Add Frequent Options dialog box opens which identifies the number of columns and the count.
### Update/History Statistics

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
</table>
| Update Statistics    | **All**: Indicates that all collect statistics are to be updated in the catalog. **All** is the default.  
**Access Path**: Indicates that updates are to be made only to the catalog table columns that provide statistics used for selecting the access path.  
**Space**: Indicates that updates are to be made only to the catalog table columns that provide statistics that help assess the status of a particular tablespace or index.  
**None**: Indicates no catalog tables will be updated. Note: This is enabled only when the Report option is set to Yes. |
| History Statistics   | This records all catalog table inserts or updates to the catalog history tables.  
**Default**: The value specified on  
**All**: Indicates all collected statistics are to be updated in the catalog history tables.  
**Access Path**: Indicates that updates are to be made only to the catalog history table columns that provide statistics used for selecting the access path.  
**Space**: Specifies that only space-related statistics are to be updated in catalog history tables.  
**None**: Specifies that no catalog history tables are updated with the statistics that have been collected. |

### Dictionary/Log

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep dictionary</td>
<td>If you specify <strong>No</strong>, the load operation is prevented from building a new compression dictionary, thereby saving you the costs related to such a build operation. <strong>No</strong> is the default.</td>
</tr>
<tr>
<td>Reuse</td>
<td>When used with Replace (on an earlier panel of the wizard), this option indicates that the load operation should reset and reuse the datasets without deleting or refining them. <strong>No</strong> is the default.</td>
</tr>
<tr>
<td>Log</td>
<td>The <strong>Default</strong> specifies normal logging during the load operation and is equivalent to <strong>Yes</strong>. All loaded records will be logged. <strong>No</strong> specifies no logging of data in the load operation and disables the copy-pending option below.</td>
</tr>
<tr>
<td>LOAD is not to set the tablespace in the COPY-pending status</td>
<td>This, the NOCOPYPEND option will not turn off any copy pending options set prior the load operation. <strong>NOTE</strong>: Specify this option only if the data in this table can be recreated by another load operation if, for some reason, the data is lost. You may not be able to recover the tablespace if you don’t take an image copy after the load operation. <strong>No</strong> is the default.</td>
</tr>
</tbody>
</table>

### Enforce/Specify

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enforce</td>
<td>Specifies whether the load operation should enforce check and referential constraints. <strong>Constraints</strong> indicates constraints will be enforced. If the operation detects a constraint violation, the errant row is deleted and identified by a message. If you elect this option, and there are referential constraints, you must define sort input and output datasets. <strong>Default</strong> enforces the constraints. <strong>No</strong> specifies constraints are not enforced. It also puts the tablespace in the Check-pending status if at least one referential or check constraint exists.</td>
</tr>
</tbody>
</table>
Would you like to specify the Work DDN?

If you elected to enforce constraints, you must specify the work DDN. Here you establish DD names for the temporary sort input and output files. The default input is SYSUT1 and output is SORTOUT.

Would you like to specify the SORTKEYS?

This specifies that the index keys are to be sorted in parallel during the sort build phase to improve performance. If you want, you can specify an integer value to provide an estimate of the number of index keys that are to be sorted. The default is 0 if the input is on tape or a cursor.

Format

This selection is for the format of the input data. **Unload** specifies that the input record format is compatible with DB2 unload format. **SQL/DS** indicates the input record format is compatible with SQL/DS format.

### Input/CCSID

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to specify the Work DDN?</td>
<td>If you elected to enforce constraints, you must specify the work DDN. Here you establish DD names for the temporary sort input and output files. The default input is SYSUT1 and output is SORTOUT.</td>
</tr>
<tr>
<td>Would you like to specify the SORTKEYS?</td>
<td>This specifies that the index keys are to be sorted in parallel during the sort build phase to improve performance. If you want, you can specify an integer value to provide an estimate of the number of index keys that are to be sorted. The default is 0 if the input is on tape or a cursor.</td>
</tr>
<tr>
<td>Format</td>
<td>This selection is for the format of the input data. <strong>Unload</strong> specifies that the input record format is compatible with DB2 unload format. <strong>SQL/DS</strong> indicates the input record format is compatible with SQL/DS format.</td>
</tr>
</tbody>
</table>

### Input/CCSID

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input data file type</td>
<td>Select between EBCDIC, ASCII, or Unicode.</td>
</tr>
<tr>
<td>CCSID(s)</td>
<td>These coded character set identifiers for the input file generally correspond to the input data file type. If that's not the case, the input data will be converted to the table CCSIDs before being loaded.</td>
</tr>
<tr>
<td>Do not accept substitution characters in a string</td>
<td>Specifies that the load operation should not accept substitution characters in the string. Use substitutions when a string is being converted from one data type to another. If you do not accept substitutions and the load operation determines that a substitution appears in the string because of a conversion, if discard processing is active, the record will be moved to the discard file. If discard processing is inactive, the load operation will terminate.</td>
</tr>
<tr>
<td>Float</td>
<td>S390 specifies that the load operation should expect floating point numbers are in the System/390 hexadecimal floating point format. IEEE specifies that the load operation should expect that floating point number are in IEEE binary floating point format. If you specify IEEE, the binary format floats will be converted to hexadecimal floating points. If there’s a conversion error, the record goes in the discard pile.</td>
</tr>
<tr>
<td>Specify the error DDN</td>
<td>Here you specify the DD statement or template for a work dataset that’s used during error processing. <strong>SYSERR</strong> is the default.</td>
</tr>
</tbody>
</table>

### Map DDN/Discard DDN

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to specify the Map DDN?</td>
<td>Specifies the DD statement or template for a work data set to be used in error processing. This is used to correlate the identifier of a table row with the input record that’s causing the error. <strong>SYSMAP</strong> is the default.</td>
</tr>
<tr>
<td>Would you like to specify the Discard DDN?</td>
<td>Specifies the DD statement or template for a discard dataset that will hold copies for records that aren’t loaded in the operation (e.g., a record with a conversion error). <strong>SYSDISC</strong> is the default.</td>
</tr>
<tr>
<td>Would you like to specify the maximum number of source records to be written on the discard data set?</td>
<td>Specify an integer with a value between 0 and 2146483647. If the maximum value is reached, the load operation abnormally terminates. When you enter 0, you are specifying that you do not want to set a maximum value and the entire input data set can be discarded.</td>
</tr>
</tbody>
</table>
Specify Sort Number/CONTINUE-IF

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to specify the sort number?</td>
<td>Here you indicate, if available, the device type for temporary datasets that are to be dynamically allocated by DFSORT.</td>
</tr>
<tr>
<td>Would you like to specify a CONTINUE-IF condition?</td>
<td>Yes indicates that you want to treat each input record as a portion of a larger record. If a record is true, the next record is concatenated with it before it is loaded.</td>
</tr>
<tr>
<td>Start/Finish</td>
<td>Specifies the column numbers of the first and last columns.</td>
</tr>
<tr>
<td>Hexidecimal</td>
<td>The byte-string value in the continuation field indicates that the next input record is a continuation of the current load record. Records with this byte-string value are concatenated until the value in the continuation field changes.</td>
</tr>
<tr>
<td>Condition</td>
<td>After the CONTINUE-IF completes, write a condition as X’byte-string’ or ‘character-string’.</td>
</tr>
</tbody>
</table>

Into Table Specification Dialog Box
At least one table INTO STATEMENT is required for each table that is to be loaded. Each Into Table Specification:

- Identifies the table that is to be loaded.
- Describes fields within the input record.
- Defines the format of the input dataset.

**NOTE:** All tables that are specified must belong to the same tablespace.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page 1:</td>
<td></td>
</tr>
<tr>
<td>Skip fields in the input data set that do not correspond to columns in the target table</td>
<td>Specifies whether or not the load operation should skip fields in the input dataset that do not correspond to tables in the target table.</td>
</tr>
<tr>
<td>Partition</td>
<td>An integer value that applies to the number of the partition where records are to be loaded. This is an option that is enabled only for partitioned tablespaces.</td>
</tr>
<tr>
<td>Would you like to preformat?</td>
<td>Specifies that any remaining pages should be preformatted up to the high-allocated RBA in the partition and its corresponding partitioning index space. Preformatting happens after the data is loaded and the indexes are built.</td>
</tr>
<tr>
<td>Load records into a non-empty partition</td>
<td>No loads records into an empty partition. No is the default. Yes loads records into a non-empty partition. If the partition is empty, you get a warning message, but the partition is loaded anyway.</td>
</tr>
<tr>
<td>Replace only the contents of the partition specified above, rather than the entire tablespace.</td>
<td>NOTE: You cannot use Load Replace with the partition integer replace option. You can either replace an entire tablespace using the load replace option or the single partition option here.</td>
</tr>
</tbody>
</table>
### Required Information

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reset and reuse DB2-managed data sets</strong></td>
</tr>
<tr>
<td>When used with the replace option above, this specifies that the load operation should reset and reuse the DB/2-managed datasets without deleting and redefining them.</td>
</tr>
<tr>
<td><strong>Would you like to keep the dictionary?</strong></td>
</tr>
<tr>
<td>This is an option only if a dictionary exists and the partition being loaded has the COMPRESS YES attribute. If the partition has the COMPRESS YES attribute but no dictionary exists, a dictionary is built and you get an error message.</td>
</tr>
<tr>
<td><strong>Would you like to specify a cursor?</strong></td>
</tr>
<tr>
<td>If desired, specify a cursor for the input dataset.</td>
</tr>
<tr>
<td><strong>Select statement for cursor</strong></td>
</tr>
<tr>
<td><strong>In DNN</strong></td>
</tr>
<tr>
<td>Specifies the data definition statement or template for the input dataset for the partition. The default is SYSREC.</td>
</tr>
<tr>
<td><strong>Discard DNN</strong></td>
</tr>
<tr>
<td>Specifies the data definition statement or template for the partition. This dataset also holds copies of records that were loaded and removed. It must be a sequential data set.</td>
</tr>
</tbody>
</table>

### Page 2:

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Do you want to set criteria for which records are loaded?</strong></td>
</tr>
<tr>
<td>If you do not want to specify load criteria, all records in the input dataset are loaded, except for data that is beyond the range of the specified partition.</td>
</tr>
<tr>
<td><strong>Do you want to use field selection criteria?</strong></td>
</tr>
<tr>
<td>Yes or No</td>
</tr>
</tbody>
</table>

### SQL/DS

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table Owner</strong> Select a table owner from the drop-down list.</td>
</tr>
<tr>
<td><strong>Table</strong> This is the name of the table that the SQL/DS table is loaded into.</td>
</tr>
<tr>
<td><strong>Field selection Criteria</strong> The criteria describes a field and a character constant. A field must contain a character or graphic string. (When compared to a string constant, no datatype conversions are performed.)</td>
</tr>
<tr>
<td>If the field and the constant are not the same length, the shorter of the two is padded with blanks before a comparison is made.</td>
</tr>
<tr>
<td><strong>Fields</strong> Click Add to open the Field Specification Dialog Box.</td>
</tr>
</tbody>
</table>

### Field Specification Dialog Box

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field Name</strong> The name of a field defined by a field specification. This can be any name you choose.</td>
</tr>
<tr>
<td><strong>Would you like to specify a position?</strong> Indicates where a field appears in the assembled load record. Start/End are the locations of the first and last field records. The first record is column 1.</td>
</tr>
<tr>
<td><strong>Datatype</strong> Pick the appropriate datatype from the drop-down list.</td>
</tr>
<tr>
<td><strong>Length/Scale</strong> Length is the length in bytes of the input field. Scale is the number of digits to the right of the decimal point and must be greater than or equal to 0. 0 is the default.</td>
</tr>
<tr>
<td><strong>Would you like to add a condition that causes the db2 column to be loaded with NULL or it's default value?</strong> NULLIF/CONDITION: Write a condition that that causes the DB/2 column to be loaded with NULL. DEFAULTIF/CONDITION: Write a condition that causes the DB2 column to be loaded with its default value. No</td>
</tr>
</tbody>
</table>
Set Dataset Info Dialog Box

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset Name</td>
<td>The name of the input dataset from which the table data should be loaded.</td>
</tr>
<tr>
<td>Template Name</td>
<td></td>
</tr>
<tr>
<td><strong>SMS</strong>: Management class</td>
<td>The management class name may not exceed 8 characters. This enables the</td>
</tr>
<tr>
<td></td>
<td>dataset to be catalogued. MGMTCLAS name.</td>
</tr>
<tr>
<td>Storage class</td>
<td>The storage class name must be valid and cannot exceed 8 characters.</td>
</tr>
<tr>
<td></td>
<td>STORCLAS name.</td>
</tr>
<tr>
<td>Data class</td>
<td>The data class name cannot exceed 8 characters. DATACLAS name.</td>
</tr>
<tr>
<td>Unit: Device Type</td>
<td>Specifies the device type or group name for the data set.</td>
</tr>
<tr>
<td>Count</td>
<td></td>
</tr>
<tr>
<td><strong>Volume</strong>: Private</td>
<td>A communications connection that's specific to DB/2.</td>
</tr>
<tr>
<td>Sequence Number</td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td></td>
</tr>
<tr>
<td>Serial no(s)</td>
<td>The serial number of the volume on which a new temporary or permanent</td>
</tr>
<tr>
<td></td>
<td>data set resides.</td>
</tr>
<tr>
<td><strong>Space</strong>: Unit</td>
<td></td>
</tr>
<tr>
<td>Primary Allocation</td>
<td></td>
</tr>
<tr>
<td>Secondary Allocation</td>
<td></td>
</tr>
<tr>
<td>Release</td>
<td></td>
</tr>
<tr>
<td><strong>Label</strong>: Sequence Number</td>
<td></td>
</tr>
<tr>
<td>Retention period</td>
<td></td>
</tr>
<tr>
<td>Expiration date</td>
<td></td>
</tr>
</tbody>
</table>

Load Utility - Tape Stacking Dialog Box

Tape stacking allows you to make four copies of the same data set simultaneously. DB2 OS390’s Copy to Copy utility allows you to make multiple image copies as well.

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Site Primary Copy</td>
<td>Name the data set where you want the output image copy to go at the local primary site. This relates to the copyddn1 parameter in the stored procedure.</td>
</tr>
<tr>
<td>Local Site Backup Copy</td>
<td>Name the data set where you want the output image copy to go at the local backup site. This relates to the copyddn2 parameter in the stored procedure.</td>
</tr>
<tr>
<td>Recovery Site Primary Copy</td>
<td>Name the data set where you want the output image copy to go at the recovery primary site. This relates to the recoveryddn1 parameter in the stored procedure.</td>
</tr>
<tr>
<td>Recovery Site Backup Copy</td>
<td>Identify the image set where you want the image copy to go at the recovery backup site. This relates to the recoveryddn2 parameter in the stored procedure.</td>
</tr>
<tr>
<td>In DDN</td>
<td>Identify the input data set that contains the table to be loaded.</td>
</tr>
<tr>
<td>Discard DDN</td>
<td>Identify the work data set that will hold copies of records not included in the load operation.</td>
</tr>
</tbody>
</table>
Data Unload Wizard for Microsoft SQL Server and Sybase ASE

DBArtisan offers a simple, five-panel wizard that can export table data via Microsoft SQL Server's BCP (bulk copy program) utility or Sybase ASE's BCP (bulk copy program) utility. After defining the data export operation, you can save the job definition for future re-use.

The Data Unload Wizard lets you:

- Create a new data unload operation.
- Reuse a previously defined data unload operation.
- Specify source server, data unload file format, and data unload specification.
- Select the tables that you want to unload.
- Save the job definition for future re-use.

For more information, see Data Unload Wizard - Panel 1.

Data Unload Wizard - Panel 1

The table below describes the options and functionality on the first panel of the Data Unload Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you wish to initiate a new data unload operation or ‘play back’ a previously saved operation?</td>
<td>New Data Unload Operation - Select to define a new data unload operation. ‘Playback’ Previous Data Unload Operation - Select to use a previously defined schema extraction operation. Then in the Data Unload Definition File Name box type the location and name of the job file or click the browse button.</td>
</tr>
</tbody>
</table>

For more information, see Unloading Data.

Data Unload Wizard - Panel 2

The table below describes the options and functionality on the second panel of the Data Unload Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In which directory should the unloaded files be placed?</td>
<td>Lets you specify the directory by typing the location and name of the directory or find the directory by clicking the browse button.</td>
</tr>
<tr>
<td>Which database has the data you wish to unload?</td>
<td>Lets you click the list, and then select the server. Then click the list, and then click the database.</td>
</tr>
<tr>
<td>What character should delimit columns in the unloaded data files?</td>
<td>Lets you click the list, and then select the character.</td>
</tr>
<tr>
<td>Use BCP Utility</td>
<td>Select if you want to use the Microsoft SQL Server's BCP (bulk copy program) utility.</td>
</tr>
</tbody>
</table>

For more information, see Unloading Data.
Data Unload Wizard - Panel 3
The table below describes the options and functionality on the third panel of the Data Unload Wizard for Microsoft SQL Server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>Lets you select the target tables to unload.</td>
</tr>
<tr>
<td>Upd Stats button</td>
<td>Select the target tables and click button to update the table row counts.</td>
</tr>
<tr>
<td>Select All button</td>
<td>Click to select all tables.</td>
</tr>
<tr>
<td>Unselect All button</td>
<td>Click to unselect all tables.</td>
</tr>
</tbody>
</table>

Data Unload Wizard - Panel 4
The table below describes the options and functionality on the fourth panel of the Data Load Wizard for Sybase ASE:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Terminator</td>
<td>Lets you enter a value for this common BCP option.</td>
</tr>
<tr>
<td>Packet Size</td>
<td>Lets you enter a value for this common BCP option.</td>
</tr>
<tr>
<td>Text/Image Size</td>
<td>By default, Sybase only copies the first 32K of data in a text or image field. To override default size, type a new size.</td>
</tr>
<tr>
<td>Client Char Set</td>
<td>Lets you enter a value for this common BCP option.</td>
</tr>
</tbody>
</table>

For more information, see Unloading Data.

Data Unload Wizard - Panel 5
Review the data unload operation description to verify its accuracy.

1. If you need to make any changes, click Back to scroll back to the appropriate panels of the wizard to make your corrections.
2. If you are satisfied with the data unload operation, click Execute.

   DBArtisan starts the Data Unload operation.

To schedule the data unload operation, click Schedule.

   DBArtisan opens your scheduling application, Embarcadero Job Scheduler or Microsoft Task Scheduler.

For more information, see Unloading Data.
Data Load Wizard for Microsoft SQL Server and Sybase ASE

DBArtisan offers a simple, six-step wizard that can load data via Microsoft SQL Server's BCP (bulk copy program) utility or via Sybase ASE's BCP (bulk copy program) utility. After defining the data import operation, you have the opportunity to save the job definition for future re-use.

The Data Load Wizard lets you:

- Create a new data load operation.
- Re-use a previously defined data load operation.
- Specify data import file.
- Specify target database.
- Specify target schema.
- Specify target table or view.
- Specify the method for selecting columns in a table and select the columns into which the table or view data should be imported.
- Specify import LOBs from separate files, and if so, then specify the location for the message file.
- Define the number of rows to skip in the data file import if the operation restarts.
- Define the number of rows imported before committing to the database.
- Specify the use of Compound SQL statements.
- Save the operation for re-use at a later time.
- Preview the definition of your data unload operation.

For more information, see Data Load Wizard - Panel 1.

Data Load Wizard - Panel 1

The table below describes the options and functionality on the first panel of the Data Load Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you wish to initiate a new data load operation or 'play back' a previously saved operation?</td>
<td>New Data Load Operation - Select to define a new data load operation. Previous Data Load Operation - Select to open a previously defined schema extraction operation. In the Data Load Definition File Name box, type the location and name of the job file or click the browse button.</td>
</tr>
</tbody>
</table>

For more information, see Loading Data.

Data Load Wizard - Panel 2

The table below describes the options and functionality on the second panel of the Data Load Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which database do you want to load the data into?</td>
<td>Lets you click the list, and then select the server. Then click the list and select the database.</td>
</tr>
</tbody>
</table>
The table below describes the options and functionality on the third panel of the Data Load Wizard for Microsoft SQL Server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which rows do you want to import?</td>
<td>All Rows</td>
</tr>
<tr>
<td></td>
<td>Range - Lets you type the range in the First Row and Last Row boxes.</td>
</tr>
<tr>
<td>What restrictions do you want to place on the import?</td>
<td>Max Errors or Batch Size.</td>
</tr>
</tbody>
</table>

For more information, see [Loading Data](#).

The table below describes the options and functionality on the fourth panel of the Data Load Wizard for Microsoft SQL Server:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data File</td>
<td>Lets you type the directory location of the data file or click the browse button to find the file.</td>
</tr>
<tr>
<td>Format</td>
<td>To use a format file, type the directory location of the format file or click the browse button to find the file.</td>
</tr>
<tr>
<td>Error File</td>
<td>To output errors to an error file, type the directory location of the format file or click the browse button to find the file.</td>
</tr>
<tr>
<td>What character delimited the columns in the datafile?</td>
<td>Lets you click the list, and then click the character.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the fourth panel of the Data Load Wizard for Sybase ASE:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Terminator</td>
<td>Lets you enter a value for this common BCP option.</td>
</tr>
<tr>
<td>Packet Size</td>
<td>Lets you enter a value for this common BCP option.</td>
</tr>
<tr>
<td>Text/Image Size</td>
<td>By default, Sybase only copies the first 32K of data in a text or image field. To override default size, type a new size.</td>
</tr>
<tr>
<td>Client Char Set</td>
<td>Lets you enter a value for this common BCP option.</td>
</tr>
<tr>
<td>Max Connections</td>
<td>Lets you specify the maximum connection of connections in the connection pool.</td>
</tr>
<tr>
<td>Show-fi</td>
<td>Provides the option to add columns for each functional index.</td>
</tr>
<tr>
<td>Hide-vcc</td>
<td>Provides the option to hide virtual computed columns</td>
</tr>
</tbody>
</table>
For more information, see Loading Data.

**Data Load Wizard - Panel 5**

The table below describes the options and functionality on the fifth panel of the Data Load Wizard for Microsoft SQL Server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you wish to re-use this data load operation?</td>
<td>To re-use the operation, type the location and name of the data load definition file or click the browse button to find the file. DBArtisan gives the operation a default extension of *.LOAD.</td>
</tr>
<tr>
<td>Do you wish to DROP the index(es) on the target table BEFORE copying data?</td>
<td>Drop Index(es) Before Loading of Data - Select to drop the indexes on the target table before copying data. Recreate Index(es) After Loading of Data - Select to recreate these indexes after the data loads.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the fifth panel of the Data Load Wizard for Sybase ASE.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data File</td>
<td>Lets you type the directory location of the data file or click the browse button to find the file.</td>
</tr>
<tr>
<td>Format</td>
<td>To use a format file, type the directory location of the format file or click the browse button to find the file.</td>
</tr>
<tr>
<td>Error File</td>
<td>To output errors to an error file, type the directory location of the format file or click the browse button to find the file.</td>
</tr>
<tr>
<td>What character delimited the columns in the datafile?</td>
<td>Lets you click the list, and then click the character.</td>
</tr>
</tbody>
</table>

For more information, see Loading Data.

**Data Load Wizard - Panel 6**

For Microsoft SQL Server, the sixth panel of the Data Load Wizard lets you preview the definition of your data load operation.

1. If you need to make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections.

2. If you are satisfied with the data load operation, click **Execute**.
   DBArtisan starts the Data Load operation.

3. To schedule the data load operation, click **Schedule**.
   DBArtisan opens your scheduling application, Embarcadero Job Scheduler or Microsoft Task Scheduler.
The table below describes the options and functionality on the sixth panel of the Data Load Wizard for Sybase.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you wish to re-use this data load operation?</td>
<td>To re-use the operation, type the location and name of the data load definition file or click the browse button to find the file. DBArtisan gives the operation a default extension of *.LOAD.</td>
</tr>
</tbody>
</table>
| Do you wish to DROP the index(es) on the target table BEFORE copying data? | Drop Index(es) Before Loading of Data - Select to drop the indexes on the target table before copying data.  
Recreate Index(es) After Loading of Data - Select to recreate these indexes after the data loads. |

**Data Load Wizard - Panel 7**

Review the data load operation description to verify its accuracy.

1. If you need to make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections.
2. If you are satisfied with the data load operation, click **Execute**. DBArtisan starts the Data Load operation.
3. To schedule the data load operation, click **Schedule**. DBArtisan opens your scheduling application, [Embarcadero Job Scheduler](https://www.embarcadero.com/jobscheduler) or [Microsoft Task Scheduler](https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/schtasks)

**Database Manager**

DBArtisan lets you view and manage your databases through Database Manager. The Database Manager lets you:

- Add, modify, and delete database files.
- Add, modify, and delete transaction logfiles.
- Add or delete database options.
- View data space and transaction log usage.
- View DDL.

**Opening the Database Manager**

1. On the **Datasource Explorer**, select the **Databases** node.
   DBArtisan displays the databases in the right pane of the Explorer window.
2. In the right pane of the Explorer window, click the database link(s).
3. On the **Utilities** menu, click **Database Manager**.
Database Backup Wizard

Implementing a plan to back up critical databases is perhaps the most important job for any database administrator. Without a sound disaster recovery plan, your company has no safeguards against hardware and software failures.

The Database Backup Wizard lets you backup your databases without knowing the underlying syntax. It provides support for backing up databases as well as transaction logs, including commands to stripe the backup across multiple devices.

The Database Backup Wizard lets you:

- Create a new dump operation.
- Reuse a previously defined operation.
- Specify the target server, database, and type of backup operation.
- Specify backup device type.
- Specify how to report the progress of the backup operation.

You can back up the Microsoft SQL Server while the database is online and in use. However, the backup will fail if you are performing one of the following operations while attempting the back up:

- Creating indexes.
- Performing non-logged operations.
- Shrinking the database.
- Creating or deleting database files.

**NOTE:** Backups created with Microsoft SQL Server version 7 cannot be restored to an earlier version of SQL Server.

Creating or Editing a Backup Operation

DBArtisan lets you create different backup operations using the Database Backup Wizard. After you have successfully created a backup operation, you can reopen and edit it using the same wizard and make the changes you want.

**To Open the Database Backup Wizard**

1. On the **Datasource Explorer**, select the **Databases** node.
   
   DBArtisan displays the databases in the right pane of the Explorer window.

2. Select the target database.

3. On the **Utilities** menu, click **Database Backup**.

In the following sections, you’ll find all the fields you’ll encounter as you create or edit a backup operation:

**Job Category Wizard**

**Database Backup Wizard**
Job Category Wizard

The Job Category Wizard panels appear ONLY when you are performing a backup operation for a MS SQL datasource. The job categories are relevant to the Backup Analyst and are described below:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Backup</td>
<td>Any backup job which is a full database, differential database, file and file group, transaction log or table backup which does not include the compression or encryption options.</td>
</tr>
<tr>
<td>Advanced Backup</td>
<td>Any backup job which is a full database, differential database, file and file group, transaction log or table backup which includes at least one of the compression or encryption options.</td>
</tr>
<tr>
<td>Recovery</td>
<td>Any recovery job created by Backup Analyst.</td>
</tr>
</tbody>
</table>
| Would you like to create the 16 popular SQL Server Alerts for common Backup and Recovery Errors? | Yes - Opens the Alerts wizard.  
No - Opens the Database Backup Wizard.                                                                                                                 |

Database Backup Wizard

The first panel of the Database Backup Wizard welcomes you to the Backup Wizard.

Subsequent panels of the wizard are as follows:

- Playback or New Operation
- Database Selection for SQL Server
- Backup Type for SQL Server
- Backup Type for SQL Server
- Backup Style
- Backup Location for MS SQL
- Sybase Server Selection
- Backup Device and Location for Sybase
- Backup and Media Information
- Specify Device-handling Options for Sybase
- Backup Options
- Backup Operation Summary
- Backup Operation Execution Selection
Playback or New Operation

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Dump Operation</td>
<td>This initiates the process of creating a new backup operation.</td>
</tr>
<tr>
<td>Previous Dump Operation</td>
<td>If you want to replay a backup operation, when you select the radio button, the browse function is enabled so you can find the .DBDump file you want to reuse.</td>
</tr>
</tbody>
</table>

Database Selection for SQL Server

Here you select the database you want to backup from a drop-down list and name the backup operation. You can keep the default name for the operation or replace it with one you type. The job description is optional and you can accept the default or write another.

Sybase Server Selection

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are the source Sybase server and database?</td>
<td>Select a server from the drop-down list and then the appropriate datasource.</td>
</tr>
<tr>
<td>What do you want to dump?</td>
<td>Database - Backs up the entire database. The size of the database is listed in parentheses.</td>
</tr>
<tr>
<td></td>
<td>Transaction Log - Backs up the inactive portion of the transaction log. The size of the transaction log is listed in parentheses.</td>
</tr>
<tr>
<td></td>
<td>Transaction Log With No Log - Removes the inactive portion of the transaction log without making a backup of it. You should only use this command when you run out of room in the database and cannot issue DUMP TRAN WITH TRUNCATE_ONLY.</td>
</tr>
<tr>
<td></td>
<td>Transaction Log With Truncate Only - Removes the inactive portion of the transaction log without making a backup of it. You should use this command prior to backing up a database.</td>
</tr>
<tr>
<td></td>
<td>Transaction Log With No Truncate - Makes it possible to dump a transaction log even if the database is inaccessible. You can use this option when the data portion of your database is damaged and the transaction log resides on a separate device.</td>
</tr>
</tbody>
</table>
Backup Type for SQL Server

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Database</td>
<td>Backs up the entire database. The size of the database is listed in parentheses.</td>
</tr>
<tr>
<td>Differential Database</td>
<td>Records all modifications made to a database after the last database backup, not just the changes made after the last differential database backup.</td>
</tr>
<tr>
<td>File and File Group</td>
<td>Select to specify a file or file group.</td>
</tr>
<tr>
<td>Transaction Log</td>
<td><strong>Full Truncate</strong> - Backs up the inactive portion of the transaction log. The size of the transaction log is listed in parentheses.</td>
</tr>
<tr>
<td></td>
<td><strong>With No Log</strong> - Removes the inactive portion of the transaction log without making a backup of it. You should only use this command when you run out of room in the database and cannot issue DUMP TRAN WITH TRUNCATEONLY.</td>
</tr>
<tr>
<td></td>
<td><strong>Full No Truncate</strong> - Makes it possible to dump a transaction log even if the database is inaccessible. You can use this option when the data portion of your database is damaged and the transaction log resides on a separate device.</td>
</tr>
<tr>
<td></td>
<td><strong>With Truncate Only</strong> - Removes the inactive portion of the transaction log without making a backup of it. You should use this command prior to backing up a database.</td>
</tr>
</tbody>
</table>

Backup Style

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Analyst Backup</td>
<td>Select to use the Backup Analyst for your backup, assuming the Backup Analyst is installed on the database you are backing up. Also lets you compress the backup and apply an encrypted password.</td>
</tr>
<tr>
<td>Standard SQL Server Backup</td>
<td>Select to continue the Database Backup Wizard.</td>
</tr>
</tbody>
</table>

Backup Location for MS SQL

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Device</td>
<td>Depending on what is available to you, you can choose to backup to a disk, floppy, pipe, or tape device.</td>
</tr>
<tr>
<td>Disk Directory and Name</td>
<td>Lets you specify a backup location on your machine or click the browse button and select a location.</td>
</tr>
<tr>
<td>View Contents</td>
<td>Click to open the Backup Media Contents Dialog Box.</td>
</tr>
</tbody>
</table>

Backup Device and Location for Sybase

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What type of dump device do you want to use?</td>
<td>Choose among tape device, disk device, or file. When you select file, you need to type the file’s name.</td>
</tr>
<tr>
<td>Where should messages about the dump operation be sent?</td>
<td>Your options are the Operator Console, Client, or Default.</td>
</tr>
<tr>
<td>Dump file</td>
<td>It's up to you to specify the name of the dump file.</td>
</tr>
</tbody>
</table>

Backup and Media Information
Here, for MS SQL, you specify the backup set name, backup description, backup set password, media name, media description, and media name password.
For Sybase, select the dump device you want to use, if applicable. Clicking **Attribute** opens the Backup Device editor. For details, see [Using the Backup Devices/Dump Devices editor (Microsoft SQL Server)](#).

### Specify Device-handling Options for Sybase

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overwrite Contents</td>
<td>Select to overwrite the contents of a tape, and in the Retain Days box, type the number of days to retain the contents of the tape.</td>
</tr>
<tr>
<td>Compress</td>
<td>Lets you compress dumps and logs using different levels of compression. The valid levels are 1-9, with level 9 compressing the file to its smallest size.</td>
</tr>
<tr>
<td>Dismount After Completion</td>
<td>Select to dismount the tape after backup.</td>
</tr>
<tr>
<td>Rewind and Unload After Completion</td>
<td>Select rewind the tape and unload after the backup.</td>
</tr>
</tbody>
</table>

### Backup Options

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expire Date</td>
<td>The date when the backup set expires and can be overwritten. If supplied as a variable (@takeover), this date is specified as either a string constant (@takeover = date), as a variable of character string data type (except for the text or text data types), a small datetime, or datetime variable, and must follow the configured system datetime format.</td>
</tr>
<tr>
<td>Retain Days</td>
<td>The number of days that must elapse before this backup media set can be overwritten. If supplied as a variable (@days_var), it must be specified as an integer.</td>
</tr>
<tr>
<td>Skip backup set expiration and name checking before overwriting</td>
<td>Microsoft SQL Server will skip the set expiration and name checking before overwriting.</td>
</tr>
<tr>
<td>Overwrite all backup sets on the device</td>
<td>Microsoft SQL Server will overwrite the contents of the device.</td>
</tr>
<tr>
<td>Format the entire media before backup operation</td>
<td>Will format the entire media before the backup operation starts.</td>
</tr>
<tr>
<td>Read and verify the integrity after the backup after finish</td>
<td>After the operation completes, the integrity of the backup is verified.</td>
</tr>
<tr>
<td>Backup Threads</td>
<td>0 is the default. Zero enables the server default value.</td>
</tr>
</tbody>
</table>

### Backup Operation Summary

Here you indicate whether or not you want to save the operation, and if so, where the file should be saved. Also displayed are the elements of the function you are creating. You can go back to earlier panels if you want to make changes before the job runs.

### Backup Operation Execution Selection

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open the backup script in an ISQL editor</td>
<td>The backup script will be opened into a new ISQL session. <strong>NOTE:</strong> This option is not available for Sybase backups.</td>
</tr>
<tr>
<td>Perform the backup now</td>
<td>The backup will be executed immediately.</td>
</tr>
<tr>
<td>Schedule the backup for later execution</td>
<td>The backup will be scheduled to run at a specified date/time.</td>
</tr>
</tbody>
</table>

For more information, see:
Executing a Backup Script in the ISQL Window

To execute the actual command DBArtisan uses to backup your database in the ISQL Window, do the following:

1. On **Backup Operation Execution Selection**, click **Open**.
   - DBArtisan opens the ISQL Window, which is used to execute SQL and Transact-SQL commands. The ISQL Window displays the procedure call that Backup Analyst makes to back up SQL Server databases, complete with the parameters needed to perform the backup.

2. To execute your backup, on the **ISQL Editor** toolbar, click **Execute**.
   - DBArtisan completes your backup and displays a message "Procedure Succeeded" on the Results tab of the ISQL Editor.

Backup Media Contents Dialog Box

Here you find a read-only display of the backup media available for the backup/restore operation you are creating.

Database Recovery Wizard

To both validate your back ups and practice the disaster recovery process, you should regularly restore your database dumps. If possible, every day, you should restore your databases on a separate server to enhance system availability. In this manner, you can maintain a warm backup of your database guarding against downtime from system failure.

The Database Recovery Wizard lets you:

- Create a new load operation.
- Reuse a previously defined operation.
- Specify the target server, database, and type of restore operation.
- Specify the recovery completion state.
- Specify how to report the progress of the load operation.
- Specify which file to load from the device.
- Specify whether to perform safety checks before replacing the database.
- Save the definition of the dump operation for future reuse.

**NOTE:** Backups created with Microsoft SQL Server version 7 cannot be restored to an earlier version of SQL Server.

**NOTE:** If you perform a Database Backup, and then Perform a Data Unload on the same database, close all open windows or disconnect from the database before starting the Database Recovery Wizard.

Creating or Editing a Recovery Operation

DBArtisan lets you create different recover operations using the Recovery Wizard. After you have successfully created a recovery operation, you can reopen and edit it using the same wizard and make the changes you want.
To Open the Database Recovery Wizard

1. On the **Datasource Explorer**, select the **Databases** node(s).
   
   DBArtisan displays the databases in the right pane of the Explorer window.

2. Select the target database.

3. On the **Utilities** menu, click **Database Restore**.

The first panel of the Database Backup Wizard welcomes you to the Backup Wizard.

Subsequent panels of the wizard are as follows:

- **Playback or New Operation**
- **Database Selection for MS SQL**
- **Database Selection for Sybase ASE**
- **Recovery Completion for MS SQL**
- **Recovery Completion for Sybase ASE**
- **Backup and Media Set Information for MS SQL**
- **Dump Device Selection for Sybase ASE**
- **Tape Handling Options for Sybase ASE**
- **Recovery Options**
- **Recover Operation Summary**
- **Action Selection**

### Playback or New Operation

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Recovery/Load Operation</td>
<td>This initiates the process of creating a new recovery operation.</td>
</tr>
<tr>
<td>Previous Recovery/Load Operation</td>
<td>If you want to replay a recovery operation, when you select the radio button, the browse function is enabled so you can find the .sqrc file you want to reuse.</td>
</tr>
</tbody>
</table>

### Database Selection for MS SQL

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Database</td>
<td>From the drop-down list, specify the target database for the recovery.</td>
</tr>
<tr>
<td>Restore Type</td>
<td>Elect to restore the database, filegroups or files, or a transaction log.</td>
</tr>
<tr>
<td>Backup Source</td>
<td>From Backup History - Find backup sets between the dates you specify and make your selection. Or, click Remove or Remove All to eliminate the device(s) from this operation.</td>
</tr>
</tbody>
</table>

### Database Selection for Sybase ASE

Here you select the Sybase server and database you want to recover from the drop-down lists.
Recovery Completion for MS SQL

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leave database operational</td>
<td>No more transaction logs can be recovered.</td>
</tr>
<tr>
<td>Leave database non-operational</td>
<td>You can restore more transaction logs.</td>
</tr>
<tr>
<td>Leave database read-only</td>
<td>You can restore more transaction logs.</td>
</tr>
<tr>
<td>Force restore over existing database</td>
<td></td>
</tr>
<tr>
<td>Eject tapes (if any) after restoring each backup</td>
<td>Eject tapes after the backup is restored.</td>
</tr>
<tr>
<td>Point in time restore</td>
<td>Select the date and time from which you want the restore operation to take place.</td>
</tr>
</tbody>
</table>

Recovery Completion for Sybase ASE

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which operation do you want to perform?</td>
<td>Load Database- Specifies that the complete database and transaction log should be restored. Load Transaction Log - Specifies that only the transaction log should be applied to the database. List Database - Validates a load database operation without actually performing the load. List Transaction Log - Validates a load transaction operation without actually performing the load.</td>
</tr>
<tr>
<td>Information type you want to see</td>
<td>Header Only, Brief Description, or Detailed Description.</td>
</tr>
<tr>
<td>If you are loading or listing a transaction log...</td>
<td>If you are loading the transaction log, in the If you are loading or listing a transaction log... box, type the specific log time you want to load.</td>
</tr>
</tbody>
</table>

Backup and Media Set Information for MS SQL

Here, for MS SQL, you specify the backup set name, backup set password, media name, and media name password. Any existing backup files are listed.

If you are creating at Leave database read-only operation, the Undo File option is enabled. Browse to find the file you want to undo.

Backup and Media Information for Sybase ASE

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What type of dump device do you want to use?</td>
<td>Choose among: Tape or Disk Device: Existing options for either of these choices become apparent on the next panel. If none are available, you won’t see any devices. File: You must type the name of the file you want use</td>
</tr>
<tr>
<td>Where should messages about the load operation be sent?</td>
<td>Select among: Default, Client, or Console.</td>
</tr>
<tr>
<td>Dump File</td>
<td>You must type the name of the file where you want to send the load operation messages.</td>
</tr>
</tbody>
</table>
Dump Device Selection for Sybase ASE
If you elected to use a tape or disk device, here you select which specific tape/disk device you want to use. You can choose one or more of the devices that appear in the panel. If none appear, you have to go Back to pick another device type. If you specify tape or disk properties by clicking each device, and then clicking the Attributes button. If you click Attribute, the Device Attribute Editor dialog box opens.

Tape Handling Options for Sybase ASE
You can opt to dismount after the operation completes or rewind and unload after the operation completes.

Recovery Options
For MS SQL, the backup file is identified here, and if you are creating a new database from another database’s backup, you can rename files in the Recover As box.

For Sybase, you see the default backup file, which you can easily change. You also have the option, for tape device operations, to load the data with decompression.

Recover Operation Summary
You can elect to save the recovery operation. You can see the default location where the recovery operation will be saved or you can browse to another location. The summary panel displays the choices you have made to this point. Go back to make changes.

Action Selection

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open the recovery script in an ISQL editor</td>
<td>The recovery script will be opened into a new ISQL session. See Executing a Recovery Script in the ISQL Window. NOTE: This option is not available for Sybase recoveries.</td>
</tr>
<tr>
<td>Perform the recovery now</td>
<td>The recovery will be executed immediately.</td>
</tr>
<tr>
<td>Schedule the recovery for later execution</td>
<td>The recovery will be scheduled to run at a specified date/time.</td>
</tr>
</tbody>
</table>

Device Attribute Editor Dialog Box
The Device Attribute Editor dialog box lets you configure tape/disk device attributes. The choices you make here apply to the device, not only to the current restore job you are configuring.

The table that follows describes the required information:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Server Name</td>
<td>Type the name of the backup server where the dump device is located.</td>
</tr>
<tr>
<td>Density</td>
<td>Use the drop-down list to select the appropriate value. This is the density of the tape, if a reel to reel tape is the storage device.</td>
</tr>
<tr>
<td>Block Size</td>
<td>Enter the appropriate number of bytes. This is amount of data to read/write in a single operation.</td>
</tr>
<tr>
<td>Capacity</td>
<td>Enter the appropriate value in megabytes. This is the backup storage capacity.</td>
</tr>
<tr>
<td>Volume Name</td>
<td>Enter name of the volume where the backup is stored.</td>
</tr>
</tbody>
</table>
Recovery Wizard Dialog Box

The Recovery Wizard dialog box records your selection for the backup device and file you want to include in the backup operation.

**NOTE:** The Backup Device and Disk Directory options are mutually exclusive.

The table below explains the information fields you encounter:

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore Type</td>
<td>From the drop-down list, choose among Database, Filegroups or files, and Transaction Log recovery operations.</td>
</tr>
<tr>
<td>Backup Device</td>
<td>Your choices from the drop-down list are Disk Device, Floppy Device, Pipe Device, and Tape Device. Only when a device is selected, and available, does the right drop-down list populate.</td>
</tr>
<tr>
<td>Disk Directory and Name</td>
<td>The default directory populates the directory and name field. You can browse or type to make changes to the default.</td>
</tr>
<tr>
<td>Backup Number</td>
<td>Indicate the backup number. When you select View Contents, the Backup Media Contents Dialog Box opens.</td>
</tr>
<tr>
<td>Password/Password Confirmation</td>
<td>Create a password if you want to restrict access to the recovery file.</td>
</tr>
</tbody>
</table>

Executing a Recovery Script in the ISQL Window

To execute the actual command DBArtisan uses to recover your database in the ISQL Window, do the following:

1. On Database Recovery Wizard - Last Panel, click **Open**.
   - DBArtisan opens the ISQL Window, which is used to execute SQL and Transact-SQL commands. The ISQL Window displays the procedure call that Backup Analyst makes to recover SQL Server databases, complete with the parameters needed to perform the recovery.

2. To execute your recovery, on the ISQL Editor toolbar, click **Execute**.
   - DBArtisan completes your recovery and displays a message "Procedure Succeeded" on the Results tab of the ISQL Editor.

Database Recovery Wizard - Last Panel

Review the database restore description to verify its accuracy.

1. If you need to make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections.

2. If you are satisfied with the database restore operation, click **Execute**.
   - For IBM DB2 for z/OS and OS/390, DBArtisan opens the Preview:Create dialog box.

For more information, see Restoring Databases.
Data Export Wizard for Oracle

DBArtisan provides comprehensive graphical support for the Oracle Export Utilities. DBArtisan offers a simple wizard that guides you through the process of extracting object definitions and table data from an Oracle database and creating binary export file. DBArtisan's Export Wizard ensures that the proper parameters and privileges are used, provides access to desired database information and resolves version compatibility differences.

The Export Wizard supports all four export modes:
- Table
- Users
- Full Database
- Point-in-time Recovery (available in Oracle8)

The benefits of using the Oracle Export Utilities range from moving data between different versions of Oracle quickly and efficiently, saving space on the database, to selectively backing up an entire or portion of a database using cumulative or incremental export functions.

To use the Export Wizard, you must specify the path and name for the Oracle Export utility in the Oracle Utility Tab of the Options Editor.

**NOTE:** If you do not have the EXP_FULL_DATABASE role assigned to you, you may not be able to access all the functionality of the export wizard.

Starting the Data Export Wizard

1. On the **Datasource Explorer**, select the database object.
   - DBArtisan displays the database objects in the right pane of the Datasource Explorer.
2. On the **Utilities** menu, click **Database Export**.
   - DBArtisan opens the **Data Export Wizard**.

Data Export Wizard - Panel 1

The first panel of the Export Wizard lets you specify the source datasource, name an existing parameter file or use the existing Oracle default parameter file.

The table below describes the options and functionality on the first panel of the Data Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which database has the data you wish to export?</td>
<td>Click the list, and then click the database.</td>
</tr>
<tr>
<td>Edit an existing parameter file</td>
<td>Type the location and name of the parameter file or click the browse button to find the file.</td>
</tr>
<tr>
<td>Default values</td>
<td>To use the Oracle default values for the parameter file, leave this box blank.</td>
</tr>
</tbody>
</table>

For more information, see [Data Export Wizard](#).
Data Export Wizard - Panel 2

The second panel of the Export Wizard lets you select the export mode that determines the database objects you will export.

- The full database mode makes available incremental, cumulative or complete export operations.
- The point-in-time export operation can export one or more tablespaces, which you can use in an import operation to recover a tablespace at a prior point in time without affecting the rest of the database.

The table below describes the objects, related objects and structures that are exported for each mode:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Object Types and Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Table definitions, object types defined on the table, table data by partition, nested table data, indexes defined on the table, table constraints, table permissions, column and table comments, table analysis information, triggers that refer to the table, and auditing information.</td>
</tr>
<tr>
<td>User</td>
<td>Libraries, object types, database links, sequence numbers, cluster definitions, table definitions, object types defined on the table, table data by partition, nested table data, indexes defined on the table, table constraints, table permissions, table analysis information, column and table comments, private synonyms, stored procedures, functions, packages, auditing information, views, cluster analysis information, referential constraints, triggers, snapshots, and snapshot logs.</td>
</tr>
<tr>
<td>Full Database</td>
<td>Tablespaces definitions, tablespace quotas, profiles, user definitions, roles, role privileges, system privileges, resource costs, rollback segment definitions, database links, sequence numbers, directories, libraries, object types, cluster definitions, password information, default and system auditing, object type tables, table definitions, table data by partition, nested table data, table indexes, table constraints, table privileges, table analysis information, column and table comments, auditing information, synonyms, views, stored procedures, functions, packages, triggers, cluster analysis information, snapshots, snapshot logs, refresh groups and children.</td>
</tr>
<tr>
<td>Point-In-Time (available in Oracle8)</td>
<td>Tablespaces.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the second panel of the Data Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which mode would you like to use for this export?</td>
<td>Click the option button that corresponds to the export mode you want to implement: <strong>Table</strong>, <strong>User</strong>, <strong>Full Database</strong>, or <strong>Point-in-Time Recovery</strong>.</td>
</tr>
</tbody>
</table>

For more information, see [Data Export Wizard](#).

Data Export Wizard - Panel 3

The third panel of the Export Wizard lets you specify the objects you want to export based on the mode you selected in the second panel of the Export Wizard.

The table below describes the options and functionality on the third panel of the Data Export Wizard for Table mode:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the tables and/or partitions you would like to export</td>
<td>To display list of available tables, click the Owner list, and then click the owner.</td>
</tr>
<tr>
<td>Tables</td>
<td>In the box, select the check boxes that correspond to the tables you want to export. If a table has partitions, select the check boxes that correspond the partitions you want to export. To select all tables, click the Select All button.</td>
</tr>
</tbody>
</table>
The table below describes the options and functionality on the third panel of the Data Export Wizard for User mode:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the users whose objects you would like to export</td>
<td>To display list of available users, select the check boxes that correspond to the objects you want to export. To select all objects, click the Select All button.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the third panel of the Data Export Wizard for Full Database mode:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Would you like to perform an incremental export  | An incremental operation exports all database objects that have changed since the last incremental, cumulative, or complete export operation.  
  NOTE: To perform an incremental export operation, you must have the EXP_FULL_DATABASE role.  
  A cumulative operation exports all database objects that have changed since the last cumulative or complete export.  
  A complete operation exports all objects. |
| Do you want to record this export in the system tables that track exports? | To record the export operation in the SYS.INEXP, SYS.INCFIL, and SYS.INVID system tables, click the Yes button. |

The table below describes the options and functionality on the third panel of the Data Export Wizard for Point-in-Time Recovery mode:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Select the tablespaces to be recovered          | To display list of available tablespaces, select the check boxes that correspond to the tablespaces you want to recover.  
  To recover all tablespaces, click the Select All button. |

For more information, see Data Export Wizard.

**Data Export Wizard - Panel 4**

The fourth panel of the Export Wizard lets specify:

- The objects you want to export.
- To view the progress meter.

The table below describes the options and functionality on the fourth panel of the Data Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Check the objects you would like to export     | NOTE: This option is not available if you are performing a full database incremental export operation. The Oracle Export Utility automatically determines if a row has been changed since the last export operation.  
  Select the check box that corresponds to the object(s) you want to export:  
  To export indexes associated with the tables, select the Indexes check box.  
  To export check constraints or referential integrity constraints, select the Constraints check box.  
  To export object privileges, select the Grants check box.  
  To export table data, select the Rows check box. |
Data Export Wizard - Panel 5

The fifth panel of the Export Wizard lets you set the following options:

- The Export Path
- The Length of the File Record
- The Buffer Size

The table below describes the options and functionality on the fifth panel of the Data Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to perform a direct path export?</td>
<td>A direct path export extracts data much faster as it bypasses the SQL Command processing layer and saves data copies whenever possible.</td>
</tr>
<tr>
<td>Would you like to accept your platform's value for BUFSIZ as the length of the file record?</td>
<td>Lets you select the Yes or No option button. If you clicked the No option button, in the Specify length box, type the KB value of the length. The highest value you can specify is 64 KB.</td>
</tr>
<tr>
<td>Would you like to accept your platform's default value for the buffer size used to fetch rows?</td>
<td>NOTE: This option is only valid if you are performing a conventional path export. Lets you select the Yes or No option button. If you clicked the No option button, in the Specify size box, type the KB value of the length; the highest value you can specify is 64 KB.</td>
</tr>
</tbody>
</table>

For more information, see Data Export Wizard.

Data Export Wizard - Panel 6

The sixth panel of the Export Wizard lets you specify:

- Handling storage of table data.
- Type of optimizer statistics to generate when the file is imported.
- Make the data remain unchanged during the export operation.

The table below describes the options and functionality on the sixth panel of the Data Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like table data to be consolidated into one initial extent upon import?</td>
<td>Lets you select the No or Yes option button.</td>
</tr>
</tbody>
</table>

For more information, see Data Export Wizard.
Data Export Wizard - Panel 7

The seventh panel of the Export Wizard lets you specify names and locations for the export, parameter and logfiles.

The table below describes the options and functionality on the seventh panel of the Data Export Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What type of database optimizer statistics should be generated upon import?</td>
<td>Lets you click the appropriate option button: Estimate, Compute, or None.</td>
</tr>
<tr>
<td>Would you like to ensure that the data seen by this export operation does not change during execution?</td>
<td>NOTE: This option is not available for an incremental export operation.</td>
</tr>
<tr>
<td></td>
<td>Lets you select the No or Yes option button.</td>
</tr>
<tr>
<td></td>
<td>Click Yes to ensure that the data remains intact throughout the export operation.</td>
</tr>
</tbody>
</table>

For more information, see Data Export Wizard.

Data Export Wizard - Panel 8

Review the data export operation description to verify its accuracy.

Completing Panel 8

Review the export operation description to verify its accuracy.

1. If you need to make any changes, click Back to scroll back to the appropriate panels of the wizard to make your corrections.

2. If you are satisfied with the export operation, click Execute.

    DBArtisan starts the Export operation.

For more information, see Data Export Wizard.
Data Import Wizard for Oracle

DBArtisan provides comprehensive graphical support for the Oracle Import Utilities. DBArtisan offers a simple wizard that guides you through the process of extracting object definitions from an Oracle Export file and inputting them into the database. The Oracle Import utility only reads export files created with the Oracle Export utility. DBArtisan's Import Wizard ensures that the proper parameters and privileges are used, provides access to desired database information and resolves version compatibility differences.

The Import Wizard supports all four import modes:

- Table
- Users
- Full Database
- Point-in-time Recovery (available in Oracle8)

The Oracle Import utilities build database objects in the proper sequence to avoid dependency problems. Tables are created first, then table data is input into the proper tables, next indexes are built, then triggers are imported and finally integrity constraints are built.

To use the Import Wizard, you must specify the path and name for the Oracle Import utility in the Oracle Utility Tab of the Options Editor.

**NOTE:** If you do not have the EXP_FULL_DATABASE role assigned to you, you may not be able to access all the functionality of the import wizard.

Starting the Import Wizard

1. On the **Datasource Explorer**, select the database object.

   DBArtisan displays the database objects in the right pane of the Datasource Explorer.

2. On the **Utilities** menu, click **Data Import**.

   DBArtisan opens the **Data Import Wizard**.

Data Import Wizard - Panel 1

The first panel of the Import Wizard asks you to specify the target datasource and to name an existing import parameter file or use the existing Oracle default parameter file.

The table below describes the options and functionality on the first panel of the Data Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Into which database would you like to import data?</td>
<td>Click the list, and then click the database.</td>
</tr>
<tr>
<td>Edit an existing parameter file</td>
<td>Type the location and name of the parameter file or click the browse button to find the file.</td>
</tr>
<tr>
<td>Default values</td>
<td>To use the Oracle default values for the parameter file, leave this box blank.</td>
</tr>
</tbody>
</table>

For more information, see **Data Import Wizard**.
Data Import Wizard - Panel 2

The second panel of the Import Wizard lets you specify the export file from which the data should be imported. You must also select the import mode you want to use which determines the database objects that DBArtisan imports. The table below describes the objects, related objects and structures that are imported for each mode:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Object Types and Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Table definitions, object types defined on the table, table data by partition, nested table data, indexes defined on the table, table constraints, table permissions, column and table comments, table analysis information, triggers that refer to the table, and auditing information.</td>
</tr>
<tr>
<td>User</td>
<td>Libraries, object types, database links, sequence numbers, cluster definitions, table definitions, object types defined on the table, table data by partition, nested table data, indexes defined on the table, table constraints, table permissions, table analysis information, column and table comments, private synonyms, stored procedures, functions, packages, auditing information, views, cluster analysis information, referential constraints, triggers, snapshots, and snapshot logs.</td>
</tr>
<tr>
<td>Full Database</td>
<td>Tablespace definitions, tablespace quotas, profiles, user definitions, roles, role privileges, system privileges, resource costs, rollback segment definitions, database links, sequence numbers, directories, libraries, object types, cluster definitions, password information, default and system auditing, object type tables, table definitions, table data by partition, nested table data, table indexes, table constraints, table privileges, table analysis information, column and table comments, auditing information, synonyms, views, stored procedures, functions, packages, triggers, cluster analysis information, snapshots, snapshot logs, refresh groups and children.</td>
</tr>
<tr>
<td>Point-In-Time (available in Oracle8)</td>
<td>Tablespaces.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the second panel of the Data Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the export file from which you want to import data?</td>
<td>Lets you type the location and name export file to be used in the import operation, or find the file by clicking the Browse button.</td>
</tr>
<tr>
<td>Which database was used to create this export file?</td>
<td>Lets you choose the database used for export file creation from the list.</td>
</tr>
</tbody>
</table>
| Which mode would you like to use for this export? | Lets you click the option button that corresponds to the import mode you want to implement.  
  Table  
  User  
  Full Database  
  ORACLE 8 OR LATER ONLY:  
  Point-In-Time |

For more information, see Data Import Wizard.

Data Import Wizard - Panel 3

The third lets you specify which objects you want to import based on the mode you selected in the second panel of the Import Wizard.

The table below describes the options and functionality on the third panel of the Data Import Wizard for Table mode:
The table below describes the options and functionality on the third panel of the Data Import Wizard for User mode:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the users whose objects you would like to import</td>
<td>To display list of available users, select the check boxes that correspond to the objects you want to import. To select all objects, click the Select All button.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the third panel of the Data Import Wizard for Full Database mode:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to perform an incremental import</td>
<td>An incremental operation imports all database objects that have changed since the last incremental, cumulative, or complete export operation.</td>
</tr>
</tbody>
</table>

For more information, see Data Import Wizard.

**Data Import Wizard - Panel 4**

The fourth panel lets you specify options to import dependent objects, privileges and table data, and to create a file for index-creation commands.

The table below describes the options and functionality on the fourth panel of the Data Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check the objects you would like to import</td>
<td>Select the check box that corresponds to the object(s) you want to export. To import object privileges, click the Grants check box. To import table data, select the Rows check box. To import indexes associated with the tables, select the Indexes check box.</td>
</tr>
<tr>
<td>Would you like to skip unusable indexes?</td>
<td>Lets you click the Yes or No option button.</td>
</tr>
<tr>
<td>Would you like specify a file to receive index-creation commands?</td>
<td>When this parameter is specified, index-creation commands for the requested mode are extracted and written to the specified file, rather than used to create indexes in the database. Tables and other database objects are not imported. If you clicked the Yes option button, type the location and name of the file or find the file by clicking the Browse button.</td>
</tr>
</tbody>
</table>

For more information, see Data Import Wizard.
Data Import Wizard - Panel 5
The fifth panel of the Import Wizard lets you:

- Set the data rows transfer buffer size.
- Indicate if the import operation should commit after each array insert.
- Specify the option to re-use existing data files.
- Specify the option to ignore creation errors.

The table below describes the options and functionality on the fifth panel of the Data Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to accept your platform's default value for buffer size used to fetch rows?</td>
<td>If you select the No option button, in the Specify length box, type the KB value of the length. The highest value you can specify is 64 KB.</td>
</tr>
<tr>
<td>Should Import commit after each array insert?</td>
<td>Lets you select the No or Yes option button.</td>
</tr>
<tr>
<td>Would you like to reuse the existing datafiles making up the database?</td>
<td>Select Yes to set the reuse option in the datafile clause of the CREATE TABLESPACE command.</td>
</tr>
<tr>
<td>Should Import ignore object creation errors when attempting to create database objects?</td>
<td>Select Yes if you selected Table as your mode of import in panel 2, and the tables exist in the database in the target database.</td>
</tr>
</tbody>
</table>

For more information, see Data Import Wizard.

Data Import Wizard - Panel 6
The sixth panel of the Import Wizard lets you specify:

- The length of the file record.
- How you want to implement SQL Analyze statements contained in the export file.
- To view the contents of the export file rather than import the database information.
- To view the progress meter.

The table below describes the options and functionality on the sixth panel of the Data Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to show the contents of the export file rather than import them?</td>
<td>Lets you select the No or Yes option button.</td>
</tr>
<tr>
<td>Would you like to view the progress meter as rows are imported</td>
<td>Lets you enter the number of rows per dot displayed in the meter box, type the value of the number of rows per dot you want displayed in the meter.</td>
</tr>
<tr>
<td>Would you like to accept your platform's value for BUFSIZ as the length of the file record?</td>
<td>If you select the No option button, in the Specify length box, type the KB value of the length; the highest value you can specify is 64 KB.</td>
</tr>
<tr>
<td>Would you like to accept your platform's default value for the buffer size used to fetch rows?</td>
<td>NOTE: This option is only valid if you are performing a conventional path export. If you click the No option button, in the Specify Size box, type the KB value of the length; the highest value you can specify is 64 KB.</td>
</tr>
<tr>
<td>Would you like Import to execute SQL Analyze statements found in the export file?</td>
<td>Lets you select the No or Yes option button.</td>
</tr>
</tbody>
</table>
For more information, see Data Import Wizard.

Data Import Wizard - Panel 7

The seventh panel of the Import Wizard lets you specify names and locations for the parameter and logfiles. If you importing from an Oracle 6 export file, you can define the character set of the original file.

The table below describes the options and functionality on the seventh panel of the Data Import Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character Set</td>
<td>For Oracle 6 export files, type the character set of the original file.</td>
</tr>
<tr>
<td>Logfile Button</td>
<td>Click to capture informational and error messages, and then type the location and name of the logfile in the corresponding box. Export logfile names use the &quot;.logfile&quot; extension. (for example, C:\Temp\EXPLJOBerlin.log).</td>
</tr>
<tr>
<td>None Button</td>
<td>Click if you do not want to capture informational and error messages.</td>
</tr>
<tr>
<td>Parameter File</td>
<td>If you expect to perform the same export operation again, type the location and name of the parameter file in the corresponding box. Parameter files are stored as .txt files. (for example, C:\Temp\EXPPARBerlin.txt).</td>
</tr>
</tbody>
</table>

For more information, see Data Import Wizard.

Data Import Wizard - Panel 8

Review the data import operation description to verify its accuracy.

Completing Panel 8

Review the import operation description to verify its accuracy.

1. If you need to make any changes, click Back to scroll back to the appropriate panels of the wizard to make your corrections.

2. If you are satisfied with the import operation, click Execute.
   
   DBArtisan starts the Import operation.

For more information, see Data Import Wizard.

Archive Activity

Oracle uses Redo logs to record changes made to a database during operation. These files, which are active or online during normal operation, are referred to as the online redo logfiles. Oracle uses these files to reapply changes made during the recovery process that were not written to datafiles before shutdown. You can operate an Oracle database in ARCHIVELOG or NOARCHIVELOG mode. If you choose ARCHIVELOG mode, then Oracle uses a background process to copy the contents of the online redo logfiles as archive redo logfiles in a separate area. ARCHIVELOG mode lets you perform online backups and database media recovery of your Oracle database.
The Archive Activity Monitor lets you analyze the status, activity levels, proper sizing and performance of your archive redo logs. At a glance, you can determine the ARCHIVELOG mode, configuration and vital statistics about redo log performance. In particular, the Archive Activity Monitor lets you assess whether or not the redo logfiles are sized properly. It is very important to choose the right size for online redo logfiles. If the redo logs are too small, the Log Writer must switch logfiles too often, which hurts performance. If the redo logs are too big, recovery times are longer.

The table below describes the Archive Activity Monitor options:

<table>
<thead>
<tr>
<th>Category</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Archiving</td>
<td>Log Mode</td>
<td>Whether or not ARCHIVELOG mode is active for the Oracle instance.</td>
</tr>
<tr>
<td>Auto Archival</td>
<td></td>
<td>Whether or not the database has been configured to archive redo logs automatically.</td>
</tr>
<tr>
<td></td>
<td>Destination</td>
<td>Directory where the archive redo logs are contained.</td>
</tr>
<tr>
<td></td>
<td>Oldest Log Sequence</td>
<td>Oldest system change number. As a precaution, your backups should include all archived logfiles starting with the oldest log sequence.</td>
</tr>
<tr>
<td></td>
<td>Current Log Sequence</td>
<td>Current system change number. It marks the start of the backup.</td>
</tr>
<tr>
<td>Online Redo Archive Activity</td>
<td>Redo Entries</td>
<td>Number entries made to the online redo logs by the Log Writer (LGWR) process.</td>
</tr>
<tr>
<td></td>
<td>Avg Redo Size</td>
<td>Average size, in kilobytes, of a redo log entry.</td>
</tr>
<tr>
<td>Redo Log Buffer Contention</td>
<td>Entry Requests</td>
<td>Number of entries made to the redo log buffer by the Log Writer (LGWR) process.</td>
</tr>
<tr>
<td></td>
<td>Entry Wait Times</td>
<td>Number of times that the Oracle Log Writer has had to wait before making an entry to the redo log buffer. This number is accumulated since the last time that the database was started. A high number of waits indicates that the log buffer size is too small. To increase the size, you need to modify the LOG_BUFFER parameter in the Oracle configuration file (e.g., INITORCL.ORA).</td>
</tr>
<tr>
<td>Redo Log Switches</td>
<td>Time</td>
<td>When the redo log switch occurred. If redo log switches occur frequently, the redo logs are probably too small. As a rule-of-thumb, redo log switches should not occur more frequently than once every 30 minutes. If they are occurring more frequently, then you should increase the size of the redo logs.</td>
</tr>
<tr>
<td></td>
<td>Record ID in Controlfile</td>
<td>Displays ID number in the control file.</td>
</tr>
</tbody>
</table>

Opening the Archive Activity Dialog Box

1. On the Utilities menu, click Archive Activity.

   DBArtisan opens the Archive Activity dialog box.

Backup Control Files

You should keep many copies of your control files to safeguard against disk failure or loss of the primary control file. DBArtisan provides a graphical interface for constructing the ALTER DATABASE command needed to backup your control files.

Using the Backup Control Files Utility

1. On the Utilities menu, click Backup Control File.

   DBArtisan opens the Backup Control File dialog box.
Backup Controlfile Dialog Box

The table below describes the options and functionality available on the Backup Controlfile dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Controlfile Name</td>
<td>Type the name of the backup control file in the box. To reuse an existing Back Up Controlfile, type the name of the existing file, and then select Reuse Existing Backup Controlfile. <strong>NOTE:</strong> When you type in the Backup Controlfile Name box, the check box is activated.</td>
</tr>
<tr>
<td>Reuse Existing Backup Controlfile check box</td>
<td>Select this check box if you want to reuse an existing file. The name of the file and the name entered in the Backup Controlfile Name box must be the same.</td>
</tr>
<tr>
<td>Trace SQL Statements</td>
<td>Click the option button that corresponds to how you want Trace SQL statements handled:</td>
</tr>
<tr>
<td>Owner</td>
<td>Lets you select the owner for whom you want to scan for chained rows.</td>
</tr>
<tr>
<td>Refresh Button</td>
<td>Click to refresh the list.</td>
</tr>
</tbody>
</table>

Chained Row Analysis

Updates can cause data rows to migrate and to chain over multiple data blocks. Chaining degrades performance because more physical reads are required to access a row of data. Consequently, you should monitor chained rows regularly to spot performance bottlenecks before they become severe.

To avoid chained rows, you can tune the Percent Free storage settings for that table or cluster. By increasing the Percent Free on each data block, you decrease the likelihood that an update causes a data row to expand over multiple data blocks.

If the row length is greater than the size of a data block, you cannot avoid chained rows.

DBArtisan provides a utility that lets you identify chained rows quickly by displaying chained rows for all or any schema in your Oracle database.

Analyzing Chained Rows

1. On the **Utilities** menu, click **Chained Row Analysis**.

DBArtisan opens the Chained Row Analysis dialog box.

The table below describes the options and functionality on the Chained Row Analysis dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Lets you select the owner for whom you want to scan for chained rows.</td>
</tr>
<tr>
<td>Refresh Button</td>
<td>Click to refresh the list.</td>
</tr>
</tbody>
</table>
Free Space Deficit

One of the central tasks of every Oracle administrator is to monitor the availability of free space in tablespaces. When a data segment (e.g., table, index or cluster) is created on a tablespace, it acquires an initial extent of a determined size. As it grows, the data segment will eventually need to acquire another extent to store additional data. If a data segment cannot acquire its next extent, all further transactions on it will fail.

The Free Space Deficit Monitor identifies any objects that will not be able to acquire their next extent. Whenever it identifies a problem data segment, you should take corrective action as soon as possible. To address the problem, you can:

1. Reduce the object's Next Extent Size from the appropriate Object Editor.
2. Increase the availability of free space in the tablespace by:
   - Adding datafiles.
   - Activating autoextend for the tablespace's datafiles. Autoextend allows the datafiles to grow to fill available room on the physical drive. This option is only available with Oracle.

The table below describes the objects:

<table>
<thead>
<tr>
<th>Category</th>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Name</td>
<td>Name of the tablespace.</td>
</tr>
<tr>
<td></td>
<td>Largest Free Section</td>
<td>Largest contiguous section of free space in a tablespace.</td>
</tr>
<tr>
<td>Objects</td>
<td>Object Name</td>
<td>Name of any object that will not be able to acquire its next extent because it will be larger than the Largest Free Section.</td>
</tr>
<tr>
<td></td>
<td>Object Type</td>
<td>Type of the object, such as a table, index or cluster, that will not be able to acquire its next extent.</td>
</tr>
<tr>
<td></td>
<td>Next Extent</td>
<td>Specified size of the object's next extent. One of the options you have to correct the problem is to reduce the next extent size so that it is smaller than the Largest Free Section.</td>
</tr>
</tbody>
</table>

Opening the Free Space Deficit Dialog Box

1. On the Utilities menu, click Free Space Deficit.

   DBArtisan opens the Free Space Deficit dialog box.

   The table below describes the options and functionality on the Free Space Deficit dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Lets you click the owner for whom you want to analyze free space.</td>
</tr>
</tbody>
</table>

Pinned Code

For Oracle, DBArtisan now offers a Pinned Code Utility that lets you pin or unpin code objects in memory. This ensures that frequently accessed database stored procedures, triggers, functions, etc., are available in memory, which improves their overall execution performance.

**NOTE:** To create the DBMS_SHARED_POOL package log on as SYS and run the DBMSPOOL.SQL script that is found in the ORACLE_HOME\RDBMS\ADMIN directory.
Opening the Pin Code Dialog Box

1. On the Utilities menu, click Pinned Code.

DBArtisan opens the Pin Code dialog box.

The table below describes the options and functionality on Pin Code dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Lets you select the object user.</td>
</tr>
<tr>
<td>Pin Button</td>
<td>To pin an object, select the target object, and then click the Pin button.</td>
</tr>
<tr>
<td>UnPin Button</td>
<td>To unpin an object, select the target object, and then click the UnPin button.</td>
</tr>
<tr>
<td>Flush SGA Button</td>
<td>Click to flush the Oracle shared pool to remove fragmentation and any unwanted code objects from memory.</td>
</tr>
</tbody>
</table>

You must install the SYS.DBMS_SHARED_POOL package on your Oracle server to use this utility.

Confirm Shared Pool Flush

The Confirmed Shared Pool Flush dialog box lets you confirm that you want to flush the Shared Pool. Flushing the Shared Pool causes all non-pinned code objects to be removed. After you flush the Shared Pool, all non-pinned objects are parsed and reloaded back into the pool on the next execution. This can extend first-time execution response times.

Maximum Extent Limits

When an object reaches its maximum extent limit, it cannot acquire any new free space. This can cause critical problems in a production database. DBArtisan’s Maximum Extent Limits for Oracle tool lets you quickly find and correct space problems.

Opening the Maximum Extent Limits Dialog Box

1. On the Utilities menu, click Maximum Extent Limits.

DBArtisan opens the Maximum Extent Limits dialog box.

The table below describes the options and functionality on Maximum Extent Limits dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Owner</td>
<td>Lets you select owner of the objects to display the Objects box.</td>
</tr>
<tr>
<td>Objects</td>
<td>Objects for the user account(s) that have reached their maximum extent limit.</td>
</tr>
<tr>
<td>Edit Button</td>
<td>To increase an object’s maximum extent limit, edit the object and adjust its maximum extent property to a higher value or to unlimited.</td>
</tr>
</tbody>
</table>
Instance Parameter Configuration

DBArtisan lets you configure and tune your Oracle databases. DBArtisan offers a utility that lets you view and edit the settings for each Oracle instance. DBArtisan derives its information from the system catalog. To reconfigure your Oracle instance, you must edit the parameter settings in the applicable configuration file (for example, INITORCL.ORA).

NOTE: After changing the settings, shut down and restart the database server for the changes to take effect.

Changing an Instance Parameter

1. On the Datasources menu, click Configure.

DBArtisan opens the Instance Parameters dialog box.

2. To modify an instance parameter, double-click the target instance parameter or click the Edit button to open the Edit Configuration dialog box.

NOTE: To modify an instance parameter, the parameter must have a Yes in the Session Modifiable or System Modifiable column of the Instance Parameter grid.

The table below describes the options and functionality on the Edit Configuration dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Value</td>
<td>Lets you type the value for the parameter.</td>
</tr>
</tbody>
</table>

Extracting Instance Configuration Parameters

DBArtisan lets you extract the statements for setting and displaying your instance configuration parameters.

From the Datasource Explorer

1. On the Datasource Explorer, select the Configuration node.

DBArtisan displays the parameters in the right pane of the Datasource Explorer.

2. In the right pane of the Datasource Explorer, click the target parameter(s).


OR

In the right-pane of the Datasource Explorer, right-click the parameters, and then click Extract.

DBArtisan extracts the definition to a text window.

Generating a Report on Configuration Parameters

DBArtisan lets you generate a detailed report containing information about your instance configuration parameters. The reports derive their contents from the information displayed in the right hand side of the Datasource Explorer. To facilitate the online publication of these reports, DBArtisan generates the report in Hypertext Mark-Up Language (HTML). You can view the report directly in DBArtisan's built-in HTML-browser.

NOTE: For more information on generating reports, see Schema Publication.
Generating a Instance Configuration Parameters Report

1. On the Datasource Explorer, select the Configuration node.
   DBArtisan displays the parameters in the right pane of the Datasource Explorer.

2. In the right pane of the Datasource Explorer, click the target parameter(s).


   OR

   In the right-pane of the Datasource Explorer, right-click the parameters, and then click Report.

   DBArtisan opens the Report dialog box.

The table below describes the options and functionality on the Report dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Home Page File Name</td>
<td>To change the report home page file name, in the box, type the location and name of the file or find the file by clicking the Browse icon.</td>
</tr>
<tr>
<td>Report Title</td>
<td>To change the report title, in the box, type the location and name of the file or click the Browse Button to find the file.</td>
</tr>
</tbody>
</table>

Configuring Resource Cost

When you create a profile, you can assign a composite resource limit to it. The composite limit is a weighted average of four individual resource limits:

- Connect time
- Private SGA
- CPU per session
- Logical reads per session

The default setting is to weight each of these measures equally. You can alter the weights by issuing an ALTER RESOURCE COST command. DBArtisan provides a visual interface for constructing such a command.

Altering Resource Cost

   DBArtisan opens the Alter Resource Cost dialog box.

The table below describes the options and functionality on the Alter Resource Cost dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weightings</td>
<td>In the box, specify the weightings for each resource limit by typing the weight value in the corresponding box: Connect time, Private SGA, CPU per session, or Logical reads per session.</td>
</tr>
<tr>
<td>Enable Resource Limits</td>
<td>Select to To enable the resource limit changes immediately.</td>
</tr>
</tbody>
</table>
Session Information

Oracle tracks detailed information about the resources used by each session on a database. This information is useful for analyzing resource utilization patterns between user sessions to monitor performance.

The information displayed is derived from the Oracle system tables, SYS.V_$STATNAME and SYS.V_$SESSTAT. Consult the Oracle documentation or detailed descriptions in the meanings of each session parameter.

**TIP:** Waits lets you analyze bottlenecks and discover what a database is waiting on.

Viewing Session Information

1. On the **Utilities** menu, click **Session Information**.
   DBArtisan opens the User Session Information dialog box.
2. Complete the **User Session Information** dialog box.
3. When you finish viewing SQL, click **Close** to return to the **User Session Information** dialog box.
4. When you finish viewing the session information, click **Close**.

Resource Limits

Resource limits are sets of parameters that let you maximize your datasource assets. By setting resource limits, you can prevent transactions and queries from congesting server resources. This ensures that you do not:

- Exceed estimated or actual input/output demands determined by the optimizer.
- Return a number of rows beyond a set limit.
- Exceed a defined elapsed time.

DBArtisan includes the Resource Limits dialog box that lets you:

- **Show Resource Limits**
- **Add Resource Limits**
- **Edit Resource Limits**
- **Drop Resource Limits**

Opening the Resource Limits Dialog Box

You must connect to a datasource running Sybase before you can open the Resource Limits dialog box. To open the Resource Limits dialog box, do the following:

1. On the **Datasource Explorer**, select the target datasource node.
2. On the **Utilities** menu, click **Resource Limits**.
   DBArtisan opens the Resource Limits dialog box.

The table below describes the columns in the Resource Limits table of the Resource Limits dialog box:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The login name for the resource limit.</td>
</tr>
</tbody>
</table>
Utilities > Resource Limits

DBArtisan displays all resource limits for the current datasource. In addition, DBArtisan lets you use the Resource Limits dialog box to search for and display specific resource limits. In the Resource Limits dialog box, you can specify search criteria for resource limits. DBArtisan displays the resource limits that meet your search criteria in the Resource Limits table of the Resource Limits dialog box.

Displaying Resource Limits

1. On the Datasource Explorer, select the target datasource node.
2. On the Utilities menu, click Resource Limits. DBArtisan opens the Resource Limits dialog box.
3. Specify the search parameters. The table below describes the search parameters in the Resource Limits dialog box:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login list</td>
<td>Lets you specify the target login or logins.</td>
</tr>
<tr>
<td>Application list</td>
<td>Lets you specify an application or applications for the limit criteria.</td>
</tr>
<tr>
<td>Limit Time check box</td>
<td>Lets you specify a time range in which to enforce the resource limit.</td>
</tr>
<tr>
<td>Limit Time list</td>
<td>Lets you specify a start time for the resource limit. The start time must be within the proceeding 24 hours. To specify the start time, you must select the Limit Time check box.</td>
</tr>
<tr>
<td>Limit Day list</td>
<td>Lets you specify the day or day(s) for the resource limit.</td>
</tr>
<tr>
<td>Scope list</td>
<td>Lets you specify the scope of the target action(s). On the Scope list, the numbers next to each scope display in the scope column of the Resource Limits table.</td>
</tr>
<tr>
<td>Action list</td>
<td>Lets you specify the action to take place when a resource limit is exceeded. On the Action list, the numbers next to each action display in the action column of the Resource Limits table.</td>
</tr>
</tbody>
</table>
4 When you finish specifying search parameters, click List Limits.

DBArtisan lists the resource limits meeting the target criteria in the Resource Limits table.

TIP: To sort the display of the Resource Limits list, click a column heading.

For more information, see Resource Limits table of the Resource Limits dialog box.

Adding Resource Limits

DBArtisan offers a simple graphical wizard that guides you through the process of creating a resource limit without requiring your to know any of the underlying commands.

Starting the Resource Limit Wizard

1 On the Datasource Explorer, select the target datasource node.

2 On the Utilities menu, click Resource Limits.

   DBArtisan opens the Resource Limits dialog box.

3 In the Resource Limits dialog box, click Add.

   DBArtisan opens the first panel of the Resource Limit Wizard.

Resource Limit Wizard - Panel 1

The first panel of the Resource Limit Wizard lets you identify the objects to which the resource limit will apply.

The table below describes the options and functionality on the first panel of the Resource Limit Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Button</td>
<td>Click to open the Add Resource Limit Object dialog box.</td>
</tr>
<tr>
<td>Edit Button</td>
<td>Click to open the Edit Resource Limit Object dialog box.</td>
</tr>
</tbody>
</table>

For more information, see the Resource Limit Wizard.

Add/Edit Resource Limit Object Dialog Box

The table below describes the options and functionality on the Add/Edit Resource Limit Object dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>Lets you click the list, and then click the target user login.</td>
</tr>
<tr>
<td>Application</td>
<td>Click the list, and then click the target application.</td>
</tr>
</tbody>
</table>

For more information, see the Resource Limit Wizard.

Resource Limit Wizard - Panel 2

The second panel of the Resource Limit Wizard lets you specify:
The type of resource limit.

• The value of the resource limit type.

• The time during which the limit will be enforced.

The table below describes the options and functionality on the second panel of the Resource Limit Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the type of resource limit</td>
<td>Lets you specify the type of resource limit by clicking one of the following option buttons: Input/output cost, Elapsed time, or Row count.</td>
</tr>
<tr>
<td>Limit Value box</td>
<td>NOTE: You must type the value for the limit to continue.</td>
</tr>
<tr>
<td>Time Range</td>
<td>Lets you click the list, and then click the target named range.</td>
</tr>
<tr>
<td>Modify Button</td>
<td>Click to add or edit a named time range. DBArtisan opens the Named Time Ranges dialog box.</td>
</tr>
<tr>
<td>Add Button</td>
<td>Click to add or edit a named time range. DBArtisan opens the Named Time Ranges dialog box.</td>
</tr>
<tr>
<td>Edit Button</td>
<td>Click to add or edit a named time range. DBArtisan opens the Named Time Ranges dialog box.</td>
</tr>
<tr>
<td>Pre-execution</td>
<td>To specify the execution times during which the limit will be enforced, select the target check box(es):</td>
</tr>
<tr>
<td>Execution</td>
<td>To specify the execution times during which the limit will be enforced, select the target check box(es):</td>
</tr>
</tbody>
</table>

For more information, see the Resource Limit Wizard.

Named Time Ranges Dialog Box

A time range is a contiguous set of days. For example if you designate a Friday start day and a Monday end day, the time range includes the weekend.

The table below describes the options and functionality on the Named Time Ranges dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>In the box, type a name for the new time range.</td>
</tr>
<tr>
<td>Start time</td>
<td>In the box, specify the start time.</td>
</tr>
<tr>
<td>End time</td>
<td>In the box, specify the end time.</td>
</tr>
<tr>
<td>Start day</td>
<td>In the box, specify the beginning of the time range.</td>
</tr>
<tr>
<td>End day</td>
<td>In the box, specify the end of the time range.</td>
</tr>
</tbody>
</table>

For more information, see the Resource Limit Wizard.

Resource Limit Wizard - Panel 3

The third panel of the Resource Limit Wizard lets you specify the scope of the resource limit and the action to perform when the limit is exceeded. The available scopes are dependent on the type of resource limit you selected on the previous panel of the wizard.
The table below describes the types and available scopes:

<table>
<thead>
<tr>
<th>Resource Limit Type</th>
<th>Available Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input/output cost</td>
<td>Query</td>
</tr>
<tr>
<td>Elapsed time</td>
<td>Query batch, Transaction, and Query batch and transaction</td>
</tr>
<tr>
<td>Row count</td>
<td>Query</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the third panel of the Resource Limit Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope</td>
<td>Click one of the option buttons to specify the scope of the resource limit:</td>
</tr>
<tr>
<td></td>
<td>Query</td>
</tr>
<tr>
<td></td>
<td>Query batch</td>
</tr>
<tr>
<td></td>
<td>Transaction</td>
</tr>
<tr>
<td></td>
<td>Query batch and transaction</td>
</tr>
<tr>
<td>Action</td>
<td>Click one the following option buttons to specify the action to take when the resource limit is exceeded: Issue a warning, Abort the query batch, Abort the transaction, or Kill the session.</td>
</tr>
</tbody>
</table>

For more information, see the [Resource Limit Wizard](#).

**Resource Limit Wizard - Last Panel**

The last panel of the Resource Limit Wizard lets you:

- Preview the SQL for function that adds the resource limit.
- Execute the SQL for function that adds the resource limit.

**Completing the Last Panel**

1. Review the SQL.
2. Click **Execute**.

For more information, see the [Resource Limit Wizard](#).

**Editing Resource Limits**

DBArtisan lets you edit resource limits. To edit resource limits, do the following:

1. On the **Datasource Explorer**, select the target datasource node.
2. On the **Utilities** menu, click **Resource Limits**.
   DBArtisan opens the Resource Limits dialog box.
3. In the **Resource Limits** grid, select the target resource limit.
4. Click **Edit**.
   DBArtisan opens the Resource Limit Editor.
5  To modify the limit value, in the Limit Value box edit the value.
6  To change the type of action, click a different option button.
7  Click OK.

Dropping Resource Limits
DBArtisan lets you drop resource limits. To drop resource limits, do the following:

1  On the Datasource Explorer, select the target datasource node.
2  On the Utilities menu, click Resource Limits.
   DBArtisan opens the Resource Limits dialog box.
3  In the Resource Limits grid, select the target resource limit.
4  Click Drop.
   DBArtisan opens a confirm drop message.
5  Click Yes.

Top Sessions
Being able to identify the top resource users in an Oracle system is a valuable tuning asset. The Top Sessions dialog box lets you quickly pinpoint the heaviest users of I/O and memory usage. The Top Sessions utility quickly finds and ranks user sessions based on their resource usage.

Opening the Top Sessions Dialog Box
1  On the Utilities menu, click Top Sessions.
   DBArtisan opens the Top Sessions dialog box.

The table below describes the options and functionality on the Top Sessions dialog box.

   TIP:  Click the Options button to open the Top Sessions Options dialog box.

Top Sessions Options Dialog Box
The table below describes the options and functionality on the Top Sessions Options dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclude Sys and System</td>
<td>Select to force the query used to display top session data to be changed.</td>
</tr>
<tr>
<td>Exclude Background Process</td>
<td>Select to force the query used to display top session data to be changed.</td>
</tr>
<tr>
<td>Limit Top Session Display to</td>
<td>Lets you specify the number of sessions to display. These options will force the query used to display top session data to be changed.</td>
</tr>
<tr>
<td>No Limit</td>
<td></td>
</tr>
<tr>
<td>Sort Top Sessions By</td>
<td>Lets you custom soft the display</td>
</tr>
</tbody>
</table>
User Session Information Dialog Box
The table below describes the options and functionality on the User Session Information dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Lets you select a user.</td>
</tr>
<tr>
<td>OS User Name</td>
<td>Displays the operating system user name.</td>
</tr>
<tr>
<td>Session ID</td>
<td>Lets you select the application.</td>
</tr>
<tr>
<td>Machine Name</td>
<td>Displays the name of the datasource.</td>
</tr>
<tr>
<td>Status</td>
<td>Displays the session status.</td>
</tr>
<tr>
<td>Terminal Name</td>
<td>Displays the terminal name.</td>
</tr>
<tr>
<td>Command</td>
<td>Displays the command.</td>
</tr>
<tr>
<td>Program</td>
<td>Displays the program used.</td>
</tr>
<tr>
<td>Client Platform</td>
<td>IBM DB2 FOR WINDOW/UNIX ONLY: Lets you select the client platform.</td>
</tr>
<tr>
<td>Client Protocol</td>
<td>IBM DB2 FOR WINDOW/UNIX ONLY: Lets you select the client protocol.</td>
</tr>
<tr>
<td>Session Parameters</td>
<td>IBM DB2 FOR WINDOW/UNIX ONLY: Lets you select the session parameters.</td>
</tr>
<tr>
<td>Statistical Details Tab</td>
<td>ORACLE ONLY: Displays statistical details.</td>
</tr>
<tr>
<td>Waits Tab</td>
<td>ORACLE ONLY: Displays waits.</td>
</tr>
<tr>
<td>Current SQL Tab</td>
<td>ORACLE ONLY: Displays the current SQL.</td>
</tr>
</tbody>
</table>

User Session Information Dialog Box - Statistical Details Tab
The table below describes the options and functionality on the Statistical Details Tab of the User Session Information Dialog Box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical Details grid</td>
<td>Displays the session parameters and value (if any).</td>
</tr>
</tbody>
</table>

User Session Information Dialog Box - Waits Tab
The table below describes the options and functionality on the Statistical Details Tab of the User Session Information Dialog Box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Historical Waits</td>
<td>If selected, displays historical waits in the waits grid.</td>
</tr>
<tr>
<td>Current Waits</td>
<td>If selected, displays current waits in the waits grid.</td>
</tr>
<tr>
<td>Include Idle Waits</td>
<td>If selected, displays idle waits in the waits grid.</td>
</tr>
<tr>
<td>Waits grid</td>
<td>Displays the wait information, including Session ID, Event, Seconds in Wait, Wait Time, State, and other information.</td>
</tr>
</tbody>
</table>
User Session Information Dialog Box - Current SQL Tab

The table below describes the options and functionality on the Statistical Details Tab of the User Session Information Dialog Box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL box</td>
<td>Displays current SQL.</td>
</tr>
</tbody>
</table>
Tools

DBArtisan incorporates a number of powerful tools to help you create, edit and manage your development environment. You can use Tools to:

- Conduct extensive database searches across multiple databases.
- Execute scripts or files across multiple databases.
- Schedule tasks.
- Identify differences in files or objects.
- Graphically build complex queries.
- Administer your ODBC data sources.

Tools is divided into sections. The table below describes each section:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find in Files</td>
<td>This section describes the Find in Files dialog box that lets you find a phrase or character in your files.</td>
</tr>
<tr>
<td>Database Search</td>
<td>This section describes the powerful database search utility that helps you to find instances of a string across multiple databases.</td>
</tr>
<tr>
<td>Script Execution Facility</td>
<td>This section describes the Script Execution Facility, a stand-alone utility that establishes multiple threads and database connections letting you simultaneously execute SQL statements against multiple Oracle, Sybase Adaptive Server, Microsoft SQL Server, and IBM DB2 for Linux, Unix, and Windows datasources.</td>
</tr>
<tr>
<td>File Execution Facility</td>
<td>This section describes the File Execution Facility, a stand-alone utility that establishes multiple threads and database connections that lets you execute parallel queries and ANSI SQL files against multiple, cross-platform datasources.</td>
</tr>
<tr>
<td>Scheduling</td>
<td>The DBArtisan scheduling programs and utilities let you schedule and execute jobs on local datasources anytime.</td>
</tr>
<tr>
<td>Visual Difference</td>
<td>DBArtisan lets you compare two files or database objects. Using the Visual Difference Utility, you can easily synchronize and analyze database objects or files across multiple database platforms.</td>
</tr>
<tr>
<td>Query Builder</td>
<td>This section describes Query Builder, a tool that lets you construct, structure, and manipulate up to five different types of queries simultaneously.</td>
</tr>
<tr>
<td>Import Data</td>
<td>How to bring data from an external source into a database table is described. You can import data from a text file or a Microsoft Excel document.</td>
</tr>
<tr>
<td>Data Editor</td>
<td>This section describes the Data Editor to edit your tables in real-time. The Data Editor supports all editable datatypes and is an alternative way to add, edit, or delete data from your tables.</td>
</tr>
<tr>
<td>Embarcadero Products</td>
<td>The Tools menu lists all installed Embarcadero Technologies products. This lets you toggle to or start another Embarcadero product.</td>
</tr>
<tr>
<td>Code Workbench</td>
<td>This section describes the Code Workbench that lets you enable table column look-up and population in the ISQL window, define auto replacement expressions that can be used to quickly insert commonly used blocks of SQL syntax or commands in any open window and to import and export Code Workbench specific settings for client sharing purposes.</td>
</tr>
</tbody>
</table>
Find in Files

The Find in Files dialog box lets you find a phrase or character in your files.

Completing the Find in Files Dialog Box

1. On the **Tools** menu, click **Find in Files**.

   DBArtisan opens the Find in Files dialog box.

The table below describes the options and functionality on the Find in Files dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find what</td>
<td>Specifies the character(s) or phrase you want to find. Use the browse arrow button next to the textbox to choose options from a pop-up list.</td>
</tr>
<tr>
<td>In files/file types</td>
<td>Specifies the files in which to search for the character(s) or phrase. Either enter the filename(s) in the drop-down box, or click the arrow to choose a file type.</td>
</tr>
<tr>
<td>In folder</td>
<td>Specifies the directory where the file(s) is located. Click the browse button to view your Windows Explorer.</td>
</tr>
<tr>
<td>Match whole word only</td>
<td>Specifies the application to find only the entire phrase.</td>
</tr>
<tr>
<td>Match case</td>
<td>Specifies the application to find only the specified phrase in the case you have entered.</td>
</tr>
<tr>
<td>Regular Expression</td>
<td>Tells the application whether the specified character(s) is a regular expression.</td>
</tr>
<tr>
<td>Look in subfolders</td>
<td>Specifies the application to search the file(s) any folders located within the specified folder.</td>
</tr>
<tr>
<td>Output to Pane 2</td>
<td>Specifies the application to display the results in another window.</td>
</tr>
</tbody>
</table>

**NOTE:** You can also use the Find feature to locate a phrase or character in an ISQL window.

Database Search

The powerful [database search utility](#) helps you to find instances of a string across multiple databases.

Starting the Database Search Wizard

1. On the **Tools** menu, click **Database Search**.

   DBArtisan opens the first panel of the Database Search Wizard.

Database Search Wizard - Panel 1

The first panel of the Database Search Wizard lets you specify the owner.

The table below describes the options and functionality on the first panel of the Database Search Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the datasource(s) you want to search</td>
<td>Search however many datasources you want. You can search more than one platform at a time and specific owners within each datasource.</td>
</tr>
</tbody>
</table>

1. Click **Next**.

   DBArtisan opens the next panel of the wizard.
Database Search Wizard - Panel 2
The second panel of the Database Search Wizard lets you specify the search criteria.

The table below describes the options and functionality on the second panel of the Database Search Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Database For</td>
<td>Lets you enter the search string. Strings can also be searched for using DB2-standard wildcards. And for multiple string searches, separate each string with a vertical bar (for example, DEPARTMENTS</td>
</tr>
<tr>
<td>Match Case</td>
<td>Select Yes to make the search case sensitive. <strong>NOTE:</strong> IBM DB2 for OS/390, searches are always case insensitive.</td>
</tr>
<tr>
<td>Search DDL of these Objects</td>
<td>In the grid, select the target object check boxes. <strong>NOTE:</strong> Event Monitors are available for IBM DB2 for Linux, Unix, and Windows only.</td>
</tr>
</tbody>
</table>

1. **Click Execute** to start the operation.

DBArtisan displays a progress dialog box while the search runs. When the search completes, DBArtisan opens the Database Search Window.

Database Search Results
DBArtisan displays Database Search operation results in a Database Search Window, listing all of the objects containing the search string in the left pane. You can browse instances of the search string by selecting different objects in the tree. The DDL of the objects displays in the right pane and the search string is highlighted.

The table below describes the buttons on the Database Search Window toolbar:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>Opens the first panel of the Database Search Wizard.</td>
</tr>
<tr>
<td>Criteria</td>
<td>Opens the Search Criteria dialog box.</td>
</tr>
<tr>
<td>Open</td>
<td>Opens the editor for the target object.</td>
</tr>
<tr>
<td>Extract</td>
<td>Lets you extract the target object.</td>
</tr>
<tr>
<td>Print</td>
<td>Lets you print the target object SQL.</td>
</tr>
</tbody>
</table>

Script Execution Facility
DBArtisan's Script Execution Facility is a stand-alone utility that establishes multiple threads and database connections letting you simultaneously execute SQL statements against multiple Oracle, Sybase Adaptive Server, Microsoft SQL Server, and IBM DB2 for Linux, Unix, and Windows datasources. After completing a scheduled job, DBArtisan generates a results report that lists errors, verifies script execution, and details the output of the job. The Script Execution Facility also works in conjunction with DBArtisan's scheduling facilities, letting you schedule script execution jobs. When used in conjunction with a scheduler, DBArtisan can automatically send the results report to any e-mail or network recipients. The Script Execution Facility is a tabbed dialog box where you set the parameters and options for the script execution. In the Script Execution Facility dialog box you can:

- Type or paste the target SQL script.
• Specify the datasources against which to execute the script.
• Specify the output mode for the results report.
• Open a scheduling program to schedule the script execution.
• Specify execution notification e-mail and Net Send addresses.

For more information, see Completing the Script/File Execution Facility.

File Execution Facility

DBArtisan’s File Execution Facility is a stand-alone utility that establishes multiple threads and database connections that lets you execute parallel queries and ANSI SQL files against multiple, cross-platform datasources. The Script Execution Facility also works in conjunction with DBArtisan’s scheduling facilities, letting you schedule script execution jobs. After completing a scheduled job, DBArtisan generates a results report that lists errors, verifies execution, and details the output of the job. When used in conjunction with a scheduler, DBArtisan can automatically send the results report to any e-mail or network recipients. The File Execution Facility is a tabbed dialog box where you set the parameters and options for the file execution. In the File Execution Facility dialog box you can:

• Specify the target files or ANSI SQL scripts.
• Specify the datasources against which to execute the files.
• Specify the output mode for the results report.
• Open one of the DBArtisan scheduling programs to schedule the file execution.
• Specify execution notification e-mail and Net Send addresses.

Completing the Script/File Execution Facility

DBArtisan lets you run parallel queries against multiple datasources with the File Execution Facility.

1. On the Tools menu, click Script Execution Facility or File Execution Facility.
   DBArtisan opens the Script or File Execution Facility dialog box.
2. Complete the Script Tab (Script Execution Facility).
3. Complete the Files Tab (File Execution Facility).
4. Complete the Target Tab.
5. Complete the Output Tab.
6. Complete the Notify Tab.

For more information, see:

File Execution Facility
Script Execution Facility
Script Execution Facility - Script Tab

The table below describes the options and functionality on the Script Tab of the File/Script Execution Facility:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Script box</td>
<td>Lets you type or paste a script.</td>
</tr>
</tbody>
</table>

For more information, see:

File Execution Facility
Script Execution Facility
Completing the Script/File Execution Facility

File Execution Facility - Files Tab

The table below describes the options and functionality on the Files Tab of the File Execution Facility:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Full File Paths</td>
<td>Select to display the full path. Deselect to display only the file name.</td>
</tr>
<tr>
<td>File Name</td>
<td>Displays the file names.</td>
</tr>
<tr>
<td>Add</td>
<td>Click to open the Select Files dialog box.</td>
</tr>
<tr>
<td>Remove</td>
<td>Click to remove the selected file.</td>
</tr>
<tr>
<td>View</td>
<td>Opens the View File dialog box.</td>
</tr>
<tr>
<td>Up</td>
<td>Click to move the selected file up in the list.</td>
</tr>
<tr>
<td>Down</td>
<td>Click to move the selected file down in the list.</td>
</tr>
</tbody>
</table>

For more information, see:

File Execution Facility
Script Execution Facility
Completing the Script/File Execution Facility

File/Script Execution Facility - Target Tab

The table below describes the options and functionality on the Target Tab of the File/Script Execution Facility:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Select the Target Datasource(s) to Execute the Script Against | Only Connected Datasources - Displays only datasources that are currently connected in the Datasource grid.  
All DBMS Types - Displays all DBMS types in the Datasource grid. |
| Datasource grid                               | Displays the target datasource(s) to execute the script/file against. Select a datasource name. If the datasource has multiple databases, type in a database in the Database box. |

For more information, see:
File Execution Facility

Script Execution Facility

Completing the Script/File Execution Facility

File/Script Execution Facility - Output Tab

The table below describes the options and functionality on the Output Tab of the File/Script Execution Facility:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical Output</td>
<td>If selected, specifies a graphical output.</td>
</tr>
<tr>
<td>File Output</td>
<td>If selected, specifies a file output. Directory - Type or browse to enter the full path and directory name in which you want to place the output file. File Type - Specifies a file type. Include column titles when saving - If selected, lets you save column titles. Open files with registered applications - If selected, opens files with registered applications.</td>
</tr>
</tbody>
</table>

For more information, see:

File Execution Facility

Script Execution Facility

Completing the Script/File Execution Facility

File/Script Execution Facility - Notify Tab

The table below describes the options and functionality on the Notify Tab of the File/Script Execution Facility:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Description</td>
<td>Lets you enter a job description. This description will be the subject of the notification E-mail.</td>
</tr>
<tr>
<td>E-mail address</td>
<td>Lets you enter E-mail addresses. Separate each E-mail address with a semicolon (;).</td>
</tr>
<tr>
<td>Net Send User Names</td>
<td>Lets you enter net send user names. Separate each name with a semicolon (;).</td>
</tr>
</tbody>
</table>

For more information, see:

File Execution Facility

Script Execution Facility

Completing the Script/File Execution Facility

Scheduling

The DBArtisan scheduling programs and utilities let database administrators schedule and execute jobs on local datasources 24-hours-a-day, 7-days-a-week. After completing a scheduled job, DBArtisan generates a results report that lists errors, verifies script execution, and details the output of the job. DBArtisan can automatically send the results report to any e-mail or network recipients.
DBArtisan offers the following programs and utilities to let you schedule routine tasks and jobs:

- Microsoft Task Scheduler
- Embarcadero ETSQLX

DBArtisan's scheduling facilities let you:

- Execute scheduled jobs without running DBArtisan.
- Run batch files to automate tasks.
- Run a script in parallel against multiple datasources.
- Automatically send result reports to e-mail or network recipients when jobs complete.

**Microsoft Task Scheduler**

DBArtisan lets you use the Microsoft Task Scheduler to schedule jobs. The Microsoft Task Scheduler is included with various Microsoft applications. If you do not have this program on your system, the first time you attempt to schedule a job, DBArtisan provides you with a link to the Microsoft Web site where you can download the Microsoft Task Scheduler at no cost.

The table below describes the options and functionality on the Schedule Action dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Name</td>
<td>Lets you type the name of the job.</td>
</tr>
<tr>
<td>Job Description</td>
<td>Lets you type a job description that will appear in the subject line in your e-mail and Net Send messages.</td>
</tr>
<tr>
<td>E-mail Address(es)</td>
<td>Lets you type the e-mail address(es) to which you want to send notifications.</td>
</tr>
<tr>
<td>Net Send</td>
<td>Lets you type the network user(s) to whom you want to send notifications.</td>
</tr>
<tr>
<td>Action Output Directory</td>
<td>Lets you type the full path for the directory in which to place the output file or click browse to open the Select Directory dialog box.</td>
</tr>
</tbody>
</table>

The table below describes the tabs on the Job Scheduler dialog box:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule</td>
<td>Lets you set your scheduling parameters.</td>
</tr>
<tr>
<td>Settings</td>
<td>Lets you set your settings parameters.</td>
</tr>
</tbody>
</table>

**NOTE:** DBArtisan’s ETSQLX command line utility runs a scheduled job even if DBArtisan is not running.

This section includes the following:

- Opening the Microsoft Scheduled Task Directory Through DBArtisan

**Select Directory**

The select directory dialog box lets you locate the directory in which to place the output file.

For more information, see Microsoft Task Scheduler.
Opening the Microsoft Scheduled Task Directory Through DBArtisan

In the Microsoft Scheduled Tasks directory, Microsoft lets you create, view, edit, and delete jobs.

1. On the **Tools** menu, click **Scheduler**.

   DBArtisan opens the Microsoft Scheduled Tasks directory.

**ETSQLX Command Line Utility**

DBArtisan’s ETSQLX command line utility, is a multi threaded, cross-platform, SQL scripting engine. You can use ETSQLX in conjunction with the Microsoft Task Scheduler to schedule and automate routine jobs. ETSQLX creates batch files (with the extension.cfg) containing commands to execute automated and scheduled jobs. ETSQLX creates a directory, CFG, in which it stores the.cfg files. You can run.cfg files directly from the command line.

   **NOTE:** ETSQLX supports .csv, .tab, .htm, and .html formats for result reports attachments.

**Visual Difference**

DBArtisan lets you compare two files or database objects. Using the Visual Difference dialog box, you can easily synchronize and analyze database objects or files across multiple database platforms. The files are displayed side by side in the Visual Difference dialog box. The Visual Difference Utility highlights any differences between two files. Viewing differences between objects and files helps you negotiate between the different phases of development as well as providing a visual aid to rapidly changing and evolving production environments.

   **NOTE:** Because contents of the Visual Difference dialog box are read-only, you will not be able to modify your files or objects directly from this dialog box.

The Visual Difference dialog box is composed of two panes; the left pane displays your source object or file and the right pane shows your target object or file. The Visual Difference dialog box also contains its own toolbar which lets you:

- Search
- Navigate differences
- Set options
- Print

**Opening the Visual Difference Dialog Box**

1. On the **Tools** menu, click **Visual Diff**.

   DBArtisan opens the Visual Difference dialog box.

For more information, see:

- Comparing Files
- Comparing Database Objects
- Navigating in the Visual Difference Dialog Box
- Printing a Pane of the Visual Difference Dialog Box
- Searching in the Visual Difference Dialog Box
- Setting Options in the Visual Difference Dialog Box
Comparing Files
You can compare two files side-by-side in the Visual Difference dialog box. The file you want to compare is called the Source. The file you want to compare the first file to is the Target.

Comparing Items
1 On the Tools menu, click Visual Diff.
   DBArtisan opens the Visual Difference dialog box.
2 On the Visual Difference toolbar, click the Source icon or click the Down arrow next to the Source icon and then click File.
   DBArtisan opens the Select the 1st File to Compare dialog box.
3 Click the file that you want to be the Source file.
4 On the Visual Difference toolbar, click the Target icon or click the Down arrow next to the Target icon and then click File.
   DBArtisan opens the Select the 2nd File to Compare dialog box.
   NOTE: The Visual Difference Utility highlights all differences between the two files.

For more information, see Visual Difference Utility.

Comparing Database Objects
The schema of database objects is automatically extracted so you can view the underlying differences between object and perform a side-by-side comparison in the Visual Difference Dialog.

Comparing Database Objects
1 On the Tools menu, click Visual Diff.
   DBArtisan opens the Visual Difference dialog box.
2 On the Visual Difference toolbar, click the Down arrow next to the Source icon and then click Database Object.
   DBArtisan opens the Select the 1st Database Object to Compare dialog box.
3 Click the datasource and then click OK to connect.
4 Navigate through the datasource tree and double-click the database object that you want to be the Source.
5 On the Visual Difference toolbar, click the Down arrow next to the Target icon and then click Database Object.
   DBArtisan opens the Select the 2nd Database Object to Compare dialog box.
   NOTE: The Visual Difference Utility highlights all differences between the two database objects.

For more information, see Visual Difference Utility.

Navigating in the Visual Difference Dialog Box
You can navigate through the Visual Difference dialog box using the up and down arrow buttons. You can move back and forth between highlighted differences in your compared files or database objects.
Going To the Next Difference
1 From the Visual Difference dialog box, click down arrow to go to the next difference.

Going To the Previous Difference
1 From the Visual Difference dialog box, click up arrow to go to the next difference.

For more information, see Visual Difference Utility.

Printing a Pane of the Visual Difference Dialog Box
You can print each pane of the Visual Difference dialog box.

Printing a Pane of the Visual Difference Dialog Box
1 Position your cursor inside the pane you want to print.
2 Click the Print icon on the Visual Difference toolbar.
   DBArtisan opens the Print Setup dialog box.
3 Click OK to print the pane.
   NOTE: You can only print one pane of the Visual Difference dialog box at a time.

For more information, see Visual Difference Utility.

Searching in the Visual Difference Dialog Box
The Visual Difference dialog box lets you search for text in your files or database objects.

Searching for Text
1 Place your cursor inside the pane you want to search.
2 Click the Find icon on the Visual Difference toolbar.
   DBArtisan opens the Find dialog box.
3 In the Find What box, enter the search string.
4 To match whole words only, select the Match whole word only check box.
5 To make the search case sensitive, select the Match case check box.
6 Click Find Next to find the next occurrence of your search string. You can also click the Find Next icon on the Visual Difference toolbar to search for the next occurrence at a later time.

For more information, see Visual Difference Utility.

Setting Options in the Visual Difference Dialog Box
The Visual Difference dialog box lets you set display and comparison options to help you customize the dialog box to view differences in a comprehensive manner.
Setting Options
1. Click the Options icon on the Visual Difference toolbar.

DBArtisan opens the Visual Diff Options dialog box.

The following table describes the check box options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Line Numbers</td>
<td>Indicates that line numbers should appear in the Visual Diff dialog box.</td>
<td>Off</td>
</tr>
<tr>
<td>Display Hidden Characters</td>
<td>Indicates that hidden characters (nonprintable) should be displayed.</td>
<td>Off</td>
</tr>
<tr>
<td>Ignore White Space</td>
<td>Indicates that White Space (such as spaces, carriage returns, line feeds, and tabs) should be ignored. If this option is set on, text will be considered equivalent regardless of white space, otherwise the text will be shown as being different.</td>
<td>On</td>
</tr>
<tr>
<td>Ignore Hidden Characters</td>
<td>Indicates that hidden characters (nonprintable) should be excluded.</td>
<td>Off</td>
</tr>
<tr>
<td>Ignore Case</td>
<td>Indicates that case should not be a differentiating factor.</td>
<td>On</td>
</tr>
</tbody>
</table>

2. Click OK.

DBArtisan accepts the options.

For more information, see Visual Difference Utility.

Query Builder

Query Builder is a database productivity tool that lets you construct, structure, and manipulate up to five different types of queries simultaneously. It includes a separate graphical interface that opens within your current workspace. You can run Query Builder against all Embarcadero Technologies supported database platforms.

Query Builder displays the interconnections of your queries as you work. The powerful visual components of Query Builder let you see your query grow and change to fit your needs. Query Builder eases the task of drawing data from tables by automatically creating correct SQL code as you build a statement. You can use Query Builder to create and execute SELECT statements for tables and views. You can also test queries, and easily adjust your information, before you save. Query Builder does not rely on knowledge of the underlying SQL code.

You can save and reopen queries in Query Builder. Query Builder automatically checks for changes in your tables or columns between the time you save the query and the time you reopen it.

The table below describes the types of queries available in Query Builder:

<table>
<thead>
<tr>
<th>Query Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>Create, manipulate and execute SELECT Statements for tables and views.</td>
</tr>
<tr>
<td>INSERT</td>
<td>Create and manipulate INSERT Statements for tables.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Create and manipulate UPDATE Statements for tables.</td>
</tr>
<tr>
<td>DELETE</td>
<td>Create and manipulate DELETE Statements for tables.</td>
</tr>
<tr>
<td>CREATE VIEW</td>
<td>Create and manipulate CREATE VIEW Statements for tables and views.</td>
</tr>
</tbody>
</table>

**NOTE:** You can execute SELECT statements directly from Query Builder. INSERT, UPDATE, DELETE, and CREATE VIEW statements must be moved to an ISQL Editor for execution.
For more information, see:

- Query Builder Design
- Using Query Builder

### Query Builder Design

Query Builder lets you build DML statements using an intuitive, graphical interface. It offers you a powerful and flexible way to quickly create complex statements without sacrificing time manipulating SQL code. Query Builder lets you add tables or columns, create joins, and change statements within the graphic display without leaving DBArtisan. It also lets you have multiple sessions working at the same time.

Query Builder includes many different features to assist you in building and manipulating your query:

- Query Builder Statement Properties
- Workspace Windows
- Query Builder Explorer
- Tool Bar
- SQL Diagram Pane
- SQL Statement Pane

For more information, see Build Query.

### Workspace Windows

The Workspace Windows provide a comprehensive view of your data structure and query. The table below describes the Workspace Windows:

<table>
<thead>
<tr>
<th>Pane</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Builder Explorer Window</td>
<td>Includes two tabs that display selected object details:</td>
</tr>
<tr>
<td></td>
<td>Tables/Views</td>
</tr>
<tr>
<td></td>
<td>DML</td>
</tr>
<tr>
<td>SQL Diagram Pane</td>
<td>Displays tables or views included in the current query.</td>
</tr>
<tr>
<td>SQL Statement Pane</td>
<td>Displays the SQL code, and when appropriate, a Results Tab.</td>
</tr>
</tbody>
</table>

For more information, see Build Query.

### Query Builder Explorer Window

The Query Builder Explorer is a separate tree that exposes all the tables and views in your target database. It also displays your current query structure. The Query Builder Explorer includes two tabs that display information about the selected objects:

- Tables/Views
- DML
Tables/Views Tab
The Tables/View Tab displays information about the selected tables or views. You can use the drop-down lists to change your table or view, and when appropriate, the owner. The table below describes each drop-down list on the Tables/Views Tab:

<table>
<thead>
<tr>
<th>List</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>Displays all databases for a target Microsoft SQL Server or Sybase ASE.</td>
</tr>
<tr>
<td>Second</td>
<td>Displays all valid owners.</td>
</tr>
</tbody>
</table>

**NOTE:** To change your current database, select the new database in the Explorer, and then open another Query Builder session. Query Builder prompts you to save the current session prior to opening a new session.

DML Tab
The DML Tab displays all the basic elements of a query statement in the SQL Statement Tree. You can access any element of the current statement display and perform SQL editing from the SQL Statement Tree.

For more information, see:
- Creating a Clause Using the SQL Statement Tree
- Build Query

SQL Diagram Pane
The SQL Diagram Pane displays tables, views, and joins included in the current query. You can manipulate elements of your query, using the mouse functionality, in the SQL Diagram Pane. From the SQL Diagram Pane you can:

- Add and Remove tables and views.
- Create and delete joins.
- Add and Subtract columns.

All changes in the SQL diagram reflect in correct SQL code in the SQL Statement Pane.

For more information, see Build Query.

SQL Statement Pane
The SQL Statement Pane displays the current query SQL code. When you run a query, Query Builder displays results to your query in the SQL Statement Pane. The SQL Statement Pane is divided into two tabs:

- SQL Tab
- Results Tab

**SQL Tab**
The SQL Tab displays the query in progress. It displays each element of your query as you build it, and updates as you do edits such as selecting or deselecting columns, adding clauses, and creating joins. DBArtisan lets you open the current statement directly into an ISQL editor or copy it to the clipboard for later use.
Results Tab

The Results Tab displays the results of your executed query in the Results grid. To edit data, use the Data Editor application from Query Builder. When you begin building a new query, the tab title changes to Old Results until you execute the new query.

For more information, see Build Query.

Query Builder Tool Bar

The Query Builder tool bar lets you access commonly used features.

The table below describes Query Builder tool bar functionality:

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>Copies the current SQL statement to the clipboard.</td>
</tr>
<tr>
<td>Statement Box</td>
<td>Displays the type of statement currently on display in the main workspace window.</td>
</tr>
<tr>
<td>Stop Execution</td>
<td>Stops an executing query.</td>
</tr>
<tr>
<td>Execute</td>
<td>Executes the current SELECT or CREATE VIEW statement. If the button is not available, the statement is not executable.</td>
</tr>
<tr>
<td>New</td>
<td>Adjusts to the target node in the Query Builder Explorer window.</td>
</tr>
<tr>
<td>Edit</td>
<td>Displays, on the DML Tab, the ORDER BY or GROUP BY dialog boxes when target node is selected.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the target object.</td>
</tr>
<tr>
<td>Auto Layout</td>
<td>Resets the main workspace to the auto layout mode.</td>
</tr>
<tr>
<td>Auto Join</td>
<td>Finds and joins, automatically, like items by name.</td>
</tr>
<tr>
<td>Statement Check</td>
<td>Checks query syntax.</td>
</tr>
<tr>
<td>Edit Data</td>
<td>Opens Data Editor.</td>
</tr>
<tr>
<td>Close</td>
<td>Closes the current query.</td>
</tr>
</tbody>
</table>

**NOTE:** Query Builder adjusts tool availability to match the current query functionality.

For more information, see Build Query.

Tables and Views Shortcut Menus

Query Builder includes a shortcut menu that lets you manipulate a table or view. The table below describes the table shortcut options:
For more information, see [Build Query](#).

### Tables and Views Keyboard Commands

Query Builder provides a number of keyboard shortcuts that let you quickly construct queries. The table below describes the keyboard commands:

<table>
<thead>
<tr>
<th>Keyboard Command</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESCAPE</td>
<td>SQL Diagram Pane</td>
<td>Breaks off a join.</td>
</tr>
<tr>
<td>F5</td>
<td>Query Builder</td>
<td>Refreshes screen and runs Schema Change Detection. In a CREATE VIEW, this key adds the new view to the Table Tree Pane.</td>
</tr>
<tr>
<td>CTRL A</td>
<td>SQL Diagram Pane</td>
<td>Selects all tables and joins in the current diagram.</td>
</tr>
<tr>
<td>F1</td>
<td>Query builder and application</td>
<td>Obtains context sensitive Help.</td>
</tr>
</tbody>
</table>

For more information, see [Build Query](#).

### Query Builder Dialog Boxes

Query Builder includes a number of dialog boxes to assist you in building and customizing your query.

<table>
<thead>
<tr>
<th>Dialog Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement Properties</td>
<td>Specifies general properties in an individual Query Builder session.</td>
</tr>
<tr>
<td>Table Properties</td>
<td>Specifies column selection and alias names for a table or view.</td>
</tr>
<tr>
<td>Column Properties</td>
<td>Specifies column functionality within SELECT and CREATE VIEW statements.</td>
</tr>
</tbody>
</table>

For more information, see [Build Query](#).
Statement Properties
The Statement Properties dialog box lets you customize properties in an individual Query Builder session. For example, you can set options to limit the number of rows returned in a query to save execution time, or turn off the auto join function to manually control all joins for an individual query. These properties override the global options set in the Options Editor for the current session.

The table below describes the options and functionality of the Statement Properties dialog box.

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Generation</td>
<td>Generate Use Database statement</td>
<td>Adds a line of SQL code indicating which database or instance is used in the statement.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Generate owner names</td>
<td>Adds a line of SQL code showing the table owner name as part of the query.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Include Row Count limits</td>
<td>Includes the output row limit set in the Execution settings.</td>
<td>Selected</td>
</tr>
<tr>
<td>Execution</td>
<td>Max Row Count in Results Set</td>
<td>Sets row count limits to build and check a query without congesting server processes when a query executes.</td>
<td>1000 rows</td>
</tr>
<tr>
<td>General</td>
<td>Show Column Data types in Query Diagram</td>
<td>Lets Query Builder reveal the data type in each column for tables in the SQL Diagram Pane.</td>
<td>Not selected</td>
</tr>
<tr>
<td></td>
<td>Confirm on Item delete</td>
<td>Lets Query Builder open a Confirm Delete dialog box when an item is deleted.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>NOTE:</strong> Clearing this function can result in unexpected changes to your query diagram and statement.</td>
<td></td>
</tr>
<tr>
<td>Auto Join</td>
<td>Require Indexes</td>
<td>Joins indexed columns automatically, and requires indexed columns for joins.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Require same data type</td>
<td>Automatically joins columns with the same data type.</td>
<td>Selected</td>
</tr>
<tr>
<td>Syntax Checker</td>
<td>Automatic Syntax Check</td>
<td>Lets Query Builder check syntax every time an execute statement, refresh or copy statement begins.</td>
<td>Selected</td>
</tr>
<tr>
<td></td>
<td>Run Automatically</td>
<td>Lets Query Builder automatically detect like names and data types and create joins for multiple tables.</td>
<td>Selected</td>
</tr>
</tbody>
</table>

**NOTE:** If you set options in the Options Editor while Query Builder is running, DBArtisan displays a warning indicating that you are about to change options or properties.
Completing the Statement Properties Dialog Box
To complete the Statement Properties dialog box, do the following:

1. On the Query Builder menu, click Statement Properties.
2. Set options.
3. Click OK.

Query Builder saves the options.

For more information, see Build Query.

Table Properties
The Tables Properties dialog box lets you set parameters for tables or views in your SQL Diagram. The table below describes the options and functionality on the Table Properties dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Alias</td>
<td>Creates an alias name for your table.</td>
</tr>
<tr>
<td>Show Datatypes</td>
<td>Shows or hides the datatype for every column in the target table.</td>
</tr>
<tr>
<td>Displayed Columns</td>
<td>Displays columns visible in the SQL Diagram.</td>
</tr>
<tr>
<td>Hidden Columns</td>
<td>Displays columns hidden in the SQL Diagram.</td>
</tr>
<tr>
<td>Hide All</td>
<td>Moves all non selected columns in the table to the Hidden Columns window.</td>
</tr>
<tr>
<td>Display All</td>
<td>Moves all columns in the table to the Displayed Columns window.</td>
</tr>
<tr>
<td>Right Arrow</td>
<td>Moves a target file from Displayed Columns to Hidden Columns.</td>
</tr>
<tr>
<td>Left Arrow</td>
<td>Moves a target file from Hidden Columns to Displayed Columns.</td>
</tr>
</tbody>
</table>

Completing the Table Properties Dialog Box
To complete the Table Properties dialog box, do the following:

1. Double click the target table or view title bar.
   OR
   Right-click target table or view, and then click Properties.
2. If you only want to hide or display columns in your table, click the arrow button on the table title bar.
3. You can also edit view properties from the Table Properties dialog box.
4. Click OK.

Query Builder saves the changes.

For more information, see Build Query.

Column Properties
The Column Properties dialog box lets you set properties for individual columns in your SELECT or CREATE VIEW statements. You can set aggregate functions and create an alias for an individual column.

The Column Properties dialog box is not available for INSERT, UPDATE or DELETE statements.
The table below describes the options and functions Columns Properties dialog box:

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables/Views</td>
<td>Displays all tables and views in the SQL Diagram Pane.</td>
</tr>
</tbody>
</table>
| Aggregate         | Specifies aggregate options for the target column.  
|                   | AVG - An average is taken for a column with an int or numeric datatype.  
|                   | COUNT - Returns the number of rows which contain data for the target column.  
|                   | MAX - Returns the highest number in a row in the column.  
|                   | MIN - Returns the lowest number in a row in the column.  
|                   | SUM - Returns the sum of the target column in all rows which contain data. This function is only operable on int or numeric datatypes. |
| Alias             | Displays the alias name for the target column. Lets you type the name of the alias.  
|                   | **NOTE:** Query Builder displays the results of an aggregate column without a column name unless you create an alias for that column. |
| Available Columns | Displays all available columns in the target table or view. |
| Selected Columns  | Displays all selected columns in the target table or view. To create an aggregate function or alias for a different column, select target column, select an aggregate function, and then type the name of the alias. |
| Select All        | Moves all columns in the Available Columns box to the Selected Columns box. |
| Clear All         | Moves all columns in the Selected Columns box to the Available Columns box. |
| Right Arrow       | Moves target column in the Available Columns box to the Selected Columns box. |
| Left Arrow        | Moves target column in the Selected Columns box to the Available Columns box. |
| Select List Statement | Displays the current query. |

**Completing the Column Properties Dialog Box**

To complete the Column Properties dialog box, do the following:

1. On the **SQL Statement Tree**, double-click target column.
2. Select options.
3. Click **OK**.

For more information, see [Build Query](#).

**Join Properties**

Query Builder lets you edit joins with the Join editor. You can edit join parameters in a SELECT, UPDATE, DELETE, and CREATE VIEW Statement.

The table below describes the options and functionality on the Join dialog box.
NOTE: For IBM DB2 for Linux, Unix, and Windows and IBM DB2 for z/OS and OS/390 servers, there is an additional join object in the SQL Statement Tree. The Join On node displays join relations between columns in IBM DB2 for Linux, Unix, and Windows and IBM DB2 for z/OS and OS/390 tables and views.

Completing the Join Dialog Box
To complete the Join dialog box, do the following:

1. In the SQL Diagram Pane, right-click the target join, and then click Properties.
   OR
   In the SQL Diagram Pane, double-click the target join.
   OR
   On the SQL Statement Tree, expand the Where and And nodes, and then double-click the target join.
2. Select options.
3. Click OK.

For more information, see Build Query.

Using Query Builder
Query Builder provides a visual display of your queries as you construct them. You can run Query Builder against any registered datasource in DBArtisan. Query Builder lets you build five separate types of queries simultaneously:

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE VIEW

You can execute a SELECT statement from Query Builder. To execute an INSERT, UPDATE, DELETE, and CREATE VIEW statement, copy them to an ISQL Editor. You can also copy the statements to the clipboard for later use in the ISQL Editor. Query Builder also lets you save a statement at any time so that you can open them later for editing or execution.
DBArtisan lets you open Query Builder with multiple tables or views with the same or different owners. If you open tables or views with different owners, Query Builder displays “All Owners” in the Owner drop-down list. You can start multiple Query Builder sessions from DBArtisan. You can use different tables and views for each query. You can also toggle back and forth among each of the queries.

You can save and reopen queries in Query Builder. Query Builder automatically checks for changes in your database or instance between the time you save the query and the time you reopen it with the Schema Change detection component.

Query Builder is integrated with Data Editor so you can edit data in real time and then continue to build your query with the new information embedded in the query.

Using Query Builder
To use Query Builder, do the following:

- Select an instance or database
- Select a statement.
- Select a table(s) or view(s).
- Select a column or columns.

**NOTE:** You can start Query Builder directly from a table or view which automatically selects the instance or database which contains that table or view.

1. On the Tools menu, click Query Builder.
   DBArtisan opens Query Builder.

Selecting a Database
To create an SQL statement, first select an instance or database.

**NOTE:** You can start Query Builder directly from a table or view which automatically selects the database which contains that table or view.

If you are working with Microsoft SQL Server or Sybase ASE, Query Builder provides two drop-down lists. The first drop-down list displays all available databases for the target server. The second drop-down list displays owners.

If you are working with Oracle, IBM DB2 for z/OS and OS/390 or IBM DB2 for Linux, Unix, and Windows the first drop-down list is unavailable.

**NOTE:** You can start Query Builder directly from a table or view which automatically selects the database which contains that table or view.

1. Start Query Builder.
2. In the database drop-down list, click the target instance or database.
3. In the owners drop-down list, select the appropriate owner.
   Query Builder is ready for Statement selection.
4. To select different instances or databases while Query Builder is running, on the Tables/Views Tab, in the database drop-down list, click the target instance or database.
   DBArtisan clears the current query and displays a warning prompt.
5 To save the current query, click **Yes**.
   DBArtisan opens the Save As dialog box.

6 To continue without saving, click **No**.
   DBArtisan clears the SQL Diagram Pane and SQL Statement Pane.

For more information on saving queries in Query Builder, see **Saving and Reopening Queries**.

Selecting a Statement
Query Builder lets you build **SELECT**, **INSERT**, **UPDATE**, **DELETE**, and **CREATE VIEW** queries simultaneously.

To select a statement, do the following:

1 On the **Query Builder** tool bar, click the statement drop-down list, and then click the target statement type.
   
   OR
   
   In the **SQL Diagram Pane**, right-click, and then click the target statement type.

For more information, see **Using Query Builder**.

Selecting Tables and Views
To build a query, open one or more tables or views in the **SQL Diagram Pane**. You can use different tables or views for each type of query.

   **TIP:** For multiple tables: Press **SHIFT+click** for adjacent tables or **CTRL+click** for nonadjacent tables.
   You can also drag the bounding line with your pointer to select multiple tables.

To select a Table or View, do the following:

1 In the **Tables/Views Tab**, drag the target table or view to the **Diagram Pane**.
   
   OR
   
   In the **Tables/Views Tab**, click target table or view and then, on the **Query Builder** tool bar, click **Add**.
   
   OR
   
   In the **Tables/Views Tab**, right-click target table or view, and then click **Add**.
   
   Query Builder displays the target table(s) and view(s) in the SQL Diagram Pane.

For more information, see **Using Query Builder**.

Selecting Columns
You must select at least one column to build a query. Query Builder displays columns in each table in the SQL Diagram window. By default, Query Builder exposes every column in a table. You can select the columns you want to use for your query. Query Builder orders them, in your statement, in the select order.

Query Builder lets you select columns in the:

   • **SQL Diagram Pane**.
   
   • **SQL Statement Tree**.

   You can select an individual column or all columns. Query Builder orders them, in your statement, in the select order.
   You can reorder columns after you set them in your diagram or statement.
For more information, see:

Selecting Columns in the SQL Diagram Pane
Selecting Columns in the SQL Statement Tree
Selecting All Columns

Selecting Columns in the SQL Diagram Pane
To select a column in the SQL Diagram Pane, do the following:

1. Select the check box to the left of the target column name.

For more information, see Using Query Builder.

Selecting Columns in the SQL Statement Tree
Query Builder lets you select and set individual properties using the Selected Column Properties Dialog Box.

For more information, see Using Query Builder.

Selecting All Columns
Query Builder uses columns in statements based on the order of selection. When you select all columns, Query Builder displays the columns in the order they appear in the table.

NOTE: Query Builder lets you select all columns in single or multiple tables.

To select all columns, do the following:

1. On the Query Builder menu, click Select Star.

For more information, see Using Query Builder.

Building a Query
Query Builder lets you build five different queries, which you can run separately or simultaneously, depending on your needs. The table below describes these queries:

<table>
<thead>
<tr>
<th>Query Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>Lets you create, manipulate and execute SELECT Statements for tables and views.</td>
</tr>
<tr>
<td>INSERT</td>
<td>Lets you create and manipulate INSERT Statements for tables.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Lets you create and manipulate UPDATE Statements for tables.</td>
</tr>
<tr>
<td>DELETE</td>
<td>Lets you create and manipulate DELETE Statements for tables.</td>
</tr>
<tr>
<td>CREATE VIEW</td>
<td>Lets you create and manipulate CREATE VIEW Statements for tables and views.</td>
</tr>
</tbody>
</table>

To build a Query, do the following:

- Select an instance or database.
- Select a Statement.
- Select your table(s) or view(s).
• **Select your columns.**

**NOTE:** You can start Query Builder directly from a table or view which automatically selects the database which contains that table or view.

Query Builder lets you build queries that include both **tables** and **views** in the SQL Diagram Pane for **SELECT** and **CREATE VIEW** statements. For the INSERT, UPDATE, and DELETE statements, use one or the other object, but you cannot use both.

Once you make your selections, you can edit, restructure, and streamline your query. Query Builder offers many options for streamlining your queries.

For more information, see:

- Working with Tables and Views in the SQL Diagram Pane
- Working with Columns in the SQL Diagram Pane
- Joins
- Creating a Clause using the SQL Statement Tree
- Moving Tables and Columns in the SQL Statement Tree
- Subqueries
- Aliases

**Building a SELECT Statement**

Query Builder lets you construct and execute simple-to-complex SELECT statements using data from any table or view. You can also create and edit **joins** for SELECT statements. Query Builder can check your query and warn you if there are syntax errors with the Syntax Checker.

To build a SELECT statement, do the following:

1. On the **Tools** menu, click **Query Builder**.
   - DBArtisan opens Query Builder.
2. In the statement drop-down list, click **SELECT**.
3. In the **Table Tree Pane**, select target table(s) or view(s) and move them to the **SQL Diagram Pane**.
4. In the target table or view, click target column(s), or click **Select Star** to select every column.
5. To check syntax, click **Check**.
6. To copy the statement, click **Copy**.
7. To execute the statement, click **Execute**.

**Copying a SELECT Statement from the SQL Statement Pane**

To copy any part of a statement from the SQL Statement Pane, do the following:

1. Open **Query Builder**, then begin a new SELECT statement.
   - OR
   - Open an existing SELECT statement.
2. In the **SQL Statement Pane**, select all, or the target portion of the statement.
On the **Query Builder** tool bar, click **Copy**.

**OR**

In the **SQL Statement Pane**, right-click, and then click **Copy**.

Query Builder makes the target statement portion available on the clipboard.

For more information, see Building a Query.

**Building an INSERT Statement**

Query Builder lets you construct and execute simple-to-complex INSERT statements using data from any table. To execute an INSERT statement, copy it to an ISQL Editor. You can also copy the statement to the clipboard for later use in the ISQL Editor. Query Builder also lets you save your statement at any time so that you can open it later for editing or execution.

**Building an INSERT Statement**

To build an INSERT Statement, do the following:

1. On the **Tools** menu, click **Query Builder**.
   
   **D**BArtisan opens Query Builder.

2. In the statement drop-down list, click **INSERT**.

3. In the **Table Tree** Pane, select target table, and move it to the **SQL Diagram Pane**.

4. In the target table, click target column(s).

**Copying an INSERT Statement from the SQL Statement Pane**

To copy any part of a statement from the SQL Statement Pane, do the following:

1. Open Query Builder, then begin a new INSERT statement.

   **OR**

   Open an existing INSERT statement.

2. In the **SQL Statement Pane**, select all, or the target portion of the statement.

3. On the **Query Builder** tool bar, click **Copy**.

   **OR**

   In the **SQL Statement Pane**, right-click, and then click **Copy**.

   Query Builder makes the target statement portion available on the clipboard.

For more information, see Building a Query.

**Building an UPDATE Statement**

Query Builder lets you construct and execute simple-to-complex UPDATE statement using data from any table. To execute an UPDATE statement, copy it to an ISQL Editor. You can also copy the statement to the clipboard for later use in the ISQL Editor. Query Builder also lets you save your statement at any time so that you can open it later for editing or execution.

**Building an UPDATE Statement**

To build an UPDATE Statement, do the following:

1. On the **Tools** menu, click **Query Builder**.
   
   **D**BArtisan opens Query Builder.

2. In the statement drop-down list, click **UPDATE**.

3. In the **Table Tree** Pane, select target table, and move it to the **SQL Diagram Pane**.

4. In the target table, click target column(s).
Building an UPDATE Statement

To build an UPDATE statement, do the following:

1. On the Tools menu, click Query Builder.
   DBArtisan opens Query Builder.
2. In the statement drop-down list, click UPDATE.
3. In the Table Tree Pane, select target table and move it to the SQL Diagram Pane.
4. In the target table, click target column(s).

Copying an UPDATE Statement from the SQL Statement Pane

To copy any part of a statement from the SQL Statement Pane, do the following:

1. Open Query Builder, then begin a new UPDATE statement.
   OR
   Open an existing UPDATE statement.
2. In the SQL Statement Pane, select all, or the target portion of the statement.
3. On the Query Builder toolbar, click Copy.
   OR
   In the SQL Statement Pane, right-click, and then click Copy.
   Query Builder makes the target statement portion available on the clipboard.

For more information, see Building a Query.

Building a DELETE Statement

Query Builder lets you construct DELETE statements using data from any table. Query Builder displays a Confirmation Option Message box when you create a DELETE statement. You can set the Statement Properties dialog box to display or hide this message when creating a DELETE statement.

To execute a DELETE statement, copy it to an ISQL Editor. You can also copy the statement to the clipboard for later use in the ISQL Editor. Query Builder also lets you save your statement at any time so that you can open it later for editing or execution.

Building a DELETE Statement

To build a DELETE statement, do the following:

1. On the Tools menu, click Query Builder.
   DBArtisan opens Query Builder.
2. In the statement drop-down list, click DELETE.
3. In the Table Tree Pane, select target table, and move it to the SQL Diagram Pane.
**Copying a DELETE Statement from the SQL Statement Pane**

To copy any part of a statement from the SQL Statement Pane, do the following:

1. Open Query Builder, then begin a new DELETE statement.
   
   OR
   
   Open an existing DELETE statement.

2. In the SQL Statement Pane, select all, or the target portion of the statement.

3. On the Query Builder toolbar, click Copy.
   
   OR
   
   In the SQL Statement Pane, right-click, and then click Copy.

Query Builder makes the target statement portion available on the clipboard.

For more information, see [Building a Query](#).

**Building a CREATE VIEW Statement**

Query Builder lets you construct and execute simple-to-complex CREATE VIEW statements using data from any table or view. You can also copy the statement to the clipboard for later use in the ISQL Editor. Query Builder also lets you save your statement at any time so that you can open it later for editing or execution.

To build a CREATE VIEW statement, do the following:

1. On the Tools menu, click Query Builder.
   
   DBArtisan opens Query Builder.

2. In the statement drop-down list, click CREATE VIEW.

3. In the Table Tree Pane, select target table or view and move it to the SQL Diagram Pane.
   
   **NOTE:** Query Builder supports multiple tables and views in a CREATE VIEW statement.

4. In the target table or view, click the target column(s).

5. To check syntax, click Check.

6. To copy the statement, click Copy.

7. To execute the CREATE VIEW Statement, click the SQL Statement Pane, and then press any key.
   
   Query Builder opens the Edit SQL dialog box.

8. Click OK.
   
   **CAUTION:** If you have used this method previously, and you selected the Please do not show me this dialog again check box, on the Edit SQL dialog box, Query Builder does not display the Edit SQL dialog box. It pastes your statement directly to the ISQL Editor.

   DBArtisan opens the ISQL Editor.

9. In the ISQL Editor, on the line, CREATE VIEW NAME AS, replace the word NAME with a name for your view.

10. On the tool bar, click Execute.
    
    DBArtisan executes the CREATE VIEW query.
11 To close the Editor, click **Close**.
    
    DBArtisan opens the ISQL Editor save message.

12 Click **No**.
    
    DBArtisan returns to Query Builder.

13 To add the view to the table tree, on the **Query Builder** menu, click **Refresh**.
    
    Query Builder adds the view to the Table Tree Pane.

### Copying a CREATE VIEW Statement from the SQL Statement Pane

To copy any part of a statement from the SQL Statement Pane, do the following:

1 Open Query Builder, then begin a new CREATE VIEW statement.
    
    OR
    
    Open an existing CREATE VIEW statement.

2 In the **SQL Statement Pane**, select all, or the target portion of the statement.

3 On the **Query Builder** tool bar, click **Copy**.
    
    OR
    
    In the **SQL Statement Pane**, right-click, and then click **Copy**.
    
    Query Builder makes the target statement portion available on the clipboard.

For more information, see Building a Query.

### Working with Tables and Views in the SQL Diagram Pane

Query Builder lets you organize your tables and views in the SQL Diagram Pane. You can also customize appearance, change visual aspects, and adjust layout while continuing to manufacture a query. You can resize or customize a selected table and view, or move them to the front or back of the diagram. The key symbol indicates a column that is indexed or participates in a primary key.

- Selecting and Deselecting Tables and Views
- Moving Tables and Views
- Moving Additional Tables and Views to the SQL Diagram Pane
- Deleting Tables and Views

Query Builder can automatically dictate a layout in the SQL Diagram Pane using the **Auto Layout** button.

### Selecting and Deselecting Tables and Views

You can select tables and views in the SQL Diagram Pane. You can make changes to more than one table or view simultaneously by selecting multiple tables or views.

To select and deselect Tables and Views, do the following:

1 To select a table, click the table title bar.

2 To select more than one table, drag the pointer to enclose all target tables with the bounding line.
    
    Query Builder selects all target tables; none have handles.
3 To select all tables, in the SQL Diagram, right-click, and then click Select All.

4 Click the SQL Diagram workspace to deselect all tables.

For more information, see Working with Tables and Views.

Moving Tables and Views
Query Builder lets you move tables and views in the SQL Diagram Pane. It also moves selections and joins with the tables and views.

To move Tables and Views, do the following:

1 To move a table or view, drag the title bar to the target location.

   NOTE: If you select more than one table or view, Query Builder moves all selected tables and views and any joins with the pointer.

For more information, see Working with Tables and Views.

Moving Additional Tables and Views to the SQL Diagram Pane
Query Builder sets tables and views in your statement in the order that you move them to the SQL Diagram Pane. Tables and views moved into the Diagram Pane appear first in your statement, including all joins connecting that table. To change the order of tables, move them back into the Table Tree and re-select them in the order in which you would like to join them.

Moving Additional Tables or Views
To move additional tables or views, do the following:

1 Click the target table or view and drag it to the Diagram Pane.

   For multiple tables or views: Use **SHIFT+click** for adjacent tables or views or use **CTRL+click** for non-adjacent tables and views.

   OR

   Click the target table or view, and then on the Query Builder tool bar, click Add.

   OR

   Right-click the target table or view, and then click Add.

   For multiple tables or views: Use **SHIFT+click** for adjacent tables or views or use **CTRL+click** for non-adjacent tables and views.

   NOTE: Moving a table or view to the SQL Diagram Pane is not available while a query is executing.

For more information, see Working with Tables and Views.
Deleting a Table or View
To delete tables from the SQL Diagram Pane, do the following:

1. Right-click the target table or view, and then click **Delete**.

   OR

   In the **SQL Diagram**, click target table or view, and then on the **Query Builder** tool bar, click **Delete**.

   OR

   In the **SQL Diagram**, right-click the target table or view, and then click **Delete**.

Query Builder deletes the table from the SQL Diagram, SQL Statement, and SQL Statement Tree.

For more information, see *Working with Tables and Views*.

---

Working with Columns in the SQL Diagram Pane
You can customize queries by selecting and deselecting columns in the SQL Diagram Pane. You can customize columns using the **Selected Column Properties dialog box**.

Selecting and Deselecting Columns
You can select and deselect columns in the SQL Diagram. Query Builder lets you select and deselect individual columns or all columns. Your results reflect the order of selection. You can change the order of columns after you set them in your diagram or statement.

**NOTE:** You can also select, re-order and deselect columns in the **SQL Statement Tree**.

**Selecting Individual Columns**
To select individual columns, do the following:

1. To select a column, in the **SQL Diagram**, select the check box to the left of the target column name.

**Deselecting Individual Columns**
To deselect individual columns, do the following:

1. To deselect a column, in the **SQL Diagram**, select the check box to the left of the target column name.

   **NOTE:** When you clear the columns, Query Builder deletes the columns and any sub clauses from the SQL Statement Pane and SQL Statement Tree.

**Selecting All Columns**
To select all columns, do the following:

1. On the **Query Builder** menu, click **Select Star**.

   **NOTE:** Query Builder uses columns in statements based on the order of selection. When you select all columns, Query Builder displays the columns as they appear in the table.

**Deselecting All Columns**
To deselect all columns, do the following:

1. On the **Query Builder** menu, click **Select None**.

   Query Builder adds or removes selected columns from the SQL Statement Tree and the SQL Statement Pane.
Selecting ALL or DISTINCT Columns

Selecting ALL or DISTINCT columns is a way to filter data in your query. Selecting ALL columns means all rows displays results in the grid regardless of duplication in non-primary key columns. The DISTINCT column function is a query process that limits duplicate data in non-primary key columns to rows with the first iteration of any identical data. For example, if there are two identical addresses for different last names, and the column with a primary key does not participate in the query, only the row with the first instance of the address displays in the results of the query.

To select ALL or DISTINCT columns, do the following:

1. In the Statement Tree pane, right-click the ALL or DISTINCT node, click Properties, and then select the ALL or DISTINCT check box.

   OR

2. In the Statement Tree pane, double click the ALL or DISTINCT node. Query Builder toggles to the opposite function.

   NOTE: You can change between ALL or DISTINCT at any time prior to executing or copying a query.

Joins

Joins let you distill the information in your database to a usable form. Query Builder lets you create, manipulate, and edit work with joins without requiring knowledge of the underlying SQL code. Query Builder lets you create any type of join for SELECT and CREATE VIEW Statements. You can create self joins for UPDATE or DELETE Statements. You cannot create joins for INSERT Statements.

Query Builder includes four types of joins. The table below describes joins and their availability in Query Builder:

<table>
<thead>
<tr>
<th>Join</th>
<th>Statement Availability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner Join</td>
<td>SELECT, CREATE VIEW, DELETE, UPDATE</td>
<td>Returns data from the joined tables that match the query’s join criteria and set a relation between tables or views. Inner joins return results where the join condition is true.</td>
</tr>
<tr>
<td>Left Outer Join</td>
<td>SELECT, CREATE VIEW</td>
<td>Returns all data from the primary table and data from the joined tables that match the query’s join criteria and set a join relation operator from a column in a primary table or view to a column in a secondary table or view.</td>
</tr>
<tr>
<td>Right Outer Join</td>
<td>SELECT, CREATE VIEW</td>
<td>Returns all data from the primary table and data from the joined tables that match the query’s join criteria and set a join relation operator from a column in a secondary table or view to a column in a primary table or view.</td>
</tr>
<tr>
<td>Self Join</td>
<td>SELECT, CREATE VIEW</td>
<td>Set a relation between columns in the same table.</td>
</tr>
</tbody>
</table>

In the Query Builder SQL Diagram Pane, you can create, edit, and delete joins. You can edit joins in the Join dialog box. You can set Query Builder options, in the Options Editor, to automatically create joins.

Joins are the way you can filter data in relational databases. Query Builder lets you change the types of joins between tables, views and columns. It is important that you have some knowledge of the data in your tables, and the datatypes for each column. This information helps you frame a better query, and filter your data for maximum effect.

For more information, see Joins.

Inner Joins

Inner joins are the most common types of joins for SELECT statements. An inner join returns information from two tables where the relation between two target columns is true for both columns.
The join operand determines the relation results, for example, if the join operand is equals, then identical data, in two columns, is the only result. If the join operand is not equals, Query Builder only returns data that is different between two columns.

For example, if you have an inner join matching territory numbers between the table dbo.Managers and dbo.Clients, running the query returns all Managers and Clients with matching territory numbers:

Query Builder displays the following results from this query with an inner join

**NOTE:** Query Builder displays results of columns in the order of selection. You can reorder columns by deselecting and selecting in the SQL Diagram Pane, the Selected Columns Properties dialog box, or the SQL Statement Tree.

For more information, see [Joins](#).

**Left Outer Joins**

Left outer joins bring back a different data set than inner joins. Left outer joins retrieve all the data in columns selected from the primary table, and only matching data from the joined or secondary table.

For example, in the same pair of tables, a left inner join from dbo.Managers to dbo.Clients, where the columns Current Territory and Territory are joined, displays different results.

**NOTE:** There is one additional manager who does not have a client, but because a left outer join includes all data from selected columns in the primary table, the last entry in the illustration is displayed.

For more information, see [Joins](#).

**Right Outer Joins**

Right outer joins return opposite results from a left outer join. In a right outer join, you are asking for all the information in the secondary table’s column, and the join operator’s matching information from the primary table.

For example, in the same set of data we used in the left outer join example, a right outer join returns all clients from dbo.Client, and only managers who match territory numbers, in the joined column.

**NOTE:** The managers are the same as the first, inner join, but a right outer join returns the additional clients without matching managers.

For more information, see [Joins](#).

**Self Joins**

A self join is a join within a single table. Query Builder lets you return specific information from a single table using a self join.

For example, in our example table, there is a column for the number of clients and another column with the goal client total for a territory.

A self join can ascertain which managers are reaching their quota. Notice that the join relation operator in the example is greater than or equal to, which shows managers exceeding quota as well.

For more information, see [Joins](#).
Adding and Deleting a Join in the SQL Diagram Pane
Query Builder lets you add and delete joins. This method adds a WHERE clause in your query. You can join different tables and or views in a SELECT or CREATE VIEW statement.

Adding a Join
To add a Join, do the following:

1. In the SQL Diagram Pane, drag the target column to the second column.

   Query Builder displays both a line joining the two columns in the SQL Diagram Pane and the corresponding SQL code in the SQL Statement Pane.

Removing a Join
Query Builder lets you remove joins from your query. Query Builder automatically deletes joins from the query in the SQL Statement Pane, when you remove them from the SQL Diagram Pane.

To remove a join, do the following:

1. Click the target join, and then on the Query Builder tool bar, click Delete.

   OR

   Right-click the target join, and then click Delete.

   Query Builder deletes the Join.

For more information, see Joins.

Editing Joins
Query Builder lets you edit joins with the Join editor. You can edit join parameters in a SELECT, UPDATE, DELETE, and CREATE VIEW Statement.

The table below describes the options in the Join dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Table Column</td>
<td>The primary column in the join.</td>
</tr>
<tr>
<td>To Table Column</td>
<td>The secondary column in the join.</td>
</tr>
<tr>
<td>Select the join relation operator</td>
<td>Click the target join operator. If it is not equals, the operator displays on the join in the SQL Diagram Pane.</td>
</tr>
<tr>
<td>Join Type: Inner</td>
<td>Click to make the join an inner join. Aggregates are only available for inner joins.</td>
</tr>
<tr>
<td>Join Type: Left Outer</td>
<td>Click to make the join a left outer join.</td>
</tr>
<tr>
<td>Join Type: Right Outer</td>
<td>Click to make the join a right outer join.</td>
</tr>
</tbody>
</table>
Completing the Join Dialog Box

1. In the SQL Diagram Pane, right-click the target join, and then click Properties.
   OR
   In the SQL Diagram Pane, double-click the target join.
   OR
   On the SQL Statement Tree, expand the Where and And nodes, and then double-click the target join.

Query Builder opens the Join dialog box.

**NOTE:** For IBM DB2 for Linux, Unix, and Windows and IBM DB2 for z/OS and OS/390 servers, there is an additional join object in the SQL Statement Tree. The Join On node displays join relations between columns in IBM DB2 for Linux, Unix, and Windows and IBM DB2 for z/OS and OS/390 tables and views.

Changing a Join Color

Query Builder lets you change the color at a join in the SQL Diagram Pane. Complex statements using many tables and multiple joins can be easier to view if joins have different colors.

To change the color of a join, do the following:

1. Right-click the target join, and then click Color.
   Query Builder opens the Color dialog box.

2. In the Basic colors grid, click a target color
   OR
   Click Define Custom Colors, then create a custom color.

   **NOTE:** Query Builder lets you save custom colors for the current color. Click Add to Custom Color to have the option of using that color for your queries.

3. Click OK.

For more information, see Joins.

Auto Layout

The Auto Layout function displays tables and views in the SQL Diagram Pane. It makes the best use of the available area in the SQL Diagram Pane by placing your tables and views in the most efficient manner. If the automatic join function is on, Query Builder displays all joins between columns in your diagram. Query Builder lets you run the automatic layout function any time you have tables or views in the SQL Diagram Pane.

Using Auto Layout

To use Auto Layout, do the following:

1. On the Query Builder menu, click Auto Layout.
   Query Builder organizes your tables in the SQL Diagram Pane.
Auto Joins
Query Builder includes an automatic join function that displays joins between selected tables and views in the SQL Diagram Pane. The Auto Join function seeks columns with the same name and data type. You can set global automatic join parameters in the DBArtisan Options Editor. You can use the Statement Properties Editor to set local join parameters for the current Query Builder session without changing the global parameters.

Using Auto Join
To use Auto Join, do the following:

1. On the Query Builder menu, click Auto Join.
   Query Builder joins columns in the SQL Diagram Pane.

Creating a Clause Using the SQL Statement Tree
Query Builder lets you build more detailed WHERE, ORDER BY, GROUP BY, and HAVING clauses using the SQL Statement Tree. Query Builder lets you add clauses to SELECT, UPDATE, DELETE, and CREATE VIEW statements.

   **NOTE:** Query Builder does not support clauses for INSERT statements.

The table below describes these clauses:

<table>
<thead>
<tr>
<th>Clause</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHERE</td>
<td>Limits rows in the query.</td>
</tr>
<tr>
<td>ORDER BY</td>
<td>Orders the results of the query to a target column.</td>
</tr>
<tr>
<td>GROUP BY</td>
<td>Groups target columns in the query.</td>
</tr>
<tr>
<td>HAVING</td>
<td>Filters out groups of data.</td>
</tr>
</tbody>
</table>

Creating a WHERE Clause
Query Builder lets you create a WHERE clause from the SQL Statement Tree which automatically displays in your query.

   **NOTE:** Any additional WHERE clauses are displayed as HAVING clauses.

The table below describes the options and functionality on the Where dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand (Left)</td>
<td>Lets you click the target column for the first part of your WHERE clause.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> Query Builder lists every column in all tables in the SQL Diagram in the Operand lists.</td>
</tr>
<tr>
<td>Operator</td>
<td>Lets you select the target operator.</td>
</tr>
<tr>
<td>Operand (Right)</td>
<td>Lets you click the target column for the second part of your WHERE clause.</td>
</tr>
<tr>
<td></td>
<td>Query Builder automatically writes the query language in the Statement option box.</td>
</tr>
</tbody>
</table>

   **NOTE:** Query Builder does not display clause phrases created from the SQL Statement Tree in the SQL Diagram Pane.
Creating a WHERE Clause
To Create a WHERE clause, do the following:

1. Click the WHERE node, and then on the Query Builder toolbar, click New.
   OR
   Right-click the WHERE node, and then click New.

For more information, see Creating a Clause using the SQL Statement Tree.

Deleting a WHERE Clause
To delete a WHERE clause, do the following:

1. Expand the AND node, and then on the Query Builder toolbar, click Delete.
   OR
   Expand the AND node, right-click target column and then click Delete.
   Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see Creating a Clause using the SQL Statement Tree.

Creating an AND Clause in a WHERE Clause
Query Builder lets you add an AND clause from the SQL Statement Tree which automatically displays in your query.

The table below describes the options and functionality on the Where dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand (Left)</td>
<td>Lets you click the target column for the first part of your WHERE clause.</td>
</tr>
<tr>
<td>Operator</td>
<td>Lets you select the target operator.</td>
</tr>
<tr>
<td>Operand (Right)</td>
<td>Lets you click the target column for the second part of your WHERE clause.</td>
</tr>
<tr>
<td></td>
<td>Query Builder automatically writes the query language in the Statement option box.</td>
</tr>
<tr>
<td>New Button</td>
<td>Click to clear your selections but remain in the Where dialog box.</td>
</tr>
<tr>
<td></td>
<td>Query Builder adds another AND clause to your query.</td>
</tr>
</tbody>
</table>

To open the Where dialog box, do the following:

1. Click the AND node, and then on the Query Builder toolbar, click New.
   OR
   Expand the WHERE node, right-click the AND node, and then click New.

For more information, see Creating a Clause using the SQL Statement Tree.
Deleting an AND Clause

To delete an AND clause, do the following:

1. Expand the **AND** node, click target column, and then on the **Query Builder** tool bar, click **Delete**.

   OR

   Expand the **AND** node, click target column, and then on the keyboard press **DELETE**.

   OR

   Expand the **AND** node, right-click the target column, and then click **Delete**.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see **Creating a Clause using the SQL Statement Tree**.

Inserting an AND or OR Clause

Query Builder lets you insert an AND or an OR WHERE clause from the SQL Statement Tree which automatically displays in your query. Query Builder lets you insert AND or OR clauses at any appropriate point in the SQL Statement Tree.

The table below describes the options and functionality on the Where dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand (Left)</td>
<td>Lets you click the target column for the first part of your WHERE clause.</td>
</tr>
<tr>
<td>Operator</td>
<td>Lets you select the target operator.</td>
</tr>
<tr>
<td>Operand (Right)</td>
<td>Lets you click the target column for the second part of your WHERE clause.</td>
</tr>
<tr>
<td></td>
<td>Query Builder automatically writes the query language in the Statement option box.</td>
</tr>
<tr>
<td>New Button</td>
<td>Click to clear your selections but remain in the Where dialog box.</td>
</tr>
<tr>
<td></td>
<td>Query Builder adds another AND clause to your query.</td>
</tr>
</tbody>
</table>

To insert an AND or OR Clause, do the following:

1. On the **SQL Statement Tree**, expand the **WHERE** node, right-click the target **AND** node, then click **Insert**, and then click **And** or **Or**.

For more information, see **Creating a Clause using the SQL Statement Tree**.

Deleting an OR Clause

To delete an OR clause, do the following:

1. Expand the **OR** node, and then on the **Query Builder** tool bar, click **Delete**.

   OR

   Expand the **OR** node, right-click the target column and then click **Delete**.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see **Creating a Clause using the SQL Statement Tree**.
Creating an ORDER BY Clause

Query Builder lets you create an ORDER BY clause from the SQL Statement Tree which automatically displays in your query.

The table below describes the Order By Columns dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Columns</td>
<td>Select target column(s) and click the right arrow. Query Builder moves target column from the Available Columns list to the Order By Columns list.</td>
</tr>
<tr>
<td>NOTES: Query Builder sorts query results based on the order that columns are placed in the ORDER BY clause.</td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td>Lets you select the target sort order. ASC - Ascending  DESC - Descending  Query Builder displays the SQL language in the Order By Statement box.</td>
</tr>
</tbody>
</table>

To open the Order By Columns dialog box, do the following:

1. On the SQL Statement Tree, click the ORDER BY node, and then on the Query Builder tool bar, click Properties.
   OR
1. On the SQL Statement Tree, right-click the ORDER BY node, and then click Properties.

For more information, see Creating a Clause using the SQL Statement Tree.

Changing the Sort Order in an ORDER BY Clause

To quickly change the sort order of a column in a query, do the following:

1. On the SQL Statement Tree, expand the ORDER BY node, and then double-click the target column.
   OR
1. On the SQL Statement Tree, expand the ORDER BY node, then right-click the target column, and then click Properties.

Query Builder opens the Order dialog box.

2. Click the target sort order, and then click OK.

Query Builder appends the Order By clause for target column with the appropriate sort order in the SQL Statement Pane.

For more information, see Creating a Clause using the SQL Statement Tree.
Deleting an ORDER BY Clause
To delete an ORDER BY clause, do the following:

1. Expand the ORDER BY node, and then on the Query Builder toolbar, click Delete.

   OR

   Expand the ORDER BY node, right-click the target column, and then click Delete.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see Creating a Clause using the SQL Statement Tree.

Creating a GROUP BY Clause
The table below describes the options and functionality on the Group By Columns dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Selected Columns     | Select target column(s) and click the right arrow. Or click the Select All button. Query Builder moves target column from the Selected Columns list to the Group By Columns list.  
                        *NOTE:* Query Builder sorts query results based on the order that columns are placed in the ORDER BY clause. |
| Clear All Button     | Click to move target column from the Group By Columns list to the Selected Columns list. Query Builder displays the SQL language in the Group By Statement window. |

Creating a GROUP BY Clause
To create a GROUP BY clause from the SQL Statement Tree which automatically displays in your query, do the following:

1. On the SQL Statement Tree, double-click the GROUP BY node.

   OR

   On the SQL Statement Tree, right-click the GROUP BY node, and then click New.

Query Builder adds all the selected columns in your table(s) to the GROUP BY node in the SQL Statement Tree, and to the appropriate location in the SQL Statement Pane.

2. On the GROUP BY node, double-click any column.

   OR

   On the GROUP BY node, click any column, then on the Query Builder menu, click New.

   OR

   On the GROUP BY node, right-click any column, then click Properties.

For more information, see Creating a Clause using the SQL Statement Tree.
Deleting a GROUP BY Clause
To delete a GROUP BY clause, do the following:

1. On the SQL Statement Tree expand the GROUP BY node, and then on the Query Builder tool bar, click Delete.

OR

On the SQL Statement Tree Expand the GROUP BY node, right-click the target column, and then click Delete.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see Creating a Clause using the SQL Statement Tree.

Creating a HAVING Clause
A HAVING clause is a type of WHERE clause. It filters additional information from your tables. Query Builder lets you create a HAVING clause from the SQL Statement Tree which automatically displays in your query. Query Builder lists every column in all tables in the SQL Diagram in the Operand lists. Query Builder displays the datatype of a column in the operand boxes.

The table below describes the options and functionality on the Having dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand (Left)</td>
<td>Lets you click the target column for the first part of your HAVING clause.</td>
</tr>
<tr>
<td>Operator</td>
<td>Lets you select the target operator.</td>
</tr>
<tr>
<td>Operand (Right)</td>
<td>Lets you click the target column for the second part of your HAVING clause.</td>
</tr>
<tr>
<td></td>
<td>Query Builder automatically writes the query language in the Statement option box.</td>
</tr>
<tr>
<td>New Button</td>
<td>Click to clear your selections but remain in the Having dialog box.</td>
</tr>
<tr>
<td></td>
<td>Query Builder adds another AND clause to your query.</td>
</tr>
</tbody>
</table>

**NOTE:** Query Builder does not display clause phrases created from the SQL Statement Tree in the SQL Diagram Pane.

To create a HAVING clause, do the following:

1. On the SQL Statement Tree, expand the HAVING node, and then expand the And node. If there is not a join listed on the And node, double-click And. If there is a join listed, use the shortcut option below.

OR

On the SQL Statement Tree, right-click the HAVING node, and then click New.

For more information, see Creating a Clause using the SQL Statement Tree.

Deleting a HAVING Clause
To delete a HAVING clause, do the following:

1. On the SQL Statement Tree expand the HAVING node, and then on the Query Builder tool bar, click Delete.

OR

On the SQL Statement Tree expand the HAVING node, right-click the target column, and then click Delete.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.
Changing Tables and Columns Location in the SQL Statement Tree

Query Builder lets you move tables and columns on the SQL Statement Tree by dragging them to new locations. You can move columns from the AND and OR nodes to an AND or OR node on the WHERE and HAVING clause nodes. Query Builder changes the query in the SQL Statement Pane to match each move. Query Builder moves tables or columns you are dragging below target table or column.

To move a table or column in the SQL Statement Tree, do the following:

1. Expand target node, then drag the target table or column to a new location.
   
   Query Builder makes the appropriate change in the query in the SQL Statement Pane.

   **NOTE:** Query Builder lets you select multiple tables or columns.

2. To move a table or column to the bottom of a node, drag it to the target node.
   
   Query Builder displays the target table or column at the bottom of target node.

For more information, see *Creating a Clause using the SQL Statement Tree*.

Subqueries

Query Builder lets you build subqueries for SELECT and CREATE VIEW statements in the WHERE or HAVING clause. The table below describes the options available for a subquery in Query Builder:

<table>
<thead>
<tr>
<th>Operand</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXISTS</td>
<td>Left operand</td>
<td>Specifies data that exists in a column.</td>
</tr>
<tr>
<td>NOT EXISTS</td>
<td>Left operand</td>
<td>Specifies data that does not exist in a column.</td>
</tr>
<tr>
<td>ANY</td>
<td>Right operand</td>
<td>Specifies data satisfying the operator parameters.</td>
</tr>
<tr>
<td>ALL</td>
<td>Right operand</td>
<td>Specifies data satisfying the operator parameters.</td>
</tr>
<tr>
<td>SELECT</td>
<td>Right operand</td>
<td>Specifies data satisfying the operator parameters.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the Where or Having dialog boxes.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand (Left)</td>
<td>Lets you click the target column for the first part of your clause.</td>
</tr>
<tr>
<td>Operator</td>
<td>Lets you select the target operator.</td>
</tr>
<tr>
<td>Operand (Right)</td>
<td>Lets you click the target column for the second part of your clause. Query Builder displays the working subquery in the Statement window.</td>
</tr>
<tr>
<td>Subquery</td>
<td>Paste or type the SUBQUERY statement.</td>
</tr>
</tbody>
</table>

To use the WHERE and HAVING dialog boxes to create subqueries, do the following:

1. On the SQL Statement Tree, expand the Where or Having node, and then expand the And node. If there is not a join listed on the And node, double-click And. If there is a join listed, use the shortcut option below.

   OR

   On the SQL Statement Tree, right-click the Where or Having node, and then click New.
For more information, see Creating a Clause using the SQL Statement Tree.

Syntax Checker
The Syntax Checker scans SQL statements for errors. You can check your syntax at any time while you are fashioning a query, or a Procedure or Function. Query Builder can automatically run a syntax check to validate your query when you are executing or copying a statement. The options for the Syntax Checker tool are set in the Options Editor.

NOTE: Query Builder lets you continue with your query even if there are errors detected in the syntax.

Using the Syntax Checker
The table below describes the possible syntax errors the Query Builder Syntax Checker tool displays, in order:

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the query contain duplicate aliases?</td>
<td>Query Builder returns an error message when it detects duplicate aliases.</td>
</tr>
<tr>
<td>If the query has a HAVING clause, is there a GROUP BY clause?</td>
<td>Query Builder returns an error message when it detects a HAVING clause without a GROUP BY clause.</td>
</tr>
<tr>
<td>If there are aggregates, or a GROUP BY clause, are all columns in one or the other?</td>
<td>Query Builder returns an error message when it detects an aggregate, or a GROUP BY clause without all columns in one or the other.</td>
</tr>
<tr>
<td>Are there joins against non-indexed columns, or columns not participating in a primary key?</td>
<td>Query Builder returns a warning when it detects a join against a non-indexed column, or a column not participating in a primary key.</td>
</tr>
<tr>
<td>Are there joins between different datatypes?</td>
<td>Query Builder returns a warning when it detects a join between different datatypes.</td>
</tr>
<tr>
<td>Are there cross-products in the query?</td>
<td>Query Builder returns a warning when it detects a cross-product in the query.</td>
</tr>
</tbody>
</table>

Saving and Reopening Queries
You can save and reopen queries in Query Builder. Saving a query saves the SQL Diagram, SQL Statement, and Query Builder Explorer view. Query Builder automatically checks for changes in your database or instance between the time you save the query and the time you reopen it. Query Builder prompts you to save any time you try to close Query Builder, or any time you attempt quitting DBArtisan.

Query Builder runs Schema Change detection any time you set a query to execute, refresh the data, or open a saved query.

DBArtisan lets you open multiple saved queries simultaneously.

Saving Queries
To save a query using standard Save and Save As functions, do the following:

1. On the File menu, click Save or Save As.
   DBArtisan opens the Save As dialog box.
2 In the **File name** box, type the name of the query.

**NOTE:** By default, the product appends the .qbl extension to Query Builder files. If there is more than one Query Builder session in progress when you save, the file is further appended with an integer, for example.qbl2.

**TIP:** DBArtisan lets you save data in text (*.txt) and XML (*.xml) file formats.

3 Click **OK**. DBArtisan saves the file and closes the Save As dialog box.

**Reopening Queries**

You can open a query using standard Open functions. Query Builder displays the Query Builder diagram, statement and Query Builder Explorer Pane and it checks the instance or database for schema changes.

The Query Builder Schema Change Detection component checks for:

- Renamed or dropped tables referenced in the query. Renamed tables that have been renamed are considered dropped.
- Renamed or dropped columns referenced in the query. Renamed columns are considered dropped and inserted.
- **Columns** added or reordered in tables referenced in the query.

If Query Builder detects a change, it opens the Schema Change Detected dialog box. The dialog box displays details of changes to your schema.

Query Builder opens an ISQL Editor with the last saved versions of the SQL statement.

**Using Data Editor with Query Builder**

SELECT statements. DBArtisan lets you open multiple Data Editor sessions so that you can continue to change your data until you find the best match for your query.

**CAUTION:** Data Editor is a real-time editor. Changes in your data using Data Editor are permanent.

**Opening the Data Editor from Query Builder**

To open the Data Editor from Query Builder, do the following:

1 On the **Tools** menu, click **Query Builder**. DBArtisan opens Query Builder.
2 Select a **database** or instance.
3 Select a **table**.
4 Select a **column**, or columns.
5 On the **Query Builder** menu, click **Edit Data**. DBArtisan opens Data Editor.

For more information on using the Data Editor, see Using Data Editor.
Import Data

It is often necessary to import data into database tables from external sources. You may need to bring sample data into a database to establish test case scenarios or import data for use in the real world. Business analysts, for example, often use spreadsheets from outside sources. Putting such data into tables can help them perform detailed queries and make informed decisions based on meaningful data.

**NOTE:** When you import a text file, the wizard may give the impression of stalling. Press ESC and you should be able to continue without further incident.

To Open the Import Data Wizard:

1. On the Tools menu, click Import Data.

The fields you may encounter as you move through the wizard are described in the table that follows:

### Data Properties

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the file to be used in this data load operation.</td>
<td>Type or browse to the file you will be importing the data from</td>
</tr>
<tr>
<td>Which table do you want to load data into?</td>
<td>Catalog: The collection of tables for Sybase or MS SQL. This is where you want to import the data to.</td>
</tr>
<tr>
<td>Catalog</td>
<td>Schema: The user.</td>
</tr>
<tr>
<td>Schema</td>
<td>Table: The particular table where you want to import the data.</td>
</tr>
<tr>
<td>First Row Contains Field Names</td>
<td>By choosing this, if the first row contains column names, these will be displayed as the first row of the file. You can still change column mappings if you want. If you do not select this, generic Field 1, Field 2, etc. will indicate column mappings.</td>
</tr>
</tbody>
</table>

### Delimeter

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What character delimitated the columns in the data file?</td>
<td>Choose between Tab, Semicolon, Comma, Space, or Other. If you choose Other, you must indicate the specific delimiting character.</td>
</tr>
</tbody>
</table>

### Column Mapping

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NOTE:</strong> You must map all mandatory (NOT NULL) column fields to a column heading before you can enable the Finish button and import the data.</td>
<td>To map columns, click the grey bar at the top of the grid and a drop-down list appears. The names in this drop-down list are columns in the table that's going to receive the data. These need to sync up to the data that's being imported. Red, or mandatory columns, turn blue in the column list when they have been assigned. A maximum of 10 rows are displayed here to make mapping columns easier. All rows will be imported when the job completes. If there are more columns in the file you are importing than in the receiving table, the extra columns will be ignored and you can create the insert statements. You cannot complete the import operation if columns are mapped more than once. The import operation will not let you generate invalid columns.</td>
</tr>
</tbody>
</table>

NOTE: You must map all mandatory (NOT NULL) column fields to a column heading before you can enable the Finish button and import the data.
Excel Column Mapping

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheet</td>
<td>If the file you selected has more than one sheet, identify the worksheet that has the data you want to import.</td>
</tr>
<tr>
<td>Start/End Cell</td>
<td>Identify the starting and ending cells of data you want to import and click Refresh. Or, accept the default cells.</td>
</tr>
<tr>
<td>First Row Contains Field Names</td>
<td>By choosing this, if the first row contains column names, these will be displayed as the first row of the file. You can still change column mappings if you want. If you do not select this, generic Field 1, Field 2, etc. will indicate column mappings.</td>
</tr>
</tbody>
</table>

**NOTE:** You must map all mandatory (NOT NULL) column fields to a column heading before you can enable the Finish button and import the data.

To map columns, click the grey bar at the top of the grid and a drop-down list appears. The names in this drop-down list are columns in the table that's going to receive the data. These need to sync up to the data that's being imported. Red, or mandatory columns, turn blue in the column list when they have been assigned.

If there are more columns in the file you are importing than in the receiving table, the extra columns will be ignored and you can create the insert statements.

You cannot complete the import operation if columns are mapped more than once. The import operation will not let you generate invalid columns.

Data Editor

The Edit Data function opens the Data Editor. You can use the Data Editor to edit your tables in real-time. The Data Editor supports all editable datatypes and is an alternative way to add, edit, or delete data from your tables.

**NOTE:** You can use Data Editor within Query Builder to edit data in tables while you create SELECT statements. You can open multiple Data Editor sessions so that you can continue to change your data until you find the best match query.

The Data Editor includes a Data Editor Filter that lets you select the columns in your table that you want to edit. You must select at least one column to use the Data Editor. The Data Editor Filter is not available for the Query Builder.

**TIP:** You can customize Data Editor options in the Options Editor.

For more information, see:

- Data Editor Design
- Using Data Editor

Data Editor Design

The Data Editor includes the following components:

- Edit Window
- ISQL Window
- Tool Bar
- Data Editor Filter
Data Editor Edit Window
Data Editor displays all the information in the target table in the Data Editor Edit Window. You can edit data directly in this window.

For more information, see:
- ISQL Window
- Tool Bar
- Data Editor Filter
- Date/Time Format Builder
- Using Data Editor

Data Editor ISQL Window
The Data Editor ISQL Window displays the active SQL statement, which uses the data from the target table.

When appropriate, Data Editor displays a History Tab. The History Tab displays all SQL Statements created in the current session. If there is an error, Data Editor displays an Error Tab. The Error Tab details any errors in data entry encountered during execution.

For more information, see:
- Edit Window
- Tool Bar
- Data Editor Filter
- Date/Time Format Builder
- Using Data Editor

Data Editor Tool Bar
The Data Editor tool bar lets you access commonly used features.

The table below describes the function of each Data Editor tool.

<table>
<thead>
<tr>
<th>Description</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop Button.</td>
<td>Stops loading data to the Data Editor. Data Editor displays rows up to the stopping point.</td>
</tr>
<tr>
<td>List of options for the target table.</td>
<td>Displays the editing mode for the target table.</td>
</tr>
<tr>
<td>Execute SQL button</td>
<td>Executes the current SQL statement for the target table.</td>
</tr>
<tr>
<td>Insert Record button</td>
<td>Inserts new record for the target table. New records display at the end of the table.</td>
</tr>
<tr>
<td>Save Current Row button</td>
<td>Saves data in the current selected row. Data Editor prompts to save when you attempt to leave a row in Live mode.</td>
</tr>
</tbody>
</table>
### Data Editor

<table>
<thead>
<tr>
<th>Description</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove Data button</td>
<td>Removes data in target row. Data Editor displays an optional prompt.</td>
</tr>
<tr>
<td>Clear SQL Text button</td>
<td>Clears SQL text from the SQL Statement Pane.</td>
</tr>
<tr>
<td>Undo button</td>
<td>Undoes the most recent operation.</td>
</tr>
<tr>
<td>Redo button</td>
<td>Redoes the most recent operation.</td>
</tr>
<tr>
<td>First Record button</td>
<td>Moves to the first record in the target table.</td>
</tr>
<tr>
<td>Last Record button</td>
<td>Moves to the final record in the target table.</td>
</tr>
<tr>
<td>Filter Data button</td>
<td>Filters table using the target cell as the filter parameter.</td>
</tr>
<tr>
<td>Refresh button</td>
<td>Reloads data for target table</td>
</tr>
<tr>
<td>Calendar button</td>
<td>Sets correct format for target date/time cell. Enables the Calendar window.</td>
</tr>
<tr>
<td>Date/Time Format Builder button</td>
<td>Opens the <a href="#">Date/Time Format Builder dialog box</a>.</td>
</tr>
<tr>
<td>Date/Time Format Undo button</td>
<td>Undoes the last date/time format display.</td>
</tr>
<tr>
<td>Date/Time Format Redo button</td>
<td>Redoes the last date/time format display.</td>
</tr>
<tr>
<td>Close button</td>
<td>Closes and exits Data Editor.</td>
</tr>
</tbody>
</table>

For more information, see:

- [Edit Window](#)
- [ISQL Window](#)
- [Data Editor Filter](#)
- [Date/Time Format Builder](#)
- [Using Data Editor](#)

### Data Editor Filter

The Data Editor Filter displays the columns of a target table and the corresponding SELECT SQL Statement. You can select columns from the filter for selective data editing.

For more information, see:

- [Edit Window](#)
- [ISQL Window](#)
- [Tool Bar](#)
- [Date/Time Format Builder](#)
- [Using Data Editor](#)
- [Extract Data as XML](#)
Using Data Editor

Data Editor lets you edit data in your tables with any editable datatype without leaving the parent application. Data Editor lets you use your new data immediately.

**CAUTION:** Data Editor is a real-time editor. Changes in your data using Data Editor are permanent.

The table below describes the functions and options of the Data Editor:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Live</td>
<td>Edits data one row at a time. You must execute when you leave the row.</td>
</tr>
<tr>
<td>Batch</td>
<td>Edits data in multiple rows before executing.</td>
</tr>
</tbody>
</table>

**NOTE:** You can also use the Data Editor to edit date and time functions in a table.

**NOTE:** If you make an incorrect edit in a cell, Data Editor displays the error on the Error Tab of the ISQL Editor. Data Editor does not incorporate this error(s) in data into the table. Data Editor saves any changes in data prior to the error message.

**CAUTION:** Data Editor is a real-time editor. Changes in your data using Data Editor are permanent.

For more information, see:

- Edit Window
- ISQL Window
- Tool Bar
- Data Editor Filter
- Date/Time Format Builder

Editing Date and Time Functions

The Data Editor lets you edit date and time functions in a table. Data Editor uses a calendar tool to guarantee accurate input for date and time data. You can also change the display of date and time using the Date/Time Format Builder.

For more information, see:

- Calendar Button
- Date/Time Format Builder

Date/Time Format Builder

The Date/Time Format Builder lets you customize your date/time display. The Data Editor uses this format to display your dates and times. You control how the Data Editor displays the dates and time by using predefined formats, or by customizing a format to fit your needs.
The Data Editor uses the default date/time format of your Operating System. If you do not make any global changes in DBArtisan, the Date/Time Format Builder displays dates and times using the default formats of your operating system. If you make changes to dates and times in the Data Editor, DBArtisan commits the changes in the format used by the database.

**NOTE:** The changes you make using the Date/Time Format Builder do not affect the way your database stores dates and times.

### Editing the Date/Time Display
You can edit the date/time display on a global, table, or column level. The table below describes the different ways you can edit your date/time format display:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>Lets you make global changes to the Data Editor date display from the Data Editor Tab of the <a href="#">Options Editor</a>.</td>
<td>Options Editor</td>
</tr>
<tr>
<td>Grid</td>
<td>Lets you make changes to the date display of the entire Data Editor grid for that session only.</td>
<td>Data Editor grid</td>
</tr>
<tr>
<td>Column</td>
<td>Lets you make changes to the date display of a single column in the Data Editor for that session only.</td>
<td>Data Editor column</td>
</tr>
</tbody>
</table>

**NOTE:** Date/Time formats changed on a table or column level are valid for that session only.

### Editing Date/Time Globally
You can use the Options Editor to make global changes to your date/time display in the Data Editor. When you change the date/time format, using the Options Editor, the Data Editor displays all dates and times in the global manner. To change the date/time display for a particular session, see [Editing Grid Date/Time Format](#) or [Editing Column Date/Time Format](#).

To edit the date and time globally, do the following:

1. On the **File** menu, click **Options**. DBArtisan opens the Options Editor.
2. On the **Options Editor**, click the **Data Editor Tab**.
3. On the **Data Editor Tab**, click **...** DBArtisan opens the Date/Time Format Builder dialog box.
4. On the **Date/Time Format Builder** dialog box, click the **Date/Time Format** list, and then click the target predefined date/time format.
5. To customize the date/time format to your specifications, click **Customize**. DBArtisan opens the Date/Time Format Builder dialog box.
6. On the **Date/Time Format Builder** dialog box, select the appropriate Date/Time Format Options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date/Time Format</td>
<td>Displays the predefined Date/Time format.</td>
</tr>
<tr>
<td>Day Format</td>
<td>Lets you choose the day display.</td>
</tr>
<tr>
<td>Separator</td>
<td>Lets you choose the display separator between the day, month, and year.</td>
</tr>
</tbody>
</table>
When you have finished selecting the Date/Time format options, click **OK**.

DBArtisan accepts the date/time format changes and closes the Date/Time Format Builder dialog box.

8. On the **Options Editor**, select the appropriate Default Date/Time Format options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Calendar Control as default</td>
<td>If selected, DBArtisan uses the <a href="#">Calendar Control</a> window.</td>
</tr>
<tr>
<td>Two-digit year system setting warning</td>
<td>If selected, DBArtisan sends a warning when you use a two-digit year system setting.</td>
</tr>
</tbody>
</table>

9. Click **OK**.

DBArtisan accepts the Default Date/Time Format changes and closes the Options Editor.

**NOTE:** To use a different format for a particular session, change the date/time at the session level.

**Editing Grid Date/Time**

You can change the date/time display for a particular session when working in the **Data Editor**. The Data Editor does not maintain the format changes once you close your session. To make this display permanent, use the **Editing Global Date/Time Format**.

To edit the grid date and time, do the following:

1. On the **Datasource Explorer**, select the target table.

2. Right-click the table, and then click **Edit Data**.

DBArtisan opens the **Data Editor**.

3. On the **Data Editor** tool bar, click **Date/Time Format Builder**.

DBArtisan opens the Date/Time Format Builder.

4. On the **Date/Time Format Builder**, click the **Date/Time Format** list, and then click the target predefined date/time format.

5. To customize the date/time format to your specifications, click **Customize**.

DBArtisan opens the Date/Time Format Builder dialog box.
6 On the **Date/Time Format Builder** dialog box, select the appropriate Date/Time Format Options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date/Time Format</td>
<td>Displays the predefined Date/Time format.</td>
</tr>
<tr>
<td>Day Format</td>
<td>Lets you choose the day display.</td>
</tr>
<tr>
<td>Separator</td>
<td>Lets you choose the display separator between the day, month, and year.</td>
</tr>
<tr>
<td>Month Format</td>
<td>Lets you choose the month display.</td>
</tr>
<tr>
<td>Year Format</td>
<td>Lets you choose the year display.</td>
</tr>
<tr>
<td>Date Order</td>
<td>Lets you choose the date order display.</td>
</tr>
<tr>
<td>Hour Format</td>
<td>Lets you choose the hour display.</td>
</tr>
<tr>
<td>Minute</td>
<td>Lets you choose the minute display.</td>
</tr>
<tr>
<td>Sec Format</td>
<td>Lets you choose the second display.</td>
</tr>
<tr>
<td>AM/PM</td>
<td>Lets you choose the AM/PM display.</td>
</tr>
<tr>
<td>Date/Time Order</td>
<td>Lets you choose the date/time order display.</td>
</tr>
<tr>
<td>Format Display</td>
<td>Displays the current format.</td>
</tr>
<tr>
<td>Sample</td>
<td>Displays a sample of the current format.</td>
</tr>
</tbody>
</table>

7 When you have finished selecting the Date/Time format options, click **OK**.

DBArtisan accepts the date/time format changes and closes the Date/Time Format Builder dialog box.

- To undo changes, on the **Data Editor** tool bar, click **Undo Change**.
- To redo changes, on the **Data Editor** tool bar, click **Redo Change**.

**NOTE:** Date/Time formats changed on a table level are valid for that session only.

### Editing Column Date/Time

You can change the date/time display for a particular column when working in the **Data Editor**. The Data Editor does not maintain the format changes once you close your session. To change the format for the entire grid, see **Editing Grid Date/Time Format**. To make this display permanent, see **Editing Global Date/Time Format**.

To edit the column date and time, do the following:

1. On the **Datasource Explorer**, select the target table.
2. Right-click the table, and click **Edit Data**.
   
   DBArtisan opens the **Data Editor**.
3. On the **Data Editor**, click the column header to select the column.
4. Right-click the column and click **Format**.
   
   DBArtisan opens the Date/Time Format Builder.
5. On the **Date/Time Format Builder** dialog box, click the **Date/Time Format** list, and then click the target predefined date/time format.
6. To customize the date/time format to your specifications, click **Customize**.
   
   DBArtisan opens the Date/Time Format Builder dialog box.
7 On the Date/Time Format Builder dialog box, select the appropriate Date/Time Format Options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date/Time Format</td>
<td>Displays the predefined Date/Time format.</td>
</tr>
<tr>
<td>Day Format</td>
<td>Lets you choose the day display.</td>
</tr>
<tr>
<td>Separator</td>
<td>Lets you choose the display separator between the day, month, and year.</td>
</tr>
<tr>
<td>Month Format</td>
<td>Lets you choose the month display.</td>
</tr>
<tr>
<td>Year Format</td>
<td>Lets you choose the year display.</td>
</tr>
<tr>
<td>Date Order</td>
<td>Lets you choose the date order display.</td>
</tr>
<tr>
<td>Hour Format</td>
<td>Lets you choose the hour display.</td>
</tr>
<tr>
<td>Minute</td>
<td>Lets you choose the minute display.</td>
</tr>
<tr>
<td>Sec Format</td>
<td>Lets you choose the second display.</td>
</tr>
<tr>
<td>AM/PM</td>
<td>Lets you choose the AM/PM display.</td>
</tr>
<tr>
<td>Date/Time Order</td>
<td>Lets you choose the date/time order display.</td>
</tr>
<tr>
<td>Format Display</td>
<td>Displays the current format.</td>
</tr>
<tr>
<td>Sample</td>
<td>Displays a sample of the current format.</td>
</tr>
</tbody>
</table>

8 When you have finished selecting the Date/Time format options, click **OK**.

DBArtisan accepts the date/time format changes and closes the Date/Time Format Builder dialog box.

- To undo changes, on the **Data Editor** tool bar, click **Undo Format**.
- To redo changes, on the **Data Editor** tool bar, click **Redo Format**.

**NOTE:** Date/Time formats changed on a column level are valid for that session only.

Page Setup
The table below describes the options and functionality on the Page Setup dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Margins</td>
<td>Lets you select the size of the left, right, top, and bottom margins.</td>
</tr>
<tr>
<td>Titles and Gridlines</td>
<td>Lets you select options.</td>
</tr>
<tr>
<td>Preview</td>
<td>Displays how the table will appear when printed.</td>
</tr>
<tr>
<td>Page Order</td>
<td>Lets you specify when to print columns and rows.</td>
</tr>
<tr>
<td>Center on Page</td>
<td>Lets you select how table floats on the page.</td>
</tr>
</tbody>
</table>

Embarcadero Products
The Tools menu lists all installed Embarcadero Technologies products. This lets you toggle to or start another Embarcadero product.
Performance Center

If you have Performance Center installed on a server, DBArtisan lets you view performance metrics across all of your Performance Center monitored datasources.

**NOTE:** This feature requires only a [Performance Center server definition](#). No client upgrade is required.

Performance Center offers an automated strategy for identifying problems that threaten the availability and performance of critical Oracle®, Sybase and Microsoft SQL Server databases. Embarcadero Performance Center ensures that relational database management systems enjoy maximum up-time and exhibit performance that exceeds user expectations. It quickly identifies performance problems, and helps prioritize performance-tuning workloads to keep your databases performing at peak efficiency.

**TIP:** On the Perf Center Tab of the Options Editor, you can specify your Performance Center server definition and whether the DBArtisan opens the standard client (PerfCntr.exe) or the Web Client URL.

For more information, [Using the Performance Center Web Client Within DBArtisan](#).

---

Using the Performance Center Web Client Within DBArtisan

To open the Performance Center web client, do the following:

1. On the **Tools** toolbar, select **Performance Center**.
   
   **OR**
   
   On the **Tools** menu, select **Performance Center**.

   The first time you launch the Performance Center Web Client from DBArtisan, DBArtisan opens the Select Performance Center Server dialog box.

2. **In Computer Name or IP Address** specify where the Performance Center Server is installed.

3. **In Port Number** specify the port for the Apache or IIS server.

4. **To verify the settings, click Test**.

   DBArtisan displays the Performance Center Server Test Results dialog box.

5. **Click Close**.

   DBArtisan returns to the Select Performance Center Server dialog box.

6. **To hold these setting for future use, click OK**.

   DBArtisan opens Performance Center.

---

Code Workbench

The Code Workbench tool reduces the time needed for the most common day-to-day coding tasks. The Code Workbench lets you:

- Enable table column auto lookup and population in the ISQL Window.
- Define auto replacement expressions that can be used to quickly insert commonly used blocks of SQL syntax or commands.
• Import and Export Code Workbench settings for client sharing purposes.

**NOTE:** No privileges are required to use the Code Workbench options.

**Important Notes**
When installing DBArtisan on a machine that already has Rapid SQL 7.3 or DBArtisan 7.3 or later, the Code Workbench settings are not installed by default. This is to protect any modifications that have been made to the current settings. To import the settings that are shipped with the latest release, select Tools, Code Workbench. Select the Import Settings option, and then select C:\Program Files\Embarcadero\DBArtisan\codeworkbench.xml.

**CAUTION:** This will overwrite the current settings.

The table below describes the options and functionality on the Code Workbench dialog box.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Settings</td>
<td>Enable Column Auto Lookup</td>
<td>Lets you easily add columns to ISQL Window queries without having to type. The ISQL Window auto-populates table columns.</td>
</tr>
<tr>
<td></td>
<td>Popup Delay</td>
<td>Lets you specify setting how long (in milliseconds) the editor should wait before populating and presenting the column list.</td>
</tr>
<tr>
<td></td>
<td>Enable Auto Replacement</td>
<td>When enabled, you can define simple keystrokes that will be automatically replaced with predefined expressions when some activation keyboard event is performed (space, new line, special character, etc.). This also lets you define correction actions for common misspellings or SQL syntax coding errors.</td>
</tr>
<tr>
<td></td>
<td>Replace substrings</td>
<td>When selected, this lets you enable automatic replacement for substrings: simple keystrokes will be automatically replaced with predefined expressions when some activation keyboard event is performed (space, new line, special character, etc.). This also lets you define correction actions for common misspellings or SQL syntax coding errors.</td>
</tr>
<tr>
<td>Auto Replace</td>
<td>Add</td>
<td>Click to open the Edit Auto Replace Expression dialog box.</td>
</tr>
<tr>
<td></td>
<td>Edit</td>
<td>Click to open the Edit Auto Replace Expression dialog box.</td>
</tr>
<tr>
<td></td>
<td>Clone</td>
<td>Click to open the Edit Auto Replace Expression dialog box.</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td>Click to delete the selected auto replace expression.</td>
</tr>
<tr>
<td></td>
<td>Import Settings</td>
<td>Click to import settings from a local or network directory.</td>
</tr>
<tr>
<td></td>
<td>Export Settings</td>
<td>Click to export your settings to a local or network directory.</td>
</tr>
<tr>
<td></td>
<td>Restore Settings</td>
<td>Click to restore the Code Workbench settings to the original settings.</td>
</tr>
</tbody>
</table>

**Common Tasks**

- Using Column Auto Lookup in the ISQL Window
- Creating and Modifying Code Workbench Auto Replace Shortcuts
- Using Code Workbench Auto Replace Shortcuts in the ISQL Window
- Importing and Exporting Settings in Code Workbench
Code Workbench Tutorial
Code Workbench creates a personal toolbox of coding accessories for the ISQL Window. This tutorial will take you through the steps of using the Code Workbench.

1. Select Tools, and then Code Workbench.
2. On the Settings Tab keep defaults.
3. On the Auto Replace Tab, Code Workbench includes a set of Auto Replace entries.
5. Select File, New, and then SQL.
6. In the ISQL Window, type a replace expression. For example, type beg.
   The application displays begin.
   1. For Microsoft SQL Server, connect to the Northwind database.
   2. Confirm that the Northwind database is selected in the database list.
   3. Select File, New, and then SQL.
   4. Type select * from dbo.Orders t1
   5. On the next line type where t1.
      The application displays a list of columns for the target table.

6. You can use your keyboard arrow keys or the mouse to select the column.
   The application pastes the column name in the ISQL Window.

Common Tasks
Using Column Auto Lookup in the ISQL Window
Creating and Modifying Code Workbench Auto Replace Shortcuts
Using Code Workbench Auto Replace Shortcuts in the ISQL Window
Importing and Exporting Settings in Code Workbench
Using Column Auto Lookup in the ISQL Window

In the ISQL Window, when Enable Auto Replacement is selected on the Setting Tab, the application uses "dot completion" auto population to display the list of columns for the target table.

1. For example, for Microsoft SQL Server, connect to the Northwind database.
2. Confirm that the Northwind database is selected in the database list.
3. Select File, New, and then SQL.
4. Type `select * from dbo.Orders t1` on the next line type `where t1`. The application uses "dot completion" auto population to display the list of columns for the target table.
5. Use your keyboard arrow keys or the mouse to select the column. The application pastes the column name in the ISQL Window.

For more information, see Code Workbench.

Creating and Modifying Code Workbench Auto Replace Shortcuts

The Edit Auto Replace Expression dialog box lets you define shortcut expressions for commonly used SQL syntax or commands. You can define what keys or key combination activate the auto replacement of the defined shortcut.

2. On the Auto Replace Tab, select Add.
3. In Expression type a new expression.
4. In Activation type activation information. For example, CTRL+SPACE.
5. In Replace With type replace with information.
6. Click OK. The replace expression is now ready for use in the ISQL Window.

For more information, see:

Using Code Workbench Auto Replace Shortcuts in the ISQL Window

Code Workbench

Using Code Workbench Auto Replace Shortcuts in the ISQL Window

When Enable Auto Replacement is selected on the Setting Tab of the Code Workbench, you can type a replace expression in the ISQL Window instead of typing a complete command.

1. Select File, New, and then SQL.
2. In the ISQL Window, type a replace expression. For example, type `beg`. The application displays begin.

For more information, see Code Workbench.
Importing and Exporting Settings in Code Workbench

The feature helps standardize your application settings. Exporting settings is useful when you want to give your auto replace expressions to another ISQL Window user. Importing settings is useful when another user wants to give you their auto replace expressions.

**Importing Settings**

1. Select **Tools, Code Workbench**.
2. Click **Import Settings**.
   
   Code Workbench opens the Open dialog box.
3. Locate the .xml file, and then click **Open**.
   
   Code Workbench imports the settings.

**Exporting Settings**

1. Select **Tools, Code Workbench**.
2. Click **Export Settings**.
   
   Code Workbench opens the Save As dialog box.
3. Specify a location, and file name, and then click **Save**.
   
   Code Workbench saves your settings as an .xml file. You can send the .xml file to the user you want to import your settings.

For more information, see **Code Workbench**.
Permissions Management

DBArtisan includes permissions management capabilities that include:

- Explicit Permissions
- Cascading Permissions
- Using Roles to Grant Permissions and Privileges
- Using Roles to Revoke Permissions and Privileges
- Grant Privilege(s) To
- Revoke Privilege(s) From
- Deny Privileges From

Explicit Permissions

Explicit permissions are permissions specifically granted by one user to another. Granting a permission is an integral part of database security and management. DBArtisan tracks explicit permission changes across your enterprise. Multiple explicit permissions can be consolidated in roles. Different groups and roles can share permissions for an object. DBArtisan lets you grant permissions with the Roles Editor. Embarcadero lets you edit permissions on the Privileges tabs of the object editors.

**TIP:** The Admin option is similar to the grantable option for privileges. If the user has the admin option, they can grant that role to other people.

**TIP:** In most cases, you will want to make all roles granted, default roles. If you do not make a role default for a user, they will have to enable it with a SQL command. To avoid this complication, assign roles as default unless business rules specify otherwise.

**TIP:** The Grantable option gives the user the ability to grant that privilege to other users. Use the Grantable option SPARINGLY.

For more information, see Permissions Management.

Cascading Permissions

Cascading permissions are the path of privileges granted from one user to another user, group, or role. Using cascading permissions is a way to manage multiple sets of permissions and privileges for groups of users. When you drop a user with a revoke cascade command, all permissions and privileges granted by the dropped user are also revoked. Embarcadero lets you edit and set grant permission parameters with the Permissions Editor.

Once a user with grant permission privileges is dropped with cascade, reinstate permissions for all affected users.

For more information, see Permissions Management.

Using Roles to Grant Permissions and Privileges

Roles are sets of user privileges you associate with access to objects within a database. Roles streamline the process of granting permissions. You can use roles to grant sets of permissions and privileges to users and groups. DBArtisan lets you grant permissions to roles with the Roles Editor.
For more information, see [Permissions Management](#).

### Using Roles to Revoke Permissions and Privileges

Roles can be effective in revoking permissions and privileges from users. Rather than individually revoke permissions from users, you can revoke groups of permissions from multiple users simultaneously using roles. DBArtisan lets you revoke permissions with the Roles Editor.

DBArtisan lets you identify existing users and their database permissions, and to detect and manage changes to user permissions by rolling change back or forward. DBArtisan lets you manage database permissions in a cross-platform database environment, and gives you the ability to compare objects/permissions and migrate them to other instances. Using the compare functionality for permissions management, DBArtisan gives you:

- Enhanced permissions management across the enterprise (Oracle, Sybase ASE, Microsoft SQL Server, and IBM DB2 for Linux, Unix, and Windows).
- Faster detection of changed user rights.
- Ability to detect database accounts (users/logins) that are invalid.
- Rollback of invalid permissions in a single click.
- Archives of users, groups, roles and their permissions.

For more information, see [Permissions Management](#).

### Grant Privilege(s) To

The Grant Privilege(s) To dialog box lets you select or clear the check boxes corresponding to the target privileges.

Permissions and privileges allow or limit access to system database administration and database objects. To manage databases, it is important to monitor who can access the enterprise structure and change the underlying schema. It is also important to monitor who can grant and revoke permissions and privileges in the enterprise. DBArtisan lets you manage permissions and privileges in the Grant Privilege(s) To dialog box.

On the Privileges Tab of the editor, you can see whether a privilege was inherited from a role or group, or if it was granted explicitly by a user. DBArtisan also shows if the privilege is grantable (granted with the `GRANT OPTION`). The table below describes the icons:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black check mark</td>
<td>User</td>
<td>The privilege(s) was granted explicitly by a user.</td>
</tr>
<tr>
<td>Blue check mark</td>
<td>Group</td>
<td>The privilege(s) was inherited from a group.</td>
</tr>
<tr>
<td>Green check mark</td>
<td>Role</td>
<td>The privilege(s) was inherited from a role.</td>
</tr>
<tr>
<td>Red check mark</td>
<td>Denied</td>
<td>MICROSOFT SQL SERVER ONLY: The privilege(s) was denied from a security account in the current database.</td>
</tr>
<tr>
<td>Two black check marks</td>
<td>User/Grant</td>
<td>The privilege(s) was granted by a user with the GRANTABLE option.</td>
</tr>
<tr>
<td>Two blue check marks</td>
<td>Group/Grant</td>
<td>The privilege(s) was granted from a group with the GRANTABLE option.</td>
</tr>
<tr>
<td>Two green check marks</td>
<td>Role/Grant</td>
<td>The privilege(s) was granted from a role with the GRANTABLE option.</td>
</tr>
</tbody>
</table>

**NOTE:** The available privileges depend on the target DBMS and object.
The table below describes the options and functionality on the Grant Privilege(s) From Dialog Box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privilege</td>
<td>Lets you select or clear the check boxes corresponding to the target privileges.</td>
</tr>
<tr>
<td>Grantable</td>
<td>Select No to prevent cascading the privileges to other users.</td>
</tr>
</tbody>
</table>

The table below describes the object permissions:

<table>
<thead>
<tr>
<th>Object</th>
<th>Permission(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>CONTROL</td>
</tr>
<tr>
<td>Packages</td>
<td>BIND, CONTROL, EXECUTE</td>
</tr>
<tr>
<td>Schema</td>
<td>ALTERIN, CREATIN, DROPIN (w/GRANT OPTION)</td>
</tr>
<tr>
<td>Tables</td>
<td>ALTER, CONTROL, DELETE, INDEX, INSERT, REFERENCES (&amp; on column), SELECT, UPDATE (&amp; on column) (w/GRANT OPTION)</td>
</tr>
<tr>
<td>Tablespaces</td>
<td>USE (w/GRANT OPTION)</td>
</tr>
<tr>
<td>Views</td>
<td>CONTROL, DELETE, INSERT, SELECT, UPDATE (w/GRANT OPTION)</td>
</tr>
</tbody>
</table>

For more information, see:

- [Completing a Dialog Box](#)
- [Permissions Management](#)

**Revoke Privilege(s) From**

The table below describes the options and functionality on the Revoke Privilege(s) From dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privilege</td>
<td>Lets you select or clear the check boxes corresponding to the target privileges.</td>
</tr>
<tr>
<td>Cascade</td>
<td>Select No to prevent cascading the revocation privileges to other users.</td>
</tr>
</tbody>
</table>

For more information, see:

- [Completing a Dialog Box](#)
- [Permissions Management](#)
Deny Privileges From

**NOTE:** Deny Privileges is available for Microsoft SQL Server only.

The Deny Privileges From dialog box lets you deny a privilege from a security account in the current database. It also shows if the privilege is grantable (granted with the GRANT OPTION.) The table below describes the icons:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black check mark User</td>
<td>User</td>
<td>The privilege(s) was granted explicitly by a user.</td>
</tr>
<tr>
<td>Blue check mark Group</td>
<td>Group</td>
<td>The privilege(s) was inherited from a group.</td>
</tr>
<tr>
<td>Green check mark Role</td>
<td>Role</td>
<td>The privilege(s) was inherited from a role.</td>
</tr>
<tr>
<td>Red check mark Denied</td>
<td>Denied</td>
<td><strong>MICROSOFT SQL SERVER ONLY:</strong> The privilege(s) was denied from a security account in the current database.</td>
</tr>
<tr>
<td>Two black check marks User/Grant</td>
<td>User/Grant</td>
<td>The privilege(s) was granted by a user with the GRANTABLE option.</td>
</tr>
<tr>
<td>Two blue check marks Group/Grant</td>
<td>Group/Grant</td>
<td>The privilege(s) was granted from a group with the GRANTABLE option.</td>
</tr>
<tr>
<td>Two green check marks Role/Grant</td>
<td>Role/Grant</td>
<td>The privilege(s) was granted from a role with the GRANTABLE option.</td>
</tr>
</tbody>
</table>

**NOTE:** The available privileges depend on the target DBMS and object.

The table below describes the options and functionality on the Deny Privilege(s) From dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privilege</td>
<td>Lets you select or clear the check boxes corresponding to the target privileges.</td>
</tr>
<tr>
<td>Cascade</td>
<td>Select No to prevent cascading the deny privileges to other users.</td>
</tr>
</tbody>
</table>

**TIP:** On the editor, the Deny privilege can be Revoked, just as a Grant permission can be revoked.

For more information, see:

- [Completing a Dialog Box](#)
- [Permissions Management](#)
Add-On Tools

Embarcadero Technologies offers add-on tools that enhance the usefulness of DBArtisan. The table below describes these add-on tools:

<table>
<thead>
<tr>
<th>Add-on Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Embarcadero SQL Debugger</strong></td>
<td>Provides detailed information about using the Embarcadero SQL Debugger. It includes a step-by-step walk through to familiarize you with the features of the Embarcadero SQL Debugger.</td>
</tr>
<tr>
<td><strong>DBArtisan PL/SQL Profiler</strong></td>
<td>Provides detailed information about using the DBArtisan PL/SQL Profiler. Includes a step-by-step walk through to familiarize you with the features of the PL/SQL Profiler.</td>
</tr>
</tbody>
</table>
Embarcadero SQL Debugger

The Embarcadero SQL Debugger lets you locate and fix bugs in the following elements:

- Procedures (IBM DB2 for Linux, Unix, and Windows, Microsoft SQL Server, Oracle, Sybase)
- Functions (Oracle)
- Triggers (IBM DB2 for Linux, Unix, and Windows, Microsoft SQL Server, Oracle, Sybase)

**NOTE:** The SQL Debugger lets you debug triggers by debugging the procedures that call them.

**NOTE:** For Oracle, you cannot debug packages, but you can debug the functions and procedures within packages.

**NOTE:** For Oracle, you cannot debug any objects contained in the Exclusion List. For more information, see Editing the Exclusion List.

The table below describes the sections of this chapter:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Debugger modules and DBMS support</td>
<td>Describes the specific SQL Debugger modules and DBMS versions supported/</td>
</tr>
<tr>
<td>Embarcadero SQL Debugger features</td>
<td>Provides an overview of SQL Debugger functionality.</td>
</tr>
<tr>
<td>Setting up the Embarcadero SQL Debugger</td>
<td>Provides details on requirements and setup tasks for each supported DBMS.</td>
</tr>
<tr>
<td>Embarcadero SQL Debugger interface</td>
<td>Describes the Embarcadero SQL Debugger graphical interface that includes an editor window and four debug view windows.</td>
</tr>
<tr>
<td>Embarcadero SQL Debugger functionality</td>
<td>This section describes the functionality on the SQL Debugger.</td>
</tr>
<tr>
<td>Using the Embarcadero SQL Debugger</td>
<td>This section describes how to run a debug session.</td>
</tr>
<tr>
<td>Tutorial sessions</td>
<td>Provides walkthrough sessions for new users.</td>
</tr>
</tbody>
</table>

**SQL Debugger modules and DBMS support**

There are four Embarcadero SQL Debugger modules, each corresponding to a supported DBMS. Modules and DBMS support are as follows:

- **Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows** - supports IBM DB2 for Linux, Unix, and Windows version 8 or later
- **Embarcadero SQL Debugger for Microsoft** - supports Microsoft SQL Server version 7.0 or later
- **Embarcadero SQL Debugger for Oracle** - supports all Oracle versions supported by DBArtisan
- **Embarcadero SQL Debugger for Sybase** - supports Sybase 12, 12.0 (special version), and 12.5

Each SQL Debugger version is an optional and separate add-on module.

**Embarcadero SQL Debugger features**

The Embarcadero SQL Debugger lets you identify problems within your code. The Embarcadero SQL Debugger lets you:
• Interactively step through the flow of script execution.
• Examine the value of variables.
• Solve logical problems with your script design.

**NOTE:** The Debugger is available on the DBArtisan main menu, the Procedures window, the DDL Editor and ISQL windows.

The Embarcadero SQL Debugger offers fundamental debugging features and options to fine tune debugging. The table below describes these features:

<table>
<thead>
<tr>
<th>Debugging Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step Into</strong></td>
<td>Lets you execute each instruction step-by-step and step inside a stored object.</td>
</tr>
<tr>
<td><strong>Step Out</strong></td>
<td>Lets you stop stepping through the current object and execute the remainder of the script. This option is only active when the pointer indicates a child dependent instruction.</td>
</tr>
<tr>
<td><strong>Step Over</strong></td>
<td>Lets you execute the current instruction without stepping into any child dependents.</td>
</tr>
<tr>
<td><strong>Insert or Remove Breakpoint</strong></td>
<td>Lets you specify positions in a program where the debugger stops execution.</td>
</tr>
</tbody>
</table>

To set specific Debugger values on DBArtisan’s Options Editor, see [Debugger Options](#).

**Setting up the Embarcadero SQL Debugger**

The following topics describe requirements and tasks for those DBMS that have setup requirements:

• [Embarcadero SQL Debugger for Linux, Unix, and Windows setup](#)
• [The Embarcadero SQL Debugger for Microsoft setup](#)
• [Embarcadero SQL Debugger for Oracle setup](#)

In addition, you should set your debugger preferences at this time. For details, see [Embarcadero SQL Debugger options](#).

**Embarcadero SQL Debugger for Linux, Unix, and Windows setup**

Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows requires proper configuration of the server and client.

For more information, see:

• [Prerequisites for Adding and Compiling Stored Procedures](#)
• [Configuring the IBM DB2 for Linux, Unix, and Windows Server for Procedures](#)
• [Prerequisites for Debugging Procedures](#)

**Prerequisites for Adding and Compiling Stored Procedures**

The Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows requires the following products and components.

**Client**

• IBM DB2 for Linux, Unix, and Windows 8 or later
The server must have a local directory structure and file C:\program files\sqlib\function\routine\sr_cpath.bat. This file is installed with IBM DB2 8 and includes the C compiler options needed to compile the procedure on the server. If it is not found, install the IBM DB2 8 Administration and Configuration Tools option on the server.

Configuring the IBM DB2 for Linux, Unix, and Windows Server for Procedures
DBArtisan lets you create procedures on the targeted server using DBArtisan.

To create or run any procedure, set up the configuration environment and enable the C compiler options on the server.

To configure your server, do the following:

1. Open a DB2 Command Window, and then type:
   
   ```
   DB2set DB2_SQLROUTINE_COMPILER_PATH="C:\program files\sqlib\function\routine\sr_cpath.bat"
   ```

   DB2 sets the DB2_SQLROUTINE_COMPILER_PATH DB2 registry variable to call the required initialization script for the C compiler on the server.
To enable the C compiler options on your server:

1. Open the file C:\program files\sql\lib\function\routine\sr_cpath.bat.
2. Remove the REM (remarks) prefix on the lines that match the version of Visual Studio that is installed on the server. VCV6 = version 6.0 and VCV5 = version 5.0.
   
   **NOTE:** Only remove the REM prefix on the lines that apply to your installation of Visual Studio.
3. Restart the DB2 services on the server.

Prerequisites for Debugging Procedures

To enable debugging on the server, do the following:

1. Open a **DB2 Command** window and type:

   ```
   Db2set DB2ROUTINE_DEBUG=ON
   ```

   **NOTE:** Client must have a licensed or evaluation copy of the Embarcadero SQL Debugger.

**The Embarcadero SQL Debugger for Microsoft setup**

Embarcadero SQL Debugger for Microsoft requires proper configuration of [Server Requirements](#) and [Client Requirements](#).

**Important Note about Microsoft SQL Server 2000 Service Pack 4 (SP4)**

When running Windows XP SP2, SQL Server 2000 SP4 is required on both the client and the server. To verify this is the case start MS Query Analyzer and press Help > About. The Version SQL should be 8.00.2039.

**Important Notes about Microsoft SQL Server 2000 Service Pack 3 (SP3)**

By default, after you install Microsoft SQL Server 2000 Service Pack 3 (SP3), you cannot use the DBArtisan's T-SQL Debugger.

You may receive the following error message if you try to use the T-SQL Debugger from DBArtisan:

```
“Server: Msg 514, Level 16, State 1, Procedure sp_sdidebug, Line 1 [Microsoft][ODBC SQL Server Driver][SQL Server]Unable to communicate with debugger on [SQL Server Name] (Error = 0x80070005). Debugging disabled for connection 53.”
```

Microsoft has disabled T-SQL Debugger for Application using earlier T-SQL Debugger clients for security reasons. To enable T-SQL Debugger for these Applications, a member of the sysadmins server role, such as sa must explicitly enable debugging by running the following code:

```sql
Exec sp_sdidebug 'legacy_on'
```

**NOTE:** You must repeat this procedure whenever you restart the server.

**Server Requirements**

Embarcadero SQL Debugger for Microsoft requires:

- Windows 2005
- Windows 2000
- Microsoft SQL Server version 7.0 or later
Setting Up the Server

There are three parts to setting up the server:

1. **Installing the Microsoft SQL Debugger Interface Subcomponent**
2. **Configuring the Service**
3. **Configuring DCOM on the Server**

**Enabling SQL Debugger for Microsoft on SQL Server SP3**

SQL Debugging is disabled by default in SQL Server SP3 and greater. Please refer to Microsoft Support for information regarding enabling the SQL Debugger for Microsoft on SQL Server SP3.

**Installing the Microsoft SQL Debugger Interface Subcomponent**

The Microsoft server must have the Development Tools, Debugger Interface subcomponent of Microsoft SQL Server. To determine if the Debugger Interface subcomponent is installed, locate the following files in the \Program Files\Common Files\Microsoft Shared\SQL Debugging directory:

- SQLDBREG.exe
- SQLDBG.dll

If these files are not in the \Program Files\Common Files\Microsoft Shared\SQL Debugging directory, install them before running the Embarcadero SQL Debugger for Microsoft.

To install the Debugger Interface subcomponent on the server after the initial installation:

1. Start **Microsoft Visual Studio, Enterprise Edition Setup**.  
   OR  
   Start **Microsoft SQL Server Setup**.
2. Select **Custom Install**.
   Microsoft SQL Server opens the Select Components dialog box.
3. In the Components box, select the Development Tools check box.
4. In the Sub-components box, select the Debugger Interface check box.
5. Click **Next**.
   Microsoft SQL Server proceeds through the Microsoft SQL Server wizard to install the components.

For more information, see **Server Requirements**.

**Configuring the Service**

To configuring the service, see the instructions for your server operating system:

- **Windows 2000**

Windows 2000

1. On the Windows taskbar, click the Start button, click **Settings**, and then click **Control Panel**.
2. Double-click **Administrative Tools**, and then click **Services**.
   Windows opens the Services explorer.
3 In the right pane of the Services explorer, right click MSSQLServer, and then click Properties. Windows opens the Net Logon Properties dialog box.

4 Click the Logon Tab.

5 Select the This Account option button.

6 In the This Account box, type (or browse to locate) the logon user account (including domain name, if necessary) of the person who will be using the Embarcadero SQL Debugger for Microsoft.

   NOTE: This person needs admin permissions on the server.

7 In the Password and Confirm Password boxes, type the password.

8 Click Apply.

9 Click the General Tab.

10 Click Start.

   Windows starts the server and applies the changes.

Important Note about Microsoft SQL Server 2000 Service Pack 4 (SP4)
When running Windows XP SP2, SQL Server 2000 SP4 is required on both the client and the server. When running Windows XP SP2, SQL Server 2000 SP4 is required on both the client and the server. To verify this is the case start MS Query Analyzer and press Help > About. The Version SQL should be 8.00.2039.

Important Notes about Microsoft SQL Server 2000 Service Pack 3 (SP3)
By default, after you install Microsoft SQL Server 2000 Service Pack 3 (SP3), you cannot use the DBArtisan’s T-SQL Debugger.

You may receive the following error message if you try to use the T-SQL Debugger from DBArtisan:

   “Server: Msg 514, Level 16, State 1, Procedure sp_sdidebug, Line 1 [Microsoft][ODBC SQL Server Driver][SQL Server]Unable to communicate with debugger on [SQL Server Name] (Error = 0x80070005). Debugging disabled for connection 53.”

Microsoft has disabled T-SQL Debugger for Application using earlier T-SQL Debugger clients for security reasons. To enable T-SQL Debugger for these Applications, a member of the sysadmins server role, such as sa must explicitly enable debugging by running the following code:

   Exec sp_sdidebug 'legacy_on'

   NOTE: You must repeat this procedure whenever you restart the server.

For more information, see Configuring the Service.

Configuring DCOM on the Server
To configure DCOM on the server, do the following:

1 After the server restarts, on the Windows taskbar, click the Start button, and then click Run.

2 In the Open box, type dcomcnfg.exe.

3 Click OK.

   Windows opens the Distributed COM Configuration Properties dialog box.

4 Click the Default Security Tab.
In the **Default Access Permissions** box, click **Edit Default**. Windows opens the Registry Value Permissions dialog box.

6  Click **Add**. Windows opens the Add Users and Groups dialog box.

7  In the **Names** box, select **SYSTEM**, and then click **Add**.

8  Click the **Type of Access** list, and then click **Allow Access**.

9  To let any user use the Embarcadero SQL Debugger for Microsoft, grant them remote access on the server. To grant remote access, configure their DCOM permissions on the server. In the **Names** box, click the target users, and then click **Add**.

   NOTE: You can add individual users or groups.

10 Click the **Type of Access** list, and then click **Allow Access**.

11 Click **OK**.

12 Restart the server to apply the changes.

For more information, see **Server Requirements**.

Client Requirements
There are three categories of client requirements for the Embarcadero SQL Debugger for Microsoft:

- **Operating System**
- **Client Connectivity**
- **Installing the Microsoft SQL Debugger Interface Subcomponent**

**Operating System**
The client must be running one of the following operating systems:

- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows XP

For more information, see **Client Requirements**.

**Client Connectivity**

**When running the Debugger on SQL Server 2005 Clients**
Before you proceed, verify that the file ssdebugps.dll is registered on the client machine. This file is REQUIRED for debugging and Microsoft only installs it on the server machine and not as a part of a client-only install.

Next you must configure DCOM by following these steps:

1  At a command prompt, type `dcomcnfg`, and then press **ENTER**. Component Services opens.
2. In Component Services, expand **Component Services**, expand **Computers**, and then expand **My Computer**.

3. On the toolbar, click the **Configure My Computer** button.

   The **My Computer** dialog box appears.

4. In the **My Computer** dialog box, click the **COM Security** tab.

5. Under **Access Permission**, click **Edit Limits**.

   The **Access Permission** dialog box appears.

6. Under **Group or user names**, click **ANONYMOUS LOGON**.

7. Under **Permissions** for **ANONYMOUS LOGON**, select the **Remote Access** check box, and then click **OK**.

Finally, you need to go to the **Default Properties** tab of the MyComputer dialog.

1. Select **None** in the **Default Authentication Level** dropdown menu.

2. Select **Impersonate** in the **Default Impersonation Level** dropdown menu.

For more detailed information, you can refer to Microsoft's Help:


When running Windows XP SP2, SQL Server 2000 SP4 is required on both the client and the server.

For Microsoft SQL the client must have the Client Connectivity component of Microsoft SQL Server.

For Microsoft SQL Server

The client must have the Development Tools, Debugger Interface subcomponent of Microsoft SQL Server. To determine if the Debugger Interface subcomponent is installed, locate the following files in the `\Program Files\Common Files\Microsoft Shared\SQL Debugging` directory:

- **SQLDBREG.exe**
- **SQLDBG.dll**

If these files are not in the `\Program Files\Common Files\Microsoft Shared\SQL Debugging` directory, install them before running the Embarcadero SQL Debugger for Microsoft.

### Installing the Microsoft SQL Debugger Interface on the Client

To install the Debugger Interface subcomponent on the client, do the following:

1. Start the Microsoft SQL Server Setup program.

2. Select **Custom Install**.

   Microsoft SQL Server opens the Select Components dialog box.

3. In the **Components** box, select the **Development Tools** check box.

4. In the **Sub-Components** box, select the **Debugger Interface** check box.

5. Click **Next**.

   Microsoft SQL Server proceeds through the Microsoft SQL Server Wizard to install the components.

For more information, see [Client Requirements](#).
Embarcadero SQL Debugger for Oracle setup

The only task required in setting up the Embarcadero SQL Debugger for Oracle is Editing the Exclusion List.

Editing the Exclusion List

Upon installation, DBArtisan sets up an Exclusion List on your computer which includes packages that the application cannot debug. The Exclusion List is located in the DBArtisan directory. You can add or remove packages from this file by editing the Exclusion List.

To Edit the Exclusion List, do the following:

1. Open the Exclusion List, deborcex.etd, in a text editor, such as Microsoft Notepad or WordPad.
2. To add a package, enter the name of the package at the end of the list. Use the following format: OWNER.OBJECT_NAME.
   **NOTE:** Press ENTER after each item on the list.
3. To remove a package from the Exclusion List, delete the package from the list.
   **NOTE:** Embarcadero SQL Debugger for Oracle does not debug a package function or package procedure listed on the Exclusion List.
4. Save the changes to deborcex.etd.

Embarcadero SQL Debugger options

You can specify debugger options from the Debug Tab of DBArtisan’s Options editor. The Debug Tab of the Options Editor lets you set the duration of your debug initialization and debug session, enable DBMS output, and refresh dependencies.

Setting Debugger Options

To set debugger options, do the following:

1. On the File menu, click Options.
   DBArtisan opens the Options Editor.
2. On the Debug Tab, specify debugger options. The table below describes the options available:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialization Timeout (seconds)</td>
<td>Specifies the number of seconds DBArtisan tries to initialize the debugger. If it cannot initialize the debugger in the specified time, a message displays in the Debug Output window.</td>
<td>60</td>
</tr>
<tr>
<td>Debug Session Timeout (seconds)</td>
<td>Specifies, in seconds, the length of your debug session.</td>
<td>7200</td>
</tr>
<tr>
<td>Enable DBMS Output</td>
<td>Toggles the print output. Enable this option if you use dbms_output.put_line calls in your procedures and you want these lines displayed.</td>
<td>Selected</td>
</tr>
<tr>
<td>Refresh Dependencies for each run</td>
<td>Refreshes dependencies for each run. This potentially time-consuming process is useful if the target procedure has rapidly varying dependencies that can require updating during the debugging process.</td>
<td>Cleared</td>
</tr>
</tbody>
</table>
3 Click **Close**.

DBArtisan closes the Options Editor.

For more information, see Embarcadero SQL Debugger interface and Embarcadero SQL Debugger functionality.

---

**Embarcadero SQL Debugger interface**

The Embarcadero SQL Debugger includes an editor window and four debug view windows. When you open a debug session, DBArtisan extracts the code for the object into a DDL Editor and opens four debug view windows at the bottom of the screen. The four debug view windows are optional, dockable windows designed to let you debug your script.

**TIP:** All Embarcadero debuggers display Performance Metrics that let you measure the execution time of each statement in the debug session.

The Embarcadero SQL Debugger includes five windows:

1. **DDL Editor Window**
2. **Watch Window**
3. **Variables Window**
4. **Call Stack Window**
5. **Dependency Tree Window**

**Working with SQL Debugger Windows**

DBArtisan lets you resize, move, dock and float the following windows:

- **Watch Window**
- **Variables Window**
- **Call Stack Window**
- **Dependency Tree Window**

1. To resize the target window, click its frame and drag it.

   DBArtisan resizes the window.

2. To move and dock the target window, click its grab bar and drag it.

   DBArtisan moves the window to its new location and docks it with surrounding windows.

3. To float the target window, press **Shift**, then click its grab bar and drag it.

   DBArtisan frames the window in its own floating frame and moves the window to its new location.

**DDL Editor Window**

The DDL Editor displays your code in read-only format. When you start debugging, the Embarcadero SQL Debugger extracts your code into a DDL Editor. The DDL Editor uses the default DBArtisan syntax coloring.

For more information, see Embarcadero SQL Debugger interface.

**NOTE:** For Oracle, DBArtisan displays LOB datatypes, and REF CURSOR variables, in the Results Tab.
Watch Window
The Watch window displays the watch variables for the database object you are debugging. The Watch window also lets you specify variables you want to evaluate or modify while debugging your program.

For example, to check what happens when a variable (x) has a value of 100, you can double-click the variable in the DDL Editor, drag it into the Watch Window, and change the value to 100. When you execute the script, the Debugger uses the value x = 100. This window is only visible when the SQL Debugger is active.

NOTE: Until you step at least once into a script, variables are not defined. Therefore, step at least once before dragging or typing a local variable in the Watch Window.

NOTE: You can type a fully qualified record variable into the Watch window.

NOTE: When you exit a debug session and reenter it, the Embarcadero SQL Debugger retains any watch variables or breakpoints you have set.

Opening and Closing the Watch Window
To open and close the Watch Window, do the following:

1. On the Debug Menu, on the Debug Views sub-menu, select or clear Watch.

   OR

2. Press ALT+3.

Setting a Watch Variable
To set a Watch Variable, do the following:

1. In the DDL Editor, double-click the target variable and drag it to the Watch window.

   NOTE: Microsoft SQL Server requires that local variables begin with @. Drag the @ to the Watch Window.

2. In the Watch window, change the value of the variable.

3. On the DDL Editor, click Debug or Go.

   The Embarcadero SQL Debugger executes the script using the new variable.

Removing a Watch Variable
To remove a Watch variable, do the following:

1. In the Watch window, click the target variable and press DELETE.

For more information, see Embarcadero SQL Debugger interface.

Variables Window
The Variables window displays the local variables and their current values during script execution.

NOTE: You cannot edit the variables in the Variables window.

If the DDL Editor displays an external database object, and that object is a dependent of the object you are debugging, then the Variables Window automatically refreshes and displays the variables for that particular object. The Variables Window is only visible when the Debugger is active.
The Embarcadero SQL Debugger also lets you monitor your variables while debugging.

**Opening and Closing the Variables Window**
To open and close the Variables Window, do the following:

1. On the **Debug Menu**, on the **Debug Views** sub-menu, select or clear **Variable**.

   **OR**

   Press **ALT+4**.

**Monitoring Variables**
To monitor the values of your variables while debugging, do the following:

1. In the **SQL Editor**, hold the pointer over the target variable.

   DBArtisan opens a ScreenTip displaying the current value of that variable.

For more information, see Embarcadero SQL Debugger interface.

**Call Stack Window**
The Call Stack window displays the stack of currently active calls. The Call Stack Window is only visible when the Debugger is active.

**Opening and Closing the Call Stack Window**
To open and close the Call Stack Window, do the following:

1. On the **Debug Menu**, on the **Debug Views** sub-menu, select or clear **Call Stack**.

   **OR**

   Press **ALT+5**.

**Using the Call Stack Window**
To display a line of code that references the call in the DDL Editor, do the following:

1. In the **Call Stack** window, double-click the target line.

   In the DDL Editor, DBArtisan displays a green arrow on the line of the referenced call.

For more information, see Embarcadero SQL Debugger interface.

**Dependency Tree Window**
The Dependency Tree window displays any external database objects the script accesses. DBArtisan displays these database objects in a hierarchical tree, with the child objects as database objects accessed by the parent objects. You can use this window to display the code for a dependent database object in the DDL Editor window. This window is only visible when the Debugger is active.
Opening and Closing the Dependency Tree Window
To open and close the Dependency Tree Window, do the following:

1. On the Debug Menu, on the Debug Views sub-menu, select or clear Dependencies.
   OR
   Press ALT+6.

Displaying Dependencies
To display the code for a dependent database object in the DDL Editor window, do the following:

1. In the Dependency Tree window, double-click the target object.

   DBArtisan displays the SQL of the target object in the DDL Editor window.

For more information, see Embarcadero SQL Debugger interface.

Embarcadero SQL Debugger functionality
The Embarcadero SQL Debugger offers the following functionality:

- Input Parameters
- Step Into
- Step Out
- Step Over
- Run to Cursor
- Insert or Remove Breakpoint
- Toggle Breakpoint
- Go
- Stop
- Restart
- Break
- Close

You must be running a debugging session to use these functions. For details, see Opening a Debugging session.

Input Parameters
Input parameters are set when you first create an object. If the object you want to debug requires input parameters, DBArtisan opens a dialog box to prompt you for the input parameters when you open a debugging session.

The Procedure Execution dialog box also lets you:

- Save input parameters as *.prm files to preserve specific input parameter configurations.
- Open *.prm files to save the effort of reentering specific input parameters.
- Reset parameters to their default setting.
The following table describes the options available in this dialog box:

<table>
<thead>
<tr>
<th>Dialog box component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner drop-down list</td>
<td>Displays the current procedure’s owner</td>
</tr>
<tr>
<td>Procedure drop-down list</td>
<td>Displays the name of the current procedure.</td>
</tr>
<tr>
<td>Parameter window</td>
<td>Specify the required input parameters in this window. If input parameters are not required for the execution of the target procedure, a message appears in this window, stating that the procedure &quot;has no input parameters. Press execute to run it.&quot;</td>
</tr>
<tr>
<td>Open button</td>
<td>Click to open an Open dialog box, from which you can open an existing *.prm file. The saved parameters immediately populate the dialog box upon opening.</td>
</tr>
<tr>
<td>Save button</td>
<td>Click to save the values of your input parameters as a *.prm file. You can reopen a saved *.prm file from this dialog box at any time.</td>
</tr>
<tr>
<td>Reset button</td>
<td>Click to reset the parameters in the Parameter window to their default values.</td>
</tr>
<tr>
<td>Execute or Continue button</td>
<td>Click to execute the procedure once you have entered values for all required parameters in the Parameter window.</td>
</tr>
</tbody>
</table>

**NOTE:** You cannot debug a script that requires input parameters until you provide input parameters.

For more information, see [Embarcadero SQL Debugger functionality](#).

**Step Into**
Step Into lets you execute the current instruction. If the current instruction makes a call to a stored SQL object, the Embarcadero SQL Debugger steps inside the nested child object.

To use the Step Into facility, do the following:

1. On the **Debug** menu, click **Step Into**.
   OR
   Press **F11**.
   The Embarcadero SQL Debugger moves the arrow to execute the current instruction.

For more information, see [Embarcadero SQL Debugger functionality](#).

**Step Out**
Step Out lets you execute the remainder of the dependent child object and resumes line-by-line, step-debugging in the parent object.

**NOTE:** Step Out is only active when the pointer indicates a child dependent instruction.

To use the Step Out facility, do the following:

1. On the **Debug** menu, click **Step Out**.
   OR
   Press **SHIFT+F11**.
   The Embarcadero SQL Debugger stops stepping through the current object and executes the remainder of the script.
For more information, see [Embarcadero SQL Debugger functionality](#).

**Step Over**

Step Over lets you execute the current instruction without stepping into a nested child object if the instruction makes a call to a dependent object.

To use the Step Over, do the following:

1. On the **Debug** menu, click **Step Over**.
   
   OR
   
   Press **F10**.
   
   The Embarcadero SQL Debugger executes the current instruction.

For more information, see [Embarcadero SQL Debugger functionality](#).

**Run to Cursor**

Run to Cursor lets you execute all instructions between the yellow arrow and the cursor.

To use the Run to Cursor facility, do the following:

1. Scroll down from the yellow arrow to the target line.
2. Click the target line.
   
   Embarcadero SQL Debugger places the cursor on the target line.
3. On the **Debug** menu, click **Run to Cursor**.
   
   OR
   
   Press **CTRL+F10**.
   
   The Embarcadero SQL Debugger executes all instructions between the pointer and the cursor.

For more information, see [Embarcadero SQL Debugger functionality](#).

**Insert or Remove Breakpoint**

A breakpoint is a position in a program where a debugger stops execution. When you start debugging, Embarcadero SQL Debugger opens the script in a DDL Editor. A yellow arrow pointer indicates which line the Embarcadero SQL Debugger executes next.

The Embarcadero SQL Debugger executes all lines of code between the yellow arrow and the first breakpoint. If no breakpoints are present, Embarcadero SQL Debugger debugs the entire script.

While debugging you can set one or more breakpoints in the currently executing object or in any object in the program call stack. You can **Toggle Breakpoints**, temporarily disable or enable breakpoints without having to add or remove breakpoints.

DBArtisan’s Embarcadero SQL Debugger displays each enabled breakpoint as a red dot in the left margin of the DDL Editor Window, and each disabled breakpoint as a red circle.
DBArtisan stores all breakpoints you set so that when you debug the same script on separate occasions, you can reuse the same breakpoints. You can insert a breakpoint on the line where your cursor is located, and you can remove a breakpoint on the line where your cursor is located.

**NOTE:** Script execution stops at the first breakpoint.

To insert and remove breakpoints, do the following:

1. In the **DDL Editor** window, click the target line of SQL.
2. On the **Debug** menu, click **Breakpoint**.
   
   **OR**
   
   Press **F9**.

   The Embarcadero SQL Debugger inserts a new breakpoint or removes an existing breakpoint on the target line of code.

For more information, see **Embarcadero SQL Debugger functionality**.

**Toggle Breakpoint**

After inserting a breakpoint, Toggle Breakpoint lets you enable or disable that breakpoint. Embarcadero SQL Debugger displays each enabled breakpoint as a red dot in the left margin of the DDL Editor Window, and each disabled breakpoint as a red circle.

You can toggle any breakpoint in the DDL Editor window. When you exit a debugging session and reenter it, the Embarcadero SQL Debugger retains any breakpoints you set.

To use the Toggle Breakpoint facility, do the following:

1. In the **DDL Editor** window, click the line of the target breakpoint.
2. On the **Debug** menu, click **Enable/Disable Breakpoint**.
   
   **OR**
   
   Press **CTRL+F9**.

   The Embarcadero SQL Debugger toggles the breakpoint indicated by the pointer.

For more information, see **Insert or Remove Breakpoint**.

**Go**

Go lets you execute all instructions stopping only when it encounters a breakpoint or when the program is complete.

To use the Go facility, do the following:

1. On the **Debug** menu, click **Go**.
   
   **OR**
   
   Press **F5**.

   The Embarcadero SQL Debugger executes all instructions.

For more information, see **Embarcadero SQL Debugger functionality**.
Stop
Stop lets you halt the script execution and terminate the session.

To use the Stop facility, do the following:

1. On the Debug menu, click Stop Debugging.
   OR
   Press SHIFT+F5.
   The Embarcadero SQL Debugger stops the script execution and terminates the session.

For more information, see Embarcadero SQL Debugger functionality.

Restart
Restart lets you terminate the current debug session and open a new one. When the new session opens, DBArtisan prompts you for new input parameters.

To use the Restart facility, do the following:

1. On the Debug menu, click Restart.
   OR
   Press CTRL+SHIFT+F5.
   The Embarcadero SQL Debugger restarts the debug session.

For more information, see Embarcadero SQL Debugger functionality.

Break
Break lets you pause the debug session.

To use the Break facility, do the following:

1. On the Debug menu, click Break.
   The Embarcadero SQL Debugger suspends the debug session.

For more information, see Embarcadero SQL Debugger functionality.

Close
Close lets you close the DDL Editor and the Embarcadero SQL Debugger.

To use the Close facility, do the following:

1. On the DDL Editor toolbar, click Close.
   OR
   In the upper right corner of the window, click Close.
   OR
   In the DDL Editor window, right-click, and then click Close.
   The Embarcadero SQL Debugger closes the debug session.
For more information, see Embarcadero SQL Debugger functionality

Using the Embarcadero SQL Debugger
This section offers a general overview of how to use Embarcadero SQL Debugger’s full range of debugging functionality.

• Opening a Debugging session
• Debugging an SQL Script with Embarcadero SQL Debugger

Opening a Debugging session
When you open a debugging session, DBArtisan opens the Embarcadero SQL Debugger interface. For details, see Embarcadero SQL Debugger interface.

If the target script requires input parameters, DBArtisan opens a dialog box prompting you for the necessary input parameters before displaying the target code in the SQL Editor window. When DBArtisan displays the target script in the SQL Editor window, you can begin debugging. For details, see Debugging an SQL Script with Embarcadero SQL Debugger.

NOTE: Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows, mbarcadero SQL Debugger for Sybase, and Embarcadero SQL Debugger for Microsoft only let you debug the SQL script of procedures or triggers.

To debug an object, do the following:

1. In the Datasource Explorer, click the object node.
   DBArtisan displays the objects in the right pane of the application.

   • Right-click the target object, and select Open.
   DBArtisan opens the Object Editor.

   OR

   • Right-click the target object and select Extract.
   DBArtisan opens the DDL Editor displaying the code of the target object.

2. On the toolbar, click Debug.

   OR

   In the DDL Editor window, right-click, and then click Debug.

   If the script requests input parameters, DBArtisan opens a dialog box prompting for parameter values. If the script does not require input parameters, DBArtisan displays the script in the DDL Editor window for you to begin debugging. For details, see Debugging an SQL Script with Embarcadero SQL Debugger.

   NOTE: You cannot use the Embarcadero SQL Debugger until it has fully initialized.

3. In the dialog box, specify the appropriate parameter values, and then click OK or Continue.

   DBArtisan displays the script in the DDL Editor window.

   NOTE: If the script requires Sybase or Oracle types (tables, records, or Booleans) as input parameters, the Embarcadero SQL Debugger generates an anonymous block.
Debugging an SQL Script with Embarcadero SQL Debugger

After you open a debugging session and enter any required input parameters, you can begin working with your script in the Embarcadero SQL Debugger.

Debugging an SQL Script
To debug a SQL Script, do the following:

1. On the Debug menu, click one of the SQL Debugger options (Step Into, Step Over, and so forth) or click Go.
   
   **NOTE:** You can monitor the progress of your debug session in the Variables window.

2. On the Debug menu, click Breakpoint.
   
   OR
   
   Press F9.
   
   **NOTE:** When you set a breakpoint, the Call Stack window shows what was called before the breakpoint.

   **NOTE:** You can use the Run to Cursor option to test the lines of code between a breakpoint and your cursor (indicated by the yellow arrow in the DDL Editor).

To check your variables:

1. In the DDL Editor, click a variable in your script and drag it to the Watch window.

2. In the Watch window, change the value of the watch variable, and then click Go to run your script and see the results of the new value.

To check a record in stored objects:

1. Drag the record to the Watch window.

2. In the Watch window, change the value of the record, then click Go to run your script and see the results of the new value.

To check the dependencies:

1. In the Dependency Tree window double-click the target dependent object to extract the code into a new DDL Editor.

2. Step through the script while monitoring the Dependency Tree Window.

3. When you finish debugging the script, click Close or Exit.

   DBArtisan closes the T-SQL Debugger DDL Editor.

   **NOTE:** When you exit a debug session and reenter it, the Embarcadero SQL Debugger for MSSQL retains any watch variables or breakpoints you have set.

Tutorial sessions
The following topics provide walkthrough sessions geared to new users of the Embarcadero SQL Debugger:

• Debugging a Sample Script with Embarcadero SQL Debugger for Microsoft

• Debugging a Sample Script with Embarcadero SQL Debugger for Oracle
Debugging a Sample Script with Embarcadero SQL Debugger for Microsoft

This Getting Started section demonstrates basic debugging functionality. You will debug two procedures using the Embarcadero SQL Debugger for Microsoft.

The section topics are designed to familiarize you with basic debugging features and functionality:

- Getting Started with Embarcadero SQL Debugger for Microsoft
- Testing a Procedure
- Starting the Debugging Session
- Breakpoints
- Step Into
- Step Out
- Correcting the Script

Getting Started with Embarcadero SQL Debugger for Microsoft

This part of Debugging the Sample Script explains how to create the following two procedures to be used for debugging:

- check_modulo
- calculate_sum_with_overflow_bug

NOTE: The procedure calculate_sum_with_overflow_bug intentionally includes a bug which prevents it from executing successfully. You will use the Embarcadero SQL Debugger for Microsoft to identify this bug.

The Getting Started section includes:

- Creating Procedure 1
- Creating Procedure 2
- Confirming the Creation of the Procedures

Creating Procedure 1

Procedure 1, check_modulo, calculates the modulo of any two user-specified numbers. The user passes the numbers into the procedure as input parameters. The procedure returns the result as an output parameter. If the modulo equals zero, procedure execution returns the output “YES”. If the modulo is not zero, procedure execution returns the output “NO”. This procedure is nested in the second procedure, calculate_sum_with_overflow_bug.
To create this procedure, open DBArtisan, connect to a MSSQL datasource, open a new SQL editor and, in the SQL editor, type or copy and paste the following code:

```sql
CREATE PROCEDURE username.check_modulo
    @p_dividend_in  INT,
    @p_divisor_in  INT,
    @result VARCHAR(3) OUTPUT
AS
IF @p_dividend_in % @p_divisor_in = 0
    SELECT @result = 'YES'
ELSE
    SELECT @result = 'NO'
go
```

**NOTE:** For the purposes of this walk-through, this procedure was created under the user name Spence. Before executing the DDL above, substitute your user name for the word “username”.

1. **Start DBArtisan.**
2. **Connect to a Microsoft SQL Server datasource.**
3. **On the Datasource menu, click the database node, and then click the target database.**
   
   **NOTE:** For this walk-through, we recommend that you select a non-production database.

4. **On the Main toolbar, click New.**
   
   OR
   
   Press **CTRL+N**.
   
   DBArtisan opens an SQL Editor in the current workspace.

5. **In the SQL Editor, type the DDL for procedure check_modulo.**

   **NOTE:** Substitute your user name once in the DDL for this procedure.

6. **On the SQL Editor toolbar, click Execute.**
   
   DBArtisan executes the script and creates Procedure 1, then opens the SQL Editor Results Tab with the results of the script execution. If you were not able to create the procedure, check the error messages to determine the problem.

Creating Procedure 2

Procedure 2, calculate_sum_with_overflow_bug, requires two user-specified numbers as input parameters. Upon execution, the procedure calculates the sum of the all numbers divisible by five between the two user-specified numbers. This procedure calls sample procedure 1 (check_modulo) to calculate the modulo of the user-specified numbers.

**NOTE:** The procedure calculate_sum_with_overflow_bug intentionally includes a bug which prevents it from executing successfully. You will use the Embarcadero SQL Debugger for Microsoft to identify this bug.

**CAUTION:** When you input parameters, enter the smaller number in the @p_num1_in int box.
To create this procedure, open DBArtisan, connect to a MSSQL datasource, open a new SQL editor and, in the SQL editor, type or copy and paste the following code:

```sql
CREATE PROCEDURE username.calculate_sum_with_overflow_bug
    @p_num1_in  INT,
    @p_num2_in  INT,
    @result TINYINT OUTPUT
/*INT-Integer (whole number) data from -2^31 (-2,147,483,648)
   through 2^31 - 1 (2,147,483,647).
TINYINT-Integer data from 0 through 255.*/
AS
DECLARE @temp INT
DECLARE @temp_1 INT
DECLARE @v_divisor INT
DECLARE @v_condition VARCHAR(3)
SET @temp = @p_num1_in
SET @temp_1 = 0
SET @v_divisor = 5
SET @v_condition = 'NO'
WHILE 1=1
BEGIN
    SELECT @temp = @temp + 1 /*Increase temp starting from p_num1*/
    IF @temp = @p_num2_in /*Check if we reached p_num2*/
        /*If yes, leave the LOOP*/
        BREAK
/*Call Procedure 2 to check if number is divisible by 5*/
EXEC username.check_modulo @temp,@v_divisor,@result=@v_condition output
    IF @v_condition = 'YES'
        SELECT @temp_1 = @temp_1 + @temp
END /*WHILE LOOP*/
SELECT @result = @temp_1
RETURN
go
```

**NOTE:** For the purposes of this walk-through, this procedure was created under the user name Spence. Before executing the DDL above, substitute your user name for the word "username".

### Creating Procedure 2

1. **Start DBArtisan.**
2. **Connect to a Microsoft SQL Server datasource.**
3. **On the Datasource menu, click the database node, and then click the target database.**
   **NOTE:** For this walk-through, we recommend that you select a non-production database.
4. **On the Main toolbar, click New.**
   OR
   **Press CTRL+N.**
   DBArtisan opens an SQL Editor in the current workspace.
5. **In the SQL Editor, type the DDL for procedure calculate_sum_with_overflow_bug.**
   **NOTE:** Substitute your user name twice in the DDL for this procedure.
6 On the SQL Editor toolbar, click Execute.

DBArtisan executes the script and creates Procedure 2, then opens the SQL Editor Results Tab with the results of the script execution. If you were not able to create the procedure, check the error messages to determine the problem.

Confirming the Creation of the Procedures
After you create Procedure 1 and Procedure 2, you can confirm their creation in DBArtisan’s Datasource Explorer.

Confirming the Creation of the Procedures
To confirm the creation of the procedures, do the following:

1 On the Explorer Tab, click the Explorer list, and then click Organize By Owner.

   The Explorer Tab refreshes with the new display configuration.

2 On the Explorer Tab, double-click the Databases node, and then double-click the target database node.

   DBArtisan displays the list of object owners.

3 Double-click your user name to display a list of your objects.

4 Double-click Procedures to display a list of procedures and confirm the creation of check_modulo and calculate_sum_with_overflow_bug.

Testing a Procedure
After you confirm the creation of the procedures, execute the procedure calculate_sum_with_overflow_bug (which includes a bug) to view its error message. This procedure requires two integer input parameters: @p_num1_in int and @p_num2_in int. For all integers between these two integers, this procedure identifies those divisible by 5, and then returns their sum.

   CAUTION: When inputting parameters, enter the smaller number in the @p_num1_in int box.

Testing a Procedure
To test a procedure, do the following:

1 On the Explorer Tab, right-click calculate_sum_with_overflow_bug, and then click Execute.

   DBArtisan opens the Procedure Execution window.

2 In the Value column of the @p_num1_in row, type 1.

3 In the Value column of the @p_num2_in row, type 11.

4 Click Execute.

   DBArtisan compiles the procedure and opens a Results Tab, displaying the sum 15. There are two numbers between 1 and 11 that are divisible by 5: 5, and 10. The sum of these two numbers is 15.

5 On the Explorer Tab, right-click calculate_sum_with_overflow_bug, and then click Execute.

   DBArtisan again opens the Procedure Execution window.

6 In the Value column of the @p_num1_in row, type 100.

7 In the Value column of the @p_num2_in row, type 121.
8. On the **Procedure Execution** window toolbar, click *Execute*.

  DBArtisan returns an error stating “Arithmetic overflow occurred”.

[Click here to continue the Getting Started section.]

**Starting the Debugging Session**

After you **test the procedure**, open the procedure in Embarcadero SQL Debugger for Microsoft and enter input parameters before debugging.

To start the debugging session, do the following:

1. On the **Explorer Tab**, right-click the procedure, `calculate_sum_with_overflow_bug`, and then click **Debug** to start the debug session.

  DBArtisan extracts the DDL for the procedure into a DDL Editor and opens the Procedure Execution dialog box.

2. In the **Value** column of the `@p_num1_in` row, type **100**.

3. In the **Value** column of the `@p_num2_in` row, type **121**.

4. Click **Continue**.

  DBArtisan closes the dialog box and opens the Embarcadero SQL Debugger interface.

The Embarcadero SQL Debugger includes the following five windows:

- **DDL Editor**
- **Watch Window**
- **Variables Window**
- **Call Stack Window**
- **Dependency Tree Window**

[Click here to continue the Getting Started section.]
Breakpoints
After you start the debugging session, insert a breakpoint into the code of the procedure calculate_sum_with_overflow_bug. Then run to the breakpoint. After you run to the breakpoint, Embarcadero SQL Debugger displays a yellow arrow on the red breakpoint icon and populates the Variables Window with values for the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>@temp</td>
<td>Current number</td>
</tr>
<tr>
<td>@p_num2_in</td>
<td>Second input parameter</td>
</tr>
<tr>
<td>@p_num1_in</td>
<td>First input parameter</td>
</tr>
<tr>
<td>@temp_1</td>
<td>Sum of the numbers, between the input parameters, divisible by 5</td>
</tr>
<tr>
<td>@result</td>
<td>Condition of the output parameter</td>
</tr>
<tr>
<td>@v_condition</td>
<td>Output parameter</td>
</tr>
<tr>
<td>@v_divisor</td>
<td>Divisor</td>
</tr>
</tbody>
</table>

1. In the DDL Editor, scroll to and click the following line:

```sql
EXEC username.check_modulo @temp,@v_divisor,@result=@v_condition output
```

**NOTE:** This line is located near the end of the procedure’s code.

2. On the Debug menu, click Breakpoint.
   OR
   Press F9.
   DBArtisan inserts a breakpoint (indicated by a dot) next to the number of the target line.

3. On the Debug menu, click Go.
   OR
   Press F5.
   DBArtisan Embarcadero SQL Debugger for Microsoft displays the value of the variables before the breakpoint in the Variables Window.

**Click here to continue the Getting Started section.**

Step Into
After setting the breakpoint, step into the dependent procedure, check_modulo.

To use the Step Into facility, do the following:

1. On the Debug menu, click Step Into.
   OR
   Press F11.
   DBArtisan extracts the DDL for the dependent, nested procedure into the DDL Editor.
2 Step Into again.

DBArtisan executes the next part of the code and displays the values for the variables in the Variables Window.

The Call Stack Window displays calls to the procedures.

Click here to continue the Getting Started section.

Step Out
After you Step Into the modulo_check (nested procedure) code, step back out and return to the calculate_sum_with_overflow_bug (outside procedure) code.

To use the Step Out facility, do the following:

1 On the Debug menu, click Step Out.

OR

Press SHIFT+F11.

DBArtisan opens the DDL Editor containing the code for calculate_sum_with_overflow_bug.

2 On the Debug menu, click Go.

OR

Press F5.

When the value of the variable, @temp is equal to the value of the variable, @p_num2_in, the WHILE LOOP is complete and the Embarcadero SQL Debugger for Microsoft continues to the next executable statement in the code.

3 While monitoring the value of the variables in the Variables Window, continue to click Go to cycle through the WHILE LOOP.

After executing the SELECT and RETURN statements, DBArtisan closes the Debugger and opens a DDL Editor to the Results Tab.

Click here to continue the Getting Started section.

Correcting the Script
When you finished Stepping Out of the nested code and encounter the error, to fully fix the bug, do the following:

1 Locate the source of the error

2 Scroll to the line in the script displaying the error

3 Analyze the code

4 Correct the error

5 Compile the corrected script

When you first executed the procedure, DBArtisan displayed the error message “Arithmetic overflow error for data type tinyint, value = 450”. According to Microsoft SQL Server Books Online: “This error occurs when an attempt is made to convert a float or real data type value into a data type that cannot store the result. This error prevents the operation from being completed.”

The data type used in this procedure (TINYINT) stores values from 0 to 255. The sum of the four numbers between 100 and 121 that are divisible by 5 (105, 110, 115, and 120) is 450. But because the TINYINT variable @result can only accept a maximum value of 255, DBArtisan returns the error message and the procedure fails.
To correct the script, do the following:

1. On the **Explorer Tab**, right-click **calculate_sum_with_overflow_bug**, and then click **Extract**.
   
   DBArtisan extracts the DDL for the procedure into a DDL Editor.

2. On the **Edit toolbar**, click **Find**.
   
   DBArtisan opens the Find dialog box.

3. In the **Find What** box, type **TINYINT**.

4. Click **Find Next**.
   
   DBArtisan selects the first occurrence of TINYINT.

5. Change the data type for @result from **TINYINT** to **INT**.

6. On the **DDL Editor toolbar**, click **Execute** to execute the modified script.
   
   DBArtisan executes the script and opens the Results Tab.

7. On the **Explorer Tab**, right-click **calculate_sum_with_overflow_bug**, and then click **Execute**.
   
   DBArtisan opens the Procedure Execution dialog box.

8. In the **Value** column of the **@p_num1_in** row, type **100**.

9. In the **Value** column of the **@p_num2_in** row, type **121**.

10. Click **Execute**.
    
    DBArtisan executes the procedure with the new data type and opens the Results Tab, returning the value 450. You successfully corrected the script and debugged the procedure.

For more information, see [Getting Started](#).

---

**Debugging a Sample Script with Embarcadero SQL Debugger for Oracle**

The DBArtisan installation includes a sample script intended to walk you through basic debugging functionality. The sample script creates a package that includes functions and procedures that you debug.

**NOTE:** To create the sample package, you must have CREATE privileges.

**Overview**

Debugging a Sample Script is divided into three sections that familiarize you with basic debugging features and functionality:

- **Getting Started** guides you through creating the package you use in Debugging Sample Script 1 and Debugging Sample Script 2.

- **Debugging Sample Script 1** guides you through debugging functionality and demonstrates the Embarcadero SQL Debugger for Oracle interface features.

- **Debugging Sample Script 2** guides you through debugging functionality and error correction.

**NOTE:** For the purposes of this walk-through we have created this package under the user name DEMO_SPENCE.
Getting Started with Embarcadero SQL Debugger for Oracle

The DBArtisan installation includes a sample script that you execute to create a package containing functions and procedures. These functions and procedures demonstrate basic debugging features available in the Embarcadero SQL Debugger for Oracle.

**NOTE:** To create the sample package, you must have CREATE privileges.

The DBArtisan installation places the script in the C:\Program Files\Embarcadero\DBA710\UsrScrpt directory.

**NOTE:** The default for the DBArtisan directory is C:\Program Files\Embarcadero. If you changed the default, the sample script is be located in the DBA710\UsrScrpt directory.

If you create the package included with the DBArtisan installation, you can delete it and its objects from your system when you finish working with them. The objects to delete:

- The package COUNT_TIME_INTERVAL
- The package function WEEKEND_DAYS_( )
- The package function WORKING_DAYS_( )
- The package function YEARS_ELAPSED_BETWEEN_( )
- The procedure YEARS_ELAPSED
- The procedure YEARS_ELAPSED_Y2K

Click here to continue the Getting Started section.

Embarcadero SQL Debugger for Oracle Overview

The Getting Started section guides you through:

- Opening the sample debug script.
- Executing sample debug script.
- Changing the Explorer Tab display.
- Confirming the creation of the package, including its functions and procedures.

**Getting Started**

1. Start DBArtisan.
2. On the **File** Menu, click **Open**.
   
   DBArtisan opens the Open File(s) dialog box.
3. In the **Open File(s)** dialog box, go to **DBA710\UsrScrpt\DEBUGGER_DEMO.sql**, and then click **Open**.
   
   **NOTE:** During the installation DBArtisan places DEBUGGER_DEMO.sql in the following directory: C:\Program Files\Embarcadero\DBA710\UsrScrpt.

   DBArtisan opens the What type of file dialog box.
4. On the **What type of file** dialog box, click **The file includes the DDL to create a database object**, and then click **OK**.
   
   DBArtisan opens the target script in an SQL Editor.
5 On the **SQL Editor** toolbar, click **Execute** to execute the script and create the package.

DBArtisan executes the target script and opens the SQL Editor Results Tab, displaying the results of the script execution. If you were not able to create the package, check the error messages to determine the problem.

6 On the **Explorer Tab** list, click **Organize by Owner**.

DBArtisan displays a list of owners in the Datasource Explorer.

7 On the **Explorer**, double-click your owner name.

DBArtisan displays a list of your schema objects.

8 Under your owner node, double-click the **Packages** node.

DBArtisan displays `COUNT_TIME_INTERVAL`, confirming the package's creation.

Click here to continue the Getting Started section.

### Debugging Sample Script 1

Sample Script 1 demonstrates Embarcadero SQL Debugger's basic features and functionality with the function `WORKING_DAYS( )`, which counts the number of business days between two dates.

Debugging Sample Script 1 is divided into five parts:

- **Starting the Debug Session**
- **Entering Input Parameters**
- **Inserting Breakpoints**
- **Stepping Into**
- **Viewing Debug Session Results**

### Sample Script 1 - Starting the Debug Session

After you open and execute `DEBUGGER_DEMO.sql`, you can begin debugging Sample Script 1. To begin debugging the function `WORKING_DAYS( )`, start a debug session.

**Starting the Debug Session**

To start the debug session, do the following:

1. On the **Explorer Tab**, under the **Packages** node, double-click the `COUNT_TIME_INTERVAL` node.

   DBArtisan opens the `COUNT_TIME_INTERVAL` node and displays the following items:

2. Under the `COUNT_TIME_INTERVAL` node, double-click **Functions**.

   DBArtisan opens the Functions node and displays the following items:

3. Under the **Functions** node, right-click `WORKING_DAYS ( )`, and then click **Debug** to start the debug session.

   DBArtisan opens the Function Execution dialog box with the current date in the boxes.

Click here to continue the Getting Started section.
Sample Script 1 - Entering Input Parameters
After you start a debugging session, you can enter input parameters. You cannot debug a script that requires input parameters until you input those parameters in the Function Execution dialog box.

Input Parameters
To enter input parameters, do the following:

1. Click the **P_START_DATE DATE** box, and then click the drop-down arrow.
   DBArtisan opens a calendar.
2. On the calendar, click the left arrow to set the month to **November 1999**.
3. Click 1.
   DBArtisan displays 11/01/1999 in the Value column of P_START_DATE.
4. Click the **P_END_DATE DATE** box, and then click the drop-down arrow.
   DBArtisan opens a new calendar.
5. On the calendar, click the left arrow to set the month to **November 1999**.
6. Click 8.
   DBArtisan displays 11/08/1999 in the Value column of P_END_DATE.
7. Click **OK**.
   DBArtisan closes the Function Execution dialog box, and then opens the following five Embarcadero SQL Debugger for Oracle interface windows:
   - SQL Editor, which displays the SQL code for the function.
   - Watch window.
   - Variables window.
   - Call Stack window.
   - Dependency Tree window, which displays the dependent objects.

Click here to continue the Getting Started section.

Sample Script 1 - Inserting Breakpoints
After you input parameters in the Input Parameters dialog box, you can begin inserting breakpoints. In this example, you insert the breakpoints in the extracted dependent object code. After you extract this code, locate the target breakpoint lines by searching for the text DBMS_OUTPUT.

Breakpoints
To insert breakpoints, do the following:

1. In the **Dependency Tree window**, double-click the **COUNT_TIME_INTERVAL** package body.
   DBArtisan displays the SQL code for the package body in the SQL Editor window.
2. On the **Edit** toolbar, click **Find**.
   DBArtisan opens the Find dialog box.
3. On the **Find** dialog box, in the **Find What** box, type **DBMS_OUTPUT**.
4 Click **Find Next**.
   In the SQL Editor, DBArtisan highlights the first occurrence of DBMS_OUTPUT, on line 22.

5 On the **SQL Editor** toolbar, click **Breakpoint**.
   DBArtisan inserts a breakpoint next to the target line number.

6 On the **Find** dialog box, click **Find Next**.
   DBArtisan highlights the next occurrence of DBMS_OUTPUT.

7 Click **Find Next** a third time.
   DBArtisan highlights the next occurrence of DBMS_OUTPUT, on line 35.

8 On the **Find** dialog box, click **Cancel**.
   DBArtisan closes the Find dialog box.

9 On the **Edit** toolbar, click **Breakpoint** to insert a second breakpoint.
   You should now have breakpoints set at lines 22 and 35.

   Click here to continue the Getting Started section.

Sample Script 1- Stepping Into
After you insert breakpoints, you can step into the function code.

**Step Into**
To use the Step Into facility, do the following:

1 On the **SQL Editor** toolbar, click **Go**.
   Embarcadero SQL Debugger for Oracle begins debugging and runs to the first breakpoint, placing the yellow arrow on line 22.

2 On the **SQL Editor** toolbar, click **Step Into**.
   Embarcadero SQL Debugger for Oracle moves the yellow arrow to the next line of the code.

3 Click **Step Into** again to enter the LOOP block.
   Embarcadero SQL Debugger for Oracle displays the value of the variables in the Variables window.

4 Click **Step Into** again to start moving through the LOOP block.
   In the Variables window, Embarcadero SQL Debugger for Oracle updates the value of variable v_currdate from 01-NOV-1999 to 02-NOV-1999.

5 Click **Step Into** two more times.
   In the Variables window, Embarcadero SQL Debugger for Oracle updates the value of v_theday from NULL to Tuesday.

   **NOTE:** If you continued stepping through the LOOP block, the Embarcadero SQL Debugger for Oracle would continue to update v_currdate and v_theday until v_currdate is greater than p_end_date.

6 On the **SQL Editor** toolbar, click **Go**.
   Embarcadero SQL Debugger runs to the next breakpoint.
7 On the SQL Editor toolbar, click Go once more.

Embarcadero SQL concludes the debug session and displays the Debug Session Results box.

Click here to continue the Getting Started section.

Sample Script 1 - Viewing Debug Session Results
After Stepping Into and running to the end of the code, Embarcadero SQL Debugger for Oracle displays a Debug Session Results box containing the following information:

- Variable Output
- DBMS_OUTPUT Results

**NOTE:** In this example, the Embarcadero SQL Debugger for Oracle displays a Debug Session Results box because the sample program includes DBMS_OUTPUT.

**Debug Session Results**
To debug session results, do the following:

1 Click OK.

DBArtisan closes the Debug Session Results box and terminates your debug session.

Click here to continue the Getting Started section.

Debugging Sample Script 2
Sample Script 2 demonstrates Embarcadero SQL Debugger for Oracle’s functionality when used on a function containing a bug which prevents it from executing successfully. The buggy function, WEEKEND_DAYS( ), requires input parameters and counts the number of weekend days between two dates. In this section, use Embarcadero SQL Debugger for Oracle to identify the bug, and then correct the script so that it can execute successfully.

Debugging Sample Script 2 is divided into six parts:

- Executing the Function
- Starting the Debug Session
- Entering Input Parameters
- Inserting Breakpoints
- Stepping Into
- Correcting the Function

Sample Script 2 - Executing the Function
After you open and execute DEBUGGER_DEMO.sql, you can begin debugging Sample Script 2. To begin debugging the function WEEKEND_DAYS ( ), first execute the function to discover the type of error it returns when it fails to execute.
Executing the Function
To execute the function, do the following:

1. On the **Explorer Tab**, under the **Packages** node, double-click the **COUNT_TIME_INTERVAL** node.
   - DBArtisan opens the **COUNT_TIME_INTERVAL** node.
2. Double-click the **Functions** node.
   - DBArtisan opens the **Functions** node.
3. Click **WEEKEND_DAYS( )**, then right-click it and click **Execute**.
   - DBArtisan opens the Function Execution dialog box.
4. In the **Value** column of the **P_START_DATE** row, type **11/01/1999**.
5. In the **Value** column of the **P_END_DATE** row, type **11/30/1999**.
6. Click **Execute**.
   - DBArtisan attempts to execute the function but returns an error indicating that the character string buffer is too small.

[Click here to continue the Getting Started section.]

Sample Script 2 - Starting the Debug Session
After you unsuccessfully execute the function **WEEKEND_DAYS( )** and determine the nature of its execution error, you can start a debugging session to determine the actual cause of the error.

Starting the Debugging Session
To start the debugging session, do the following:

1. On the **Explorer Tab**, under the **COUNT_TIME_INTERVAL** node, under the **Functions** node, right-click **WEEKEND_DAYS( )**, and then click **Debug** to start the debug session.
   - DBArtisan opens the Function Execution dialog box.

[Click here to continue the Getting Started section.]

Sample Script 2 - Entering Input Parameters
After you **start the debug session**, you can enter input parameters in the Function Execution dialog box.

Entering Input Parameters
To enter input parameters, do the following:

1. At the end of the **P_START_DATE** row, click the drop-down arrow.
   - DBArtisan opens the calendar.
2. On the calendar, click **Left Arrow** to set the month to **November 1999**.
3. Click **1**.
   - DBArtisan displays 11/01/1999 in the Value column of the **P_START_DATE** row.
4. At the end of the **P_END_DATE** row, click the drop-down arrow.
   - DBArtisan opens a new calendar.
5 On the calendar, click Left Arrow to set the month to November 1999.

6 Click 30.
   DBArtisan displays 11/08/1999 in the Value column of the P_END_DATE row.

7 Click Continue.
   DBArtisan closes the Function Execution dialog box, and then opens the following five Embarcadero SQL Debugger for Oracle interface windows:
   • SQL Editor, which displays the SQL code for the function
   • Watch window.
   • Variables window.
   • Call Stack window.
   • Dependency Tree window, which displays the dependent objects.

Sample Script 2 - Inserting Breakpoints
After you enter input parameters, you can begin inserting breakpoints. In this example, you insert the breakpoints in the extracted dependent object code. After you extract this code, locate the target breakpoint lines by searching for a particular line of code.

Breakpoints
To insert breakpoints, do the following:

1 In the Dependency Tree Window, double-click the COUNT_TIME_INTERVAL package body.
   DBArtisan displays the SQL code for the package body in the SQL Editor.

2 On the Edit toolbar, click Find.
   DBArtisan opens the Find dialog box.

3 On the Find dialog box, in the Find What box, type Function weekend_days, and then click Find Next.
   Embarcadero SQL Debugger for Oracle highlights the first occurrence of Function weekend_days.

4 On the Find dialog box, click Cancel.
   DBArtisan closes the Find dialog box.

5 Click line 60, the first line of executable code:

6 On the SQL Editor toolbar, click Breakpoint.
   DBArtisan inserts a breakpoint next to the line number.

7 Click Go to start debugging and run to the breakpoint.
   Embarcadero SQL Debugger for Oracle places the yellow arrow on line 60 and populates the Variables window with the first set of variables in the function code.
   Embarcadero SQL Debugger for Oracle also populates the Call Stack window with everything called before the breakpoint.

Click here to continue the Getting Started section.
Sample Script 2- Stepping Into
After you set and run to the breakpoint, you can step into the function to locate the cause of the error. To locate the cause of the error, monitor the Variables window. As you step through the code, the Variables window updates with the value of the variables.

**Step Into**
1. On the SQL Editor toolbar, click **Step Into**.
   DBArtisan moves the yellow arrow to the next line of the code, line 64.
2. On the SQL Editor toolbar, click **Step Into**.
   Embarcadero SQL Debugger for Oracle’s Variables window updates the value of v_currdate to 02-NOV-1999.
3. On the SQL Editor toolbar, click **Step Into**.
   DBArtisan moves the yellow arrow to the next line of the code, line 66.
4. On the SQL Editor toolbar, click **Step Into**.
   DBArtisan moves the yellow arrow to the next line of the code, line 67, and, in the Variables window, updates the value of v_theday to Tuesday.
5. On the SQL Editor toolbar, click **Step Into**.
   DBArtisan moves the yellow arrow back to line 64 to repeat the loop.
6. On the SQL Editor toolbar, click **Step Into**.
   Embarcadero SQL Debugger for Oracle’s Variables window updates the value of v_currdate to 03-NOV-1999.
7. On the SQL Editor toolbar, click **Step Into**.
   DBArtisan moves the yellow arrow to the next line of the code, line 66.
8. On the SQL Editor toolbar, click **Step Into**.
   The Embarcadero SQL Debugger for Oracle locates the error. The application terminates the debug session, returns an error indicating that the numeric or value character string buffer is too small, extracts the COUNT_TIME_INTERVAL package code into an SQL Editor, and returns an error indicating the line on which the code failed.

[Click here to continue the Getting Started section.]

Sample Script 2 - Correcting the Script
After you step through the SQL code and locate the error, you can correct the bug in Sample Script 2. When Embarcadero SQL Debugger for Oracle locates an error, it extracts the target package body into an SQL Editor. To correct this script:

- Scroll to the incorrect line in the script
- Analyze the code
- Correct the error
- Execute the corrected SQL script
- Execute the WEEKEND_DAYS ( ) function
The code in Sample Script 2 fails on line 66, returning an error when the variable v_theday increments from the value Tuesday to the value Wednesday. The cause of this error is found in the declarations section of the function script, where the width of the VARCHAR2 variable v_theday is set to 8. Because "Wednesday" includes nine characters, the value of the variable v_theday fails when it attempts to place a nine-character value in an eight-character variable. To correct this error, increase the width of the variable v_theday to accommodate nine characters.

Correcting the Script
To correct the script, do the following:

1. On the Explorer Tab, under the Packages node, under the COUNT_TIME_INTERVAL node, right-click Package Body, and then click Extract.
   DBArtisan extracts the package body into an SQL Editor.
2. In the SQL Editor, scroll to line 57, the line defining the variable v_theday.
3. On line 57, change the value of the width from 8 to 9.
4. On the SQL Editor toolbar, click Execute to execute the script.
   DBArtisan successfully executes the script.
5. On the Explorer Tab, under the COUNT_TIME_INTERVAL package node, under the Functions node, click WEEKEND_DAYS ( ).
6. Right-click WEEKEND_DAYS ( ), and then click Execute.
   DBArtisan opens the Function Execution dialog box.
7. In the Value column of the P_START_DATE row, type 11/01/1999.
8. In the Value column of the P_END_DATE row, type 11/30/1999.
9. Click Execute.
   DBArtisan successfully executes the corrected function.
DBArtisan PL/SQL Profiler

The DBArtisan PL/SQL Profiler module lets Oracle 8i developers capture metrics of various PL/SQL programmable objects as they are executed in the database. Developers can use data collected in profile sessions to improve performance of PL/SQL code execution. DBArtisan PL/SQL Profiler collects and stores data in database tables that let you identify and isolate performance problems and provide code coverage information. The DBArtisan PL/SQL Profiler lets you:

- Graphically browse PL/SQL profiling data within the Explorer Tab
- View profiling data in the right pane of the application, which is populated as you navigate the Explorer Tab
- Start and stop PL/SQL profiling sessions with a single click
- Graphically analyze time spent in each programmable object (unit)
- Graphically analyze time spent in each source code line of a unit

Requirements:

- Oracle 8.1.5 or later
- Oracle built-in package DBMS_PROFILER
- Oracle tables:
  - PLSQL_PROFILER_RUNS
  - PLSQL_PROFILER_UNITS
  - PLSQL_PROFILER_DATA (user's schema)

The table below describes the sections of this chapter:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting Up the Profiler</td>
<td>This section describes the process of setting up DBArtisan PL/SQL Profiler.</td>
</tr>
<tr>
<td>Profiler Functionality</td>
<td>This section describes the functionality of DBArtisan PL/SQL Profiler.</td>
</tr>
<tr>
<td>Using the Profiler</td>
<td>This section describes how to run a profile session.</td>
</tr>
</tbody>
</table>

**NOTE:** The DBArtisan PL/SQL Profiler is an optional add-on module.

For more information, see:

- Setting Up DBArtisan PL/SQL Profiler
- DBArtisan PL/SQL Profiler Explorer
- DBArtisan PL/SQL Profiler Functionality
- Using DBArtisan PL/SQL Profiler

Setting Up DBArtisan PL/SQL Profiler

The Oracle profiling tables must be on the Oracle server before you can use the DBArtisan PL/SQL Profiler. The first time you open the PL/SQL Profiler, DBArtisan checks the server for the profiling tables. If the profiling tables are not on the server, DBArtisan automatically starts Oracle SQL*Plus, which installs profiling tables on the Oracle server.

For DBArtisan to run Oracle SQL*Plus (which installs the profiling tables) your Oracle server and client must meet the following conditions:
The Oracle server and the client have the same version of Oracle.

The client has the Oracle\BIN directory on the path.

The client has the Oracle file, SQLPLUS.exe in the Oracle\BIN directory.

The following Oracle files are in the Oracle\RDBMS\ADMIN directory:

- DBMSPBP.sql
- PROFLOAD.sql
- PROFTAB.sql
- PRVTPBP.blp

**NOTE:** If the Oracle server and the client machines are running different versions of Oracle, after running SQL*Plus, DBArtisan displays the following error message: Version of package is incompatible.

### DBArtisan PL/SQL Profiler Explorer

The DBArtisan PL/SQL Profiler displays profiling data in the right pane of the application, which is populated as you navigate the Explorer Tab.

The table below describes the nodes of the DBArtisan PL/SQL Profiler Explorer and the corresponding information in the right pane of the application:

<table>
<thead>
<tr>
<th>Node</th>
<th>Right pane information</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL/SQL Code Profiling</td>
<td>Contain all Comment, Run ID and Run Date/time data that is current stored in the Profiling tables.</td>
</tr>
<tr>
<td>Label\Comment level</td>
<td>Contains all Run ID and Run Date/time data for the specific Label\Comment.</td>
</tr>
<tr>
<td>Run level</td>
<td>Contains all Unit, Unit Name, Unit Type, Run Date/time data for the specific Run ID.</td>
</tr>
</tbody>
</table>

### DBArtisan PL/SQL Profiler Functionality

DBArtisan PL/SQL Profiler offers the following functionality:

- **Start**
- **Flush**
- **Run Summary**
- **Run Detail**
- **Unit Summary**
- **Unit Detail**
- **Clear Profile Table**
- **Stop**
Start
DBArtisan PL/SQL Profiler lets you begin a new profiling session or open a previous profiling session with the Start command.

Starting a New Profile Session
To start a new profile session, do the following:

1. On the Utilities menu, click PL/SQL Profiler, and then click Start.
   DBArtisan opens the PL/SQL Profiler - Start dialog box.
2. In the Profile Label box, type the name of the new profile.
   
   NOTE: Each user can own one or more Profiles.
3. Click OK.
   DBArtisan begins profiling.

Starting an Existing Profile Session
1. On the Utilities menu, click PL/SQL Profiler, and then click Start.
   DBArtisan opens the PL/SQL Profiler - Start dialog box.
2. Click the Profile Label list, and then click the existing profile.
3. Click OK.
   DBArtisan begins profiling.

For more information, see DBArtisan PL/SQL Profiler Functionality.

Flush
DBArtisan PL/SQL Profiler lets you move the data from the dynamic tables into analysis tables with the flush command.

The table below describes the options and functionality on the PL/SQL Profiler - Flush dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flush</td>
<td>Click to delete the data in a running profile.</td>
</tr>
<tr>
<td>Flush &amp; Analyze</td>
<td>Click to open the PL/SQL Profiler Run Detail window.</td>
</tr>
<tr>
<td>Cancel Button</td>
<td>Click to abort the flush and continue the profiling session.</td>
</tr>
</tbody>
</table>

NOTE: You can only Flush a running Profile.

Flushing a Profile
To flush a profile, do the following:

1. On the Utilities menu, click PL/SQL Profiler, and then click Flush.
   DBArtisan opens the PL/SQL Profiler - Flush dialog box.

For more information, see DBArtisan PL/SQL Profiler Functionality.
Run Summary
The DBArtisan PL/SQL Profiler Run Summary window lets you to view the following information for each of your profiles:

- Run ID
- Run Date
- Total Time

Opening the Run Summary Window
To open the Run Summary Window, do the following:

1. On the Utilities menu, click PL/SQL Profiler, and then click Run Summary.
   DBArtisan opens the PL/SQL Profiler - Run Summary window.
2. In the PL/SQL Profiler - Run Summary window, click the Label list, and then click the target profile to populate the table.

For more information, see DBArtisan PL/SQL Profiler Functionality.

Run Detail
The DBArtisan PL/SQL Profiler Run Detail window lets you to view the following information for each of your profiles:

- Run Number
- Run Date
- Run Time

The Run Detail Tab lets you:

- View the information for all runs or you can view profile information based on the unit type or unit owner.
- View results in milliseconds, seconds and minutes.
- View graphical displays of the profiling data that let you can navigate to the specific unit within the summary portion of the window.
- Specify the number of top lines to display in the graphical portion of the interface.

TIP: Each graph is a working object. You can select data in a graph and the corresponding line of source displays in the lower pane of the interface.

Opening the Run Detail Window
To open the Run Detail Window, do the following:

1. On the Utilities menu, click PL/SQL Profiler, and then click Run Detail.
   DBArtisan opens the PL/SQL Profiler - Run Detail window.
2. In the PL/SQL Profiler - Run Detail window:
   - Click the Label list box, and then click the target profile.
   - Click the Run list, and then click the target run.
   - Click the Unit Type list, and then click the target unit type(s).
   - Click the Unit Owner list, and then click the target unit owner(s) to populate the table.
For more information, see DBArtisan PL/SQL Profiler Functionality.

Unit Summary
The DBArtisan PL/SQL Profiler Unit Summary window lets you view the following information for each of your profiles:

- Run ID
- Run Date
- Run Time
- Unit Time
- Percentage of Run Time

The DBArtisan PL/SQL Profiler Unit Summary window lets you view results in milliseconds, seconds and minutes. The Unit Summary window also displays graphs of execution statistics for the top N runs and associated units. You can use the graphical displays to navigate to the specific run within summary portion of the window.

Opening the Unit Summary Window
To open the Unit Summary Window, do the following:

1. On the Utilities menu, click PL/SQL Profiler, and then click Unit Summary.
   DBArtisan opens the PL/SQL Profiler - Unit Summary window.
2. In the PL/SQL Profiler - Unit Summary window:
   - Click the Unit Owner list, and then click the target unit owner.
   - Click the Unit Name list, and then click the target unit name to populate the table.

For more information, see DBArtisan PL/SQL Profiler Functionality.

Clear Profile Table
DBArtisan PL/SQL Profiler lets you delete data from the user’s profile tables with the command Clear Profile Table.

Clearing a Profile Table
To clear a Profile Table, do the following:

1. On the Utilities menu, click PL/SQL Profiler, and then click Clear Profile Table.
   DBArtisan clears the profile table.
2. In the DBArtisan dialog box, if you are sure that you want to clear out the profiler tables, click Yes.

For more information, see DBArtisan PL/SQL Profiler Functionality.

Unit Detail
The DBArtisan PL/SQL Profiler Unit Detail window lets you view the following information for each of your profiles:

- Average Time
- Source
• PL/SQL Script

The DBArtisan PL/SQL Profiler Unit Detail window lets you view results in milliseconds, seconds and minutes. The Unit Detail window also provides two calculation options for viewing unit execution time as a percentage of total execution time (total run vs unit run). Additionally, also displays graphs of execution statistics for the top N run. You can use the graphical displays to navigate to the specific line within source code portion of the window. The graphical display portion of the window contains options for viewing advanced statistics.

The Advanced View of the DBArtisan PL/SQL Profiler Unit Detail window lets you view the following information for each of your profiles:

• Hit Lines
• Missed Lines
• Line Number
• Calls
• Total Time
• Percentage of the Total Time
• Average Time
• Minimum Time
• Maximum Time

Opening the Unit Detail Window
To open the Unit Detail Window, do the following:

1. On the Utilities menu, click PL/SQL Profiler, and then click Unit Detail.
   DBArtisan opens the PL/SQL Profiler - Unit Detail window.
2. In the PL/SQL Profiler - Unit Detail window, do any of the following:
   • Click the Label list, and then click the target profile.
   • Click the Run list, and then click the target run.
   • Click the Unit list, and then click the target unit to populate the table.
   • Right-click, and then click Show Only Hit Lines to populate the table with the Average Time and Source for hit lines.
   • Right-click, and then click Show Only Missed Lines to populate the table with the Average Time and Source for missed lines.

Opening the Unit Detail Window Advanced View
To open the Unit Detail Window Advanced View, do the following:

1. In the Unit Detail window, right-click, and then click Advanced View to populate the table with Advanced View information.

For more information, see DBArtisan PL/SQL Profiler Functionality.

Stop
DBArtisan PL/SQL Profiler Stop command pauses the data gathering operation. Stop & Analyze populates the summary tables so that you can view the Unit Detail and Run Summary windows.
The table below describes the options and functionality on the PL/SQL Profiler - Stop dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>Click to stop the profiling session.</td>
</tr>
<tr>
<td>Stop &amp; Analyze</td>
<td>Click to open the PL/SQL Profiler Run Detail window.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Click to continue the profiling session.</td>
</tr>
</tbody>
</table>

Stopping a Profiling Session
To stop a Profiling Session, do the following:

1. On the Utilities menu, click PL/SQL Profiler, and then click Stop.
   DBArtisan opens the PL/SQL Profiler - Stop dialog box.

For more information, see DBArtisan PL/SQL Profiler Functionality.

Using DBArtisan PL/SQL Profiler
The steps in this section provide a high level overview of running a profiling session, and cover the following processes:

- Starting the Session.
- Executing the Sample Script.
- Stopping and Analyzing the Session.

   NOTE: The first execution of a PL/SQL unit can take more time to execute because the code is loading into memory; subsequent runs take less time.

Using the DBArtisan PL/SQL Profiler
1. On the Utilities menu, click PL/SQL Profiler, and then click Start.
   DBArtisan opens the PL/SQL Profiler - Start dialog box.

2. In the Profile Label box, type the name of the new profile.
   NOTE: Each user can own one or more Profiles.

3. Click OK.
   DBArtisan begins profiling.

4. On the Datasource Explorer, execute on one of the following PL/SQL database objects:
   - Procedure
   - Function
   - Package Procedure
   - Package Function
   Profiler displays profiling data in the right pane of the application.
5 On the Utilities menu, click PL/SQL Profiler, and then click Stop.

DBArtisan opens the PL/SQL Profiler - Stop dialog box.

The table below describes the options and functionality on the PL/SQL Profiler - Stop dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>Click to stop the profiling session.</td>
</tr>
<tr>
<td>Stop &amp; Analyze</td>
<td>Click to open the PL/SQL Profiler Run Detail window.</td>
</tr>
<tr>
<td></td>
<td>Click the Label list, and then click the target profile.</td>
</tr>
<tr>
<td></td>
<td>Click the Run list, and then click the target run.</td>
</tr>
<tr>
<td></td>
<td>Click the Unit Type list, and then click the target unit type(s).</td>
</tr>
<tr>
<td></td>
<td>Click the Unit Owner list, and then click the target unit owner(s) to populate the table.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Click to continue the profiling session.</td>
</tr>
</tbody>
</table>

6 Use the Utilities menu to open any of the following PL/SQL Profiler windows:

- PL/SQL Profiler Run Summary
- PL/SQL Profiler Unit Summary
- PL/SQL Profiler Unit Detail

For more information, see:
Sample Profiling Session
Click here to continue the Using DBArtisan PL/SQL Profiler section.

Sample Profiling Session
The DBArtisan installation includes two scripts for the sample profiling session:

- PROFILER_BUILD_DEMO.SQL
- PROFILER_DEMO.SQL

The PROFILER_BUILD_DEMO.SQL creates the objects that you profile in the walk through, and the PROFILER_DEMO.SQL is what you profile during the walk through.

**NOTE:** To create the objects in the PROFILER_BUILD_DEMO.SQL script, you need CREATE privileges.

The sample script demonstrates the following features of the DBArtisan PL/SQL Profiler:

- Unit Detail
- Run Detail
- Show Only Hit Lines
- Advanced View

During the installation, DBArtisan places the scripts in the C:\Program Files\Embarcadero\Dba710\UsrScrpt directory.

**NOTE:** The default for the DBArtisan directory is C:\Program Files\Embarcadero. If you changed the default, the sample scripts are in the Dba710\UsrScrpt directory.
Overview
Sample Profiling Session is divided into six parts:

- Getting Started
- Starting the Session
- Executing the Sample Script
- Stopping the Session
- Re-running & Re-executing the Session
- Stopping & Analyzing

Click here to continue the Using DBArtisan PL/SQL Profiler section.

Sample Profiling Session - Getting Started
In this step of Sample Profiling Session, you create the objects that you profile in the walk through.

Overview
The Getting Started section guides you through:

- Opening PROFILER_BUILD_DEMO.SQL.
- Changing the Datasource Explorer Display.
- Confirming the Creation of the Package.

Getting Started
1. Start DBArtisan.
2. On the File menu, click Open.
   DBArtisan opens the Open Files dialog box.
   
   **NOTE:** The default for the DBArtisan directory is C:\Program Files\Embarcadero. If you changed the default, the sample scripts are located in the Dba710\UsrScrpt directory.

3. In the Open Files dialog box, type the path to the UsrScrpt directory, press ENTER, and then double-click PROFILER_BUILD_DEMO.SQL to open the script in a SQL Editor window.
   DBArtisan opens the PROFILER_BUILD_DEMO.SQL script in an SQL Editor window.

4. On the SQL Editor window, click Execute.
   DBArtisan executes the script and create the package.

5. On the Datasource Explorer window list, click Organize by Owner.
6. On the Datasource Explorer window, click the node of your owner name.
   DBArtisan displays your schema objects.

7. Double-click the Packages node to display PF_COUNT_TIME_INTERVAL and confirm its creation.
   
   **NOTE:** If you were not able to create the package, check the error messages to determine the problem.

Click here to continue the Using DBArtisan PL/SQL Profiler section.
Sample Profiling Session - Starting the Session
In this step of Sample Profiling Session, you start the profiling session.

To start the session, do the following:

1. On the **File** Menu, click **Open**.
   DBArtisan opens the Open Files dialog box.
2. In the **Open Files** dialog box, type the path to the UsrScrpt directory, press ENTER, and then double-click **PROFILER_DEMO.SQL**.
   DBArtisan opens the script in a SQL Editor window.
3. On the **Utilities** menu, click **PL/SQL Profiler**, and then click **Start**.
   DBArtisan opens the PL/SQL Profiler - Start dialog box.
4. In the **Profile Label** list, enter **DemoProfile**.
5. Click **OK**.
   DBArtisan begins the profiling session.

**NOTE:** If this is the first time you start the PL/SQL Profiler, DBArtisan displays a dialog box.

**NOTE:** Click Yes and DBArtisan opens SQL*Plus to create the tables. You need to start the profiling session again (see step 3 above.)

Click here to continue the Using DBArtisan PL/SQL Profiler section.

Sample Profiling Session - Executing the Sample Script
In this step of Sample Profiling Session, you execute the DEMO script.

To execute the sample script, do the following:

1. On the **SQL Editor** window toolbar, click **Execute**.
   DBArtisan executes the script and opens a Results Tab.

Click here to continue the Using DBArtisan PL/SQL Profiler section.

Sample Profiling Session - Stopping the Session
In this step of Sample Profiling Session, you stop the profiling run.

To stop the session, do the following:

1. On the **Utilities** menu, click **PL/SQL Profiler**, and then click **Stop**.
   DBArtisan opens the PL/SQL Profiler - Stop dialog box.
2 Click **Stop**.

**Click here to continue the Using DBArtisan PL/SQL Profiler section.**

**Sample Profiling Session - Re-running & Re-executing the Session**
In this step of Sample Profiling Session, you run the same profile session and execute the DEMO script again.

**Sample Profiling Session - Re-running & Re-executing the Session**
To re-run and re-execute the session, do the following:

1. In the **SQL Editor**, click the **Query Tab**.
2. On the **Utilities** menu, click **PL/SQL Profiler**, and then click **Start**.
   
   DBArtisan opens the PL/SQL Profiler - Start dialog box again.
3. Click the down arrow on the **Profile Label** list, and then click **DemoProfile**.
4. Click **OK**.
   
   DBArtisan begins the profiling session.
5. On the **SQL Editor** toolbar, click **Execute**.
   
   DBArtisan executes the script again and opens the Results Tab.

**Click here to continue the Using DBArtisan PL/SQL Profiler section.**

**Sample Profiling Session - Stopping & Analyzing**
In this step of Sample Profiling Session, you stop profiling and analyze the runs.

**Sample Profiling Session - Stopping & Analyzing**
To stop and analyze the sample profiling session, do the following:

1. On the **Utilities** menu, click **PL/SQL Profiler**, and then click **Stop**.
   
   DBArtisan opens the PL/SQL Profiler - Stop dialog box again.
2. Click **Stop & Analyze**.
   
   DBArtisan opens the PL/SQL Profiler - Run Detail window.
3. Click the **Run** list, and then click Run#x.
   
   **NOTE:** DBArtisan assigns a number to each profiling session. These numbers increase incrementally each time you run a profiling session. x= the number that was assigned to your first run.
   
   DBArtisan PL/SQL Profiler populates the grid with information on the procedure, package body and package specification.

   **NOTE:** For the purposes of this walk though we have created this package under the account SCOTT.
4. Click the **Run** list again, and then click Run#x for your second run.

   Notice this time there is no information on the package specification. It was created in the first run.
5 Right-click, and then click **Detail**.

DBArtisan PL/SQL Profiler opens the PL/SQL Profiler - Unit Detail window and populates the grid with the average time to execute each unit and the source code. Notice the time to execute SELECT object_name, in the example is 126 ms.

6 In the **PL/SQL Profiler - Unit Detail** window, click the **Run** list, and then click **Run** for your first run.

7 Click the **Unit** list, and then click **user name.PF_COUNT_SYSTEM_OBJECTS**.

Notice the time to execute SELECT object_name is considerably greater: in the example it is 24476 ms.

8 Right-click, and then click **Show Only Hit Lines**.

The DBArtisan PL/SQL Profiler shows only the lines of code that executed.

9 Right-click, and then click **Advanced View**.

The DBArtisan PL/SQL Profiler opens the **Advanced View** window.

10 Continue clicking the **Run** and **Unit** lists to compare the performance of each run and each session.

This concludes the Sample Profiling Session. You can delete the objects created during the Sample Profiling Session. They are:

- Check Constraints, PLSQL_PROFILER_UNITS, PLSQL_PROFILER_DATA
- Foreign Keys, PLSQL_PROFILER_UNITS, PLSQL_PROFILER_DATA
- Package, PF_COUNT_TIME_INTERVAL
- Package functions, WEEKEND_DAYS_( ), WORKING_DAYS_( ), YEARS_ELAPSED_BETWEEN_()
- PL/SQL code Profiles, DemoProfile
- Primary Keys, PLSQL_PROFILER_RUNS, PLSQL_PROFILER_UNITS, PLSQL_PROFILER_DATA
- Procedure, PF_COUNT_SYSTEM_OBJECTS
- Sequence, PLSQL_PROFILER_RUNNUMBER
- Tables, PLSQL_PROFILER_RUNS, PLSQL_PROFILER_UNITS, PLSQL_PROFILER_DATA
Capacity Analyst

Welcome to the Embarcadero Capacity Analyst. The Capacity Analyst collects critical database storage, object, and performance metrics at scheduled intervals and provides visual analysis and reporting mechanisms that allow for historical trend analysis and forecasting.

Capacity Analyst lets you:

- Create and maintain Capacity Collections that serve as containers for related statistics.
- Collect statistics in Capacity Collections.
- View graphic and report-styled presentations of trend and predictive analysis scenarios.
- Maintain functions regarding the handling of stored statistics.

The table below describes the major sections of Help.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting started with the Capacity Analyst</td>
<td>Provides information about installing, opening, and licensing Capacity Analyst.</td>
</tr>
<tr>
<td>Capacity Analyst tutorial</td>
<td>Helps you get started with Embarcadero Capacity Analyst. Provides the foundation you need to explore the many features and benefits of Capacity Analyst.</td>
</tr>
</tbody>
</table>

Geting started with the Capacity Analyst

Welcome to the Embarcadero Capacity Analyst. The Capacity Analyst helps you analyze current usage trends in your key databases and lets you forecast where your databases are heading in the future.

For more information, see:

- [What You Should Know Before Installation](#)
- [Capacity Analyst Repository Installation](#)
- [Uninstalling Capacity Analyst Repository](#)
- [Managing Repository Security](#)
- [Repository Information](#)
- [Licensing Capacity Analyst](#)

What You Should Know Before Installation

When you install the Analyst, several Embarcadero-generated stored objects are pushed onto the database server: These objects are as follows:
**SQL SERVER**

**Procedures**

<table>
<thead>
<tr>
<th>eas_apply_collection_filter</th>
<th>eas_apply_collection_schedule</th>
<th>eas_apply_purge_condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_apply_refresh_stats_condition</td>
<td>eas_create_collection</td>
<td>eas_delete_collection</td>
</tr>
<tr>
<td>eas_get_collection_metric_info</td>
<td>eas_get_metric_query</td>
<td>eas_get_metric_query_ex</td>
</tr>
<tr>
<td>eas_get_metric_query_ex1</td>
<td>eas_get_purge_condition</td>
<td>eas_get_purge_status</td>
</tr>
<tr>
<td>eas_get_refresh_stats_condition</td>
<td>eas_get_refresh_stats_status</td>
<td>eas_is_admin</td>
</tr>
<tr>
<td>eas_is_auto_purge_set</td>
<td>eas_is_collection_available</td>
<td>eas_is_collection_scheduled</td>
</tr>
<tr>
<td>eas_is_filter_all_set</td>
<td>eas_list_perf_stats</td>
<td>eas_list_wait_events</td>
</tr>
<tr>
<td>eas_purge_data</td>
<td>eas_refresh_stats</td>
<td>eas_resolve_collection_objects</td>
</tr>
<tr>
<td>eas_resolve_datastruct</td>
<td>eas_resolve_indexes</td>
<td>eas_resolve_perfstats</td>
</tr>
<tr>
<td>eas_resolve_tables</td>
<td>eas_resolve_waitevents</td>
<td>eas_run_collection</td>
</tr>
<tr>
<td>eas_run_datastruct</td>
<td>eas_run_perfstats</td>
<td>eas_run_waitevents</td>
</tr>
<tr>
<td>eas_set_auto_purge_status</td>
<td>eas_set_collection_status</td>
<td>eas_set_refresh_stats_status</td>
</tr>
<tr>
<td>eas_update_schedule_dates</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Foreign Keys**

<table>
<thead>
<tr>
<th>eas_collection_fk1</th>
<th>eas_collection_fk2</th>
<th>eas_collection_object_fk1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_collection_object_fk2</td>
<td>eas_filter_fk1</td>
<td>eas_filter_fk2</td>
</tr>
<tr>
<td>eas_group_recfk</td>
<td>eas_map_fk1</td>
<td>eas_object_fk</td>
</tr>
<tr>
<td>eas_object_type_fk</td>
<td>eas_option_fk1</td>
<td>eas_property_fk1</td>
</tr>
<tr>
<td>eas_property_fk2</td>
<td>eas_schedule_fk</td>
<td>eas_statistic_fk1</td>
</tr>
<tr>
<td>eas_statistic_fk2</td>
<td>eas_statistic_fk3</td>
<td></td>
</tr>
</tbody>
</table>

**Tables**

<table>
<thead>
<tr>
<th>eas_collection</th>
<th>eas_collection_object</th>
<th>eas_collection_sql</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_database</td>
<td>eas_filter</td>
<td>eas_group</td>
</tr>
<tr>
<td>eas_map</td>
<td>eas_object</td>
<td>eas_object_type</td>
</tr>
<tr>
<td>eas_option</td>
<td>eas_option_type</td>
<td>eas_property</td>
</tr>
<tr>
<td>eas_schedule</td>
<td>eas_statistic</td>
<td>eas_version</td>
</tr>
</tbody>
</table>

**Metadata**

<table>
<thead>
<tr>
<th>eas_column_map_insert</th>
<th>eas_database_insert</th>
<th>eas_group_insert</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_object_type_insert</td>
<td>eas_option_insert</td>
<td>eas_option_type_insert</td>
</tr>
<tr>
<td>eas_version_insert</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## DB2
### Foreign Keys

<table>
<thead>
<tr>
<th>eas_col_obj_fk1</th>
<th>eas_col_obj_fk2</th>
<th>eas_collection_fk1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_collection_fk2</td>
<td>eas_filter_fk1</td>
<td>eas_filter_fk2</td>
</tr>
<tr>
<td>eas_group_recfk</td>
<td>eas_map_fk1</td>
<td>eas_object_fk</td>
</tr>
<tr>
<td>eas_object_type_fk</td>
<td>eas_option_fk1</td>
<td>eas_property_fk1</td>
</tr>
<tr>
<td>eas_property_fk2</td>
<td>eas_schedule_fk</td>
<td>eas_statistic_fk1</td>
</tr>
<tr>
<td>eas_statistic_fk2</td>
<td>eas_statistic_fk3</td>
<td></td>
</tr>
</tbody>
</table>

### Index

<table>
<thead>
<tr>
<th>eas_collection_idxs</th>
<th>eas_collection_object_idxs</th>
<th>eas_filter_idxs</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_group_idxs</td>
<td>eas_map_idxs</td>
<td>eas_object_type_idxs</td>
</tr>
<tr>
<td>eas_option_idxs</td>
<td>eas_property_idxs</td>
<td>eas_schedule_idxs</td>
</tr>
<tr>
<td>eas_statistic_idxs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Metadata

<table>
<thead>
<tr>
<th>eas_column_map_insert</th>
<th>eas_group_insert</th>
<th>eas_object_type_insert</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_option_insert</td>
<td>eas_option_type_insert</td>
<td>eas_version_insert</td>
</tr>
</tbody>
</table>

### Tables

<table>
<thead>
<tr>
<th>eas_collection_object</th>
<th>eas_collection_sql</th>
<th>eas_database</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_group</td>
<td>eas_map</td>
<td>eas_object_type</td>
</tr>
<tr>
<td>eas_option</td>
<td>eas_option_type</td>
<td>eas_schedule</td>
</tr>
<tr>
<td>eas_version</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## ORACLE
### Foreign Keys

<table>
<thead>
<tr>
<th>eas_collection_fk1</th>
<th>eas_collection_fk2</th>
<th>eas_collection_object_fk1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_collection_object_fk2</td>
<td>eas_collection_sql_fk1</td>
<td>eas_filter_fk1</td>
</tr>
<tr>
<td>eas_filter_fk2</td>
<td>eas_map_fk1</td>
<td>eas_object_fk</td>
</tr>
<tr>
<td>eas_option_fk1</td>
<td>eas_property_fk1</td>
<td>eas_property_fk2</td>
</tr>
<tr>
<td>eas_schedule_fk</td>
<td>eas_statistic_fk2</td>
<td>eas_statistic_fk3</td>
</tr>
</tbody>
</table>

### Indexes

<table>
<thead>
<tr>
<th>eas_collection_fk1</th>
<th>eas_collection_fk2</th>
<th>eas_filter_fk1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_filter_fk2</td>
<td>eas_object_fk</td>
<td>eas_object_fk</td>
</tr>
<tr>
<td>eas_option_fk1</td>
<td>eas_property_fk1</td>
<td>eas_property_fk2</td>
</tr>
<tr>
<td>eas_schedule_fk</td>
<td>eas_statistic_fk1</td>
<td></td>
</tr>
</tbody>
</table>
### Metadata

<table>
<thead>
<tr>
<th>eas_column_map_insert</th>
<th>eas_object_type_insert</th>
<th>eas_option_insert</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_option_type_insert</td>
<td>eas_version_insert</td>
<td></td>
</tr>
</tbody>
</table>

### Tables

<table>
<thead>
<tr>
<th>eas_collection</th>
<th>eas_collection_object</th>
<th>eas_collection_sql</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_database</td>
<td>eas_filter</td>
<td>eas_map</td>
</tr>
<tr>
<td>eas_object</td>
<td>eas_object_type</td>
<td>eas_option</td>
</tr>
<tr>
<td>eas_option_type</td>
<td>eas_property</td>
<td>eas_schedule</td>
</tr>
<tr>
<td>eas_statistic</td>
<td>eas_version</td>
<td></td>
</tr>
</tbody>
</table>

### SYBASE

**Foreign Keys**

<table>
<thead>
<tr>
<th>eas_collection_fk1</th>
<th>eas_collection_fk2</th>
<th>eas_collection_object_fk1</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_collection_object_fk2</td>
<td>eas_filter_fk1</td>
<td>eas_filter_fk2</td>
</tr>
<tr>
<td>eas_group_recfk</td>
<td>eas_map_fk1</td>
<td>eas_object_fk</td>
</tr>
<tr>
<td>eas_object_type_fk</td>
<td>eas_option_fk1</td>
<td>eas_property_fk1</td>
</tr>
<tr>
<td>eas_property_fk2</td>
<td>eas_schedule_fk</td>
<td>eas_statistic_fk1</td>
</tr>
<tr>
<td>eas_statistic_fk2</td>
<td></td>
<td>eas_statistic_fk3</td>
</tr>
</tbody>
</table>

### Indexes

<table>
<thead>
<tr>
<th>eas_collection_idxs</th>
<th>eas_collection_object_idxs</th>
<th>eas_filter_idxs</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_group idxs</td>
<td>eas_map_idx</td>
<td>eas_object_idx</td>
</tr>
<tr>
<td>eas_object_type_idx</td>
<td>eas_option_idx</td>
<td>eas_property_idx</td>
</tr>
<tr>
<td>eas_schedule_idx</td>
<td>eas_statistic_idx</td>
<td></td>
</tr>
</tbody>
</table>

### Metadata

<table>
<thead>
<tr>
<th>eas_column_map_insert</th>
<th>eas_database_insert</th>
<th>eas_group_insert</th>
</tr>
</thead>
<tbody>
<tr>
<td>eas_object_type_insert</td>
<td>eas_option_insert</td>
<td>eas_option_type_insert</td>
</tr>
<tr>
<td>eas_version_insert</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Procedures

<table>
<thead>
<tr>
<th>sp_embt_ca_check_syb_version</th>
<th>sp_embt_ca_db_obj_space</th>
<th>sp_embt_ca_dbspace</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_embt_ca_delete_collection</td>
<td>sp_embt_ca_get_col_metric_info</td>
<td>sp_embt_ca_get_metric_query_ex</td>
</tr>
<tr>
<td>sp_embt_ca_get_purge_cond</td>
<td>sp_embt_ca_get_purge_status</td>
<td>sp_embt_ca_get_refresh_cond</td>
</tr>
<tr>
<td>sp_embt_ca_get_refresh_status</td>
<td>sp_embt_ca_is_admin</td>
<td>sp_embt_ca_is_auto_purge_set</td>
</tr>
<tr>
<td>sp_embt_ca_is_col_available</td>
<td>sp_embt_ca_is_col_scheduled</td>
<td>sp_embt_ca_is_filter_all_set</td>
</tr>
<tr>
<td>sp_embt_ca_purge_data</td>
<td>sp_embt_ca_run_collection</td>
<td>sp_embt_ca_run_datastruct</td>
</tr>
</tbody>
</table>
Opening the Capacity Analyst
Capacity Analyst runs as a tab in the main DBArtisan window.

To open the Capacity Analyst:
1. Start DBArtisan and connect to the datasource that you want to run reports or analysis against. For details, see Datasource Management.
2. From the Analyst menu select Capacity Analyst.
   OR
   On the Analyst Series toolbar click the Capacity Analyst button.

Capacity Analyst opens, displaying any existing collections.

Capacity Analyst Repository Installation
The Capacity Analyst Repository Installation Wizard lets you install the repository on each database the Capacity Analyst runs against. The repository holds all the captured metrics that you want to collect. For example, this can be the space used in an Oracle tablespace or the number of physical reads on a Microsoft SQL Server.

The wizard installs a set of objects to manage the captured data, and a security role to manage access to the repository and objects.

The Analyst Repository Installation Wizard lets you:
- Create a new repository.
- Upgrade an existing repository.
- Assign a repository owner.
Assign object placement in the database.

View feedback on the repository install actions.

Completing the Analyst Repository Installation Wizard

To install the repository, do the following:

1. On the **DBArtisan** toolbar, click the **Capacity Analyst** icon.

   DBArtisan opens Capacity Analyst. Capacity Analyst detects that no repository is present and opens the Analyst Repository Installation Wizard.

2. Use the following table as a guide to understanding and setting options in the Wizard,

<table>
<thead>
<tr>
<th>Panel</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create a new repository</td>
<td>Lets you create a new repository under a new schema.</td>
</tr>
<tr>
<td></td>
<td>Upgrade existing repository</td>
<td>Lets you upgrade the version of a current repository. Updates the stored procedures, and adds any new columns to tables. Also lets you use an existing repository that was created for a different Analyst.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> Available only if the wizard detects the presence of an earlier version.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use current login information</td>
<td><strong>IBM DB2 ONLY:</strong> Select to use the default connectivity information as the login id/password.</td>
</tr>
<tr>
<td></td>
<td>Login Name</td>
<td>Lets you type the server administrator ID.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> An administrator ID is not required after the initial repository setup.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Password</td>
<td>Lets you type the server administrator password.</td>
</tr>
<tr>
<td></td>
<td>Login As</td>
<td><strong>ORACLE ONLY:</strong> Lets you select one of three Oracle login modes: Default - Establishes connection with no system roles. SYSDBA - Lets you perform all DBA-related functions, like startup and shutdown.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>NOTE:</strong> SYSDBA is required for the SYS id when you log into a 9i instance. SYSOPER - Lets you perform many DBA-related functions, like startup and shutdown.</td>
</tr>
<tr>
<td></td>
<td>Using</td>
<td><strong>MICROSOFT SQL SERVER:</strong> Lets you specify authentication type, either SQL Server Authentication or Windows Authentication.</td>
</tr>
<tr>
<td></td>
<td>Database</td>
<td><strong>MICROSOFT SQL SERVER ONLY:</strong> Lets you select a database. SYBASE ASE ONLY: The following system databases are excluded from the list of potential targets for Repository installation:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Replication</em> <em>model</em> <em>sybsecurity</em> <em>sybsystemdb</em> <em>sybsystemprocs</em> <em>tempdb</em> <em>dbcdb</em> <em>sybdiag</em></td>
</tr>
<tr>
<td></td>
<td>Login</td>
<td><strong>MICROSOFT SQL SERVER and SYBASE ASE ONLY:</strong> Lets you select an owner. <strong>NOTE:</strong> Should have the CREATE PROCEDURE privilege on the storage device that is selected. For Oracle, the storage for tables and storage for indexes are tablespaces. The owner must have privileges for the tablespaces.</td>
</tr>
<tr>
<td></td>
<td>Owner</td>
<td><strong>ORACLE ONLY:</strong> Select a new user to own the repository or click New to open the User Wizard. For details, see User Wizard for Oracle.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>NOTE:</strong> Should have the CREATE PROCEDURE privilege on the storage device that is selected. For Oracle, the storage for tables and storage for indexes are tablespaces. The owner must have privileges for the tablespaces.</td>
</tr>
<tr>
<td></td>
<td>Storage for tables (and its available space)</td>
<td>Lets you separate the storage placement of repository tables and indexes. For Oracle, the storage for tables are tablespaces. Select from the list. Or click New to open the creation wizard.</td>
</tr>
</tbody>
</table>
3 When ready, click **Install**.

Capacity Analyst creates the repository. Capacity Analyst prompts you to grant other user accounts the privilege of using Capacity Analyst.

4 Select a basic security role:
   - Administrator – Users granted this privilege can create, edit, remove, and view capacity collection information.
   - User - Users granted this privilege can only view capacity collection information.

   **TIP:** You can easily grant or remove users from these roles at repository installation time or afterwards. For details, see [Managing Repository Security](#).

### Uninstalling Capacity Analyst Repository

The Capacity Analyst Repository Uninstall Wizard reverses the process completed by the Analyst Repository Installation Wizard.

#### Completing the Capacity Analyst Repository Uninstall Wizard

1 In the DBArtisan **Datasource Explorer**, connect to a datasource that has repository installed.

2 On the **Analyst** toolbar, click the **Capacity Analyst** button.

3 On the **Capacity Collection Editor** toolbar, select **Command, Repo Management**, and then **Uninstall Analyst Repo**.

   Capacity Analyst opens the Capacity Analyst Repository Uninstall Wizard.

---

<table>
<thead>
<tr>
<th>Panel</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Storage for indexes (and its available space)</strong></td>
<td>Lets you separate the storage placement of repository tables and indexes. For Oracle, the storage for indexes are tablespaces. Select from the list. Or click New to open the creation wizard.</td>
</tr>
<tr>
<td></td>
<td><strong>Password</strong></td>
<td>Lets you specify a password for the repository owner.</td>
</tr>
<tr>
<td></td>
<td><strong>Schema</strong></td>
<td><strong>IBM DB2 ONLY:</strong> Lets you select a schema. All the tables in the repository will be created for this schema. The default value is EMBTCA.</td>
</tr>
<tr>
<td></td>
<td><strong>Tablespace for Repository Tables</strong></td>
<td><strong>IBM DB2 ONLY:</strong> Lets you select the tablespace on which the repository tables are created.</td>
</tr>
<tr>
<td></td>
<td><strong>Tablespace for Repository Indexes</strong></td>
<td><strong>IBM DB2 ONLY:</strong> Lets you select the tablespace on which the repository tables are created.</td>
</tr>
<tr>
<td></td>
<td><strong>New</strong></td>
<td><strong>IBM DB2 ONLY:</strong> Click to open the Tablespace Wizard. For details, see <a href="#">Tablespace Wizard for IBM DB2 for Linux, Unix, and Windows</a>.</td>
</tr>
<tr>
<td></td>
<td><strong>Repository Login</strong></td>
<td>Displays repository login</td>
</tr>
<tr>
<td></td>
<td><strong>Execution Log</strong></td>
<td>Displays installation log.</td>
</tr>
<tr>
<td></td>
<td><strong>Execution Errors</strong></td>
<td>Displays any errors if they occur.</td>
</tr>
<tr>
<td></td>
<td><strong>Finish</strong></td>
<td>Click to complete the installation. Capacity Analyst installs the necessary files and displays the installation process. When the installation finishes, it displays a finished message. Click Finish to open the Capacity Analyst Tab. If the installation fails, Capacity Analyst lets you stop the repository installation. Capacity Analyst deletes the objects that it created.</td>
</tr>
</tbody>
</table>
4 Complete the wizard panels.
5 Click Finish.

Managing Repository Security
The Manage Repository Security dialog box lets you manage repository security.
The table below describes the options and functionality on the Manage Repository Security dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant</td>
<td>Click to grant privileges to target user.</td>
</tr>
<tr>
<td>Revoke</td>
<td>Click to revoke privileges from target user.</td>
</tr>
</tbody>
</table>

Completing the Manage Repository Security Dialog Box
1 In the DBArtisan Datasource Explorer, connect to a datasource that has repository installed.
2 On the Analyst toolbar, click the Capacity Analyst button.
3 On the Capacity Collection Editor toolbar, select Command, Repo Management, and then Manage Analyst Repo.
4 To grant privileges, select the target user, and then click Grant.
   Capacity Analyst opens the Grant Privilege(s) To dialog box. The Grant Privilege(s) To dialog box lets you select or clear the check boxes corresponding to the target privileges.
5 To revoke privileges, select the target user, and then click Revoke.
   Capacity Analyst opens the Revoke Privilege(s) From dialog box. The Revoke Privileges(s) From dialog box lets you select or clear the check boxes corresponding to the target privileges.
6 Click OK.

Repository Information
The Repository Information dialog box displays a tree view to present the information about the repository. The table below describes the standard nodes all datasource platforms:

<table>
<thead>
<tr>
<th>Node</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository Owner</td>
<td>Displays the login name of the repository owner.</td>
</tr>
<tr>
<td>Object Placement</td>
<td>Displays the database (for SQL Server and Sybase), tablespace (for Oracle), placement for tables and indexes (filegroup name - SQL Server, segment name - Sybase, tablespace - Oracle and DB2).</td>
</tr>
<tr>
<td>Repository Version</td>
<td>Schema and logic version for the tables/indexes and procedures respectively.</td>
</tr>
</tbody>
</table>

Microsoft SQL Server
The Repository Information dialog box displays the following information for Microsoft SQL Server:
• Repository owner
• Repository version
• Object placement
  • Database
  • Filegroup name where tables are stored
  • Filegroup name where indexes are stored.
• Dates
  • Date Created

**Sybase ASE**
The Repository Information dialog box displays the following information for Sybase ASE:
• Repository owner
• Repository version
• Object placement
  • Database
  • Segment name where tables are stored
  • Segment name where indexes are stored.
• Dates
  • Date Created

**Oracle**
The Repository Information dialog box displays the following information for Oracle:
• Repository owner
• Repository version
• Object placement
  • Tablespace name where indexes and tables are stored
• Dates
  • Date Created
  • Date Updated

**IBM DB2**
The Repository Information dialog box displays the following information for IBM DB2:
• Repository schema
• Repository owner
• Repository version
• Object placement
  • Tablespace name where indexes and tables are stored
• Dates
  • Date Created

Completing the Repository Information Dialog Box
1 In the DBArtisan **Datasource Explorer**, connect to a datasource that has repository installed.
2 On the Analyst toolbar, click the Capacity Analyst button.
3 On the Capacity Collection Editor toolbar, select Command, Repo Management, and then Manage Analyst Repo.
   Capacity Analyst opens the Repository Information dialog box.
4 To save repository information as a report, click Save.
   Capacity Analyst opens the Save Grid dialog box.
5 In Save In, select the location for the repository file.
6 In File Name, type the file name.
7 In Save as Type, select a file type.
8 Click Save.
   Capacity Analyst generates the report and returns to the Repository Information dialog box.
9 Click OK.
   Capacity Analyst closes the Repository Information dialog box.

Licensing Capacity Analyst
The Capacity Analyst is licensed per named server. You are issued one license key for the number of databases/servers you want to license. For example, if you purchase ten licenses for Capacity Analyst, you only need to obtain one license key from Embarcadero Technologies.

For more information, see Capacity Analyst License Manager.

Capacity Analyst License Manager
The Analyst License Manager lets you manage licenses.

The table below describes the options and functionality on the Analyst License Manager:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Summary</td>
<td>Displays basic licensing metadata for the product.</td>
</tr>
<tr>
<td>Available Datasources</td>
<td>Lets you select a datasource from the list of datasource available to license.</td>
</tr>
<tr>
<td>License</td>
<td>Click to add a license to a datasource.</td>
</tr>
<tr>
<td>Licensed Datasource(s)</td>
<td>Displays datasources already licensed.</td>
</tr>
<tr>
<td>Remove</td>
<td>Click to remove a license from a datasource.</td>
</tr>
<tr>
<td>Apply</td>
<td>Click to make you license choices permanent.</td>
</tr>
</tbody>
</table>
Completing the Capacity Analyst License Manager
To complete the Analyst License Manager, do the following:

1. Open DBArtisan.
2. On the Help menu, click Licensing.
   
   DBArtisan opens the License Configuration Dialog box.
3. Click License.
   
   DBArtisan opens the Analyst License Manager.
4. Select options.
5. To make your license choices permanent, click Apply.
6. To close the Analyst License Manager, click Cancel.

Capacity Analyst Add an Analyst License Dialog Box
The Add an Analyst License Details dialog box lets you add a license.

The table below describes the options and functionality on the Add a Analyst License Details dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a DBMS platform</td>
<td>Lets you select the DBMS platform.</td>
</tr>
</tbody>
</table>

For more information, see Capacity Analyst License Manager.

Capacity Analyst tutorial
Embarcadero Capacity Analyst offers intelligent insight into the past, current, and future inner workings of a database. By collecting historical database and performance metrics at user-defined intervals, Capacity Analyst provides historical trend analysis along with smart forecasting abilities that let you plan for the future needs of key databases. Capacity Analyst runs inside the Embarcadero DBArtisan console so you can perform all your database administration and analysis in one place.

Embarcadero Capacity Analyst lets you analyze current trends in your database storage and systems performance and helps you understand when corrective action is necessary. Capacity Analyst also helps you forecast the future needs of the your database so that your key systems can continue to meet the growing needs of its users.
About this Capacity Analyst tutorial
This short tutorial helps you get started with Embarcadero Capacity Analyst. After you complete this exercise, you will have the foundation you need to explore the many features and benefits of Capacity Analyst. You will learn how to set up Capacity Analyst and become familiar with creating and scheduling statistical collections. You will also understand how to analyze database trends and perform future forecasts of your database’s needs.

Summary of the tasks in this tutorial
The main categories of Capacity Analyst activities are:

- **Creating and editing collections** - a collection is basically a specified set of metrics. Capacity Analyst lets you use a wizard to create collections, specifying DBMS-specific metrics as content. You can edit and clone collections as well.

- **Executing collections** - based on counters, timers, and statistics packages available from the DBMS you are working against, collections store calculated totals. Executing a collection updates the totals in the collection. Depending on your needs, you can execute collections in an ad hoc fashion or you can schedule collections to be executed at regular intervals.

- **Analysis and reporting** - Capacity Analyst provides prepackaged reports and a visual analysis tool.

This tutorial walks you through a sample of these activities. You will create a collection with a small variety of metrics, schedule the collection to be updated each hour, and use the analysis and reporting tools.

Choosing an appropriate database for the tutorial exercises
The full power of the analysis and reporting tools can only be realized if you work through the exercises against an active database. For best results, use an active test bed or a production system.

Rationale for a two-part tutorial
While a collection stores historical statistics, totals cannot reflect database activities that occurred before the collection was created. In order to view meaningful analysis and reports, enough time must pass to allow a bare minimum of database activity. With that in mind, this tutorial is presented in two parts:

- **Session 1: Creating and scheduling a collection**
- **Session 2: Using analysis and reporting tools**

After completing the exercises in session 1, you should wait minimally four hours before starting the second session. This should allow sufficient database activity to allow for meaningful analysis.

Session 1: Creating and scheduling a collection
In Session 1, you will first ensure that you are properly set up to evaluate the Capacity Analyst. Subsequently, you will run the Capacity Analyst, build a small statistics collection, and ensure that it is updated on an hourly basis.

Getting Started with Capacity Analyst (optional)
The following topics prepare you for first use of the Capacity Analyst:

- **Downloading and installing DBArtisan**
• Starting DBArtisan and connecting to a datasource
• Installing the Capacity Analyst repository

When you have performed these tasks, or if you have already performed them, you can proceed to Creating and Scheduling a Capacity Collection Using Capacity Analyst.

Downloading and Installing DBArtisan

You can obtain the latest version of Capacity Analyst by downloading the DBArtisan software from the Embarcadero Web site at http://www.embarcadero.com. Capacity Analyst runs fully-contained within DBArtisan so you will need the DBArtisan product to use Capacity Analyst.

To download and install DBArtisan:

1. Locate the download area of the Web site and follow the steps indicated.
2. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for the duration of the trial period. After that time, a permanent license is needed.

Starting DBArtisan and connecting to a datasource

After installing DBArtisan, you must set up datasources to establish reusable connections to your database servers. For now, you can establish a single connection to the database/server you will use to evaluate Capacity Analyst. DBArtisan can detect and let you register any datasource in your network.

To connect to a database/server:

1. Start DBArtisan. For a default installation, Start > Programs > Embarcadero DBArtisan version number > DBArtisan (where version number corresponds to the version of DBArtisan that you downloaded).
2. On the Datasource menu, select Discover Datasources.
   
   The Discover Datasources dialog opens. After searching your network it lists all datasources to which you can connect.
3. Select the check box corresponding to the target datasource and click Register.

   The datasource is added to the left-hand pane of the datasource explorer, under the relevant DBMS folder. For example:
4 Right-click the datasource and select Connect from the context menu.

A login dialog opens.

5 Provide valid user name and password credentials and click OK.

Installing the Capacity Analyst repository
Capacity Analyst makes use of a repository to store historical statistics about your database. A repository is required for each database/server that you use with Capacity Analyst.

To install a repository on the datasource you connected to:
1 From the Analyst menu, select Capacity Analyst.

OR

On the Analyst Series toolbar, click the Capacity Analyst button.

DBArtisan opens Capacity Analyst. Capacity Analyst detects that no repository is present and opens the Analyst Repository Installation Wizard.

Depending on the type of DBMS type of the datasource you connected to, specific options will differ. In general, the Wizard prompts for login/password credentials, as well as server, database, and storage details.

2 Provide the information that the Wizard prompts you for and when complete, click the Install button and provide additional information when prompted.

Capacity Analyst creates the necessary repository objects.

Creating and Scheduling a Capacity Collection Using Capacity Analyst
When you open Capacity Analyst, Capacity Analyst displays a simple Capacity Collection Manager interface that lets you view existing collections, create new collections, and perform many other management functions.

Creating a New Capacity Collection Using Capacity Analyst
To get started collecting information for your database, you first need to create a new collection. A collection is essentially a set of related metrics or statistics that you want to report on. As you begin to make use of Capacity Analyst, you will develop a strategy for metrics you will store in each collection. For the purposes of this tutorial, you can build an arbitrary set of statistics into a collection.

You create a collection using the Capacity Collection Wizard.
To open the Capacity Collection Wizard:

1. On the Capacity Analyst toolbar, click the **New Collection Wizard** button.

Capacity Analyst opens the Capacity Collection Wizard.

The first panel is devoted to naming your capacity collection and specifying various collection properties like descriptions and optional job completion notifications.
To complete the Wizard:

1. Provide the information in the fields provided and click the Next button to continue.

   Capacity Analyst opens the Selection Types panel of the Capacity Collection Wizard. The second panel asks you what type of information you would like to collect for your database. While this panel prompts you to choose general categories, subsequent panels will let you provide more specific information about the choices you make here.

   The specific options differ according to the type of DBMS datasource you are connected to, but in general the categories are:

   • Storage structures - Depending on the DBMS you are working with, your analysis and reports can contain space and usage statistics on items such as databases and database devices, logs and backup space, partitions, tablespaces, filegroups, and files.
   • DB objects - Your analysis and reports can contain space and usage statistics on tables, indexes, and for some supported DBMS types, other database objects.
   • Performance-based statistics - if available for the DBMS you are connected to, this lets you use product-specific performance-based statistic packages.
   • Wait events - if available for the DBMS you are connected to, this lets you work with wait-based event information provided by the DBMS.

2. For the purposes of this tutorial, select all options on the panel and then click Next.

   The next set of panels let you select specific details. Each panel corresponds to one of the options you chose on the Selection Types panel.

   Most panels offer categories that let you drill down by expanding the category to expose specific options. You can then select specific options and move them from the Available list to the Selected list and then click Next to move to the next panel.

3. Complete all panels up until the Statistics tab is displayed.

   TIP: Since table and index statistics are supported for all DBMS, as an aid in subsequent sessions, select a variety of indexes and tables on the DB Objects panel.

   The Statistics tab prompts you to update the statistics you chose. In order to use the analysis tools offered by Capacity Analyst, you must execute the collection. This performs calculations and rollups on the metrics in the collection, letting you produce up-to-date reports and charts.

4. Look over the options available on the Statistics panel, enable statistics update and select one or more update options, and then click Next.

   The AutoPurge panel is displayed. It offers the option to delete older data, each time that you execute a collection.

5. Enable AutoPurge and click the Finish button.

Scheduling a Collection in Capacity Analyst

Immediately after your collection has been saved to the repository, Capacity Analyst prompts you if you would like to schedule the collection to run at specified intervals. It is not necessary that you schedule your collection to actually collect statistics in the repository. You can execute a collection in an ad-hoc manner at any time through the Capacity Collection Manager.

Depending on your requirements however, scheduling a collection may be the preferred route. For example, if you want to run packaged reports on a scheduled basis, scheduling collection of your collections will ensure up-to-date statistics in your scheduled reports.

For the purposes of this tutorial, you want the statistics in your collection updated every hour.
To schedule updates for your new collection:
1. Select Yes when asked if you would like to schedule your new collection.
   The Microsoft Task Manager scheduler opens.
2. Click Advanced to open the Advanced Schedule Options dialog.
3. Select Repeat task, specify that the tasks is to be repeated every hour, ensure that the task is repeated until a time at least four hours from now, and click OK to close the Advanced Schedule Options dialog.
4. Click OK to schedule your collection and dismiss the dialog.

TIP: You can add, edit, or remove any schedule for any collection at a later time by using the management functions supplied in the Capacity Collection Manager interface.

Reviewing other collection management functions
While you have completed all required tasks for Session 1, it is worthwhile looking at some of the options Capacity Analyst offers in creating and updating collections. Later, as you develop a strategy for setting up the collections for your enterprise, you will use additional Capacity Analyst features in managing your collections. With respect to maintaining your collections you have the following options:

Editing collections After creating a collection, you can modify general settings and the AutoPurge option. Since a collection stores totals and rolled up statistics that can be compromised, you cannot modify your choice of statistics or the objects you generate those statistics against.

Cloning collections Cloning lets you create a collection based on an existing collection and modify the metrics used and options specified in the original.

Deleting collections If you no longer have use for a collection or you are going to replace it, you can delete that collection.

Similarly, when you created your collection, you selected the option to update your statistics on an hourly basis. In some cases, you may require ad hoc reports or analysis. If so, you can update your collection manually.

To manually update a collection:
1. In the Capacity Collection Manager, select a collection.
2. On the Capacity Analyst toolbar, click the Execute button.

Depending on the volume of statistics you specified when creating the collection, this may take a few seconds. When complete the Last Run field will reflect the update.

There are other functions contained inside the Capacity Collection Manager, however those described above are likely the ones you will use the most often when managing your capacity collections. You will gain experience using these features as you learn more about Capacity Analyst.

Session 2: Using analysis and reporting tools
After several hourly updates of the statistics in the collection you created in Session 1, you are ready to start using the analysis tools. In Session 2, you will experiment with the Collection Analysis Viewer and generate one of the Capacity Analyst predefined reports.
Performing Trend Analysis Using Capacity Analyst

The Collection Analysis Viewer presents selected statistics in a graphical format and lets you view historical trends and extrapolate that data into the future.

To open the Collection Analysis Viewer, do the following:

1. Start DBArtisan again, and connect to the datasource you created earlier. If you need help, refer to Starting DBArtisan and connecting to a datasource.

2. On the Analyst menu, select Capacity Analyst.

3. Select the Capacity Collection that you recently created.

4. On the Capacity Analyst toolbar, click the Collection Analysis Viewer button.

Choosing the charted metrics you want to view

The first time that you open the Collection Analysis Viewer, it prompts you select the metrics you want to work with. You specify the metrics using the displayed Capacity Metric dialog. For example:
To specify the metrics that are to appear in the analysis:

1. From the Aggregation dropdown, select an aggregation type: Average, Maximum, or Minimum.
2. From the Type dropdown, select the type of metric you want to add, such as a table or index.
   
   **NOTE:** Certain Metric types allow you to further qualify using the Owner and Database dropdowns.
3. From the Available Object of Type list, choose a specific object or statistics type.
   
   The Metrics list is updated with the metrics available for the Type and Available Objects of Type selections you made.
4. Select a metric from the Metrics list and click Add.
   
   Capacity Analyst adds the metric to the Current Metrics list.
5. To add additional metrics, repeats step 1 through step 4. Keep in mind that the Collection Analysis View will calibrate its chart according to the metric with the largest value. If you are going to add multiple metrics, choose metrics that you expect will have similar ranges.
6. When finished adding metrics for the graphical portion of your analysis, click OK.

**Viewing Trends in Capacity Analyst**

When you finish adding metrics to the Collection Analysis Viewer, Capacity Analyst presents the data in graphical and grid form in the Collection Analysis Viewer. Your first step should be to ensure that the displayed timescale reflects the fact that you have been updating metrics for a matter of hours. You do this using the controls in the Capacity Metric Attributes area.
To optimize viewing of the metrics in your collection:

1. From the Metric Time Scale dropdown, choose Hours.

2. In the Last box, enter the number of hours since you scheduled your collection for hourly updates.

Having only statistics that cover a matter of hours, your viewing options are currently somewhat restrictive. However, during a later session you may want to experiment with the following viewing options:

- By default, the No Trend Analysis option is selected. This option shows historical data only. If you want statistics extrapolated into the future, you can use the Trend to Date or Trend to Value options to help you predict potential problems.

- You are currently using the Auto Period. It lets you quickly select an interval from the current day or the last statistics update. The Manual option lets you to have more granular control over the exact dates/times that Capacity Analyst uses to plot the data.

- To display the actual data values, position the pointer over the trend lines in the graph. Pop up values display the exact points on the chart.

- Each metric’s statistic displays in the Analysis viewer’s bottom grid with the high, low, and average value for the time period being shown for each metric.

- After your data has been displayed, you can continue to add or remove metrics from the Collection Analysis Viewer by using the Add or Remove toolbar buttons, and perform other functions using the various options available on the Collection Analysis viewer.

3. When ready to proceed to the next exercise, on the Capacity Analyst toolbar click the Close the Active Tab button.

Using Capacity Analyst predefined reports

An alternative forecasting method is through use of predefined Analysis reports. Capacity Analyst provides reports on growth trends of devices, databases, tables, rows, and other objects, as well as reports on fragmentation details and failure prediction. Like the Collection Analysis View, the reports show historical data and can be configured to extrapolate to a future date.

To generate a report on the fastest growing database objects:

1. Ensure that the collection you created is selected.

2. Click the Report button.

   The Capacity Analyst opens the Select Report Type dialog box.

3. To create a pre-defined trend analysis report, select Analysis Report and click OK.

4 Accept the default to create a New Analysis Report.

5 Click Next.

6 From the report drop down list, select ‘N’ Fastest Growing Objects. Note that there are many other reports available.

7 Accept the default Auto Entry time period of 1 day.

8 Accept the Day default rollup.

9 Specify the future date you want Capacity Analyst to use for its future ‘stop’ date.

10 Click Next.

11 Specify what databases/objects to include in the report.

12 Click Next.

13 Specify a Report File Name and a Report Title.

14 Click Execute.

Capacity Analyst opens the report in a new tab and stores an HTML-based version of the report in the location you specified in the Report File Name field.

The HTML report can be viewed in any web browser so you can build internal web sites of your forecasts for all interested parties, including project leads, and operation managers.

Using Capacity Analyst

The general categories of tasks performed using the Capacity Analyst are:

- **Creating and editing collections** - Each collection stores a set of statistics. How you set up your collections will depend on your own reporting and monitoring requirements. Capacity Analyst lets you choose the statistics in your collections and control options such as automatic purging and archiving of data.

- **Updating statistics in collections for use in reports and charts** - In order to produce up-to-date reports and charts, the rolled up data in collections must be updated. Capacity Analyst lets you update your collections manually or to schedule regular, automated updates.

- **Viewing charts and reports on collections** - The charts and reports available from the Capacity Analyst let you view trends and help you make forecasts regarding growth of your databases. The Collection Analysis Viewer lets you view graphical representation of the metrics in a collection as a function of time. Standard reports provide information on growth trends for devices, databases, tables, rows, and other objects, as well as fragmentation details and failure prediction.

Creating and editing collections

Typically, how you set up collections will be dictated by your own requirements. If your enterprise has several applications for example, each using multiple databases, you might be interested in viewing graphs and reports on an application-by-application basis, with each collection comprising statistics for several databases. At a lower level, you may be interested in viewing statistics on collections of table or indexes.

Similarly, how you set up collections can also be impacted by Capacity Analyst functionality. For example, a collection option lets you schedule statistics update at specified intervals. If your requirements include reports with different intervals, hourly as opposed to weekly for example, you could set up your collections accordingly. Or since the Collection Analysis Viewer calibrates a graph to the range of the highest valued statistic displayed, you might group statistics with similar expected values.
In short, you should consider all functionality available in Capacity Analyst before deciding on a collection setup. For details see Content and options in a collection and Viewing charts and reports on collections.

The following topics provide details on creating and editing Capacity Analyst collections:

- Creating a new collection
- Editing a capacity collection
- Cloning a collection
- Deleting a collection

Content and options in a collection

The reports and analysis you work with using Capacity Analyst include such statistics as space usage and access count details on database objects, databases, files and other storage mechanisms. They can also include performance-based and wait based statistics. The specific content and options available depend on the DBMS that DBArtisan is working with. The following topics provide summaries of available options and content on a product-by-product basis:

- IBM DB2 for Windows, Unix, and Linux capacity collection content and options
- Microsoft SQL Server capacity collection content and options
- Oracle capacity collection content and options
- Sybase capacity collection content and options

IBM DB2 for Windows, Unix, and Linux capacity collection content and options

When creating or editing a Capacity Analyst collection, you specify the statistics you want in that collection and specify additional notification and update options. The table below summarizes the available statistics and options on the tabs/panels of collection wizards and editors for this DBMS.

<table>
<thead>
<tr>
<th>Content/Option Category</th>
<th>Editor Tab or Wizard panel</th>
<th>Description or specific DBMS feature support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification/Notification</td>
<td>General</td>
<td>Lets you provide a name and description for the collection and optionally, provide an email address or net send name for notifications.</td>
</tr>
<tr>
<td>Types of statistics collected</td>
<td>Selection Types</td>
<td>Lets you select the types of content and options you want to include in the collection. You provide specifics in subsequent panels/tabs.</td>
</tr>
<tr>
<td>Database/storage structure space statistics</td>
<td>Storage Structures</td>
<td>A DB2 capacity collection can contain usage statistics on database (partition) space, DMS tablespace space, and SMS tablespace space.</td>
</tr>
<tr>
<td>Specific database object types for which to collect statistics</td>
<td>DB Objects</td>
<td>A DB2 capacity collection can contain usage statistics on indexes and tables</td>
</tr>
<tr>
<td>Update and rollup options</td>
<td>Statistics</td>
<td>A DB2 capacity collection offers the option to update statistics with additional specific choices for tables (distribution and column options), indexes (collection of extended and sample statistics), and access options (allowing read/write access during collection).</td>
</tr>
<tr>
<td>Automatic purging of old data</td>
<td>AutoPurge</td>
<td>Offers the option to automatically purge data older than a specified age, each time the collection is updated.</td>
</tr>
</tbody>
</table>
For information on how to create and edit collections, see:

- Creating a new collection
- Editing a capacity collection
- Cloning a collection

Microsoft SQL Server capacity collection content and options
When creating or editing a Capacity Analyst collection, you specify the statistics you want in that collection and specify additional notification and update options. The table below summarizes the available statistics and options on the tabs/panels of collection wizards and editors for this DBMS.

<table>
<thead>
<tr>
<th>Content/Option Category</th>
<th>Editor Tab or Wizard panel</th>
<th>Description or specific DBMS feature support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification/Notification</td>
<td>General</td>
<td>Lets you provide a name and description for the collection and optionally, provide an email address or net send name for notifications.</td>
</tr>
<tr>
<td>Types of statistics collected</td>
<td>Selection Types</td>
<td>Lets you select the types of content and options you want to include in the collection. You provide specifics in subsequent panels/tabs.</td>
</tr>
<tr>
<td>Database/storage structure space statistics</td>
<td>Storage Structures</td>
<td>An SQL Server capacity collection can contain size statistics on database backup space and database log backup space, and usage and size statistics on database space, filegroups, and files.</td>
</tr>
<tr>
<td>Specific database object types for which to collect statistics</td>
<td>DB Objects</td>
<td>An SQL Server capacity collection can contain usage and size statistics on indexes and tables.</td>
</tr>
<tr>
<td>Wait-event-based statistics</td>
<td>Wait Events</td>
<td>An SQL Server capacity collection can contain statistics on SQL Server Wait Types</td>
</tr>
<tr>
<td>Performance-based statistics</td>
<td>Performance Statistics</td>
<td>Provides statistics in the following performance counter categories: Access Methods, Buffer Manager, Buffer Partition, Databases, General Statistics, Latches, Locks, Memory Manager, SQL Statistics</td>
</tr>
<tr>
<td>Update and rollup options</td>
<td>Statistics</td>
<td>Provides the option to update statistics with additional options to specify a scan range, specify index or columns update, and recompute statistics</td>
</tr>
<tr>
<td>Automatic purging of old data</td>
<td>AutoPurge</td>
<td>Offers the option to automatically purge data older than a specified age, each time the collection is updated.</td>
</tr>
</tbody>
</table>

For information on how to create and edit collections, see:

- Creating a new collection
- Editing a capacity collection
- Cloning a collection
Oracle capacity collection content and options

When creating or editing a Capacity Analyst collection, you specify the statistics you want in that collection and specify additional notification and update options. The table below summarizes the available statistics and options on the tabs/panels of collection wizards and editors for this DBMS.

<table>
<thead>
<tr>
<th>Content/Option Category</th>
<th>Editor Tab or Wizard panel</th>
<th>Description or specific DBMS feature support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification/Notification</td>
<td>General</td>
<td>Lets you provide a name and description for the collection and optionally, provide an email address or net send name for notifications.</td>
</tr>
<tr>
<td>Types of statistics collected</td>
<td>Selection Types</td>
<td>Lets you select the types of content and options you want to include in the collection. You provide specifics in subsequent panels/tabs.</td>
</tr>
<tr>
<td>Database/storage structure space statistics</td>
<td>Storage Structures</td>
<td>All tablespace, all non-temporary tablespaces, or select specific tablespaces</td>
</tr>
<tr>
<td>Specific database object types for which to collect statistics</td>
<td>DB Objects</td>
<td>Lets you include statistics on clusters, indexes, index partitions, rollback segments, tables, and table partitions.</td>
</tr>
<tr>
<td>Wait-event-based statistics</td>
<td>Wait Events</td>
<td>Lets you include statistics based on Oracle Wait Events.</td>
</tr>
<tr>
<td>Performance-based statistics</td>
<td>Global Stats</td>
<td>Lets you include Global Database Statistics.</td>
</tr>
<tr>
<td>Update and rollup options</td>
<td>Statistics</td>
<td>Offers the option to update statistics with additional options to compute or estimate statistics.</td>
</tr>
<tr>
<td>Automatic purging of old data</td>
<td>AutoPurge</td>
<td>Offers the option to automatically purge data older than a specified age, each time the collection is updated.</td>
</tr>
</tbody>
</table>

For information on how to create and edit collections, see:

- [Creating a new collection](#)
- [Editing a capacity collection](#)
- [Cloning a collection](#)

Sybase capacity collection content and options

When creating or editing a Capacity Analyst collection, you specify the statistics you want in that collection and specify additional notification and update options. The table below summaries the available statistics and options on the tabs/panels of collection wizards and editors for this DBMS.

<table>
<thead>
<tr>
<th>Content/Option Category</th>
<th>Editor Tab or Wizard panel</th>
<th>Description or specific DBMS feature support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification/Notification</td>
<td>General</td>
<td>Lets you provide a name and description for the collection and optionally, provide an email address or net send name for notifications.</td>
</tr>
<tr>
<td>Types of statistics collected</td>
<td>Selection Types</td>
<td>Lets you select the types of content and options you want to include in the collection. You provide specifics in subsequent panels/tabs.</td>
</tr>
<tr>
<td>Database/storage structure space statistics</td>
<td>Storage Structures</td>
<td>Lets you use Global Space Summary statistics for the datasource or space and usage statistics for specific database devices and databases.</td>
</tr>
<tr>
<td>Specific database object types for which to collect statistics</td>
<td>DB Objects</td>
<td>Lets you include statistics on indexes and tables.</td>
</tr>
</tbody>
</table>
For information on how to create and edit collections, see:

- Creating a new collection
- Editing a capacity collection
- Cloning a collection

Creating a new collection
Capacity Analyst has a Wizard that walks you through specifying the statistics and options in a collection. Before creating a new collection, you should be familiar with the material in Content and options in a collection.

To create a new collection:
1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. On the Capacity Analyst toolbar click the New Collection Wizard button.

The Capacity Collection Wizard opens.
3. Provide the information required on each panel of the Wizard. For more information on these options, see the relevant DBMS-specific topic under Content and options in a collection.
4. When done click Finish.

Capacity Analyst offers the option to schedule updates for the statistics in this collection.
5. If you decide to schedule statistics updates, use the Windows Task Scheduler to provide update details.

Capacity Analyst adds the new collection to the Capacity Collection list.

See the following tasks for information on tasks you can perform after creating a collection:

- Editing a capacity collection
- Updating statistics in collections for use in reports and charts
- Viewing charts and reports on collections
Editing a capacity collection

Because a collection stores rolled up statistics, all gathered and calculated over a specific interval, editing statistics in a collection could corrupt the data in a collection. Therefore, you cannot modify the statistics in a collection or the devices, databases or database objects that statistics are gathered against. You can only modify the basic identifiers for the collection and the refresh and autopurge options.

To edit an existing collection:

1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select the collection to be edited and then on the Capacity Analyst toolbar click the Open Collection button.
3. On the General tab, edit the name, description, email address, or net send name.
4. On the Statistics tab, enable or disable update of statistics when you save your changes.
5. On the AutoPurge tab, enable or disable autopurge of older data when you save your changes.
6. Click OK.

Cloning a collection

The Capacity Analyst lets you create a new collection from an existing collection. This lets you build a new collection starting from properties defined in the existing collection.

To make a copy of an existing collection:

1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select the collection to be cloned and then on the Capacity Analyst toolbar click the Clone Collection button.
  
  Capacity Analyst opens the Capacity Analyst Collection Wizard. With the exception of the collection name, all statistics and options specified for the new collection are identical to those for the existing collection.
3. Minimally, provide a name for the new collection. Complete each panel of the wizard, changing settings as required. For more information on these options, see the relevant DBMS-specific topic under Content and options in a collection.
4. When done click Finish.
  
  Capacity Analyst offers the option to schedule updates for the statistics in this collection.
5. If you decide to schedule statistics updates, use the Windows Task Scheduler to provide update details.
  
  Capacity Analyst adds the new collection to the Capacity Collection list.

See the following tasks for information on tasks you can perform after cloning a collection:

- Editing a capacity collection
- Updating statistics in collections for use in reports and charts
- Viewing charts and reports on collections
Deleting a collection
If you no longer use the reports or analysis associated with a collection, Capacity Analyst lets you delete the collection.

To delete a collection:
1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select a collection and then on the Capacity Analyst toolbar click the Delete button.
   
   Capacity Analyst prompts you to verify that you want to delete the collection.
3. Click Yes.

Updating statistics in collections for use in reports and charts
Once you have installed and set up Capacity Analyst on a server, raw statistics are gathered and stored in the repository on that server. In order to view up-to-date charts and reports, you must execute a collection. Executing the collection calculates totals and performs all rollups for that collection in preparation for reporting.

There are two methods of updating statistics in a collection:
- Manually updating collections, useful prior to running an ad hoc report
- Scheduling updates of collections to run regularly, useful when reporting is performed on a regular basis

Manually updating collections
You can manually execute a statistics collection. This calculates totals and statistical indicators such as averages used for reporting for that collection.

To manually execute a collection:
1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select a collection and then on the Capacity Analyst toolbar click the Execute button.
   
   When complete, the Last Run field reflects the time that the collection was updated.

After executing a collection, you can run reports or view charts for that collection, that are up-to-date as of the time that you executed the collection. For details, see Viewing charts and reports on collections.

Related topics:
- Creating and editing collections

Scheduling updates of collections
If you run reports or want to regularly archive statistics for intervals, you can schedule execution of statistics collections. This updates the totals, rollups, and statistical indicators such as averages, on a regular basis. You can schedule collection updates to occur regularly such as daily weekly, or monthly, or tie execution to events such as system startup, logon, or during idle time.
The following topics provide details on the scheduling tasks you can perform against a collection:

- Creating an update collection schedule
- Editing an update collection schedule
- Deleting an update collection schedule

Creating an update collection schedule
You can create an update collection schedule to have statistics in that collection updated at regular intervals or have updates triggered by events such as system startup or logon.

In setting up your schedule, keep the following points in mind:

- Some statistical collections depend on the underlying dynamics of the database objects being collected. For example, storage structure space metrics, table and index statistics, etc., are oftentimes best collected only once a day. Other statistics, such as wait events and global database metrics can be collected on a much more granular level (every hour, etc.)
- If you routinely stop and start your database at a certain time, you may want to ensure that your collection job is not scheduled to run during its downtime. Also realize that stopping and starting a database resets the counters used for wait events and global database metrics.
- If you use the Microsoft scheduler on your machine, remember to not turn your machine off during the times your collection is scheduled to run or the job will not run.

**TIP:** You can add, edit, or remove any schedule for any collection at a later time by using the management functions supplied in the Capacity Collection Manager interface.

To schedule updates for a collection:
1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select a Capacity Collection that currently does not have an associated execution schedule.
3. Click the Schedule button.

Capacity Analyst opens the Job Scheduler.
4. Use the Job Scheduler to specify the frequency of the report or the event that triggers the update, and other details of the schedule.
5. When finished, click OK.

Related topics:
- Editing an update collection schedule
- Viewing charts and reports on collections

Editing an update collection schedule
After creating an update schedule for a statistics collection, you can revise the schedule. You may wish to change the frequency of updates or the event which triggers an update, or more specific details of the update schedule.
To modify the update schedule for a collection:
1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select a Capacity Collection that currently has an associated execution schedule.
3. Click the Edit Schedule button.

Capacity Analyst opens the Job Scheduler.
4. Use the Job Scheduler to change the frequency of the report or the event that triggers the update, and other details of the schedule.
5. When finished, click OK.

Related topics:
• Scheduling updates of collections
• For information on modifying the statistics in a collection, see Editing a capacity collection
• Viewing charts and reports on collections

Deleting an update collection schedule
If you no longer want statistics in a collection to be calculated on a regular basis or if you want to create a new, completely reworked schedule, you can delete the existing schedule for a collection.

To unschedule a Capacity Collection:
1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select a Capacity Collection.
3. On the Capacity Analyst toolbar, click the Unschedule button.

Capacity Analyst unschedules the Capacity Collection.

Related topics:
• Scheduling updates of collections
• Creating an update collection schedule

Purging old data in a collection
Policy or practicality may dictate that the metrics in your analysis and reports not include data older than a specified date. Capacity Analyst lets you manually purge data older than a specified number of months or years.

NOTE: In addition to manually purging old data, when creating or editing a collection, you can set up the collection to automatically purge data older than a specified date on every update. For details, see Creating and editing collections.
To manually purge collection data older than a specified number of months or years:
1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select a Capacity Collection.
3. On the toolbar, click the Purge button.
   
   The Perform Manual Data Purge dialog opens.
4. Use the controls in the Purge data older than area to provide a specific number of months or years.
5. Click Execute and when prompted to confirm, click Yes.

Viewing charts and reports on collections
The preliminary steps in setting up Capacity Analyst are Creating and editing collections and Updating statistics in collections for use in reports and charts. Once your collections are defined and the statistics updated, Capacity Analyst lets you view forecasting/trends charts and reports on your collections. Key options are:

• Using the Collection Analysis Viewer - lets you view charted representations of collection metrics as a function of time.
• Working with Capacity Analyst reports - lets you work with predefined reports that focus on growth of databases, storage mechanisms, and specific database objects, as well as failure prediction and fragmentation.

Using the Collection Analysis Viewer
The Collection Analysis Viewer lets you perform ad hoc analysis of statistics in a collection. It presents the metrics in two formats: a graphical representation that shows values over time and a tabular format that includes metrics such as average, maximum, and minimum values.
The Collection Analysis Viewer lets you:

- View historical trends for a metric and extrapolate into the future
- Compare and contrast multiple metrics
- Vary the time scale for fine-grained or coarse-grained views

The first time that you open the Collection Analysis Viewer on a particular collection, you are prompted to select the metrics you want to view for that collection. Subsequently, when using the Collection Analysis Viewer against that collection, you can add metrics to and delete metrics from the collection.

To open the Collection Analysis Viewer on a collection:

1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select a Capacity Collection. For more information, see Creating and editing collections.
3. On the toolbar, click the Collection Analysis Viewer button.

**NOTE:** If this is the first time you have opened this particular collection, you are prompted to specify the statistics to display. For help with this step, see Adding metrics to the Collection Analysis Viewer for a collection.

Capacity Analyst opens the Collection Analysis View or the selected collection.

Viewing options available when working with the Collection Analysis Viewer include:
• Using the Delete Metric button to remove a statistic from the displayed metrics for a collection

• Using the zoom buttons on the Collection Analysis Viewer toolbar to zoom in and out on the report

• Using the refresh button on the Collection Analysis Viewer toolbar to update displayed values

• Using the Collection dropdown to open the Collection Analysis Viewer on a different collection

• Using the Metric Time Scale dropdown to choose between displayed time scales of hours, days, months, or years

• Using the trending and value dropdowns to extrapolate metric values out to specified values or specific dates

• Using the Period controls to change the date/time range displayed

• Viewing values for metrics at specific locations on the graph by floating the cursor over specific locations on the curve

Adding metrics to the Collection Analysis Viewer for a collection

There are two situations in which you can add metrics viewed using the Collection Analysis Viewer:

• The first time that you open the Collection Analysis Viewer for a given collection

• When you want to manually add metrics by clicking the Collection Analysis Viewer toolbar’s Add Metric button:
In both cases, Capacity Analyst opens a **Capacity Metric** dialog to let you add metrics.

For each metric you want to add to the Capacity Analyst Analysis Viewer:

1. From the **Aggregation** dropdown, select an aggregation type: **Average**, **Maximum**, or **Minimum**.
2. From the **Type** dropdown, select the type of metric you want to add.
   
   **NOTE:** Certain Metric types allow you to further qualify using the **Owner** and **Database** dropdowns.

3. From the **Available Object of Type** list, choose a specific object or statistics type.
   
   The **Metrics** list is updated with the metrics available for the **Type** and **Available Objects of Type** selections you made.

4. Select a metric from the **Metrics** list and click **Add**.
   
   Capacity Analyst adds the metric to the **Current Metrics** list.

5. To add additional metrics, repeat step 1 through step 4.

6. When finished adding metrics for the Capacity Analyst Collection Analysis Viewer, click **OK**.

   Capacity Analyst updates the Collection Analyst Viewer with the new metrics.

**Working with Capacity Analyst reports**

Capacity Analyst provides two report options. For details, see the following topics:

- **Running a Capacity Collection Summary Report** - lets you view a summary of the metrics and options currently defined for a collection.
- **Running an Analysis Report** - lets you view one of several predefined reports focusing on growth trends.
Running a Capacity Collection Summary Report
You can produce a high-level summary of the content and options associated with a collection. It includes:

- Basic identification information such as the name, associated datasource, and owner of the collection
- A listing of the options, refresh and auto-purge for example, chosen for that collection
- A summary of the types of statistics available for use in the collection
- A detailed listing of the objects for which statistics are collected

**NOTE:** For a detailed listing of the specific statistics and options available for the DBMS you are working with, see the DBMS-specific heading under Content and options in a collection.

To run a Capacity Collection Summary Report:
1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select a Capacity Collection. For more information, see Creating and editing collections.
3. On the Capacity Analyst toolbar, click the Report button.

   Capacity Analyst opens the Select Report Type dialog.
4. Select Capacity Collection Summary Report and click OK.

   Capacity Analyst opens the summary report in a new tab.

Running an Analysis Report
Capacity Analyst lets you run a set of predefined reports against a collection. The reports and their availability by DBMS are:

- Database growth trends (Sybase, SQL Server, DB2)
- Tablespace failure prediction (Oracle)
- Tablespace growth trends (DB2, Oracle)
- Device growth trends (Sybase)
- Filegroup growth trends (SQL Server)
- File growth trends (SQL Server)
- Object growth trends (Sybase, SQL Server, DB2, Oracle)
- Object extent trends (Sybase, SQL Server)
- Object extent failure prediction
- Database backup growth trends (SQL Server)
- ‘N’ Fastest growing databases (Sybase, SQL Server, DB2)
- ‘N’ Fastest growing tablespaces (Oracle)
- ‘N’ Fastest growing objects (Sybase, SQL Server, DB2, Oracle)
- Table rows trend (Sybase, SQL Server, DB2, Oracle)
- Table chained/migrated rows trend (Oracle)
• Table forwarded records trend (Sybase, SQL Server)
• Index level trend (Oracle)
• Object fragmentation trends (Sybase, SQL Server, DB2)
• Tablespace fragmentation trend (Oracle)
• Database failure prediction (Sybase, SQL Server)
• DMS tablespace failure trends (Oracle)

To run an analysis report:
1. Open the Capacity Analyst. See Opening the Capacity Analyst.
2. Select a Capacity Collection. For more information, see Creating and editing collections.
3. On the Capacity Analyst toolbar, click the Report button.

Capacity Analyst opens the Select Report Type dialog.
4. Select Analysis Report and click OK.

5. Use the following table as a guide to understanding the settings and options for this wizard

<table>
<thead>
<tr>
<th>Panel</th>
<th>Tasks and settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel 1</td>
<td>Do you wish to create a new analysis report or ‘playback’ a previously saved report definition?</td>
<td>As the last step in generating a report, you are offered the option to save the report definition, for the purpose of running the report again. If you have already saved report definitions, you can use this panel to enable the Previously Saved Report option and select the existing report definition. Otherwise, to run an ad hoc report or start a new report definition, you can use the New Analysis Report option.</td>
</tr>
<tr>
<td>Panel 2</td>
<td>What report would you like to create?</td>
<td>Lets you select one of the predefined report types listed above.</td>
</tr>
<tr>
<td></td>
<td>Enter the time period sample to use for the report</td>
<td>Auto Entry lets you specify a number of days, months or years while Manual Entry lets you provide specific start and end dates.</td>
</tr>
<tr>
<td></td>
<td>Enter the time rollup to use for the report.</td>
<td>Lets your report display statistics rolled up by day, month, or year.</td>
</tr>
<tr>
<td></td>
<td>Enter the date to use for forward predictions.</td>
<td>Lets you enter a future end date that the Capacity Analyst uses to forecast.</td>
</tr>
<tr>
<td></td>
<td>Select options for the report.</td>
<td>The options offered differ by the type of report you selected and the DBMS that DBArtisan is working with.</td>
</tr>
<tr>
<td>Panel 3</td>
<td>Include objects for report.</td>
<td>Expand or collapse databases or owner lists to display specific object types and use the arrow buttons to move selected objects between the Available and Selected lists.</td>
</tr>
<tr>
<td>Panel 4</td>
<td>Data will be presented in text/grid format.</td>
<td>Lets you specify file name and path where the report will be stored and provide a title for the report.</td>
</tr>
<tr>
<td></td>
<td>Save this report definition?</td>
<td>Lets you save the report definition, specifying the file name and path for the definition.</td>
</tr>
</tbody>
</table>

6. When complete, click Schedule or Execute to run the report.
When complete, the report opens in a new DBArtisan tab.

**NOTE:** When viewing the report in DBArtisan, you can use the toolbar’s **Refresh** button to update values displayed.

A static, HTML version is also saved to with the file name and path you specified in the **Report File Name** field.
Performance Analyst

Performance Analyst is a robust client-side monitoring product that presents a complete picture of a database's performance. Performance Analyst is a quick, lightweight tool that lets you investigate performance bottlenecks.

The table below describes the major sections of Help.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
<td>Provides information about installing, opening, and licensing Performance Analyst.</td>
</tr>
<tr>
<td><strong>Application Basics</strong></td>
<td>Provides conceptual information about Performance Analyst's product design and navigation.</td>
</tr>
<tr>
<td><strong>Using Performance Analyst</strong></td>
<td>Shows how to use each component of the application. Using Performance Analyst breaks each process and procedure into simple step-by-step instructions.</td>
</tr>
<tr>
<td><strong>Performance Analyst for Oracle Tutorial</strong></td>
<td>Helps you get started with Embarcadero Performance Analyst. Provides the foundation you need to explore the many features and benefits of Performance Analyst.</td>
</tr>
<tr>
<td><strong>Performance Analyst for Oracle Expert Guide</strong></td>
<td>Performance Analyst includes expert help for all supported database platforms.</td>
</tr>
<tr>
<td><strong>Performance Analyst for Microsoft SQL Server Tutorial</strong></td>
<td>Helps you get started with Embarcadero Performance Analyst. Provides the foundation you need to explore the many features and benefits of Performance Analyst.</td>
</tr>
<tr>
<td><strong>Performance Analyst for Microsoft SQL Server Expert Guide</strong></td>
<td>Performance Analyst includes expert help for all supported database platforms.</td>
</tr>
<tr>
<td><strong>Performance Analyst for Sybase Tutorial</strong></td>
<td>Helps you get started with Embarcadero Performance Analyst. Provides the foundation you need to explore the many features and benefits of Performance Analyst.</td>
</tr>
<tr>
<td><strong>Performance Analyst for Sybase ASE Expert Guide</strong></td>
<td>Performance Analyst includes expert help for all supported database platforms.</td>
</tr>
<tr>
<td><strong>Performance Analyst for IBM DB2 for Linux, Unix, and Windows Tutorial</strong></td>
<td>Helps you get started with Embarcadero Performance Analyst. Provides the foundation you need to explore the many features and benefits of Performance Analyst.</td>
</tr>
<tr>
<td><strong>Performance Analyst for IBM DB2 for Linux, Unix, and Windows Expert Guide</strong></td>
<td>Performance Analyst includes expert help for all supported database platforms.</td>
</tr>
</tbody>
</table>
Welcome to Performance Analyst

Performance Analyst is a client-side only visual database monitor that displays current database performance data, including key performance metrics, bottleneck analysis, and user-configurable thresholds.

Running inside the Embarcadero DBArtisan console, or stand-alone in a Web browser, Performance Analyst lets you identify performance problems that threaten database availability or response times.

Supported DBMS
Embarcadero Technologies currently offers Performance Analyst for Oracle 8i - 10g. Performance Analyst does not support Oracle Clients 7.3.4, 8.1.7, and 9.2. Performance Analyst also supports Microsoft SQL Server and DB2.

For more information, see:
- What You Should Know Before Installation
- Operating System Prerequisites
- Opening the Performance Analyst
- Licensing Performance Analyst

What You Should Know Before Installation
When you install the Analyst, several Embarcadero-generated stored objects are pushed onto the database server:

These objects are as follows:

**ORACLE PROCEDURES**

<table>
<thead>
<tr>
<th>Procedures</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL_TABLESPACES</td>
</tr>
<tr>
<td>INSTANCESTARTUP</td>
</tr>
<tr>
<td>IO_DD_ARCHIVELOGMODE</td>
</tr>
<tr>
<td>IO_DD_DATAFILEIO</td>
</tr>
<tr>
<td>IO_DD_JOBQUEUESUMMARY</td>
</tr>
<tr>
<td>IO_DD_REDOWASTAGE</td>
</tr>
<tr>
<td>IO_DD_RUNNINGJOBSDETAIL</td>
</tr>
<tr>
<td>IO_DD_TABLESPACEIO</td>
</tr>
<tr>
<td>MEMORY_DD_BUFFERPOOLHITRAT</td>
</tr>
<tr>
<td>MEMORY_DD_CACHEDBLOCKSBYTS</td>
</tr>
<tr>
<td>MEMORY_DD_LATCHDETAILS</td>
</tr>
<tr>
<td>MEMORY_DD_LATCHRATIOS</td>
</tr>
<tr>
<td>MEMORY_DD_LIBCACHEPERF</td>
</tr>
<tr>
<td>MEMORY_DD_PGADETAIL</td>
</tr>
<tr>
<td>MEMORY_DD_PGAWORKAREA</td>
</tr>
<tr>
<td>MEMORY_DD_SHAREDPOOLUSAGE</td>
</tr>
<tr>
<td>OBJECT_DD_GLOBALOBJSUM</td>
</tr>
<tr>
<td>OBJECT_DD_INDEXES</td>
</tr>
<tr>
<td>ALL_USERS</td>
</tr>
<tr>
<td>IO_DD_ACCESSPATTERNDETAIL</td>
</tr>
<tr>
<td>IO_DD_CURRENTDISKSORTS</td>
</tr>
<tr>
<td>IO_DD_DBWRDETAIL</td>
</tr>
<tr>
<td>IO_DD_JOBQUEUESUMMARY</td>
</tr>
<tr>
<td>IO_DD_LOGWRDETAIL</td>
</tr>
<tr>
<td>IO_DD_REDOWASTAGE2</td>
</tr>
<tr>
<td>IO_DD_SESSIONROLLBACKACTIVITY</td>
</tr>
<tr>
<td>MEMORY_DD_BUFFERCACHERATIO</td>
</tr>
<tr>
<td>MEMORY_DD_BUFFERCACHEUTIL</td>
</tr>
<tr>
<td>MEMORY_DD_BUFFERPOOLMEMCO</td>
</tr>
<tr>
<td>MEMORY_DD_CACHEDTABLES</td>
</tr>
<tr>
<td>MEMORY_DD_LATCHRATIOS</td>
</tr>
<tr>
<td>MEMORY_DD_LIBCACHEHITRATIO</td>
</tr>
<tr>
<td>MEMORY_DD_PGAOVERVIEW</td>
</tr>
<tr>
<td>MEMORY_DD_PINNEDOBJECTS</td>
</tr>
<tr>
<td>MEMORY_DD_SHAREDPOOLFRAG</td>
</tr>
<tr>
<td>MEMORY_DD_SHAREDPOOLUTIL</td>
</tr>
<tr>
<td>OBJECT_DD_HOTCODE</td>
</tr>
<tr>
<td>OBJECT_DD_HOTTABLES</td>
</tr>
<tr>
<td>HOTTESTDBFILES</td>
</tr>
<tr>
<td>IO_DD_ACTIVEROLLBACKDETAILS</td>
</tr>
<tr>
<td>IO_DD_DAILYARCHIVELOGSUMMARY</td>
</tr>
<tr>
<td>IO_DD_ESTPHYSICALIOBYTEABLE</td>
</tr>
<tr>
<td>IO_DD_PROBLEMJOBS</td>
</tr>
<tr>
<td>IO_DD.RollbackIO</td>
</tr>
<tr>
<td>IO_DD_SESSIONSORTDETAIL</td>
</tr>
<tr>
<td>MEMORY_DD_BUFFERCACHEUTIL</td>
</tr>
<tr>
<td>MEMORY_DD_BUFFERPOOLOBJASS</td>
</tr>
<tr>
<td>MEMORY_DD_DATADICTDETAILS</td>
</tr>
<tr>
<td>MEMORY_DD_LIBCACHEHITRATIO</td>
</tr>
<tr>
<td>MEMORY_DD_PGAOVERVIEW</td>
</tr>
<tr>
<td>MEMORY_DD_SHAREDPOOLFRAG</td>
</tr>
<tr>
<td>MEMORY_DD_SHAREDPOOLUTIL</td>
</tr>
<tr>
<td>NOT_WAITING_SESSIONS_COUNT</td>
</tr>
<tr>
<td>OBJECT_DD_INVALIDOBJDET</td>
</tr>
<tr>
<td>OBJECT_DD_INVALIDOBJSUM</td>
</tr>
<tr>
<td>OBJECT_DD_REORGANALYSIS</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>OBJECTBUFFERPOOLPLACEMENT</td>
</tr>
<tr>
<td>RAC_LOCKS_LATCH</td>
</tr>
<tr>
<td>RAC_MEMORY_DATACACHE</td>
</tr>
<tr>
<td>RAC_OVERVIEW_CONFLICTS</td>
</tr>
<tr>
<td>RAC_OVERVIEW_WAITS</td>
</tr>
<tr>
<td>RAC_PING_INSTANCE</td>
</tr>
<tr>
<td>RAC_SESSIONS_TOP</td>
</tr>
<tr>
<td>RESPONSETIME_CURRENT_RESTIME</td>
</tr>
<tr>
<td>RESPONSETIME_HISTORY</td>
</tr>
<tr>
<td>RESPONSETIME_HISTSESSION</td>
</tr>
<tr>
<td>RESPONSETIME_LASTHOUR</td>
</tr>
<tr>
<td>RESPONSETIME_SESBOTTLENECK</td>
</tr>
<tr>
<td>RESPONSETIME_SESSIONTIME</td>
</tr>
<tr>
<td>RESPONSETIME_SYSTEMTIME</td>
</tr>
<tr>
<td>RETRIEVEHOMEPAGE</td>
</tr>
<tr>
<td>RETRIEVEMEMORYPAGE</td>
</tr>
<tr>
<td>RETRIEVOBJECTRATIOS</td>
</tr>
<tr>
<td>RETRIEVEUSERRATIOS</td>
</tr>
<tr>
<td>SESSION_DD_CRNTWAITS</td>
</tr>
<tr>
<td>SESSION_DD_HISTWAITS</td>
</tr>
<tr>
<td>SESSION_DD_OVERVIEW</td>
</tr>
<tr>
<td>SESSION_LAST_TEN_WAIT</td>
</tr>
<tr>
<td>SPACE_DD_DATAFILEDETAIL</td>
</tr>
<tr>
<td>SPACE_DD_SPACEBYOBJECT</td>
</tr>
<tr>
<td>SPACE_DD_TABLESPACEGROWTH</td>
</tr>
<tr>
<td>SPACE_DD_TOPUSERS</td>
</tr>
<tr>
<td>SQLANALYSIS_IO</td>
</tr>
<tr>
<td>SQLANALYSIS_SPACE</td>
</tr>
<tr>
<td>TOP_SYSTEM_BOTTLENECKS_10G</td>
</tr>
<tr>
<td>TOPDISKSORTPROCESS</td>
</tr>
<tr>
<td>TOPIOPROCESS</td>
</tr>
<tr>
<td>TOPOFILEWAIT</td>
</tr>
<tr>
<td>TOPPARSEPROCESS</td>
</tr>
<tr>
<td>TOPSESSIONS_10G</td>
</tr>
<tr>
<td>TOPSQL_RECENT_HISTORY</td>
</tr>
</tbody>
</table>
### SQL Server Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Procedure</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_embt_pa_all_database_enumerrorlog</td>
<td>sp_embt_pa_all_database_list</td>
<td>sp_embt_pa_current_sq</td>
</tr>
<tr>
<td>sp_embt_pa_all_database_analysis</td>
<td>sp_embt_pa_all_database_bottleneck</td>
<td>sp_embt_pa_database_drill_backups_db</td>
</tr>
<tr>
<td>sp_embt_pa_all_database_drill_backups_detail</td>
<td>sp_embt_pa_database_drill_backups_log</td>
<td>sp_embt_pa_database_drill_config_detail</td>
</tr>
<tr>
<td>sp_embt_pa_all_database_drill_config_server</td>
<td>sp_embt_pa_database_drill_error_log_contents</td>
<td>sp_embt_pa_database_drill_error_log_detail</td>
</tr>
<tr>
<td>sp_embt_pa_all_database_drill_error_log_volume</td>
<td>sp_embt_pa_database_drill_obj_index_det</td>
<td>sp_embt_pa_database_drill_obj_summary</td>
</tr>
<tr>
<td>sp_embt_pa_all_database_drill_object_table_detail</td>
<td>sp_embt_pa_database_drill_overview</td>
<td>sp_embt_pa_database_drill_replication_agent</td>
</tr>
<tr>
<td>sp_embt_pa_all_database_drill_replication_throughput</td>
<td>sp_embt_pa_database_drill_sqlagent_airdet</td>
<td>sp_embt_pa_database_drill_sqlagent_alerts</td>
</tr>
<tr>
<td>sp_embt_pa_all_database_sqlagent_jobdget</td>
<td>sp_embt_pa_database_drill_sqlagent_jobsu</td>
<td>sp_embt_pa_database_drill_sqlagent_statu</td>
</tr>
<tr>
<td>sp_embt_pa_all_database_errorlog</td>
<td>sp_embt_pa_db_requiring_backup</td>
<td>sp_embt_pa_db_workload_analysis</td>
</tr>
<tr>
<td>sp_embt_pa_home_active_processes</td>
<td>sp_embt_pa_home_activity_metrics</td>
<td>sp_embt_pa_home_bottleneck_waits</td>
</tr>
<tr>
<td>sp_embt_pa_home_bottlenecks</td>
<td>sp_embt_pa_home_storage_database_count</td>
<td>sp_embt_pa_home_storage_totals</td>
</tr>
<tr>
<td>sp_embt_pa_instance_startup</td>
<td>sp_embt_pa_io_access_ratios</td>
<td>sp_embt_pa_io_activity_metrics</td>
</tr>
<tr>
<td>sp_embt_pa_io_bottlenecks</td>
<td>sp_embt_pa_io_drill_database</td>
<td>sp_embt_pa_io_drill_file</td>
</tr>
<tr>
<td>sp_embt_pa_io_drill_system_ave</td>
<td>sp_embt_pa_io_drill_system_io</td>
<td>sp_embt_pa_io_drill_system_physical</td>
</tr>
<tr>
<td>sp_embt_pa_io_drill_system_space</td>
<td>sp_embt_pa_io_drill_user_lock</td>
<td>sp_embt_pa_io_drill_user_obj</td>
</tr>
<tr>
<td>sp_embt_pa_io_drill_user_sql</td>
<td>sp_embt_pa_io_drill_user_temp</td>
<td>sp_embt_pa_io_get_file_stats</td>
</tr>
<tr>
<td>sp_embt_pa_io_hottest_files</td>
<td>sp_embt_pa_io_ratios</td>
<td>sp_embt_pa_io_top_processes</td>
</tr>
<tr>
<td>sp_embt_pa_memory_analysis</td>
<td>sp_embt_pa_memory_drill_bcache</td>
<td>sp_embt_pa_memory_drill_buffer</td>
</tr>
<tr>
<td>sp_embt_pa_memory_drill_buffer_page_activity</td>
<td>sp_embt_pa_memory_drill_log_details</td>
<td>sp_embt_pa_memory_drill_log_instances</td>
</tr>
<tr>
<td>sp_embt_pa_memory_drill_proc_detail</td>
<td>sp_embt_pa_memory_drill_proc_summary</td>
<td>sp_embt_pa_memory_latch_stats</td>
</tr>
<tr>
<td>sp_embt_pa_memory_ratio_mem</td>
<td>sp_embt_pa_memory_sql_analysis</td>
<td>sp_embt_pa_memory_top_latch_waits</td>
</tr>
<tr>
<td>sp_embt_pa_memory_workload_analysis</td>
<td>sp_embt_pa_space_bottlenecks</td>
<td>sp_embt_pa_space_drill_filegroups</td>
</tr>
</tbody>
</table>
### SYBASE

#### Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_embt_pa_actsql_drill</td>
<td>sp_embt_pa_actsql_drill_det</td>
<td>sp_embt_pa_current_sql</td>
</tr>
<tr>
<td>sp_embt_pa_db_backup</td>
<td>sp_embt_pa_db_bnecks</td>
<td>sp_embt_pa_db_drill_cb_filter</td>
</tr>
<tr>
<td>sp_embt_pa_db_drill_cb_idx</td>
<td>sp_embt_pa_db_drill_cb_tbl</td>
<td>sp_embt_pa_db_drill_cb_filter</td>
</tr>
<tr>
<td>sp_embt_pa_db_drill_dbs</td>
<td>sp_embt_pa_db_drill_engines</td>
<td>sp_embt_pa_db_drill_obj_filter</td>
</tr>
<tr>
<td>sp_embt_pa_db_drill_obj_idx</td>
<td>sp_embt_pa_db_drill_obj_suspect</td>
<td>sp_embt_pa_db_drill_obj_tbl</td>
</tr>
<tr>
<td>sp_embt_pa_db_drill_opt</td>
<td>sp_embt_pa_db_drill_overview</td>
<td>sp_embt_pa_db_drill_sus_filter</td>
</tr>
<tr>
<td>sp_embt_pa_db_engines</td>
<td>sp_embt_pa_db_max_engines=</td>
<td>sp_embt_pa_db_metadata</td>
</tr>
<tr>
<td>sp_embt_pa_db_obj_space</td>
<td>sp_embt_pa_db_srv_bnecks</td>
<td>sp_embt_pa_db_workload</td>
</tr>
<tr>
<td>sp_embt_pa_dbspace</td>
<td>sp_embt_pa_home_act_metrics</td>
<td>sp_embt_pa_home_bottlenecks</td>
</tr>
<tr>
<td>sp_embt_pa_home_storage_tot</td>
<td>sp_embt_pa_instance_startup</td>
<td>sp_embt_pa_io_analysics</td>
</tr>
<tr>
<td>sp_embt_pa_io_bottlenecks</td>
<td>sp_embt_pa_io_drill_dbs</td>
<td>sp_embt_pa_io_drill_device_rt</td>
</tr>
<tr>
<td>sp_embt_pa_io_drill_device_sum</td>
<td>sp_embt_pa_io_drill_device_typ</td>
<td>sp_embt_pa_io_drill_engine</td>
</tr>
<tr>
<td>sp_embt_pa_io_drill_net_eng</td>
<td>sp_embt_pa_io_drill_net_ses</td>
<td>sp_embt_pa_io_drill_object</td>
</tr>
<tr>
<td>sp_embt_pa_io_hottest_devices</td>
<td>sp_embt_pa_io_key_system</td>
<td>sp_embt_pa_io_network</td>
</tr>
<tr>
<td>sp_embt_pa_io_trans_log</td>
<td>sp_embt_pa_io_workload</td>
<td>sp_embt_pa_io_activity</td>
</tr>
<tr>
<td>sp_embt_pa_mem_analysis</td>
<td>sp_embt_pa_mem_bottleneck</td>
<td>sp_embt_pa_mem_eff_ratios</td>
</tr>
<tr>
<td>sp_embt_pa_mem_mdcache</td>
<td>sp_embt_pa_mem_workload</td>
<td>sp_embt_pa_mem_drill_dc_act</td>
</tr>
<tr>
<td>sp_embt_pa_memory_drill_dc_cfg</td>
<td>sp_embt_pa_memory_drill_dc_eff</td>
<td>sp_embt_pa_memory_drill_dc_obj</td>
</tr>
<tr>
<td>sp_embt_pa_memory_drill_pc_act</td>
<td>sp_embt_pa_memory_drill_pc_det</td>
<td>sp_embt_pa_memory_drill_pc_use</td>
</tr>
<tr>
<td>sp_embt_pa_monitor</td>
<td>sp_embt_pa_space_bottlenecks</td>
<td>sp_embt_pa_space_db_analysis</td>
</tr>
</tbody>
</table>
Operating System Prerequisites
Performance Analyst requires prerequisites for performance monitoring Windows 2000/XP and UNIX and Linux. The connectivity requirements for the various servers are minimum requirements. The other requirements are optional.

Windows 2000/XP Prerequisites

Connectivity
Performance Analyst requires the following minimum conductivity prerequisites:

- Remote Registry Service should be running on the server if Performance Analyst is connecting to a remote machine. For local machine this does not matter.
- Adequate permissions to connect to the remote computer's registry.

Other Requirements
Performance Analyst requires the following optional prerequisites:

- Diskperf should be running on the server.

  NOTE: You can start diskperf by executing diskperf -y from the command line.

UNIX and Linux Prerequisites

Connectivity
Performance Analyst requires the following minimum conductivity prerequisites:

- Ability to connect to the server and execute commands on the server using one of the following protocols:
  - Telnet
  - SSH
  - Rlogin
- Make sure that the appropriate daemon is running on the server and that the userid used for connecting has the permission to connect using the appropriate protocol.
Commands
The table below describes the commands that the connection Userid needs to have access to, to collect performance stats. The default path for the Userid on UNIX should be setup so you can execute the commands without qualifying them with a path.

<table>
<thead>
<tr>
<th>Command Type</th>
<th>Description</th>
<th>Commands/Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Commands</td>
<td>Required on all UNIX and Linux servers.</td>
<td><code>ps, wc, cat, grep, vmstat, uname, uptime, who, df, netstat</code></td>
</tr>
<tr>
<td>Linux Commands</td>
<td>User should be able to access data from these files</td>
<td><code>/proc/meminfo</code> <code>PROC/cpuinfo</code> <code>PROC/partitions</code> or <code>iostat</code></td>
</tr>
<tr>
<td>AIX Commands</td>
<td>Required for AIX</td>
<td><code>lsdev</code> <code>pmcycles</code> <code>usr/sbin/prtconf</code> <code>iostat</code> <code>lsps</code> <code>pagesize</code></td>
</tr>
<tr>
<td>Solaris Commands</td>
<td>Required for Solaris</td>
<td><code>/usr/sbin/psrinfo</code> <code>iostat</code> <code>/usr/sbin/prtconf</code> <code>/usr/sbin/swap</code> <code>pagesize</code></td>
</tr>
<tr>
<td>HP Commands</td>
<td>Required for HP</td>
<td><code>/etc/swapinfo</code> <code>sar</code> permission to access the following: <code>/dev/mem</code> <code>/dev/kmem</code> <code>/var/adm/syslog/syslog.log</code></td>
</tr>
</tbody>
</table>

Opening Performance Analyst for the First Time
For Performance Analyst to work, a platform-specific package needs to be installed by the SYS account. When Performance Analyst first starts, it checks for the presence of this package and if it has not been installed, Performance Analyst prompts you to install it. Once created, any account with access to this package can monitor databases using Performance Analyst. You do not need to install any other objects on a monitored database.

**NOTE:** You must have SYS privileges to install the required packages. After the packages are installed, anyone can use them.

To open the Performance Analyst for the first time, connect to a database through DBArtisan and then select Performance Analyst from the Analyst drop-down menu.

Subsequently, you can connect to Performance Analyst by going to **Start>Programs>EmbarcaderoDBArtisan>Performance Analyst**.
When you open the Performance Analyst this way, you can only review one datasource at a time, as is the case when you open Performance Analyst remotely.

**NOTE:** If you are connecting through your client machine, it will save you time and give you greater flexibility to connect to Performance Analyst through the DBArtisan console. If you connect through the console, you can open multiple Performance Analyst windows for as many datasources as you like and switch between them by clicking the target tab.

You can also open Performance Analyst in stand-alone mode from your Web client. To do this, see: Opening Performance Analyst in Stand-Alone Mode from Your Web Client

### Opening Performance Analyst in Stand-Alone Mode from Your Web Client

The Embarcadero Performance Analyst Login dialog box lets you open Performance Analyst in stand-alone mode from a Web client. This way you can check the performance of any or all datasources that are registered through DBArtisan when you are working remotely.

To connect to the Performance Analyst when you are not using the DBArtisan console, simply go to `<installation directory>\PAnalyst\PerfAnalyst.html`

When you've made the connection, a login dialog box opens.

If you are connecting to the Performance Analyst from your Web client, you can only see one database at a time.

**NOTE:** Keep in mind that when you connect to Performance Analyst remotely, the Auto Connect option you may have selected when you registered the target datasource in DBArtisan is not functional. You must supply a valid user ID and password to log in to Performance Analyst successfully.

**Login As:** is only an option when you are logging into an Oracle database. Sybase, SQL Server and DB2 log in as default. The properties for the login roles are as follows:

- **Default** - Establishes a connection with no system roles.
- **SYSDBA** - Lets you perform *all* DBA-related functions. SYSDBA is required for the SYS id when you log into a 9i or later instance.
- **SYSOPER** - Lets you perform many DBA-related functions, like startup and shutdown.

### Licensing Performance Analyst

The Performance Analyst is licensed per named server. You are issued one license key for the number of databases/servers you want to license. For example, if you purchase ten licenses for Performance Analyst, you only need to obtain one license key from Embarcadero Technologies.

For more information, see Analyst License Manager.

### Performance Analyst License Manager

The Analyst License Manager lets you manage licenses.

The table below describes the options and functionality on the Analyst License Manager:
Completing the Performance Analyst License Manager

To complete the Analyst License Manager, do the following:

1. Open DBArtisan.
2. On the Help menu, click Licensing.
   
   DBArtisan opens the License Configuration Dialog box.
3. Click License.
   
   DBArtisan opens the Analyst License Manager.
4. Select options.
5. To make your license choices permanent, click Apply.
6. To close the Analyst License Manager, click Close.

For more information, see Analyst License Manager.

Performance Analyst Add an Analyst License Dialog Box

The Add an Analyst License Details dialog box lets you add a license.

The table below describes the options and functionality on the Add a Analyst License Details dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Summary</td>
<td>Displays basic licensing metadata for the product.</td>
</tr>
<tr>
<td>Available Host(s)</td>
<td>Lets you select a host from the list of hosts available to license.</td>
</tr>
<tr>
<td>License</td>
<td>Click to add a license to a host.</td>
</tr>
<tr>
<td>Licensed Host(s)</td>
<td>Displays hosts already licensed.</td>
</tr>
<tr>
<td>Remove</td>
<td>Click to remove a license from a host.</td>
</tr>
<tr>
<td>Apply</td>
<td>Click to make your license choices permanent.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Click to close the Licensing Dialog box.</td>
</tr>
</tbody>
</table>

**NOTE:** If unsaved license changes have been made, the Analyst asks if you want to discard their changes before exiting.

For more information, see Completing the Analyst License Manager.
Performance Analyst Application Basics
Performance Analyst displays a complete picture of a database's performance. Performance Analyst is a quick, lightweight tool that lets you investigate performance bottlenecks.

For more information, see Product Design.

Performance Analyst Product Design
The Performance Analyst product design includes Home Pages.

Home Pages for Oracle
Performance Analyst displays home pages of various performance categories with detailed performance data. The Performance Analyst product design includes the following home pages:

- Home Page Statistics
- I/O Page Statistics
- Memory Page Statistics
- Objects Page Statistics
- OS Page Statistics
- RAC Details (Oracle 10g only)
- Space Page Statistics
- Users Page Statistics

Home Pages for Microsoft SQL Server
Performance Analyst displays home pages of various performance categories with detailed performance data. The Performance Analyst product design includes the following home pages:

- Home Page Statistics
- Database Page Statistics
- Memory Page Statistics
- I/O Page Statistics
- Space Page Statistics
- Users Page Statistics
- OS Page Statistics

Home Pages for IBM DB2 for Linux, Unix, and Windows
Performance Analyst displays home pages of various performance categories with detailed performance data. The Performance Analyst product design includes the following home pages:

- Home Page Statistics
- Configuration Page Statistics
- I/O Page Statistics
Home Pages for Sybase ASE
Performance Analyst displays home pages of various performance categories with detailed performance data. The Performance Analyst product design includes the following home pages:

- Home Page Statistics
- Database Page Statistics
- I/O Page Statistics
- Memory Page Statistics
- OS Detail Statistics
- Space Page Statistics
- Users Page Statistics

Performance Analyst Product Navigation
Navigating Performance Analyst is easily accomplished through two main mechanisms:

1. The main Performance Analyst toolbar buttons open the major performance analysis views (the toolbar buttons you see are self-explanatory) within the product as well as frequently used actions like Refresh and Performance Threshold Customizations.

2. If you right-click any area within Performance Analyst, the application displays a shortcut menu that lets you easily move between any of the main Performance Analyst home pages and drill-down views. The shortcut menu also contains context sensitive options like drilling down further into a particular statistic, copying the display to a text file, and more.

For more information, see:

- Toolbar Icons

Toolbar Icons
Performance Analyst has several unique icons that take you directly to some functions and statistical groups that can help you see and sort through groupings of data. The functional icons you see on the Performance Analyst toolbar are as follows:

- **Top Sessions** - This takes you directly to the Top Sessions tab of the Users Detail view where you see the Top I/O, Memory, CPU, and Sessions statistics.

- **Top SQL** - Opens the Top SQL detail view and lets you move among the Summary, Criteria, SQL Details, and Recent History tabs.
• **Response Time (Oracle 10 g only)** - Oracle 10g collects response time statistics, which are displayed when you click the icon.

• **Alarms** - Goes directly to the Alarms log.

• **Threshold Maintenance** - Clicking here opens the Threshold Template Maintenance dialog box. Thresholds are explained in the Using chapter.

• **Refresh** - Click this to update the stats/information on whatever page you are currently scrutinizing.

• **Options** - To open the Performance Analyst options dialog box, where you can change the values for Oracle, SQL Server, Sybase, or DB2 options, click this cheerful checkerboard.

• **Exit** - This closes Performance Analyst for the database you are reviewing. If you have multiple Performance Analysts open, clicking this does not close all Analysts.

• **Maximize Window** - This icon appears on each pane of a multi-pane window and lets you expand the pane of choice to fill the screen.

• **Minimize Window** - When you have seen enough of an expanded pane, you can bring it back to its original size by clicking this icon.

• **Help** - Another in this series of intuitive icons, click this to open the context-sensitive Help.

### Alarm Sound

When you register an alarm in Performance Analyst, it creates a periodic alarm sound. To turn off the alarm sound, do the following:

1. Click **Start**, **Programs**, **Accessories**, **Entertainment**, and then **Volume Control**. Windows opens the Volume Control dialog box.

2. Under **Wave**, select **Mute**.

   **NOTE:** Hiding the Performance Analyst icon in the task bar does not mute the sound.
Using Performance Analyst

Performance Analyst allows you to organize performance metrics in the particular way you want to see them. You can drill-down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. You can set alarm triggers and thresholds for multiple databases across several platforms.

For example, key ratios in Performance Analyst are calculated using *delta* statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. This chapter tells you how to set the refresh intervals that best suit your needs.

So, this Using chapter explains *how* to reveal the layers of information you want to see. The platform-specific Expert Guides explain *what* you are seeing as they discuss individual statistics, the associated metrics, and troubleshooting suggestions:

- Performance Analyst for Oracle Expert Guide
- Performance Analyst for Microsoft SQL Expert Guide
- Performance Analyst for IBM DB2 Expert Guide for Linux, Unix, and Windows
- Performance Analyst for Sybase ASE Expert Guide

Performance Analyst Tools

Performance Analyst offers tools that help you configure what you can see when you survey target databases. It's an easy thing to set up an alarm notification for individual statistic thresholds based on parameters you designate. And, when you work through DBArtisan, you can easily correct problems if you notice something is amiss.

Performance Analyst supports the following functionalities:

<table>
<thead>
<tr>
<th>Alarm Log</th>
<th>Change Graph Type</th>
<th>Copy</th>
<th>Details</th>
<th>Explain SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage with DBArtisan</td>
<td>Options</td>
<td>Response Time (ORACLE 10g ONLY)</td>
<td>Save</td>
<td>Thresholds</td>
</tr>
<tr>
<td>Top Sessions</td>
<td>Top SQL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alarm Log

The Alarm Log displays every performance violation found in your currently monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst gives you the option to see all alarms for each database you are monitoring in one central log so you can see every problem in a single place. And, of course, you have the ability to determine alarm levels by setting the Threshold Properties.

To open an Alarm Log:

- Open the Performance Analyst and then scroll down to Alarms on the drop-down menu.
- Move your mouse over the Performance Analyst toolbar and click the exclamation point (you'll see Alarms appear in a tool tip under the hand icon).
Organizing the Alarm Log

You can arrange the data in the Alarm Log in several ways:

**NOTE:** Changes you make to one Alarm Log are not enacted globally. This allows you to customize each log based on your needs database by database if you want.

- **Move columns left or right** to place them in an order that makes the most sense to you. Do this by dragging-and-dropping a column heading to re-place it in the target location.

- **Sort column rows up and down or sideways.** Right-click a column heading and select:
  - **Sort Ascending:** For example, you can see the alarms as they move from Critical to Cleared.
  - **Sort Descending:** For example, you can see the alarms as they move from Cleared to Critical.
  - **Sort Multiple Columns:** Lets you group multiple columns in ascending or descending order. You cannot mix and match, however. That is, you cannot sort some columns to present data in ascending order and others to present data in descending order.
  - **Sort Clear:** Lets you clear the relationships you created if you sorted multiple columns.

The table below describes the columns of the Alarm Log:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm Type</td>
<td>Displays alarm types: Cleared, Critical, Fair and Poor.</td>
</tr>
<tr>
<td>Datasource</td>
<td>Datasource where the alarm was identified.</td>
</tr>
<tr>
<td>DBMS Type</td>
<td>Datasource DBMS type.</td>
</tr>
<tr>
<td>Date/Time</td>
<td>Date/time of when the alarm was observed.</td>
</tr>
<tr>
<td>Category</td>
<td>Displays metric category: memory, I/O, space, objects, or users.</td>
</tr>
<tr>
<td>Statistic</td>
<td>Metric name.</td>
</tr>
<tr>
<td>Value</td>
<td>Value of the statistic that caused the violation.</td>
</tr>
<tr>
<td>Explanation</td>
<td>Displays a text explanation of the problem. These text explanations can be very generic in nature and just plug in certain threshold metrics, etc.</td>
</tr>
</tbody>
</table>

Alarm Log Activities

From within the Alarms Window, right-clicking any row gives you more opportunities to get the most out of what you see. The Menu selections are:

- **Applicable Page** - Takes you to the row-specific Category page where you can see the problem metric and what’s happening with the surrounding statistics. For example, if you click a critical alarm in the Databases category, you connect immediately to the Databases home page.

- **Options** - Takes you to the Alarm Log Options dialog box where you adjust some general alarm log properties and alarm filters.

- **Clear List** - When you select this, the alarm log is cleared of all entries.

- **Help** - Opens the Help discussion for the target element.

- **Refresh** - Refreshes the display to show you any updates.

- **Copy** - Places the data in the alarm log at that point in time on the Windows clipboard. From there you can paste it into a Word or Excel document.
• **Save As** - Lets you name and save the Alarm Log to the location of your choice as a Microsoft Excel worksheet. You can then manipulate the worksheet as you would any other.

• **Print** - Enables you to print the alarm log.

**Alarm Log Options Dialog Box**
The Alarm Log Options dialog box is command central for how and where you display and manipulate previously configured alarms. Alarms themselves are configured in the [Threshold Properties](#), but this is where you arrange the particulars of the Alarm Log your databases. Keep in mind that changes you make to this dialog box are reflected globally across all platforms.

The table below describes the columns of the Alarm Log Options dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Properties</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Do not log alarms to the alarm log</strong></td>
<td>When you select this option, alarms will not be saved to an alarm log. Depending on the filters you select, you will still see alarms in the alarm window.</td>
</tr>
<tr>
<td><strong>Log alarms to the alarm log</strong></td>
<td>This option lets you save alarms to an alarm log. You can type or browse to choose a directory where you want to save the log and you can also set the maximum alarm log size in KB.</td>
</tr>
<tr>
<td><strong>Filters</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Show alarms for only current datasource</strong></td>
<td>When you select this option, you see only alarms for the current datasource in the alarm window. So, if you have multiple datasources open and you go from one to another, the alarm window will display only the alarms for the database you are observing at that very moment.</td>
</tr>
<tr>
<td><strong>Show alarms for all currently modified datasources</strong></td>
<td>This option lets you see what alarms are occurring across all databases you are connected to at the moment.</td>
</tr>
<tr>
<td><strong>Show only active alarms</strong></td>
<td>If you select this option, you’ll see only those alarms that are active at that moment in time. These are alarms that you decided to include as part of a threshold template.</td>
</tr>
<tr>
<td><strong>Show only alarms where the Explanation contains the following text</strong></td>
<td>Lets you fine-tune the alarms you see depending on what you are most interested in seeing. For example, if you are particularly interested in seeing Bottleneck alarms, type Bottlenecks. <strong>NOTE:</strong> The text search for this function is case-sensitive.</td>
</tr>
<tr>
<td><strong>Show only alarms that are of the following severity</strong></td>
<td>Select to display alarms that are critical, poor, fair, or cleared.</td>
</tr>
</tbody>
</table>

**Changing Graph Type**
When you are looking at a statistic that is displayed in graph form, you can very easily change how that data is displayed. When you choose to switch from one type to another, the changes are limited to the specific graph and its data that you are manipulating. So, for example, the same page can display a bar graph and a pie chart.

The graph types available to you are:

• Line Graph
• Area Graph
• Pie Graph
• Bar Graph
• Horizontal Bar Graph
To change the graph type:

- Right-click the target graph, and from the drop-down menu, select Change Graph Type.

Copy
The Copy functionality lets you copy data in a tabular grid display into the Windows clipboard. The data can then be saved and used for later comparisons as you continue to tweak your databases' performance.

To Copy data, do the following:
1. Hover the pointer over the target section or metric, and then right-click and select Copy. Performance Analyst copies data into the Windows clipboard.
2. Select an existing, or new, Word or Excel document where you want to Paste the data.
3. Save, or Save As, the document to the target location.

Details
The Details functionality lets you drill-down into the target metric data. Generally you are taken to the metric's category detail page.

To see section or metric details:
- Right-click in the target section or metric and select Details from the drop-down menu.
  After you are taken to your destination, right-click again to see if another level of detail is available to you.

Explain SQL
Performance Analyst lets you run Explain SQL Plans from the Top SQL window. If you are seeing problem metrics, you can take steps to correct the SQL statements.

Opening the Explain SQL/Explain Plan:
1. Click the SQL button on the Performance Analyst toolbar to open the Top SQL details page.
2. Click the SQL Details tab.
  You can also right-click anywhere and select Navigate to...>Top SQL>SQL Details from the drop-down menus.
3. Move down the list of SQL text in the grid and the SQL text appears in the Selected SQL Text window.
4. Press the Explain SQL button to open the Explain Plan.

Manage with DBArtisan
Manage with DBArtisan opens the DBArtisan editor for the object that has been selected for editing in a grid/tabular display of data.

By moving directly to DBArtisan from Performance Analyst, you have an opportunity to fix database problems as soon as you see them. This option is not available, however, when you open the Performance Analyst from a Web client.
Performance Analyst Options Dialog Box

The Performance Analyst Options dialog box is where you can specify various refresh intervals, threshold polling, and datasource overrides among other settings. You can set or change defaults that are both global and platform-specific.

**NOTE:** The settings on the Global and Datasource tabs apply to all datasource platforms. The SQL Server, Sybase, and DB2 tab settings are platform-specific. Some options on the platform-specific tabs may look as if they are available to you, but in reality they are not. For example, if you are connected to an Oracle datasource, changing the Monitoring Refresh interval on the Sybase tab is a hollow gesture.

To open the Performance Analyst Options dialog box:

- Open the Performance Analyst and then scroll down to **Options** on the drop-down menu.
- Right-click in the Performance Analyst window for the target datasource and select **Options**.
- Move your mouse over the Performance Analyst toolbar and click the multicolored square (you’ll see Options appear in a tool tip under the hand icon).

The table below describes the tabs of the Performance Analyst Options dialog box:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>Engine</td>
<td>SQL metric refresh interval</td>
<td>Lets you set the time, in seconds, between collect cycles for the SQL metrics reflected in Performance Analyst. SQL metrics include all non-operating system metrics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OS metric refresh interval</td>
<td>Lets you set the time, in seconds, between collect cycles for all operating system-related metrics reflected in Performance Analyst. OS metrics include all metrics shown on the OS page.</td>
</tr>
<tr>
<td>Threshold polling</td>
<td></td>
<td></td>
<td>Lets you determine which metrics, if any, you want to test against your pre-set thresholds. The possible settings for the Threshold polling box are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All Metrics</td>
<td>- Polls all metrics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Current Page Only</td>
<td>- Polls only those metrics reflected on the currently open page in Performance Analyst</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off</td>
<td>- Turns polling off. If polling is turned off, Performance Analyst does not activate Alarms.</td>
</tr>
<tr>
<td>Threshold polling</td>
<td></td>
<td>interval</td>
<td>Lets you set the time, in seconds, between collect cycles for threshold-related metrics. Once collected, the polled metrics are compared to the pre-set thresholds. If you select Off in the Threshold polling box, Performance Analyst does not poll these metrics, regardless of the interval.</td>
</tr>
<tr>
<td>Run queries for all tabs</td>
<td></td>
<td></td>
<td>Lets you determine whether or not you want Performance Analyst to refresh all tabs according to the SQL metric refresh interval. If Run queries for all tabs is not checked, Performance Analyst runs queries only for the current tab.</td>
</tr>
<tr>
<td>Visual</td>
<td></td>
<td>Show alarms icon in system tray</td>
<td>Lets you determine whether or not you want Performance Analyst alarms to appear in the Windows System Tray on the lower right-hand side of your screen. If you select Show alarms icon in system tray, Performance Analyst places an icon in the Windows System Tray each time an alarm is raised.</td>
</tr>
<tr>
<td></td>
<td>Number of Historic Values in Graphs</td>
<td></td>
<td>Lets you specify the number of values in the graphs.</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Tab</th>
<th>Field</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasource</td>
<td>Operating System</td>
<td>OS type</td>
<td>Lets you select the operating system type for the current datasource. The options are: MS Windows - Select MS Windows for all supported types of Microsoft Windows Unix or Linux - Select Unix or Linux if your operating system is any flavor of Unix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Host name</td>
<td>Lets you specify the name of the host for the current datasource, if necessary. Performance Analyst automatically fills in any detectable information, including the Host name box.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Username</td>
<td>Lets you select a local or network username that Performance Analyst can use to connect to the datasource. If you select the Use Datasource Username and Password check box, the Username box is unavailable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Password</td>
<td>Lets you select a local or network password that Performance Analyst can use to connect to the datasource. If you select the Use Datasource Username and Password check box, the Password box is unavailable.</td>
</tr>
<tr>
<td></td>
<td>Protocol</td>
<td>Protocol</td>
<td>Lets you specify a protocol for communicating with datasources on Unix-based operating systems. The options are: SSH Telnet Rlogin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use Datasource Username and Password</td>
<td>Lets you specify whether or not you want to use the datasource username and password to connect to the datasource as opposed to a network or operating system username and password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Save Username and Password</td>
<td>Lets you save the username and password that appear in the Username and Password boxes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Overrides</td>
<td>Lets you override the SQL metric refresh interval in the Global Tab for the current datasource. The SQL metric refresh interval specifies the time, in seconds, between collect cycles for the SQL metrics reflected in Performance Analyst. SQL metrics include all non-operating system metrics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OS metric refresh interval</td>
<td>Lets you override the OS metric refresh interval in the Global Tab for the current datasource. The OS metric refresh interval specifies the time, in seconds, between collect cycles for all operating system-related metrics reflected in Performance Analyst. OS metrics include all metrics shown on the OS page.</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Global Options</td>
<td>SQL Profiling Enabled</td>
<td>Performance Analyst requires some privileges that go along with SQL Server Profiler in order to return some statistics for SQL Server databases. Check SQL Profiling Enabled to ensure that Performance Analyst can collect all the necessary metrics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start SQL Profiling</td>
<td>Lets you set new datasources to start SQL Profiling when they are monitored by Performance Analyst.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SQL Profiling refresh interval</td>
<td>Lets you set the default time for new datasources, in seconds, between collect cycles for the SQL metrics that require the SQL Server Profiler.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Current Datasource Settings</td>
<td>Lets you set current datasources to start SQL Profiling when they are monitored by Performance Analyst.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SQL Profiling refresh interval</td>
<td>Lets you set the default time for current datasources, in seconds, between collect cycles for the SQL metrics that require the SQL Server Profiler.</td>
</tr>
</tbody>
</table>
The Response Time functionality lets you analyze response times. You can review historical data and pinpoint who was on a database and what they were doing during a specific period of time.

**To see Response Time statistics:**
- Right-click in the target and select Navigate to...>Response Time Analysis from the drop-down menu.
- Hover your mouse over the Performance Analyst toolbar and click the magnifying glass (you’ll see Response Time appear in a tool tip under the hand icon).

**Save As...**
The Save As option gives you an opportunity to save data in a tabular format in the location of your choice. What this means is that the current data will be saved to an existing or a new Microsoft Excel worksheet. You can then manipulate and compare data with all the functionality Microsoft Excel gives you.

**To Save As**
- Right-click anywhere on the metric or in the Alarm Log, for example, and choose Save As... from the drop-down menu.

---

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybase</td>
<td>Global Options</td>
<td>Monitoring Table refresh interval</td>
<td>Lets you specify, in seconds, the amount of time between refreshes for the Monitoring Table. The Monitoring Table is table installed by Performance Analyst used to store various metric-related information. The default is 15 seconds.</td>
</tr>
<tr>
<td>Datasource Overrides</td>
<td>Monitoring Table refresh interval</td>
<td></td>
<td>When you select this option and enter a time, in seconds, this will override-existing refresh times.</td>
</tr>
<tr>
<td>Sample Query</td>
<td>Use custom sample query</td>
<td></td>
<td>You can use this option to build a custom sampling query and check the database response time by reviewing the Sample Query Response Time graph on the Database home page.</td>
</tr>
<tr>
<td>DB2</td>
<td>Monitor Settings</td>
<td>Monitor Switches</td>
<td>You are able to set monitor switches ON for the current session. When check marks are visible, that means they were previously set as defaults. Check or uncheck the boxes as you wish. Monitor settings are available for Buffer Pool, Statement, Lock, Table, Sort, Unit of Work. Metrics for all checked elements are gathered in the current session.</td>
</tr>
<tr>
<td></td>
<td>SQL Statement Cache</td>
<td></td>
<td>You can collect SQL statements on each refresh, collect SQL statement only when you are on the Top SQL pages, or choose not to collect any SQL statements.</td>
</tr>
<tr>
<td></td>
<td>Database System Monitor Heap</td>
<td></td>
<td>The database system monitor heap size shows the current value, in pages, allocated for the database system monitor data. The amount of memory required for monitoring depends on the number of monitoring applications, switch settings, and the level of database activity. If you see frequent heap overflow errors in the output message window when monitoring DB2 through Performance Analyst, adjust the setting here.</td>
</tr>
<tr>
<td></td>
<td>Save settings as default</td>
<td></td>
<td>Gives you the option to set the monitoring settings as a default for the current DB2 database or for the current session only.</td>
</tr>
</tbody>
</table>

---

**Response Time for Oracle 10g**
The Response Time functionality lets you analyze response times. You can review historical data and pinpoint who was on a database and what they were doing during a specific period of time.
Thresholds

Thresholds are the point that must be exceeded to begin producing a user-defined effect or result, or to elicit a response, such as an alarm. Performance Analyst provides default threshold templates for each database platform. The default template is applied unless you choose to apply a customized template. There are two methods to place thresholds on statistics in Performance Analyst:

- **Threshold Templates** - Threshold templates are logical containers used to define and deploy thresholds to datasources. You can apply many templates at a single time and you can apply a single threshold template to many datasources. To customize a template, use the Threshold Template Maintenance dialog box. As you add or edit thresholds for a customized template, you make your changes in the Threshold Template Properties dialog box. And finally, when you have customized a threshold template to your satisfaction, you Select Datasource(s) where you want the template to be applied.

- **Datasource Thresholds** - Datasource Thresholds are default or user-defined values that mark performance boundaries. You can apply Datasource Thresholds to specific statistics on a datasource. To make changes to a datasource threshold, use the Datasource Thresholds Maintenance dialog box. From here, the Threshold Templates dialog box gives you the opportunity to use a previously created template to override the target datasource’s user thresholds.

Whenever you make changes to a threshold, whether through the Threshold Template functions or from the Datasource Thresholds, you are presented with the Threshold Properties dialog box where you can specify when, and what type of alarm you want to trigger.

Additionally, you can Copy a template.

Threshold Properties Dialog Box

Thresholds are the point that must be met to begin producing a user-defined effect or result, or to elicit a response, such as an alarm. The Threshold Properties dialog box lets you modify properties for any threshold.

**To open the Threshold Properties dialog box:**

- Choose **Threshold Maintenance**... from the Performance Analyst drop-down menu to open a Threshold Template Maintenance dialog box, and then click **Add** or **Edit**.

- You can also choose **Datasource Thresholds**... from the Performance Analyst drop-down menu to open the Datasource Thresholds Maintenance dialog box, and then click **Add** or **Edit**.

The table below describes the options and functionality on the Threshold Properties dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistic</td>
<td>Lets you choose the target statistic where you want to add a threshold. <strong>NOTE:</strong> If you are adding a statistic to a template or datasource threshold, use the drop-down list to find the statistic you want to add. When you select <strong>Edit</strong> for any statistic showing in the User Thresholds window, the target statistic is displayed.</td>
</tr>
<tr>
<td>Generate Alarm If</td>
<td>Lets you choose or change the target alarm condition. The seven options are: greater than, greater than or equal to, is equal to, is not equal to, less than, and less than or equal to.</td>
</tr>
<tr>
<td>Value amount</td>
<td>Lets you set or change the numerical value for the threshold. You cannot leave the value text box blank when you first add the threshold.</td>
</tr>
<tr>
<td>Severity</td>
<td>Lets you select the target severity level: Fair, Poor, or Critical.</td>
</tr>
</tbody>
</table>
Threshold Templates Dialog Box

Threshold Templates are logical containers used to define and deploy thresholds to datasources. Performance Analyst offers a default template for each platform. You can copy or customize and apply templates to your heart's content. You cannot, however, delete the default template.

To open the Threshold Templates dialog box:

- Choose Datasource Thresholds... from the Performance Analyst drop-down menu to open the Datasource Thresholds Maintenance dialog box, and then click Change.

The Threshold Templates dialog box displays the available Threshold Templates. The table below describes the options and functionality of the Threshold Templates dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Displays the name(s) of the available platform-specific Threshold Template(s).</td>
</tr>
<tr>
<td>Description</td>
<td>Displays the description(s), if any of available Threshold Template(s).</td>
</tr>
<tr>
<td>Override Datasource's User Thresholds</td>
<td>Lets you choose which thresholds to use on the target datasource - the thresholds in the Threshold Template or the user-defined thresholds on the target datasource. If you select the check box, Performance Analyst will override any Datasource Thresholds on the target datasource with the thresholds in the Thresholds Template.</td>
</tr>
</tbody>
</table>

Threshold Template Properties

Threshold Templates are logical containers used to define and deploy thresholds to datasources. The Threshold Template Properties dialog box is where you create a custom template or where you edit an existing one.

To open the Threshold Template Properties dialog box:

- Choose Threshold Maintenance... from the Performance Analyst drop-down menu to open a Threshold Template Maintenance dialog box, and then click Add or Edit.

The table below describes the options and functionality on the Threshold Template Properties dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Lets you name the threshold template you are creating. If you are editing a template, target template’s name appears here.</td>
</tr>
<tr>
<td>Description</td>
<td>Lets you type or review the target template’s description.</td>
</tr>
<tr>
<td>Platform</td>
<td>Offers a drop-down list of supported platforms. The selection box displays the target template’s platform, but you can use this list to see thresholds from other platforms.</td>
</tr>
<tr>
<td>Threshold/Severity/Condition/Template</td>
<td>Displays each threshold contained in each template for a given platform. Here you see the severity and condition of each template as well.</td>
</tr>
<tr>
<td>Arrow</td>
<td>Lets you select thresholds for the Threshold Template. When you click the arrow, Performance Analyst moves selected thresholds to the Threshold Template you are creating or editing.</td>
</tr>
<tr>
<td>Add</td>
<td>Use this function to add statistics to the threshold template you are creating/editing. When you click this button, Performance Analyst opens the Threshold Properties dialog box.</td>
</tr>
<tr>
<td>Edit</td>
<td>Lets you edit the selected Datasource Thresholds appearing in the Current Template window. When you click this button, Performance Analyst opens the Threshold Properties dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Lets you delete the selected Datasource Thresholds. When you click this button, Performance Analyst deletes the selected thresholds.</td>
</tr>
<tr>
<td>Save</td>
<td>Saves any changes you made in this dialog box to create the new or modified template.</td>
</tr>
</tbody>
</table>
Threshold Template Maintenance Dialog Box

Threshold Templates are logical containers used to define and deploy thresholds to datasources. The Threshold Template Maintenance dialog box lets you add or edit templates, as well as configure template maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your needs. You can control which statistics are sampled for performance violations, as well as the levels of severity to report. When you are satisfied with your Template, you can save it, and then, in one action, apply it to one or more datasources, thereby immediately enforcing your performance standards.

To open the Threshold Template Maintenance dialog box:
• Choose Threshold Maintenance... from the Performance Analyst drop-down menu.
• Hover your mouse over the red and green cylinder on the Performance Analyst toolbar and look for the Threshold Maintenance tool-tip.

The table below describes the options and functionality on the Threshold Template Maintenance dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td>This is a drop-down list of the platforms. The default is the platform of the datasource you connected to most recently, but you can make another selection.</td>
</tr>
<tr>
<td>Name</td>
<td>Displays the name(s) of available Threshold Template(s).</td>
</tr>
<tr>
<td>Description</td>
<td>Displays the description(s) of available Threshold Template(s). For customized templates, the description is merely a repetition of the template name unless you wrote something when you created or edited the template.</td>
</tr>
<tr>
<td>Add</td>
<td>Lets you create Threshold Templates. When you click this button, Performance Analyst opens the Threshold Template Properties dialog box.</td>
</tr>
<tr>
<td>Edit</td>
<td>Lets you edit the selected Threshold Templates. When you click this button, Performance Analyst opens the Threshold Template Properties dialog box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Lets you delete the selected Threshold Templates.</td>
</tr>
<tr>
<td>Copy</td>
<td>Lets you create copies of the selected Threshold Template. When you click this button, Performance Analyst opens the Copy Template dialog box.</td>
</tr>
<tr>
<td>Apply</td>
<td>Lets you select datasources to apply the selected Threshold Templates. When you click this button, Performance Analyst opens the Select Datasource(s) dialog box.</td>
</tr>
<tr>
<td>Select/Unselect All</td>
<td>Lets you make global additions to your templates. <strong>NOTE:</strong> You cannot edit or delete Default templates.</td>
</tr>
</tbody>
</table>

Datasource Thresholds Maintenance Dialog Box

Datasource Thresholds are default or user-defined values that mark performance boundaries. You can apply Datasource Thresholds to specific statistics on a datasource. You can also combine Datasource Thresholds to create a template.

Performance Analyst offers the Datasource Thresholds Maintenance dialog box to let you modify Datasource Thresholds. There are two panes in the Datasource Thresholds Maintenance dialog box:
• Applied Threshold Template
• User Thresholds

To open the Datasource Thresholds Maintenance dialog box:
• Choose Datasource Thresholds... from the Performance Analyst drop-down menu.
• Right-click in any open Performance Analyst window and choose **Datasource Thresholds**... from the drop-down list.

The table below describes the options and functionality in the Applied Threshold Template box of the Datasource Thresholds Maintenance dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Displays the name of the target Threshold Template.</td>
</tr>
<tr>
<td>Description</td>
<td>Displays any description of the Threshold Template.</td>
</tr>
<tr>
<td>Change</td>
<td>Lets you view and edit all platform-specific Threshold Templates. When you click this button, Performance Analyst opens the Threshold Templates dialog box.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes all thresholds in the Applied Threshold Template grid.</td>
</tr>
<tr>
<td>Threshold/Severity/Condition/</td>
<td>Displays the statistic on which the threshold is placed, whether it is critical, fair, or poor, and the condition that triggers the alarm.</td>
</tr>
<tr>
<td>Block</td>
<td>Prevents that threshold statistic from being triggered in the template. It’s almost as if that threshold never existed in the template. You can, however, reinstate the threshold trigger by highlighting and deleting it from the User Thresholds pane.</td>
</tr>
<tr>
<td>Override</td>
<td>Copies the threshold settings from the selected Threshold Template and applies the change to the user-defined Datasource Thresholds. Click this button if the template has the correct Severity and Threshold, but you want to modify the condition for triggering the alarm. When you create a &quot;User Threshold,&quot; you are overriding the threshold specified by applied template or you are including an additional threshold not included in the applied template. This may be useful if you have a group of individuals all monitoring the same primary statistics (and thus using the same template) but want to be able to give individual users the ability to modify/add/block a threshold used by the template, without affecting the original template.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality in the User Threshold box of the Datasource Thresholds Maintenance dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold</td>
<td>Displays the statistic on which the threshold is placed.</td>
</tr>
<tr>
<td>Severity</td>
<td>Displays the severity for the threshold.</td>
</tr>
<tr>
<td>Condition</td>
<td>Displays the condition for the threshold.</td>
</tr>
<tr>
<td>Add</td>
<td>Lets you create Datasource Thresholds. When you click this button, Performance Analyst opens the Threshold Properties dialog box.</td>
</tr>
<tr>
<td>Edit</td>
<td>Lets you edit the selected Datasource Thresholds. When you click this button, Performance Analyst opens the Threshold Properties dialog box. If you Edit a Blocked threshold, it loses its blocked status.</td>
</tr>
<tr>
<td>Delete</td>
<td>Lets you delete the selected Datasource Thresholds. When you apply this to a blocked threshold, it is removed from the user thresholds.</td>
</tr>
</tbody>
</table>

**Select Datasource(s)**
When you have created or edited a threshold template, and you are ready to distribute it, you have the opportunity to apply it to one or more target datasources using the Select Datasources dialog box.

**To open the Select Datasource(s) dialog box:**
• Choose **Threshold Maintenance**... from the Performance Analyst drop-down menu, and then click **Apply**.
The table below describes the options and functionality on the Select Datasources dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasources</td>
<td>You can apply the target threshold template to one or more datasources. You can only apply the target threshold template to the platform where it was created.</td>
</tr>
<tr>
<td>Override Datasource's User Thresholds</td>
<td>Lets you choose which thresholds to use on the target datasource - the thresholds in the Threshold Template or the user-defined thresholds on the target datasource. If you select the check box, Performance Analyst will override any Datasource Thresholds on the target datasource with the thresholds in the Thresholds Template.</td>
</tr>
</tbody>
</table>

Copy Template

The Copy Template dialog box lets you copy an existing Threshold Template, rename it, and save it for later use.

To open the Copy Template dialog box:
1. Choose Threshold Maintenance... from the Performance Analyst drop-down menu, to open the Threshold Template Maintenance dialog box.
2. Select the template you want to replicate, and then click Copy.

The table below describes the options and functionality of the Copy Template dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Lets you type the name for the new Threshold Template. The default name is Copy of Frankenstein, or whatever it is you named your template.</td>
</tr>
<tr>
<td>Description</td>
<td>Lets you type the description of the new Threshold Template. The default description is simply a repetition of the template name.</td>
</tr>
</tbody>
</table>

Top Sessions

The Top Sessions functionality lets you drill-down into the target metric data and quickly pinpoint the heaviest users of I/O and memory. Generally you are taken to the database-specific Expert Guide when you need an explanation of what you are seeing.

To open Top Sessions:
• Hover your mouse over the little head on the Performance Analyst toolbar and look for the Top Sessions tool-tip.

To open the Session Details View
• Double-click any entry in the Top Sessions grid of the Top Sessions view, and the Session Details view opens.

Top SQL

The Top SQL view displays all SQL running or executed in the current datasource. You can use the lists and boxes to reorder the display according to your specific needs.

To view Top SQL, do the following:
• On the Performance Analyst toolbar, click the SQL button.
Performance Analyst for Oracle Tutorial

Embarcadero Performance Analyst provides key insight into the complex world of database performance. Running inside the Embarcadero DBArtisan console, or stand-alone in a Web browser, Performance Analyst helps you pinpoint performance problems that threaten a database’s availability or response times. The powerful analysis facilities will satisfy advanced experts, yet novices will find Performance Analyst’s interface easy to use and the expert help a welcome aid to troubleshooting difficult performance issues.

About This Performance Analyst for Oracle Tutorial

This tutorial is intended to help you get started with Embarcadero Performance Analyst. After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Performance Analyst. You will have learned how to accurately monitor your database using the most popular performance methodologies in use today. You will know how to drill down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. Finally, you’ll understand how to fix performance problems using the integration that exists between Performance Analyst and Embarcadero DBArtisan.

This tutorial is divided into 8 sections.

Session 1: Getting Started with Performance Analyst

Session 2: Using Ratio-based Performance Analysis

Session 3: Using Bottleneck/Wait-based Performance Analysis

Session 4: Using Workload Performance Analysis

Session 5: Using SQL Performance Analysis

Session 6: Using Operating System Performance Analysis

Session 7: Identifying and Getting Help with Performance Problems

Session 8: Fixing Identified Problems

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see Additional Evaluation Resources.

Session 1: Getting Started with Performance Analyst for Oracle

Download and Install

You can obtain the latest version of Performance Analyst by downloading the DBArtisan software from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Performance Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Performance Analyst (although Performance Analyst can also run in standalone fashion within a Microsoft Internet Explorer Web browser).

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.
Overview for Performance Analyst for Oracle

The graphic below illustrates all the main elements of the DBArtisan User Interface.

When DBArtisan is connected to an Oracle database, you can open Performance Analyst two ways:

1. On the toolbar, click the Performance Analyst icon.
2. On Utilities menu, select Performance Analyst.

DBArtisan opens Performance Analyst.

Session 2: Using Ratio-Based Performance Analysis in Performance Analyst for Oracle

Ratio-based analysis has been around for many years, and used to be the only technique database administrators used when they were called on to diagnose the cause of a database slowdown. Performance ratios are very good roll-up mechanisms for busy DBAs, making possible the analysis-at-a-glance approach, however ratios must be computed correctly for them to be meaningful. Key ratios in Performance Analyst are calculated using delta statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. Performance Analyst makes it easy to see the ratios that are computed with delta statistics vs. cumulative. A triangle symbol denotes a delta-based statistic, while square symbols represent a cumulative-based measure.

Ratio-based analysis is available for all performance categories (memory, etc.) where ratios can assist in determining the performance level of a database. For example, if you would like to see memory-specific ratios, on the Performance Analyst toolbar click Memory. Performance Analyst displays the Memory home page that showcases all important memory-related ratios.

Obtaining Detail Information for a Ratio Using Performance Analyst for Oracle

A ratio gives you a global picture for a particular performance measure. However, you often need more information regarding a statistical measure so you can make an informed decision. Performance Analyst has easy-to-use drill down features that make obtaining detailed information for various ratios a breeze.

For example, to view more information for the Buffer Cache Hit Ratio statistic, do the following:

1. On either the Main Home page or the Memory Home page, locate the Buffer Cache Hit Ratio statistic.
2. Position the pointer over the statistic.
   The pointer changes to a hand symbol.
3. Double-click the left mouse button.
   Performance Analyst opens the Buffer Cache Tab of the memory drill down view.
4. Or right-click the statistic, select Details or Navigate to… > Memory > Buffer Cache.
   Performance Analyst opens the Buffer Cache Tab of the memory drill down view.

Session 3: Using Bottleneck/Wait-Based Performance Analysis in
Performance Analyst for Oracle
When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things, or it can be an indicator that a database bottleneck exists. Bottleneck analysis is a valid method of measuring performance because it helps you track where a database and user sessions have been spending their time. If latch contention or heavy table-scan activity has been dragging a database's performance down, you can use bottleneck analysis to confirm the actual root cause.

NOTE: For Performance Analyst to show wait times for Oracle wait events, the database configuration parameter TIMED_STATISTICS must be set to TRUE.

Performance Analyst uses two broad areas of metrics to form its bottleneck analysis. The first is the Oracle wait event interface, which displays where the database and user sessions have been bottlenecked. But, when using bottleneck analysis you can not rely only on the information contained in the wait event views that Oracle provides. For example, an object may attempt to extend into another extent of space in a tablespace and yet, be denied if no such free space exists. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. For this reason, Performance Analyst also shows non-wait event based bottlenecks that represent a very real threat to a database's availability and performance.

Performance Analyst prominently displays bottleneck analysis on every performance home page. The Main home page, displays a summary of all key bottlenecks threatening the database. You can also see specific bottlenecks for memory, I/O, space, objects, and users by navigating to each home page and viewing the Bottleneck Analysis section for that home page. For example, to see bottlenecks that specifically relate to I/O, do the following
1. On the toolbar, click the I/O button.
2. Locate the Bottleneck Analysis section at the lower-left portion of the screen.

The I/O bottleneck analysis section displays the top database and log-related Oracle wait events (ranking them by wait time and not actual wait count), along with the most active database files.

Obtaining Detail on System Bottlenecks Using Performance Analyst for Oracle
You can easily obtain more information regarding global bottlenecks that are plaguing a database by drilling down into a particular bottleneck metric. For example, if you would like to see more data regarding the current top system bottlenecks, do the following:
1. Open the Main home page.
2. Double-click the Top System Bottlenecks area.

Performance Analyst opens the System Waits view.
3. Or right-click anywhere in Performance Analyst, point to Users, and then click System Waits.

Performance Analyst opens the System Waits view.

The System Waits view displays critical details regarding all the wait events that Oracle has recorded. You can easily see what events that occurred the most often as well as the events that have accumulated the most wait time.

Viewing Detail for Session Bottlenecks in Performance Analyst for Oracle
Not only can you obtain detail information for system bottlenecks, but you can also use Performance Analyst to get drill down information regarding session-level bottlenecks. There are several layers to session-level details you can view.
To see more information regarding session-level waits, do the following:

1. Open the Main home page.
2. Double-click the Top Session Bottlenecks area.
   Performance Analyst opens the Session Waits drill down view.
3. Or right-click anywhere in Performance Analyst, point to Users, and then click Session Waits.
   Performance Analyst opens the Session Waits drill down view.

When you are in the Session Waits drill down view, to drill further into a session's details, do the following:

1. In the Historical Waits or Current Waits grid, select a session that appears to be experiencing critical waits.
2. Double-click the session or right-click, and then select Details.
   Performance Analyst opens a drill down view that shows key statistics for the selected session, including its current work activities, wait events, and SQL statements.

Session 4: Using Workload Performance Analysis in Performance Analyst for Oracle

Key ratios give you a global perspective on database activity. Bottleneck analysis gives you insight into things that are holding up user activity and throughput. But another technique is necessary if a database professional is to really get a handle on what's occurring inside a badly performing database.

Workload analysis involves the investigation of two critical areas of a database's performance:
- Session resource consumption and activity
- SQL execution

Without looking at these two key performance categories, you can miss a lot of what could be responsible for perceived performance problems. When performance on a database takes a sudden nosedive, it is not uncommon to find one or two sessions that are causing the bulk of the workload. Finding these processes can be easily accomplished by viewing session metadata coupled with resource consumption and statistical execution statistics.

Pinpointing Top Resource Consumers in Performance Analyst for Oracle

Performance Analyst identifies top resource consumers in a couple of different ways. On the main home page, Performance Analyst highlights the top resource sessions across physical and logical I/O, memory usage, and CPU consumption.

The percentage used across all statistical categories are displayed so you can immediately pinpoint a session that is using all or most of a particular resource. The top resource sessions are also displayed on the memory, I/O, and other home pages as well with a listing for the top sessions for that category. For example, the top memory users appear on the memory home page.

Getting Workload Details Using Performance Analyst for Oracle

Performance Analyst offers several details on top resource sessions in addition to the summaries provided on each home page. A detailed and flexible top sessions view lets you view a summary of all the major resource sessions on the database as well as every critical detail regarding each session connected to the database.
Further drill downs are available that highlights specific details regarding each session’s activities and submitted SQL. To see detailed information regarding a session identified in one of Performance Analyst’s Workload Analysis sections, do the following:

1. In a **Workload Analysis** display, position the pointer over one of the processes.
2. Double-click the process or right-click, and then select **Details**.

If you want like to see all key workload analysis metadata for sessions, do the following:

1. On the toolbar, click the **Top Sessions** button.
2. Or right-click anywhere in Performance Analyst, point to **Users**, and then click **Session Waits**.

### Session 5: Using SQL Performance Analysis in Performance Analyst for Oracle

SQL-based performance analysis is really a subset of Workload Analysis. Understanding current and historical SQL execution patterns lets you to have the second set of data points necessary to properly perform workload analysis. You may find that optimizing SQL code produces some of the best performance-enhancing boosts available for a database.

### Uncovering Resource-Intensive SQL in Performance Analyst for Oracle

Performance Analyst offers plenty of insight into SQL performance and execution patterns. Each one of the home pages depicts the most costly SQL statements as determined by a variety of different metrics for that performance category (most physical, logical I/O, etc.) For example, Performance Analyst might show that a single SQL statement is responsible for almost 60 percent of all the physical I/O on the database.

### Getting Details on Top SQL with Performance Analyst for Oracle

Getting details on identified SQL statements is easy in Performance Analyst. For example, if you would like to see further information on a SQL statement identified on the Performance Analyst Home page, do the following:

1. In the **SQL Analysis** section, position the pointer over **Hash Values**.
2. Double-click or right-click, and then select **Details**.

Performance Analyst opens the Top SQL summary page that shows a breakdown of all top SQL statements along with their execution statistics.

If you would like to see the full SQL text of a SQL statement, do the following:

1. In the **Top SQL Details** section, double-click a statement.

Performance Analyst opens the SQL Details Tab of the Top SQL view to display the entire SQL statement and the execution metrics. While on this pane, you can also perform an EXPLAIN of the SQL statement and examine its access path information.

**NOTE:** To automatically tune a problem SQL statement if you have Embarcadero SQL Tuner installed on your workstation, do the following:

2. On the **SQL Details Tab**, select a problem statement, and then click the **Tune SQL** button.

SQL Tuner opens and you can begin a tuning session.
Session 6: Using Operating System Performance Analysis in Performance Analyst for Sybase

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O or Memory), a slow CPU, insufficient I/O devices, and insufficient memory. Performance Analyst allows you to examine the relevant operating system metrics for Windows Platforms (NT /2000/XP/2003) as they relate to Oracle.

To view the operating system metrics of the machine hosting the Sybase server you are monitoring, do the following:

1. Click the OS Performance Analyst toolbar button.
   OR
2. Navigate to the OS Home Page by going to OS, and then Home.

   NOTE: The first time you invoke the OS feature you will be prompted for an operating system login. Please specify the correct operating system, host name, username and password, and protocol.

The Operating System home page has been designed in a similar fashion as the datasource home page. On this page you can see the pertinent operating system metrics displayed in five quadrants broken down into the following categories:

- Resource Usage
- Memory Analysis
- Disk Analysis
- Workload Analysis
- Bottleneck Analysis

Obtaining Details on the Operating System Using Performance Analyst for Sybase

You can easily obtain more information regarding operating system problems that are degrading database performance by drilling down into a particular analysis quadrant. For example if you'd like to see more information regarding the processes running on the host machine, do the following:

1. Navigate to the Performance Analyst Operating System home page.
2. Double-click the Top Memory Process in the Workload Analysis quadrant.

   Performance Analyst opens the Processes detail view.
   OR
3. Right-click anywhere in Performance Analyst and select the Navigate to, OS, and then Processes.

The Processes detail tab shows all the processes on the host machine. From this view you can easily determine which processes are showing high marks in I/O or CPU, for example, by simply double clicking on the column headers and sorting the processes in ascending or descending order.

Session 7: Identifying and Getting Help with Performance Problems Using Performance Analyst for Oracle

You do not need to be an advanced expert in performance tuning to spot trouble in a database if you use Performance Analyst. There are a number of mechanisms in the tool that help you quickly locate areas of concern within a database.
Using the Alarm Log
Performance Analyst has a central and flexible alarm log that lists every performance violation found in all monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst lists all alarms for each database you are monitoring in one central log so you can see every problem in one place. To view the Performance Analyst Alarm log, do the following:

1. On the toolbar, click the **Alarms** button.

Once in the Alarm log, you can perform a number of actions including:

- Selecting an alarm and viewing the application page that shows details regarding the metric in violation.
- Filtering the types of alarms shown through the Options dialog.
- Clearing all current alarms.
- Copying or saving the Alarm log into a different format.

To select any of these options, do the following:

1. Position the pointer over any area of the **Alarm Log**.
2. In the shortcut menu, select an option.

Recognizing Performance Violations
Performance Analyst makes it easy to visually pinpoint performance violations in a database. Any statistic that does not conform to your predefined performance standards is colorized in any view that it appears so you can easily see potential troubleshoots.

The colors used to highlight a particular performance violation depend on the severity of the problem. Performance Analyst uses several layers of performance standards/threshold violations with each layer being represented by a different color. You can control everything regarding threshold violations, including how thresholds are determined, their color representation in Performance Analyst's display, and more by using the Threshold Template feature of Performance Analyst.

How can Performance Analyst inform you of performance violations in your database if you minimize the product to your desktop and are working on other things? Even if Performance Analyst isn't right in front of you, it can still let you know that a problem has been found in your database through the sister alarm icon. When Performance Analyst detects a problem in one of your databases, one of the things the product does is enable an alarm icon in your workstation’s systray area that communicates the presence of alarms in your database. So even if the Performance Analyst interface isn't in sight, the product can let you know that something in your database requires attention.

Creating Threshold Templates
How does Performance Analyst know what to signal as a problem in a database? It uses predefined performance standards that you define and enforce. All of this is accomplished by using Performance Analyst's Threshold templates. A template is a set of performance metrics and their thresholds that can be reused to enforce performance standards across any number of databases. Embarcadero provides a default performance template that contains nearly all of the most-watched performance statistics so you can begin monitoring with confidence right out of the box. However, you can create your own Threshold templates so you only are alerted to performance violations that are important to you and your particular environment.

To view or create Threshold templates, do the following:

1. On the toolbar, click the **Thresholds** button.

   Performance Analyst opens the Threshold maintenance dialog box.
The Threshold Template Properties dialog box lets you select the options to add or edit templates, as well as other maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your own needs. You can control what statistics are sampled for performance violations as well as the levels of severity you would like reported. Once you are satisfied with your template, you can save it and then apply it to one or more datasources in one action to immediately begin enforcing your performance standards.

Getting Help with Performance Problems
As databases become more complex, so do the problems that plague them. Knowing what to do for a particular performance problem can be hard in some cases, but Performance Analyst helps by providing a complete set of expert help that you can use as a guide when performance problems occur. To obtain expert help for any statistic shown in Performance Analyst, do the following:

1. Position the pointer over a statistic that you need help.
2. Right-click and select Help.

The Expert Help gives you a general description of the statistic, the range of values that are considered optimal, and some suggestions on how to fix a problem situation. Expert help can also be invoked for any of the Ratio, Bottleneck, SQL, or Workload Analysis sections shown on every home page by clicking on the help icon for that section, which is located at the top right corner of the section.

Session 8: Fixing Performance Problems Using Performance Analyst for Oracle
Finding performance problems is one thing, but fixing them is another. Many performance monitors point out a particular problem in a database, but then offer no way to actually correct the situation. Such is not the case with Performance Analyst. Because of its tight integration with DBArtisan, certain problems found in Performance Analyst can be visually corrected by DBArtisan.

If an Oracle tablespace is about to run out of free space, Performance Analyst can issue an alarm that a space shortage exists. Once the space problem has been identified, do the following:

1. On the Space home page toolbar, click the Space button.
2. In the Space Analyst grid, right-click a tablespace, and then select Manage with DBArtisan.

   DBArtisan opens an editor where you can perform corrective actions, such as enabling Autoextend for the tablespace, resizing the tablespace’s existing datafile(s), or adding a new datafile for the tablespace.
Performance Analyst for Oracle Expert Guide

This section includes expert help for all Oracle categories and statistics in Performance Analyst views. For detailed information on using the application, see Using Performance Analyst. This guide includes the following sections:

<table>
<thead>
<tr>
<th>Home Page Statistics</th>
<th>I/O Page Statistics</th>
<th>Memory Page Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects Page Statistics</td>
<td>OS Page Statistics</td>
<td>RAC Detail</td>
</tr>
<tr>
<td>Response Time Statistics (Oracle 10g Only)</td>
<td>Session Details</td>
<td>Space Page Statistics</td>
</tr>
<tr>
<td>Users Page Statistics</td>
<td>Top SQL</td>
<td></td>
</tr>
</tbody>
</table>

HomePage Statistics

The Home view lets you review availability and overall performance of all monitored databases from a single window. The Home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key Ratio Analysis</th>
<th>SQL Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Analysis</td>
<td>Workload Analysis</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

I/O Page Statistics

Memory Page Statistics

Objects Page Statistics

OS Page Statistics

Space Page Statistics

Users Page Statistics

Key Ratio Analysis - Home

Database performance analysts typically use one of two methods for examining the performance levels of a database: Ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as roll-up mechanisms for at-a-glance performance analysis. Succinctly presented performance ratios can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to which rules to follow, there are some standards to adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios a you examine come from the v$sysstat performance view. The performance view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impacts how a particular ratio is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.
While there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Home page to succinctly communicate the general overall performance levels of the monitored database:

<table>
<thead>
<tr>
<th>Buffer Cache Hit Ratio</th>
<th>Library Cache Hit Ratio</th>
<th>Dictionary Cache Hit Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Sort Ratio</td>
<td>Parse/Execute Ratio</td>
<td>Buffer Busy Waits</td>
</tr>
<tr>
<td>Rollback Contention Ratio</td>
<td>Latch Miss Ratio</td>
<td>Parallel Query Busy Ratio</td>
</tr>
<tr>
<td>Free Shared Pool Percent</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For more information, see Home Page Statistics.

**Bottleneck Analysis - Home**

When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting can mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs used this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps you track where a database has been spending its time. If latch contention or heavy table scan activity has been dragging a database's performance down, you can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, you can determine which sessions and objects are causing the problem.

For wait event analysis to be properly carried out, it is imperative that the timed_statistics initialization parameter be set to TRUE. By default this parameter is set to FALSE, which disallows the collection of wait times for each wait event defined to the Oracle engine. For one to really understand the impact of wait events on database performance, you need to discover what the database is or has been waiting on, and the durations of the waits. Having both allows a complete picture to be formed regarding the magnitude of wait-initiated performance degradations. Almost all Oracle experts now agree that allowing the collection of timing statistics adds little if anything to database overhead, so setting timed_statistics to TRUE should not be a worry. The Performance Analyst Home page identifies the top system and session waits that can decrease performance.

When using bottleneck analysis, you cannot rely only on the information contained in the wait event views that Oracle provides. For example, an object can attempt to extend into another extent of space in a tablespace and yet be denied if no such free space exists. Such a failure is not reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that you cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Home page displays space-related bottlenecks as well as other sources of contention/bottlenecks that can decrease the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

<table>
<thead>
<tr>
<th>Top System Bottlenecks</th>
<th>Top Session Bottlenecks</th>
<th>Problem Tablespaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Objects</td>
<td>Current Object Blocks</td>
<td>Enqueue Waits</td>
</tr>
<tr>
<td>Free List Waits</td>
<td>Storage Analysis</td>
<td></td>
</tr>
</tbody>
</table>
SQL Analysis - Home

Much of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times. SQL statements that use much memory can also cause a problem in a database.

Before you can identify problem SQL in your database, you have to ask the question, “What is bad SQL?” What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Response (Elapsed) Time</td>
<td>The amount of time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i or later.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The amount of CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.</td>
</tr>
<tr>
<td>Physical I/O</td>
<td>This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.</td>
</tr>
<tr>
<td>Logical I/O</td>
<td>The number of memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.</td>
</tr>
<tr>
<td>Repetition</td>
<td>The number of times the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute can still cause a headache on your system if it is executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over again.</td>
</tr>
</tbody>
</table>

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

Fortunately, Oracle records all the above measures (some only in 9i), which makes tracking the SQL that has been submitted against an Oracle database much easier.

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you want answered:

1. What HAS been the worst SQL that has historically run in my database?
2. What IS the worst SQL that is running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It can be that the query is in an inefficient PL/SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.
There is the possibility that the SQL statement just is not tuned well. To determine that, you can go further into Performance Analyst’s Top SQL view and, if you have Embarcadero SQL Tuner installed, you can port the SQL over to SQL Tuner to better optimize the statement.

**Storage Analysis - Home**

While DBAs focus on memory settings and tuning SQL, storage problems can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

1. Hit-the-wall, which can bring things to a complete standstill.
2. Performance vampire, which slowly drains the performance of a database over time.

Storage problems have the capability to bring the curtain down on a database very quickly, as in the case of an archive log destination running out of free space. But storage problems can also silently work behind the scenes to slowly but surely rob a database of its performance. For example, you can access a hub table in a busy database when an application is first given life, but over time, if it develops a heavy migrated row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page reports on the total used and free space of all tablespaces in the database as well as whether the database is running in archivelog mode. For more detailed information on database space, use the Performance Analyst Space home page.

For more information, see [Home Page Statistics](#).

**Workload Analysis - Home**

When your database experiences performance slowdowns, you should answer the following:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Home page provides insight into the leading resource hogs of a database as well as a count of total active and inactive processes. You can view detailed information into what each leading session is currently involved with.

The Workload Analysis section of the Home page includes:

- Top Processes table
- User Processes bar

For more information, see [Home Page Statistics](#).

The buffer cache hit ratio is an indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. Data read from memory produces user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the Oracle buffer cache’s purposes in life.
The table below describes the key counters Performance Analyst uses to calculate the buffer cache hit ratio:

<table>
<thead>
<tr>
<th>Key Counter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB BLOCK GETS</td>
<td>Data read from memory for DML operations.</td>
</tr>
<tr>
<td>CONSISTENT GETS</td>
<td>Data read from rollback segments in memory.</td>
</tr>
<tr>
<td>PHYSICAL READS</td>
<td>Data read physically from disk.</td>
</tr>
<tr>
<td>Direct Reads</td>
<td>Data read physically from disk that bypasses the buffer cache. Direct reads are filtered out of overall physical reads so an accurate cache hit ratio can be determined.</td>
</tr>
</tbody>
</table>

Dividing the data read from memory by data read from disk yields the cache hit ratio.

**NOTE:** This statistic is also available on the Memory home page.

**Metrics**

To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, every database has its own ‘personality’ and can exhibit excellent performance with below average readings for the cache hit ratio. Excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

Investigate consistent low readings of 60% or less.

**NOTE:** For Oracle8i or earlier, the adjustment of the db_block_buffers tuning parameter is required. For Oracle9i and later, the db_cache_size parameter is the parameter that needs attention. Any increases in db_block_buffers to take effect on Oracle8i or earlier, the database must be cycled. The db_cache_size parameter in Oracle9i or later, however, is dynamic and can be altered without stopping and starting the database instance.

**Troubleshooting**

If a problem is found in Oracle8i or earlier, do the following:

- Edit the Init.ora file for the database.
- Increase the amount of db_block_buffers to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later, do the following:

- Increase the size of the db_cache_size parameter through use of the ALTER SYSTEM SET db_cache_size command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

**Library Cache Hit Ratio**

- **Metrics**
- **Troubleshooting**
Oracle’s shared pool offers a number of tuning possibilities and is made up of two main memory areas:

1. Library Cache
2. Data Dictionary Cache

The library cache hit ratio offers a key indicator in determining the performance of the shared pool. It holds commonly used SQL statements - basically database code objects. The library cache hit ratio shows how often SQL code is reused by other database users vs. the number of times an SQL statement is broken down, parsed, and then loaded (or reloaded) into the shared pool.

You can improve performance by encouraging the reuse of SQL statements so expensive parse operations can be avoided. The library cache assists this tuning effort.

**NOTE:** This statistic is available on the Home page, the Memory home page and on the Library Cache tab of the Memory Detail.

**Metrics**
A high library cache hit ratio is a desirable thing. Strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse.

**NOTE:** When a database is first started, the library cache hit ratio is not at an optimal level because all code being used is relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, increase the shared_pool_size parameter.

Other red flags that can indicate a too small shared pool include:

- A wait count for the event 'latch free' of 10 or greater.
- The library cache wait count of two or greater.

These indicators can be tracked with Performance Analyst’s Bottleneck and Wait detail views.

You can improve the library cache hit ratio by encouraging SQL code reuse through the implementation of bind variables. Discouraging hard coded literals in application code and instead making use of variables bound at run time aids in the reuse of SQL code that is maintained in Oracle’s shared pool.

**NOTE:** Bind variables can have an affect on the cost-based optimizer though.

A second way is to pin frequently used code objects in memory so they are available when needed, by using the system supplied DBMS_SHARED_POOL package. You can use Performance Analyst to view objects in the shared pool that are always present and/or have increasing reload numbers to help identify objects that are good candidates for pinning.

**Troubleshooting**
If a problem is found in Oracle8i or earlier, do the following:

- Edit the Init.ora file for the database.
- Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later, do the following:
• Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.

• Monitor the new value to see if performance improves.

• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

If you determine that SQL literals are causing SQL to not be reused, do the following (in Oracle 8.1.6 and later):

• Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.

• Monitor database to see if parse activity is reduced and library cache reloads shrink.

• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.

Dictionary Cache Hit Ratio

• Metrics

• Troubleshooting

Oracle’s shared pool offers an number of tuning possibilities and is made up of two main memory areas:

1 Library Cache

2 Data Dictionary Cache

The dictionary cache hit ratio offers another key indicator in determining the performance of the shared pool. It shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much of this vital reference information be kept in RAM as possible.

NOTE: This statistic is also available on the Memory home page.

Metrics

Just as with the library cache, a high data dictionary cache hit ratio is desirable. Strive for a hit ratio between 90-100%, with 95% being a good performance benchmark.

NOTE: When a database is first started, the data dictionary cache hit ratio is not at an optimal level because all references to object definitions are relatively new, and as such, must be placed into the shared pool. Look for hit ratios in the eighty’s for new database startups. If, however, after a solid hour or two of steady database time, the data dictionary cache hit ratio has not increased to desirable levels, increase the shared_pool_size parameter.

Databases supporting applications that involve large number of objects (such as an Oracle Financials installation) should have larger than normal shared pools to support the required object definitions.

Although each parameter is not individually tunable (it was in Oracle6), you can see which area of the dictionary cache could be pulling the overall hit ratio down.

Troubleshooting

If a problem is found in Oracle8i or earlier, do the following:

• Edit the Init.ora file for the database.
• Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.

• Cycle the Oracle server when possible to allow the new value to take effect.

• Monitor new value to see if performance improves.

If a problem is found in Oracle9i or later, do the following:

• Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.

• Monitor the new value to see if performance improves.

• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

Memory Sort Ratio

• Metrics

• Troubleshooting

Oracle's SGA is not the only memory structure used by Oracle for database work. One of the other memory areas used by Oracle8i and earlier for normal activity is an area set aside for sort actions. When a sort operation occurs, Oracle attempts to perform the sort in a memory space that exists at the operating system level. If the sort is too large to be contained within this space, it continues the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE. Oracle records the overall number of sorts that are satisfied in memory as well as those that end up being finalized on disk. Using these numbers, you can calculate the percentage of memory sorts vs. disk sorts and get a feel for how fast your sort activity is being resolved. Obviously, memory sorts completes many times faster than sorts forced to use physical I/O to accomplish the task at hand.

Oracle9i or later now has the option of running automatic PGA memory management. Oracle has introduced a new Oracle parameter called pga_aggregate_target. When the pga_aggregate_target parameter is set and you are using dedicated Oracle connections, Oracle ignores all of the PGA parameters in the Oracle file, including sort_area_size, hash_area_size and sort_area_retained_size. Oracle recommends that the value of pga_aggregate_target be set to the amount of remaining memory (less a 10% overhead for other server tasks) on a server after the instance has been started.

NOTE: This statistic is available on the Home page, Memory home page, and the Users home page.

Metrics

If the memory sort ratio falls below 90%, and you are on Oracle8i or earlier, increase the parameters devoted to memory sorts - sort_area_size and sort_area_retained_size.

For Oracle9i or later, investigate the use of pga_aggregate_target. Once the pga_aggregate_target has been set, Oracle automatically manages PGA memory allocation, based on the individual needs of each Oracle connection. Oracle9i or later allows the pga_aggregate_target parameter to be modified at the instance level with the alter system command, thereby lets you dynamically adjust the total RAM region available to Oracle9i.

Oracle9i also introduced a new parameter called workarea_size_policy. When this parameter is set to automatic, all Oracle connections benefits from the shared PGA memory. When workarea_size_policy is set to manual, connections allocates memory according to the values for the sort_area_size parameter. Under the automatic mode, Oracle tries to maximize the number of work areas that are using optimal memory and uses one-pass memory for the others.

Troubleshooting

If you find a problem, do the following:
• Edit the Init.ora or SPFILE file for the database.

• Increase the amount of sort_area_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value. EVERY user receives this amount for sorting.

• Cycle the Oracle server when possible to allow the new value to take effect.

• Monitor new value to see if performance improves.

In addition to increasing the amount of memory devoted to sorting, find inefficient SQL that cause needless sorts. For example, UNION ALL does not cause a sort whereas UNION does in an SQL query (to eliminate duplicate rows). DISTINCT is frequently misapplied to SQL statements and causes unnecessary sort actions.

There are times you simply cannot stop sort activity. This being the case, try to keep it in memory whenever possible. However, large data warehousing systems oftentimes simply exhaust RAM sort allotments, so if disk sorts must occur, try to ensure three things:

1. Your user's TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.

   NOTE: For Oracle9i or later, you can specify a default tablespace other than SYSTEM for every user account that is created.

2. The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.

3. The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.

Parse/Execute Ratio

Each time a new SQL statement is submitted to Oracle, the kernel must 'parse' the statement, which involves syntax checks, security checks, and object validations. The Parse/Execute Ratio shows the percentage of SQL executed that did not incur a hard parse.

   NOTE: This statistic is available on the Home page, Memory home page, and the Users home page.

Metrics

Seeing low values might indicate that users are firing SQL with many hard-coded literals instead of using bind variables within an application. High values (90% and greater) generally indicate Oracle is saving precious CPU by avoiding heavy parse tasks.

Troubleshooting

The best way to reduce unnecessary parse activity is to encourage SQL statement reuse. This can be done by promoting SQL execution through the use of stored procedures or applications where bind variables can be used. Oftentimes, literals in otherwise identical SQL statements can cause unneeded parse work for Oracle. The use of bind variables can counter that problem.

If you determine that SQL literals are causing SQL to not be reused, do the following (in Oracle 8.1.6 and later):

• Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.

• Monitor database to see if parse activity is reduced and library cache reloads shrink.

• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.
Buffer Busy Waits

Buffer busy waits occur when a process needs to access a data block in the buffer cache, but cannot because it is being used by another process. So it must wait. A wait event generally happens because a buffer is being read into the buffer cache by another process or the buffer is in the buffer cache, but cannot be switched to a compatible mode immediately.

Metrics

Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables.

Troubleshooting

On Oracle8i or earlier, the remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.

For Oracle9i or later, you can use the automatic segment management feature in Oracle9i locally-managed tablespaces to help make free list problems a thing of the past. Using an UNDO tablespace in 9i or later can help remedy any rollback contention problem.

You can also obtain which objects have actually experienced buffer busy waits in Oracle9i or later by querying the sys.v_$segment_statistics. This view is not populated unless the configuration parameter statistics_level is set to TYPICAL or ALL.

Rollback Contention Ratio

Rollback segments are used by Oracle to hold data needed to rollback (or undo) any changes made through inserts, updates, or deletes to various Oracle objects. They also allow Oracle to have read consistency for long running queries, are used for recovery purposes, and play a role during exports of database information. In a heavy transaction processing environment, rollback segments are accessed continuously and therefore are subject to contention problems. The Rollback Contention Ratio helps identify contention occurring on the system relating to rollbacks.

Metrics

Overall, if the rollback contention ratio approaches 1% or more, create more rollback segments. Also consider creating a specialized, larger, rollback segment to be used by long running transactions. Doing so alleviates dynamic rollback extensions and cuts down heavily on ORA-01555 Snapshot Too Old errors.

Troubleshooting

Begin by creating new rollback segments and altering them to be online for use. Then monitor the overall contention ratio to see if it begins to drop.

If you are using Oracle9i or later, consider using an UNDO tablespace and allowing Oracle to automatically control rollback segment management.

Latch Miss Ratio

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism - so either you get it or you do not and are forced to continually retry.

The latch miss ratio defines the number of times a process obtained a willing-to-wait latch vs. missing the attempt.
Metrics
If the latch miss ratio exceeds 1%, you should take action to resolve the amount of latch contention.

Troubleshooting
Examine the details regarding the latch contention. Increasing the shared_pool_size can assist in latch problems also. The table below describes latches:

<table>
<thead>
<tr>
<th>Latch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache buffer chain latch</td>
<td>Protects paths to database block buffers in the buffer cache. High I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers or db_cache_size parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.</td>
</tr>
<tr>
<td>Library cache latches</td>
<td>Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.</td>
</tr>
</tbody>
</table>

Other routine latch contention problems used to include the redo allocation and redo copy latches, but these have pretty much been made obsolete in Oracle 8.1.5 and later.

Parallel Query Busy Ratio
Oracle's parallel query feature, when used properly, allows for terrific increases in performance for many SQL and utility operations. Parallel operations can be introduced through SQL hints, specified in an object's DDL, or used in command line arguments for utility programs (like SQL*Loader). To effectively service parallel query requests, ensure that enough query servers exist in the database instance. The Parallel Query Busy Ratio is an indicator of how busy all the servers are on the database in question.

Metrics
If the Parallel Query Busy Ratio approaches 80-90%, add more query servers to the database, or examining parallel requests to ensure they are being used in an efficient and necessary manner.

Troubleshooting
To fix, do the following:

- Edit the Init.ora file or SPFILE for the database.
- Increase the amount of parallel_max_servers to a higher value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

You can also investigate the use of the parallel_automatic_tuning parameter in Oracle 8.1 and later.

Free Shared Pool Percent
Oracle's shared pool need not use all of the memory given to it through the shared_pool_size parameter. If the database does not have many object and code definitions to reference, then the shared pool can contain an abundance of free memory that is not being used.
Metrics
Under-allocating the shared pool size can have a serious impact on your database's performance, but over-allocating the shared pool can have run time ramifications as well. If you have a good chunk of memory allocated to the Oracle shared pool that is never used, it might be more of a performance enhancement to reduce the shared pool amount and instead give the memory to the buffer/data cache, or even back to the operating system itself. In terms of knowing when to reduce the shared pool, a good benchmark is continually seeing 2-3MB of free memory.

On the other hand, if after an hour or so of beginning database operation, you see that virtually no free memory is left in the shared pool, or you are seeing ORA-4031 errors (that indicate definitions cannot find enough contiguous free space in the shared pool), increase the pool by 10% or more.

Troubleshooting
If you continuously see little or no free memory in the shared pool, do the following:

For Oracle8i or earlier:
- Edit the Init.ora file for the database.
- Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

For Oracle9i or later:
- Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

You can also use the ALTER SYSTEM FLUSHED SHARED_POOL command to remove all objects from the shared pool and start with a clean slate.

Problem Tablespaces
The Problem Tablespaces metric is a count of all tablespaces in the database whose free space percentage amount has fallen below a user-defined limit.

Metrics
A rule of thumb for dynamic tablespaces (those with growing objects) is to keep a minimum of 10-15% free space available for object growth.

Troubleshooting
There are two ways to prevent a tablespace from running out of available free space:

1. Turn AUTOEXTEND on for the underlying tablespace's datafiles. This allows them to automatically grow when free space in the datafile has been exhausted.

2. Using the ALTER TABLESPACE ... ADD DATAFILE... command, you can manually add a new datafile to a tablespace that is about to run out of available free space.
Problem Objects
The Problem Objects statistic is a count of all objects in the database that are in danger of reaching their maximum extent limit or cannot allocate their next extent of free space because of a lack of overall or contiguous free space in their parent tablespace.

Metrics
Modify any object approaching their maximum extent limit or unable to allocate a new extent of space so they can continue to grow in size.

Troubleshooting
Depending on the situation, there are a number of things you can do to prevent object extent problems:

- Turn AUTOEXTEND on for the underlying parent tablespace’s datafiles. This allows a tablespace to automatically grow when free space in the datafile has been exhausted, and allows an object to extend even when little or no current free space is available.
- Using the ALTER TABLESPACE … ADD DATAFILE… command, you can manually add a new datafile to a tablespace that is about to run out of available free space.
- You can alter an object that is at or near their maximum extent limit so that the object has unlimited extents.
- With Oracle 8.1.5 and later, you can use locally-managed tablespaces to ensure that no object ever reaches its maximum extent limit, because all objects are allowed unlimited extents.
- An object can be reorganized into another tablespace or reorganized in general to reduce the number of extents the object currently takes up.

Top System Bottlenecks
When viewing wait statistics, there are many levels of detail that you can view. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. Viewing these numbers can help you determine which wait events have caused the most commotion in a database thus far. The Top System Bottlenecks section identifies the top waits that have occurred on the Oracle database based on the number of waits per event.

For Oracle 10g only you see a graph that provides a historical look at the last hour and allows you to see what wait classes experienced the most wait activity in terms of time waited. To get detail into which wait events have accumulated the most time, you can drill down into the System Waits view.

Metrics
None.

Troubleshooting
Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. For example, a ‘db file scattered read’ event is typically indicative of table scan operations. If you see many of these events, then you can begin to see if large table scans are occurring in the database. Like the ‘db file scattered read’ event, each wait event has its own meaning and individual end-resolution diagnosis.

After looking at system-level wait activity, you can discover which current connections are responsible for waits at the system level. Performance Analyst reports on historical and current wait events at the session level, making this investigation easy to accomplish.
Top Session Bottlenecks
When viewing wait statistics, there are many levels of detail that you can view. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. The second level is a historical look at waits from a session level. The third is which sessions are currently experiencing waits. The Top Session Bottlenecks section identifies the top sessions that are currently waiting, based on their wait time in seconds.

Metrics
None.

Troubleshooting
Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics.

The most common wait viewed at the session level is an ‘enqueue’ wait, which typically identifies lock contention. If enqueue waits are observed, then you can check the “Current Object Blocks” count on the Performance Analyst Home page, as well as the Users page which displays locks and blocking locks detail.

As with an enqueue event, each wait event has its own meaning and individual end-resolution diagnosis.

Current Object Blocks
A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

NOTE: This statistic is also called Sessions Blocked on the Users home page and Session Blocks on the Objects home page.

Metrics
Investigate any indicator above zero immediately before the situation has a chance to mushroom.

Troubleshooting
Once discovered, a blocking lock situation can normally be quickly remedied. You can issue a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. You can look at the Users Detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT … FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.
Enqueue Waits
An enqueue is an advanced locking device that allows multiple database processes to share certain resources. Enqueue waits typically occur when sessions wait to be granted a requested lock. Sometimes these locks are internal Oracle locks while other times they could be locks for rows of data in a table. Enqueues are issued implicitly by Oracle.

NOTE: This statistic is available on the Home page and the Objects home page.

Metrics
Investigate any enqueue waits that read consistently above one or more (delta statistics).

Troubleshooting
Removing contention for enqueues is almost always an application design issue. If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. Examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations.

Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).

Free List Waits
Free lists are lists of Oracle data blocks that contain free space for an Oracle object. Every table has at least one free list. Free lists are used to locate free blocks of space when a request is made of a table for the insertion of a row. Free list contention can reduce the performance of applications when many processes are involved in the insertion of data to the same table.

NOTE: This statistic is available on the Home page and the Objects home page.

Metrics
If consistent numbers for free list waits continues to appear, add additional free lists to the most dynamic database objects through the use of the STORAGE parameter. Another indicator of insufficient free lists are consistent, non-zero numbers for the buffer busy wait event.

Troubleshooting
If free list waits are observed, add more free lists to tables and indexes with high insert rates. For Oracle9i or later, objects can be created inside of locally-managed tablespaces that use the automatic segment management feature, which entirely does away with free lists.

Total Used Space/Total Free Space
These statistics represent the total used and free space available in all tablespaces/datafiles in the database. Although good to know, a more detailed listing by tablespace is needed to determine where any actual space shortages exist in the database.

NOTE: These statistics are in the Storage Analysis section of the Performance Analyst Home page and the Space home page.

Metrics
If any one tablespace begins to approach 90% used (and the objects contained within it are dynamic and growing as opposed to static), take action to prevent any future space allocation errors.
Troubleshooting
Here are a some things you can do to prevent a tablespace from running out of available free space:

- Turn AUTOEXTEND on for the underlying tablespace's datafiles. This allows them to automatically grow when free space in the datafile has been exhausted.
- Using the ALTER TABLESPACE ... ADD DATAFILE... command, you can manually add a new datafile to a tablespace that is about to run out of available free space.

For more information, see Storage Analysis.

Archive Log
Oracle can be placed into archivelog mode, which tells the database to make copies of the online redo log files for point-in-time recovery purposes. The Archive Log statistic displays whether the database is running in archivelog mode or not. This information is displayed in the Storage Analysis section of the Performance Analyst Home page.

Metrics
None.

For more information, see Storage Analysis.

Top Processes
When the database population as a whole experiences a system slowdown, it is not uncommon to find one or two users who are responsible for bringing the system to its knees. In the best of worlds, users have an evenly divided amount of memory usage, disk I/O, CPU utilization, and parse activity. Unfortunately, this is not usually the case. Users submit large batch jobs during peak OLTP activity, or when sessions that are firing off untuned queries on a critical system.

If you are seeing a slowdown in your database, and cannot seem to find a root cause, one thing to examine is the resource consumption of the leading sessions on a system. A handful of users can overwhelm the I/O capabilities of Oracle (through untuned queries or runaway batch jobs) or hammer the CPU or memory structures of the database and server.

Performance Analyst makes it easy to pinpoint the top sessions by showing the leading processes at the physical I/O, logical I/O, memory, and CPU usage levels.

Metrics
If any one session uses more than 50% of a total resource (CPU, memory, etc.) go into the session to find out what they are currently executing.

Active User Processes
The Active User Processes statistic is the total number of active and open threads reported in the database. Active Sessions displays the number of processes actively performing work.

Metrics
None.

For more information, see Inactive User Processes.
Inactive User Processes
The Inactive User Processes statistic is the total number of threads logged on to the database that are idle at the current time.

Metrics
A large number of inactive users could indicate user sessions that have mistakenly been left logged on. Because each user thread consumes a portion of memory on the Oracle server, sever any sessions not needing a connection to reduce resource usage.

Troubleshooting
Double-click this statistic to open the Top Sessions tab of the Users Detail page. On this tab you can check sessions that have many seconds idle and/or that have been logged on for very long periods of time as indicated by the logon time column. After verifying that the session is no longer necessary, you can KILL the session.

For more information, see Active User Processes.

Memory Page Statistics
The Memory home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key Ratio Analysis</th>
<th>SGA Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Analysis</td>
<td>Workload Analysis - Top Memory Hogs</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
Memory Detail
I/O Page Statistics
Objects Page Statistics
OS Page Statistics
Space Page Statistics
Users Page Statistics

Key Ratio Analysis - Memory
Object-related database activity can be examined using both ratio-based and wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios and statistical readings that can be used to indicate how active certain object types are. Performance ratios serve as roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis.

When using ratio-based analysis, there are some standards to adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that you examines come from the v$sysstat performance view. The performance view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impacts how a particular ratio is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.
A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database’s individual ‘personality’ with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead you to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following memory ratios are used on the Performance Analyst Memory home page to succinctly communicate the general overall memory performance levels of the monitored database:

<table>
<thead>
<tr>
<th>Buffer Cache Hit Ratio</th>
<th>Library Cache Hit Ratio</th>
<th>Dictionary Cache Hit Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Sort Ratio</td>
<td>Free Shared Pool Percent</td>
<td></td>
</tr>
</tbody>
</table>

For more information, see Memory Page Statistics.

Bottleneck Analysis - Memory

When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. You can use Bottleneck Analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Memory bottlenecks can definitely cause performance degradation in an otherwise well-running database. Typically, these bottlenecks center around Oracle’s buffer/data cache, library cache, and occasionally log buffer memory regions. To help you identify such problems, the following statistics are presented on the Memory home page:

<table>
<thead>
<tr>
<th>Buffer Busy Waits</th>
<th>Free Buffer Wait Average</th>
<th>Object Reloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redo Log Space Wait Time</td>
<td>Redo Log Space Waits</td>
<td>Top Latch Misses</td>
</tr>
</tbody>
</table>

For more information, see:

Memory Home Page

Buffer Busy Ratio

SQL Analysis - Memory

Much of a database’s overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times. SQL statements that use much memory can also cause a problem in a database. The SQL Analysis for memory shows what SQL statements have consumed the largest percentages of shareable, persistent, and runtime memory.

Before you can identify problem SQL in your database, you have to ask the question, “What is bad SQL?” What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The table lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Response (Elapsed) Time</td>
<td>The time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i or later.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.</td>
</tr>
</tbody>
</table>
There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

Fortunately, Oracle records all the above measures (some only in 9i), which makes tracking the SQL that has been submitted against an Oracle database much easier.

### Metrics
When you begin to look for inefficient SQL in a database, there are two primary questions you want answered:

1. What HAS been the worst SQL that has historically been run in my database?
2. What IS the worst SQL that is running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It could be that the query is in an inefficient PL/SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. You can view Performance Analyst’s Top SQL view and, if you have Embarcadero SQL Tuner installed, you can port the SQL over to SQL Tuner to better optimize the statement.

### SGA Analysis - Memory
Most DBAs know all about the Oracle System Global Area (SGA). The SGA is Oracle’s memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database. The table below describes Oracle memory structures:

<table>
<thead>
<tr>
<th>Memory Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default buffer cache</td>
<td>Maintains data blocks when they are read from the database. If you do not specifically place objects in another data cache, then any data requested by clients from the database is placed into this cache. The memory area is controlled by the <code>db_block_buffers</code> parameter in Oracle8i and earlier and <code>db_cache_size</code> in Oracle9i or later.</td>
</tr>
<tr>
<td>Keep buffer cache</td>
<td>For Oracle 8 or later, you can assign various objects to a special cache that retains those object’s requested blocks in RAM for as long as the database remains up. The keep cache’s main is for often-referenced lookup tables that should be kept in memory at all times for fast access. The <code>buffer_pool_keep</code> parameter controls the size of this cache in Oracle8, while the <code>db_keep_cache_size</code> parameter handles the cache in Oracle9i or later. The keep pool is a sub-pool of the default buffer cache.</td>
</tr>
</tbody>
</table>
Oracle also maintains a fixed area in the SGA that contains a number of atomic variables, pointers, and other miscellaneous structures that reference areas of the SGA.

For more information, see Memory Home Page.

Workload Analysis - Top Memory Hogs

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Oftentimes, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system level (ORA-4030 errors).

If your database server does not have an overabundance of memory, periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 25-50% of the total memory usage, investigate the sessions further to see the activities they are performing.

For more information, see Memory Home Page.

Buffer Cache Hit Ratio

- Metrics
- Troubleshooting

Oracle also maintains a fixed area in the SGA that contains a number of atomic variables, pointers, and other miscellaneous structures that reference areas of the SGA.

For more information, see Memory Home Page.
The buffer cache hit ratio is an indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. Data read from memory produces user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the Oracle buffer cache's purposes in life.

The table below describes the key counters Performance Analyst uses to calculate the buffer cache hit ratio:

<table>
<thead>
<tr>
<th>Key Counter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB BLOCK GETS</td>
<td>Data read from memory for DML operations.</td>
</tr>
<tr>
<td>CONSISTENT GETS</td>
<td>Data read from rollback segments in memory.</td>
</tr>
<tr>
<td>PHYSICAL READS</td>
<td>Data read physically from disk.</td>
</tr>
<tr>
<td>Direct Reads</td>
<td>Data read physically from disk that bypasses the buffer cache. Direct reads are filtered out of overall physical reads so an accurate cache hit ratio can be determined.</td>
</tr>
</tbody>
</table>

TIP: Dividing the data read from memory by data read from disk yields the cache hit ratio.

Metrics
To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, every database has its own ‘personality’ and can exhibit excellent performance with below average readings for the cache hit ratio. Excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

You should investigate consistent low readings of 60% or lower.

NOTE: For Oracle8i or lower, the adjustment of the db_block_buffers tuning parameter is required. For Oracle9i or higher, the db_cache_size parameter is the parameter that needs attention. Any increases in db_block_buffers to take effect on Oracle8i or lower, the database must be cycled. The db_cache_size parameter in Oracle9i or later, however, is dynamic and can be altered without stopping and starting the database instance.

To view individual session hit ratios that can be depressing the overall cache hit ratio for the database, drill down into the overall buffer cache hit ratio.

Troubleshooting
If a problem is found in Oracle8i or earlier:

• Edit the Init.ora file for the database.
• Increase the amount of db_block_buffers to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
• Cycle the Oracle server when possible to allow the new value to take effect.
• Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later:

• Increase the size of the db_cache_size parameter through use of the ALTER SYSTEM SET db_cache_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
• Monitor the new value to see if performance improves.
• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.
Library Cache Hit Ratio

- **Metrics**
- **Troubleshooting**

Oracle's shared pool offers a number of tuning possibilities and is made up of two main memory areas:

1. Library Cache
2. Data Dictionary Cache

The library cache hit ratio offers a key indicator in determining the performance of the shared pool. It holds commonly used SQL statements - basically database code objects. The library cache hit ratio shows how often SQL code is reused by other database users vs. the number of times an SQL statement is broken down, parsed, and then loaded (or reloaded) into the shared pool.

You can improve performance by encouraging the reuse of SQL statements so expensive parse operations can be avoided. The library cache assists this tuning effort.

**NOTE:** This statistic is also available on the Home page, the Memory home page and on the Library Cache tab of the Memory Detail.

**TIP:**

**Metrics**
A high library cache hit ratio is a desirable thing. Strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse.

**NOTE:** When a database is first started, the library cache hit ratio is not at an optimal level because all code being used is relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, increase the shared_pool_size parameter.

Other red flags that can indicate a too small shared pool include:

- A wait count for the event 'latch free' of 10 or greater.
- The library cache wait count of two or greater.

These indicators can be tracked with Performance Analyst's Bottleneck and Wait detail views.

A way of improving the library cache hit ratio is by encouraging code reuse through the implementation of bind variables. Discouraging hard coded literals in application code and instead making use of variables bound at run time aids in the reuse of SQL code that is maintained in Oracle's shared pool.

**NOTE:** Bind variables can have an affect on the cost-based optimizer though.

A second way is to pin frequently used code objects in memory so they are available when needed, by using the system supplied DBMS_SHARED_POOL package. You can use Performance Analyst to view objects in the shared pool that are always present and/or have increasing reload numbers to help identify objects that are good candidates for pinning.

**Troubleshooting**
If a problem is found in Oracle8i or earlier:

- Edit the Init.ora file for the database.
- Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
• Cycle the Oracle server when possible to allow the new value to take effect.
• Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later:
• Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
• Monitor the new value to see if performance improves.
• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

If you determine that SQL literals are causing SQL to not be reused, you can (in Oracle 8.1.6 and later):
• Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.
• Monitor database to see if parse activity is reduced and library cache reloads shrink.
• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.

Dictionary Cache Hit Ratio

Metrics
• Troubleshooting

Oracle's shared pool offers an number of tuning possibilities and is made up of two main memory areas:

1. Library Cache
2. Data Dictionary Cache

The dictionary cache hit ratio offers another key indicator in determining the performance of the shared pool. It shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much of this vital reference information be kept in RAM as possible.

NOTE: This statistic is also available on the Memory home page.

TIP:

Metrics
Just as with the library cache, a high data dictionary cache hit ratio is desirable. Strive for a hit ratio between 90-100%, with 95% being a good performance benchmark.

NOTE: When a database is first started, the data dictionary cache hit ratio is not at an optimal level because all references to object definitions are relatively new, and as such, must be placed into the shared pool. Look for hit ratios in the eighty's for new database startups. If, however, after a solid hour or two of steady database time, the data dictionary cache hit ratio has not increased to desirable levels, increase the shared_pool_size parameter.

Databases supporting applications that involve large number of objects (such as an Oracle Financials installation) should have larger than normal shared pools to support the required object definitions.
Although each parameter is not individually tunable (it used to be in Oracle6), you can see which area of the dictionary cache could be pulling the overall hit ratio down.

**Troubleshooting**

If a problem is found in Oracle8i or earlier:

- Edit the Init.ora file for the database.
- Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

If a problem is found in Oracle9i or later:

- Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

**Memory Sort Ratio**

- **Metrics**
- **Troubleshooting**

Oracle's SGA is not the only memory structure used by Oracle for database work. One of the other memory areas used by Oracle8i and earlier for normal activity is an area set aside for sort actions. When a sort operation occurs, Oracle attempts to perform the sort in a memory space that exists at the operating system level. If the sort is too large to be contained within this space, it continues the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE. Oracle records the overall number of sorts that are satisfied in memory as well as those that end up being finalized on disk. Using these numbers, you can calculate the percentage of memory sorts vs. disk sorts and get a feel for how fast your sort activity is being resolved. Obviously, memory sorts completes many times faster than sorts forced to use physical I/O to accomplish the task at hand.

Oracle9i or later now has the option of running automatic PGA memory management. Oracle has introduced a new Oracle parameter called pga_aggregate_target. When the pga_aggregate_target parameter is set and you are using dedicated Oracle connections, Oracle ignores all of the PGA parameters in the Oracle file, including sort_area_size, hash_area_size and sort_area_retained_size. Oracle recommends that the value of pga_aggregate_target be set to the amount of remaining memory (less a 10% overhead for other server tasks) on a server after the instance has been started.

**NOTE:** This statistic is available on the Home page, Memory home page, and the Users home page.

**TIP:**

If your memory sort ratio falls below 90%, and you are on Oracle8i or earlier, increase the parameters devoted to memory sorts - sort_area_size and sort_area_retained_size.
For Oracle9i or later, investigate the use of pga_aggregate_target. Once the pga_aggregate_target has been set, Oracle automatically manages PGA memory allocation, based upon the individual needs of each Oracle connection. Oracle9i or later allows the pga_aggregate_target parameter to be modified at the instance level with the alter system command, thereby letting you dynamically adjust the total RAM region available to Oracle9i.

Oracle9i also introduces a new parameter called workarea_size_policy. When this parameter is set to automatic, all Oracle connections benefit from the shared PGA memory. When workarea_size_policy is set to manual, connections allocate memory according to the values for the sort_area_size parameter. Under the automatic mode, Oracle tries to maximize the number of work areas that are using optimal memory and uses one-pass memory for the others.

**Troubleshooting**

If you find a problem, do the following:

- Edit the Init.ora or SPFILE file for the database.
- Increase the amount of sort_area_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value. EVERY user receives this amount for sorting.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

In addition to increasing the amount of memory devoted to sorting, find inefficient SQL that cause needless sorts. For example, UNION ALL does not cause a sort whereas UNION does in an SQL query (to eliminate duplicate rows). DISTINCT oftentimes is misapplied to SQL statements and causes unnecessary sort actions.

There are times you simply cannot stop sort activity. This being the case, try to keep it in memory whenever possible. However, large data warehousing systems oftentimes simply exhaust RAM sort allotments, so if disk sorts must occur, try to ensure three things:

- Your user’s TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.

  *NOTE:* For Oracle9i or later, you can specify a default tablespace other than SYSTEM for every user account that is created.

- The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.
- The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.

**Free Shared Pool**

**Percent**

- **Metrics**
- **Troubleshooting**

Oracle's shared pool need not use all of the memory given to it through the shared_pool_size parameter. If the database does not have many object and code definitions to reference, then the shared pool can contain an abundance of free memory that is not being used.
Metrics
Under-allocating the shared pool size can have a serious impact on your database's performance, but over-allocating the shared pool can have run time ramifications as well. If you have a good chunk of memory allocated to the Oracle shared pool that is never used, it might be more of a performance enhancement to reduce the shared pool amount and instead give the memory to the buffer/data cache, or even back to the operating system itself. In terms of knowing when to reduce the shared pool, a good benchmark is continually seeing 2-3MB of free memory.

On the other hand, if after an hour or so of beginning database operation, you see that virtually no free memory is left in the shared pool, or you are seeing ORA-4031 errors (that indicate definitions cannot find enough contiguous free space in the shared pool), increase the pool by 10% or more.

Troubleshooting
If you continuously see little or no free memory in the shared pool, you can do the following:

If a problem is found in Oracle8i or earlier:

- Edit the Init.ora file for the database.
- Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later:

- Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

You can also use the ALTER SYSTEM FLUSHED SHARED_POOL command to remove all objects from the shared pool and start with a clean slate.

Buffer Busy Waits
Buffer busy waits occur when a process needs to access a data block in the buffer cache, but cannot because it is being used by another process. So it must wait. A wait event generally happens because a buffer is being read into the buffer cache by another process or the buffer is in the buffer cache, but cannot be switched to a compatible mode immediately.

Metrics
Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables.

Troubleshooting
On Oracle8i or earlier, the remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.

For Oracle9i or later, you can use the automatic segment management feature in Oracle9i locally-managed tablespaces to help make free list problems a thing of the past. Using an UNDO tablespace in 9i or later can help remedy any rollback contention problem.
You can also obtain which objects have actually experienced buffer busy waits in Oracle9i or later by querying the sys.v_$segment_statistics. This view is not populated unless the configuration parameter statistics_level is set to TYPICAL or ALL.

Free Buffer Wait Average
There are times when a user must wait to obtain a clean buffer in the buffer/data cache. The free buffer wait average provides a metric that communicates the average number of tree buffer waits.

**Metrics**
Seeing consistent non-zero numbers for the free buffer wait average might indicate an undersized data-buffer cache.

**Troubleshooting**
If a problem is found in Oracle8i or earlier:
- Edit the Init.ora file for the database.
- Increase the amount of db_block_buffers to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later:
- Increase the size of the db_cache_size parameter through use of the ALTER SYSTEM SET db_cache_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them when the database is stopped and re-started.

Object Reloads
When a SQL statement or code object (stored procedure, trigger, etc.) is executed for the first time, Oracle loads the code definition into the shared pool. If the statement or code object is executed again, Oracle can reuse the object definition in memory, and thereby shorten execution time.

If a code object is aged out of the shared pool, and then re-requested, Oracle must load the object back into the pool. Such activity can lead to reduced response times. The Object Reloads statistic provides you with a count of how many objects have been reloaded back into the shared pool.

**Metrics**
Seeing consistent non-zero numbers for object reloads could indicate a too small shared pool or that help identify that there are hot code objects that should be pinned in the shared pool so they cannot be removed.

**Troubleshooting**
If the Object Reloads statistic, along with other shared pool metrics, indicates an undersized shared pool, you can do the following depending on the version of Oracle you are running:

If a problem is found in Oracle8i or earlier:
• Edit the Init.ora file for the database.

• Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.

• Cycle the Oracle server when possible to allow the new value to take effect.

• Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later:

• Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.

• Monitor the new value to see if performance improves.

• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

If you suspect that there are hot code objects that would benefit from being pinned into the shared pool, then you can go into shared pool details using Performance Analyst’s monitors. You can easily spot the general code object types or specific objects that should be pinned in the pool from these monitors. Once identified, you can use the dbms_shared_pool package to pin the hot code objects into RAM.

Redo Log Space Waits

The Oracle database is able to manage recovery by recording all changes made to a database through the use of redo log files. Oracle writes modifications made to a database to these files that have the capability to be archived off to another medium for disaster recovery. The background process that performs these operations is Oracle’s Log Writer (LGWR). There is a buffer area in Oracle’s System Global Area (SGA) that is used to reduce redo log file I/O, whose size, or lack thereof, can affect performance in a busy system. Sometimes a user process must wait for space in this redo log buffer. Oracle uses the log buffer area to cache redo entries prior to writing them to disk, and if the buffer is not large enough for the redo entry load, waits can occur.

Metrics

The two main numbers to watch are the redo log space requests and the redo log wait time. If either statistic strays too far from zero, increase the log_buffer parameter and add more memory to the redo log buffer.

Troubleshooting

To fix, do the following:

1. Edit the Init.ora file or SPFILE for the database.

2. Increase the amount of log_buffer to a higher value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value.

3. Cycle the Oracle server when possible to allow the new value to take effect.

4. Monitor new value to see if performance improves.

On some Oracle platforms, when adjusting the log_buffer parameter, make sure you make the amount a multiple of the block size. Otherwise, on database startup, Oracle returns an error stating that you have entered an invalid amount for the redo log buffer.

NOTE: Oracle increases the log_buffer parameter if you make it smaller than its default size for a given platform.
Redo Log Space Wait Time
The Oracle database is able to manage recovery by recording all changes made to a database through the use of redo log files. Oracle writes modifications made to a database to these files that have the capability to be archived off to another medium for disaster recovery. The background process that performs these operations is Oracle's Log Writer (LGWR). There is a buffer area in Oracle's System Global Area (SGA) that is used to reduce redo log file I/O, whose size, or lack thereof, can affect performance in a busy system. Sometimes a user process must wait for space in this redo log buffer. Oracle uses the log buffer area to cache redo entries prior to writing them to disk, and if the buffer is not large enough for the redo entry load, waits can occur.

Metrics
The two main numbers to watch are the redo log space requests and the redo log wait time. If either statistic strays too far from zero, increase the log_buffer parameter and add more memory to the redo log buffer.

Troubleshooting
To fix, do the following:

1. Edit the Init.ora file or SPFILE for the database.
2. Increase the amount of log_buffer to a higher value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
3. Cycle the Oracle server when possible to allow the new value to take effect.
4. Monitor new value to see if performance improves.

On some Oracle platforms, when adjusting the log_buffer parameter, make sure you make the amount a multiple of the block size. Otherwise, on database startup, Oracle returns an error stating that you have entered an invalid amount for the redo log buffer.

NOTE: Oracle increases the log_buffer parameter if you make it smaller than its default size for a given platform.

Top Latch Misses
Latches protect the many memory structures in Oracle's SGA. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. More restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism.

The Top Latch Misses chart indicates the top latch waits as ranked by total misses.

Metrics
Seeing increasing numbers for the same latch miss can require further accessing session latch details, which can be accomplished inside Performance Analyst.

Troubleshooting
The remedy for latch misses depends on the latch itself. Buffer chain latches, for example, are indicative of high I/O rates in databases. Increasing the buffer/data cache can help as might adding more buffer LRU chain latches by increasing the db_block_lru_latches parameter.

Library cache latch misses are oftentimes produced by non-reused SQL in the shared pool. Increasing the shared_pool_size can help, but a better remedy might be changing SQL to use bind variables or setting the cursor_sharing parameter to FORCE (Oracle 8.1.6 and later).
Memory Detail
The following tabbed pages are available in the Memory Detail page:

<table>
<thead>
<tr>
<th>Buffer Cache</th>
<th>Buffer Pools</th>
<th>Dictionary Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latch Activity</td>
<td>Library Cache</td>
<td>Shared Pool</td>
</tr>
<tr>
<td>PGA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Buffer Cache Tab
The Buffer Cache tab includes the following sections:

<table>
<thead>
<tr>
<th>Buffer Cache Hit Ratio</th>
<th>Buffer Cache Utilization</th>
<th>Cached Blocks by Tablespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cached Tables</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Buffer Cache Hit Ratio
Data read from memory will produce end-user response times many times faster than when that same data is read from disk. Keeping physical I/O's to an absolute minimum is one of the purposes of the Oracle buffer/data cache.

The buffer cache hit ratio is a terrific indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. A number of key counters in Oracle are used to arrive at this statistic. The table below lists them:

<table>
<thead>
<tr>
<th>Counter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB BLOCK GETS</td>
<td>Data read from memory for DML operations.</td>
</tr>
<tr>
<td>CONSISTENT GETS</td>
<td>Data read from rollback segments in memory.</td>
</tr>
<tr>
<td>PHYSICAL READS</td>
<td>Data read physically from disk.</td>
</tr>
<tr>
<td>DIRECT READS</td>
<td>Data read physically from disk that bypasses the buffer cache. Direct reads are filtered out of overall physical reads so an accurate cache hit ratio can be determined.</td>
</tr>
</tbody>
</table>

Dividing the data read from memory by data read from disk yields the cache hit ratio.

Metrics
To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, you should be aware that every database has its own 'personality' and may exhibit excellent performance with below average readings for the cache hit ratio. You should also be aware that excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

Consistently viewed low readings (60% or less) may require tuning attention on the part of the DBA.

For Oracle8i or earlier, the adjustment of the db_blockBuffers tuning parameter is required. For Oracle9i and later, the db_cache_size parameter is the parameter that needs attention.

**NOTE:** For any increases in db_blockBuffers to take effect on Oracle8i or earlier, the database must be cycled. The db_cache_size parameter in Oracle9i and later, however, is dynamic and may be altered without stopping and starting the database instance.

Troubleshooting
If a problem is found in Oracle8i or earlier, then:

- Edit the INIT.ORA file for the database.
• Increase the amount of db_block_buffers to a higher value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).

• Cycle the Oracle server when possible to allow the new value to take effect.

• Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later, then:

• Increase the size of the db_cache_size parameter through use of the ALTER SYSTEM SET db_cache_size command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).

• Monitor the new value to see if performance improves.

• If using an SPFILE, save the new configuration values so Oracle will reuse them each time the database is stopped and re-started.

Cached Blocks by Tablespace
Oracle’s buffer/data cache contains blocks read in from disk at the user’s request. The Cached Blocks by Tablespace section displays the amount of data held currently in the cache by tablespace.

Metrics
None.

Cached Tables
One of the ways a DBA can help data from a table stay in the buffer/data cache as long as possible is to enable the table’s CACHE property. CACHE specifies that the blocks retrieved for a table be placed at the most recently used end of the LRU list in the buffer cache when a full table scan is performed. This is particularly useful for small lookup tables.

The Cached Tables section displays the data for all tables that have their CACHE property enabled. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>The owner of the table.</td>
</tr>
<tr>
<td>Table Name</td>
<td>The name of the table.</td>
</tr>
<tr>
<td>Table Size</td>
<td>The size of the table in bytes.</td>
</tr>
</tbody>
</table>

Metrics
CACHE is best suited for small lookup tables. Therefore, if you see a large table that has been cached, you may want to reevaluate whether it is smart to cache the contents of that table in Oracle’s buffer/data cache.

Troubleshooting
If you wish to disable the CACHE property of a table, you can alter the table and set the property to false.

Buffer Cache Utilization
The Buffer Cache Utilization section shows the state and use of the buffers currently in the buffer cache along with a count of how many there are for each state. Typically, the statuses of the buffers are: being read, free, read and modified, and read and not modified.
Metrics
This grouping of statistics is quite helpful in determining if you have an overabundance of block buffers allocated. Consistently seeing large number of free buffers will clue you in to the fact that you can reduce the amount of block buffers in the cache and give memory back to the operating system.

If, however, you see no free buffers within the first hour of bringing your Oracle database up, then you may consider adding more buffers to the cache.

Troubleshooting
If a problem is found, then:

1. Edit the Init.ora file or spfile for the database.
2. Increase the amount of db_block_buffers (Oracle 8i and earlier) or db_cache_size (Oracle9i and later) to a higher value if free buffers are not found (take caution to not over-allocate; ensure enough free memory exists on server before increasing value). Reduce the number of an overabundance of FREE buffers are present.
3. Cycle the Oracle server (for Oracle 8i and earlier) when possible to allow the new value to take effect.
4. Monitor new value to see if more numbers of free buffers show up for the FREE status.

Dictionary Cache Tab

Oracle’s data dictionary is a component of the shared pool that contains system elements necessary for the parsing of SQL statements, security resolution, object definitions, and more.

The overall data dictionary cache hit ratio provides a key indicator in determining the performance of the shared pool, and shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much as possible of this vital reference information be kept in RAM.

The Dictionary Cache tab shows the individual elements of the data dictionary along with their associated hit ratios. In versions 6.x of Oracle, these individual elements could be tuned, but in versions 7.x and later, the only main method for tuning them involves the adjustment of the entire shared pool setting. Although not tunable from an individual parameter level, each displayed element gives insight into which area of the data dictionary is either adding to or detracting from overall performance.

The table below describes the information available on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>The name of the individual data dictionary element.</td>
</tr>
<tr>
<td>Usage</td>
<td>The number of cache entries that contain valid data.</td>
</tr>
<tr>
<td>Gets</td>
<td>The number of requests for this element.</td>
</tr>
<tr>
<td>Get Misses</td>
<td>The number of requests resulting in a cache miss.</td>
</tr>
<tr>
<td>Hit Ratio</td>
<td>The ratio of cache hits versus misses of total requests. The maximum is 100%.</td>
</tr>
<tr>
<td>Scans</td>
<td>The number of scan requests.</td>
</tr>
<tr>
<td>Scan Misses</td>
<td>The number of times that a scan failed to find the needed data in the cache.</td>
</tr>
<tr>
<td>Scan Completes</td>
<td>The number of times the list was scanned completely.</td>
</tr>
<tr>
<td>Modifications</td>
<td>The number of insert, update, and delete actions.</td>
</tr>
<tr>
<td>Flushes</td>
<td>The number of disk flushes.</td>
</tr>
</tbody>
</table>
Metrics
An overall high data dictionary cache hit ratio is desirable, as are high hit ratios in each individual parameter. You should strive for a hit ratio between 90-100%, with 95% being a good performance benchmark.

**NOTE:** When a database is first started, the data dictionary cache hit ratio is not at an optimal level because all references to object definitions are relatively new, and as such, must be placed into the shared pool. Look for hit ratios between 80-90% for new database startups. If, however, after an hour or two of steady database time, the data dictionary cache hit ratio and individual hit ratios, have not increased to desirable levels, you should look into the possibility of increasing the `shared_pool_size` parameter.

**NOTE:** Note that databases supporting applications that involve large number of objects (such as an Oracle Financials installation) should have larger than normal shared pools to support the required object definitions.

Buffer Pools Tab
The Buffer Pools tab includes the following sections:

- **Buffer Pool Hit Ratio**
- **Buffer Pool Memory Configuration**
- **Buffer Pool Object Assignments**

Buffer Pool Hit Ratio
**Metrics**
Because data objects can be referenced with different usage patterns, Oracle8 and later offers the option to intelligently place objects into one of three buffer caches. The table below describes the types of buffer caches:

<table>
<thead>
<tr>
<th>Cache</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEEP</td>
<td>Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.</td>
</tr>
<tr>
<td>RECYCLE</td>
<td>Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>The traditional cache for all other data objects.</td>
</tr>
</tbody>
</table>

**NOTE:** Unless you specify the KEEP or RECYCLE buffer cache, Oracle automatically places objects into the DEFAULT buffer cache.

The Buffer Pool Hit Ratio section the hit ratios for each of the Oracle8 and later buffer caches so you can easily see how often the objects placed into the various caches are being referenced in memory. Examining how often data is satisfied from memory vs. disk will help you determine if the caches are large enough and if they are being used in an optimal manner. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer pool name</td>
<td>The name of the Oracle buffer pool.</td>
</tr>
<tr>
<td>Buffer pool hit ratio</td>
<td>The overall hit ratio for the particular cache.</td>
</tr>
</tbody>
</table>

**NOTE:** The DEFAULT buffer cache will only be shown for those installations not using the specialized caches available in Oracle8 and later.
Metrics
The KEEP buffer pool should maintain a hit ratio as close to 100% as possible. However, the buffer pool hit ratio is not near 100% until the database has been up and running in a typical state for a short time.

A poor hit ratio for the RECYCLE buffer pool may not be a bad thing since there is little chance of reusing a block stored in the buffer pool before it is aged out.

**NOTE:** If you place objects into the KEEP buffer pool, you should periodically reexamine their object sizes to ensure that they are not growing to a physical state that will jeopardize the performance of the KEEP pool.

Buffer Pool Object Assignments
Because data objects can be referenced with different types of usage patterns, Oracle8 and later offers the option to intelligently place objects into one of three buffer caches. The table below describes these three buffer caches:

<table>
<thead>
<tr>
<th>Cache</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEEP</td>
<td>Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.</td>
</tr>
<tr>
<td>RECYCLE</td>
<td>Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>The traditional cache for all other data objects.</td>
</tr>
</tbody>
</table>

Objects not specified to go into either the KEEP or RECYCLE caches will automatically be placed into the DEFAULT buffer cache.

The Buffer Pool Object Assignments section provides a summary with respect to which types of objects have been assigned to the various buffer caches. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Type</td>
<td>The type of object (table, index, etc.).</td>
</tr>
<tr>
<td>Buffer Pool</td>
<td>The buffer cache name (KEEP, etc.).</td>
</tr>
<tr>
<td>Total Objects Assigned</td>
<td>The number of objects that have been assigned to the particular cache.</td>
</tr>
</tbody>
</table>

Metrics
When looking at overall database I/O activity, you should keep an eye out for objects that may lend themselves to being placed into a particular buffer cache. Consider using a KEEP cache for relatively small, frequently accessed tables that require fast response times. Large tables with random I/O activity and are scanned from beginning to end a lot are good candidates for a RECYCLE cache.

Buffer Pool Memory Configuration
Because data objects can be referenced with different types of usage patterns, Oracle8 and later offers the option to intelligently place objects into one of three buffer caches. The table below describes these three buffer caches:

<table>
<thead>
<tr>
<th>Cache</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEEP</td>
<td>Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.</td>
</tr>
</tbody>
</table>
Objects not specified to go into either the KEEP or RECYCLE caches will automatically be placed into the DEFAULT buffer cache.

### Metrics

When looking at overall database I/O activity, you should keep an eye out for objects that may lend themselves to being placed into a particular buffer cache. Consider using a KEEP cache for relatively small, frequently accessed tables that require fast response times. Large tables with random I/O activity and are scanned from beginning to end a lot are good candidates for a RECYCLE cache.

### Troubleshooting

Objects can be placed into different buffer pools at object creation time (using the STORAGE...BUFFER_POOL option) or existing objects can be set into a different pool with the ALTER command. Note that tables, partitions, and indexes may be placed into the different caches.

If you just want to use the DEFAULT buffer pool and not enable any special caches, you can still encourage Oracle to keep certain objects in the buffer cache as long as possible using the CACHE parameter. For example, issuing the command `ALTER TABLE ... CACHE` specifies that the blocks retrieved for this table be placed at the most recently used end of the LRU list in the DEFAULT buffer cache when a full table scan is performed. The CACHE hint may also be used in SQL statements to cache a table, but used in this form, the blocks will only be cached until the next time the database is shut down. Once the database comes back up, the CACHE hint would have to be issued in a SQL statement again to cache the needed blocks of data.

### Latch Activity Tab

The Latch Activity tab includes the following sections:

<table>
<thead>
<tr>
<th>Cache</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECYCLE</td>
<td>Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>The traditional cache for all other data objects.</td>
</tr>
</tbody>
</table>

## Latch Hit Ratio

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry. The latch hit ratio defines the number of times a process obtained a willing-to-wait.

### Metrics

If the latch hit ratio falls below 99%, then action should be taken to resolve the amount of latch contention occurring.
Troubleshooting
The details regarding the latch contention should be examined. Increasing the shared_pool_size can assist in latch problems also. There are a few specialized cases of latch contention that can be rectified pretty easily. The table below describes them:

<table>
<thead>
<tr>
<th>Contention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache buffer chain latch</td>
<td>This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers/db_cache_size parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.</td>
</tr>
<tr>
<td>Library cache latches</td>
<td>Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.</td>
</tr>
</tbody>
</table>

Other routine latch contention problems used to include the redo allocation and redo copy latches, but these have pretty much been made obsolete in Oracle 8.1.5 and later.

Immediate Latch Hit Ratio
Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry. The latch immediate miss ratio defines the number of times a process obtained a not-willing-to-wait latch.

Metrics
If the immediate latch hit ratio falls below 99%, then action should be taken to resolve the amount of latch contention occurring.

Troubleshooting
The details regarding the latch contention should be examined. Increasing the shared_pool_size can assist in latch problems also. There are a few specialized cases of latch contention that can be rectified pretty easily. The table below describes them:

<table>
<thead>
<tr>
<th>Contention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache buffer chain latch</td>
<td>This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers/db_cache_size parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.</td>
</tr>
<tr>
<td>Library cache latches</td>
<td>Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.</td>
</tr>
</tbody>
</table>

Other routine latch contention problems used to include the redo allocation and redo copy latches, but these have pretty much been made obsolete in Oracle 8.1.5 and later.
Latch Sleep Ratio
Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry.

A sleep indicates that a latch could not be obtained for a process, and that the attempt will be retried. A low ratio indicates many processes that had to sleep multiple times before obtaining a requested latch.

Metrics
Sleeps should be kept as low as possible. If the overall sleep ratio falls below 99%, then action should be taken to resolve the amount of latch contention that is occurring.

Troubleshooting
The details regarding the latch contention should be examined. Increasing the shared_pool_size can assist in latch problems also. There are a few specialized cases of latch contention that can be rectified pretty easily. The table below describes them:

<table>
<thead>
<tr>
<th>Contention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache buffer chain latch</td>
<td>This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers/db_cache_size parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.</td>
</tr>
<tr>
<td>Library cache latches</td>
<td>Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.</td>
</tr>
</tbody>
</table>

Other routine latch contention problems used to include the redo allocation and redo copy latches, but these have pretty much been made obsolete in Oracle 8.1.5 and later.

Latch Detail
Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry.

The Latch Detail section presents a detailed view of latch activity. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latch Name</td>
<td>The name of the latch.</td>
</tr>
<tr>
<td>Gets</td>
<td>The number of times the latch was requested by a process.</td>
</tr>
<tr>
<td>Misses</td>
<td>The number of failed attempts to acquire the latch on the first attempt.</td>
</tr>
<tr>
<td>Immediate Gets</td>
<td>The number of nowait requests for a latch.</td>
</tr>
<tr>
<td>Immediate Misses</td>
<td>The number of failed nowait attempts to acquire the latch on the first attempt.</td>
</tr>
<tr>
<td>Sleeps</td>
<td>The number of requests that paused while waiting for a latch.</td>
</tr>
</tbody>
</table>
Metrics

Common indicators of latch contention are a latch miss ratio (which records willing-to-wait mode latch requests) and latch immediate miss ratio (which records no-wait mode latch requests). These statistics reflect the overall health of how often latch requests were made and satisfied without waiting. If either of these exceed 1%, then latch contention may be causing system slowdowns.

There are a few latch contention situations that you can recognize and get to the bottom of quickly. The table below describes them:

<table>
<thead>
<tr>
<th>Contention Situation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache buffer chain latch</td>
<td>This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.</td>
</tr>
<tr>
<td>Library cache latches</td>
<td>This latch protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.</td>
</tr>
</tbody>
</table>

Other routine latch contention problems used to include the redo allocation and redo copy latches, but these have pretty much been made obsolete in Oracle 8.1.5 and later.

Library Cache Tab

The Library Cache tab includes the following sections:

Library Cache Hit Ratio
Library Cache Performance
Pinned Objects

For more information, see Memory Home Page.

Library Cache Hit Ratio

Metrics

Oracle's shared pool is made up of two main memory areas - the Library Cache and the Data Dictionary Cache, and offers a number of tuning possibilities.

The library cache holds commonly used SQL statements - basically database code objects. A method for improving performance in Oracle is to encourage the reuse of SQL statements so expensive parse operations may be avoided. The library cache assists this tuning effort.

The library cache hit ratio offers a key indicator in determining the performance of the shared pool. It shows how often SQL code is being reused by other database users vs. the number of times a SQL statement is broken down, parsed, and then loaded (or reloaded) into the shared pool.

**NOTE:** This statistic is also available on the Home page, the Memory home page and on the Library Cache tab of the Memory Detail.

Metrics

A high library cache hit ratio is a desirable thing. You should strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse. Note that when a database is first started, the library cache hit ratio will not be at an optimal level because all code being used will be relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, you should look into the possibility of increasing the shared_pool_size parameter.

Other red flags that can indicate a too small shared pool include:
• A wait count for the event 'latch free' of ten or greater
• The library cache wait count of two or greater.

These indicators can be tracked with Performance Analyst's Bottleneck and Wait detail views.

A way of improving the library cache hit ratio is by encouraging code reuse through the implementation of bind variables. Discouraging hard coded literals in application code and instead making use of variables bound at run time aids in the reuse of SQL code that is maintained in Oracle's shared pool. Note that bind variables can have an affect on the cost-based optimizer though.

A second way is to pin frequently used code objects in memory so they will always be there when needed. This can be accomplished through the use of the system supplied DBMS_SHARED_POOL package. You can use Performance Analyst to view objects in the shared pool that are always present and/or have increasing reload numbers to help identify objects that are good candidates for pinning.

Troubleshooting
If a problem is found in Oracle8i or earlier, then:

• Edit the Init.ora file for the database.
• Increase the amount of shared_pool_size to a higher value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).
• Cycle the Oracle server when possible to allow the new value to take effect.
• Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later, then:

• Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).
• Monitor the new value to see if performance improves.
• If using an SPFILE, save the new configuration values so Oracle will reuse them each time the database is stopped and re-started.

If you determine that SQL literals are causing SQL to not be reused, you can (in Oracle 8.1.6 and later):

• Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.
• Monitor database to see if parse activity is reduced and library cache reloads shrink.
• If using an SPFILE, save the new configuration values so Oracle will reuse them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.

Library Cache Performance
The library cache holds commonly used SQL statements - basically database code objects. A method for improving performance in Oracle is to encourage the reuse of SQL statements so expensive parse operations may be avoided. The library cache assists this tuning effort.

The Library Performance sectioned tab provides insight into how efficiently the library cache is operating. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Namespace</td>
<td>The region of the library cache.</td>
</tr>
</tbody>
</table>
An overall high library cache hit ratio is a desirable thing. You should strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse. When a database is first started, the library cache hit ratio, along with the individual region hit ratios, will not be at an optimal level because all code being used will be relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, you should look into the possibility of increasing the shared_pool_size parameter.

To keep important code objects from being aged out of the library cache, you can use the DBMS_SHARED_POOL package to pin frequently used code objects in memory so they will always be there when needed.

### Pinned Objects

To keep important code objects from being aged out of the library cache, you can use the DBMS_SHARED_POOL package to pin frequently used code objects in memory so they will always be there when needed. The Pinned Objects section displays all objects in the library cache that have pinned. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>The user account that owns the object.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of the object: INDEX, TABLE, CLUSTER, VIEW, SET, SYNONYM, SEQUENCE,</td>
</tr>
<tr>
<td></td>
<td>PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, TRIGGER, CLASS, OBJECT, USER,</td>
</tr>
<tr>
<td></td>
<td>DBLINK.</td>
</tr>
<tr>
<td>Shareable Memory</td>
<td>The amount of memory consumed by the object in the shared pool.</td>
</tr>
<tr>
<td>Loads</td>
<td>The number of times the object has been loaded into the cache. Note that this count also increases when an object has been invalidated.</td>
</tr>
<tr>
<td>Executions</td>
<td>The number of times the object has been executed by a session thread.</td>
</tr>
<tr>
<td>Locks</td>
<td>The number of users actively locking the object.</td>
</tr>
<tr>
<td>Pins</td>
<td>The number of user actively pinning the object.</td>
</tr>
<tr>
<td>Pinned?</td>
<td>Indicates (YES or NO) if this object has been pinned in memory with the DBMS_SHARED_POOL package.</td>
</tr>
</tbody>
</table>

To keep important code objects from being aged out of the library cache, you can use the DBMS_SHARED_POOL package to pin frequently used code objects in memory so they will always be there when needed.

### Metrics

**NOTE:** An overall high library cache hit ratio is a desirable thing. You should strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse. When a database is first started, the library cache hit ratio, along with the individual region hit ratios, will not be at an optimal level because all code being used will be relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, you should look into the possibility of increasing the shared_pool_size parameter.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets</td>
<td>The number of times a lock was requested for objects in the particular namespace.</td>
</tr>
<tr>
<td>Get Hit Ratio</td>
<td>The percentage of times (with 100% being the maximum) that the object was found in the cache.</td>
</tr>
<tr>
<td>Pins</td>
<td>The number of times a pin was requested for objects of this namespace.</td>
</tr>
<tr>
<td>Pin Hit Ratio</td>
<td>The percentage of times (with 100% being the maximum) that pin requests were successful.</td>
</tr>
<tr>
<td>Reloads</td>
<td>The number of times a piece of the object had to be brought back in from disk to the cache, most likely because it was flushed from the shared pool.</td>
</tr>
<tr>
<td>Invalidations</td>
<td>The number of times objects in this namespace were marked invalid because a dependent object was modified.</td>
</tr>
</tbody>
</table>
Shared Pool Tab
The main components of the shared pool memory are the library and dictionary caches. Many of the caches in the shared pool increase or decrease in size automatically as necessary. The Shared Pool tab of the Memory Detail view presents the following sections:

<table>
<thead>
<tr>
<th>Shared Space Pool Utilization</th>
<th>Shared Pool Fragmentation</th>
<th>Shared Pool Usage</th>
</tr>
</thead>
</table>

Shared Space Pool Utilization
Here you see a graphical representation of the amount of used and free memory.

Shared Pool Fragmentation
The Shared Pool Fragmentation section displays the following statistics:
- **Bucket**: A work area defined by its optimal memory requirement.
- **Free Space**: The amount of free space in the bucket.
- **Free Chunks**: The number of free chunks that are available to be allocated from the shared pool.
- **Average Size**: The average size of a chunk.
- **Largest**: The size of the largest chunk.

Shared Pool Usage
The Shared Pool Usage section shows the following statistics:
- **Shared Pool Component Name**
- **Amount Used (bytes)**
- **Percent Used**

PGA Tab
The PGA tab displays statistical measures and techniques that let you manage a session's Program Global Area (PGA).

**NOTE**: This information is available for Oracle9i and later.

The PGA tab includes the following sections:

<table>
<thead>
<tr>
<th>PGA Overview</th>
<th>PGA Session Detail</th>
<th>PGA Workarea Summary</th>
</tr>
</thead>
</table>

For more information, see Memory Home Page.

PGA Overview
Oracle's Program Global Area (PGA) is a private memory region containing data and control information for every connected process. Examples of PGA information are the runtime area of a cursor and memory sort space. If shared servers are used, then part of the runtime area can be located in the SGA.

Intensive queries cause a big portion of the PGA to be dedicated to work areas supporting memory intensive operators, such as the following:

- Sort-based operators (for example, ORDER BY, GROUP BY, ROLLUP, window functions)
• Hash-join
• Bitmap merge
• Bitmap create
• Write buffers used by bulk load operations

A sort operator uses a sort area to perform the in-memory sort of a set of rows. Similarly, a hash-join operator uses a hash area to build a hash table.

NOTE: This information is available for Oracle9i and later.

Metrics
The size of a PGA area can be controlled and tuned. Larger work areas can significantly improve performance of session work, with the trade off being higher memory consumption. The size of a session’s PGA should be big enough so that it can accommodate the input data and auxiliary memory structures allocated by its requested SQL actions. This is known as the optimal size of a PGA work area. When the size of a PGA work area is smaller than optimal, the response time increases, because an extra pass is performed over part of the input data. This is known as the one-pass size of the PGA work area. Under the one-pass threshold, when the size of a work area is smaller than its input data size, multiple passes over the input data are needed. This could drastically increase the response time of the session. This is known as the multi-pass size of the work area.

The goal is to have most work areas running with an optimal size (more than 90% or more for OLTP systems), while a smaller amount of them are running with a one-pass size (less than 10%). Multi-pass execution should be completely avoided.

Prior to Oracle9i, the PGA was controlled using the SORT_AREA_SIZE, HASH_AREA_SIZE, BITMAP_MERGE_AREA_SIZE and CREATE_BITMAP_AREA_SIZE parameters. Setting these parameters is difficult, because the maximum PGA size is ideally selected based on the data input size and the total number of work areas active in the system. These two factors vary a lot from one work area to another and from one point in time to another.

Troubleshooting
With Oracle9i, you can simplify and improve the way PGA memory is allocated, by enabling automatic PGA memory management. In this mode, Oracle dynamically adjusts the size of the portion of the PGA memory dedicated to work areas, based on an overall PGA memory target explicitly set by the DBA. To enable automatic PGA memory management, you have to set the initialization parameter PGA_AGGREGATE_TARGET.

The table below describes two metrics you should to watch:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Over allocation count</td>
<td>Over-allocating PGA memory can happen if the value of PGA_AGGREGATE_TARGET is too small to accommodate the PGA other component in the previous equation plus the minimum memory required to execute the work area workload. When this happens, Oracle cannot honor the initialization parameter PGA_AGGREGATE_TARGET, and extra PGA memory needs to be allocated. If over-allocation occurs, you should increase the value of PGA_AGGREGATE_TARGET.</td>
</tr>
<tr>
<td>Cache hit percentage</td>
<td>This metric is computed by Oracle to reflect the performance of the PGA memory component. A value of 100% means that all work areas executed by the system since instance start-up have used an optimal amount of PGA memory. When a work area cannot run optimally, one or more extra passes is performed over the input data. This reduces the cache hit percentage in proportion to the size of the input data and the number of extra passes performed.</td>
</tr>
</tbody>
</table>
PGA Workarea Summary
Oracle's Program Global Area (PGA) is a private memory region containing data and control information for every connected process. Examples of PGA information are the runtime area of a cursor and memory sort space. If shared servers are used, then part of the runtime area can be located in the SGA.

Intensive queries cause a big portion of the PGA to be dedicated to work areas supporting memory intensive operators, such as the following:

- Sort-based operators (for example, ORDER BY, GROUP BY, ROLLUP, window functions)
- Hash-join
- Bitmap merge
- Bitmap create
- Write buffers used by bulk load operations

A sort operator uses a sort area to perform the in-memory sort of a set of rows. Similarly, a hash-join operator uses a hash area to build a hash table.

**NOTE:** This information is available for Oracle9i and later.

Metrics
The size of a PGA area can be controlled and tuned. Larger work areas can significantly improve performance of session work, with the trade off being higher memory consumption. The size of a session's PGA should be big enough so that it can accommodate the input data and auxiliary memory structures allocated by its requested SQL actions. This is known as the optimal size of a PGA work area. When the size of a PGA work area is smaller than optimal, the response time increases, because an extra pass is performed over part of the input data. This is known as the one-pass size of the PGA work area. Under the one-pass threshold, when the size of a work area is smaller than its input data size, multiple passes over the input data are needed. This could drastically increase the response time of the session. This is known as the multi-pass size of the work area.

The goal is to have most work areas running with an optimal size (more than 90% or more for OLTP systems), while a smaller amount of them are running with a one-pass size (less than 10%). Multi-pass execution should be completely avoided.

Prior to Oracle9i, the PGA was controlled using the SORT_AREA_SIZE, HASH_AREA_SIZE, BITMAP_MERGE_AREA_SIZE and CREATE_BITMAP_AREA_SIZE parameters. Setting these parameters is difficult, because the maximum PGA size is ideally selected based on the data input size and the total number of work areas active in the system. These two factors vary a lot from one work area to another and from one point in time to another.

Troubleshooting
With Oracle9i, you can simplify and improve the way PGA memory is allocated, by enabling automatic PGA memory management. In this mode, Oracle dynamically adjusts the size of the portion of the PGA memory dedicated to work areas, based on an overall PGA memory target explicitly set by the DBA. To enable automatic PGA memory management, you have to set the initialization parameter PGA_AGGREGATE_TARGET.

If multi-pass percentages are consistently non-zero, you should increase the size of PGA_AGGREGATE_TARGET.

PGA Session Detail
Oracle's Program Global Area (PGA) is a private memory region containing data and control information for every connected process. Examples of PGA information are the runtime area of a cursor and memory sort space. If shared servers are used, then part of the runtime area can be located in the SGA.

Intensive queries cause a big portion of the PGA to be dedicated to work areas supporting memory intensive operators, such as the following:
• Sort-based operators (for example, ORDER BY, GROUP BY, ROLLUP, window functions)
• Hash-join
• Bitmap merge
• Bitmap create
• Write buffers used by bulk load operations

A sort operator uses a sort area to perform the in-memory sort of a set of rows. Similarly, a hash-join operator uses a hash area to build a hash table.

The PGA Session Detail section displays PGA usage for all connected sessions. The table below describes the information in this section:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>The unique identifier given the session by Oracle.</td>
</tr>
<tr>
<td>Username</td>
<td>The user account a session is using.</td>
</tr>
<tr>
<td>O/S ID</td>
<td>The operating system ID of the process.</td>
</tr>
<tr>
<td>Logon Time</td>
<td>The time the session logged into Oracle.</td>
</tr>
<tr>
<td>Machine Name</td>
<td>The name of the machine running the process.</td>
</tr>
<tr>
<td>PGA Alloc Memory</td>
<td>The amount of allocated PGA memory for the process.</td>
</tr>
<tr>
<td>PGA Used Memory</td>
<td>The amount of freeable or usable memory for the process.</td>
</tr>
<tr>
<td>PGA Max Memory</td>
<td>The maximum amount of PGA memory used by the process.</td>
</tr>
</tbody>
</table>

**NOTE:** This information is available for Oracle9i and later.

### I/O Page Statistics

The I/O home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key Systems Analysis</th>
<th>I/O Access Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Analysis</td>
<td>Workload Analysis - Top I/O Hogs</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

* I/O Detail
* Home Page Statistics
* Memory Page Statistics
* Objects Page Statistics
* OS Page Statistics
* Space Page Statistics
* Users Page Statistics
Key Systems Analysis - I/O

When complaints begin to surface about your database's performance, oftentimes the root cause can be traced to one or more issues with I/O. The thing to keep in mind when you begin to monitor the I/O of your database is that you are actually reviewing the success of your physical design model.

All the physical storage characteristics and placements, the table and index designs, and the speed with which it all works are on display when I/O is monitored. Because a database's main index of performance is measured by how fast I/O needs are satisfied, it is your responsibility to quickly interrogate Oracle to determine if a reported database slowdown is I/O related.

The following key I/O values are reported in Key Numerical Analysis:

<table>
<thead>
<tr>
<th>Consistent Reads</th>
<th>Logical Changes</th>
<th>Logical Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Reads</td>
<td>Physical Writes</td>
<td>Redo Writes</td>
</tr>
</tbody>
</table>

For more information, see I/O Page Statistics.

Bottleneck Analysis - I/O

When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting can mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. You can use Bottleneck Analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps you track where a database has been spending its time. If heavy table scan activity has been dragging a database’s performance down, you can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, you can discover a fair amount of detail about which sessions and objects are causing the problem.

For wait event analysis to be properly carried out, it is imperative that the timed_statistics initialization parameter be set to TRUE. By default this parameter is set to FALSE, which disallows the collection of wait times for each wait event defined to the Oracle engine. For one to really understand the impact of wait events on database performance, you need to not only discover what the database is or has been waiting on, but the durations of the waits. Having both allows a complete picture to be formed regarding the magnitude of wait-initiated performance degradations. Almost all Oracle experts now agree that allowing the collection of timing statistics adds little if anything to database overhead, so setting timed_statistics to TRUE should not be a worry. The Performance Analyst I/O home page identifies the top database and log file waits that can cause decreased performance.

The Bottleneck Analysis section includes:

| Hottest Database Files | Top Database File Waits | Top Log File Waits |

For more information, see I/O Page Statistics.

SQL Analysis - I/O

Much of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times. SQL statements that use much memory can also cause a problem in a database.
Before you can identify problem SQL in your database, you have to ask the question, “What is bad SQL?” What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The SQL Analysis for I/O shows what SQL statements have consumed the largest percentages of physical and logical I/O, sort activity, and rows processed. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Response (Elapsed) Time</td>
<td>The amount of time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i or later.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The amount of CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.</td>
</tr>
<tr>
<td>Physical I/O</td>
<td>This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statement's logical I/O as well.</td>
</tr>
<tr>
<td>Logical I/O</td>
<td>This is a measure of how many memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.</td>
</tr>
<tr>
<td>Repetition</td>
<td>This is a measure of how often the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute may still be causing a headache on your system if it has executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over again.</td>
</tr>
</tbody>
</table>

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

Fortunately, Oracle records all the above measures (some only in 9i), which makes tracking the SQL that has been submitted against an Oracle database much easier.

**Metrics**
When you begin to look for inefficient SQL in a database, there are two primary questions you want answered:

1. **What HAS been the worst SQL that has historically been run in my database?**
2. **What IS the worst SQL that is running right now in my database?**

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It could be that the query is in an inefficient PL/SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can go further into Performance Analyst's Top SQL view and, if you have Embarcadero SQL Tuner installed, you can port the SQL over to SQL Tuner to better optimize the statement.
I/O Access Patterns - I/O
There are many different routes that Oracle can take to access various objects contained in a database. The I/O Access Patterns section summarizes the methods Oracle is using to satisfy end-user requests for data.

Metrics
Long table scans are typically an activity to avoid as they have the capability to cause needless physical and logical I/O as well as flood the buffer cache with seldom referenced blocks of data. You can discover which sessions have been causing the most large table scans.

The table fetch continued row statistic is indicative of chained/migrated row I/O. Such activity is not desired because chained/migrated row access can cause twice the I/O needed to access a table. Oracle must pick do two or more I/Os to read a chained/migrated row in a table. You can discover which sessions have been accessing tables with chained/migrated rows.

For more information, see I/O Page Statistics.

Workload Analysis - Top I/O Hogs
It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. The Top I/O Hogs table shows the top sessions that have caused the most physical I/O on the system (both reads and writes).

Metrics
If you see one or two users who have caused more than 25-50% of the total memory usage, go further into the sessions to see the activities they are performing.

For more information, see I/O Page Statistics.

Physical I/O

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Reads</td>
<td>Physical Reads is the total number of physical reads performed on all datafiles since the last refresh.</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>Physical Writes is the total number of times the DBWR process has performed writes to various database datafiles since the last refresh.</td>
</tr>
<tr>
<td>Redo Writes</td>
<td>Oracle's redo logs are hotbeds of I/O activity in databases that store heavily modified data. Redo log files are used to perform database recovery in the event of a system crash. Redo logs are written to in a cyclical fashion - each log file is filled up before Oracle moves on to the next file. The redo writes statistic reflects the total number of redo writes by the LGWR process since the last refresh.</td>
</tr>
</tbody>
</table>
Metrics
The table below describes metrics for Physical I/O statistics:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Reads</td>
<td>Large numbers of physical reads could reflect a too small data/buffer cache. The buffer cache hit ratio is a better indicator of overall logical vs. physical I/O.</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>Wait events related to I/O activity are good indicators of physical I/O problems. These events include db file parallel write and db file single write.</td>
</tr>
</tbody>
</table>

Troubleshooting
Doing the following can negate large numbers of continuous physical reads:

- Increasing the size of the data/buffer cache.
- Pinning often-referenced objects in memory by using the KEEP buffer pool (Oracle 8 and higher.)
- Placing heavily scanned objects in larger blocksize tablespaces (16-32KB). For Oracle9i or later.
- Tune SQL statements for better efficiency.

Logical I/O
- Metrics
- Troubleshooting

Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. You can also use other processes like the checkpoint and archive processes (CKPT and ARCH). Embarcadero Performance Analyst shows three key indicators of logical I/O. The table below describes these indicators:

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Reads</td>
<td>Logical Reads is the total number of db block gets and consistent gets (data read from memory) since the last refresh.</td>
</tr>
<tr>
<td>Logical Changes</td>
<td>Logical Changes is the total number of changes that were made to all blocks in the SGA that were part of an update or delete operation. These changes generate redo log entries and are permanent if the transaction is committed. The number of logical changes is an approximate indication of total database work.</td>
</tr>
<tr>
<td>Consistent Reads</td>
<td>Consistent Reads is the total number of times a consistent read was requested for a database block. Such a read is performed from Oracle’s rollback segments.</td>
</tr>
</tbody>
</table>

Metrics
Regarding raw logical I/O counts, no hard-core metrics exist. However, because physical I/O takes longer to complete than logical (memory) I/O, you should minimize physical read operations when possible. The buffer cache hit ratio is a better indicator of overall logical vs. physical I/O.

Troubleshooting
While logical I/O is still up to 1,400 times faster than physical disk access, it would be wise to investigate the top logical I/O process using Performance Analyst and see what SQL it is executing. If one process on the system is consuming between 25-50% of the overall amount, their SQL might require tuning.
Top Database File Waits
When viewing wait statistics, there are many levels of detail. The first level is the system view, which provides a
global, cumulative snapshot of all the waits that have occurred on a system. The Top Database File Waits section
filters all system wait events to present waits that are associated with database file activity. It ranks the top events by
the count of total waits.

Metrics
None.

Troubleshooting
Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in
Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based
event metrics.

The most common database file wait is a db file scattered read, which indicates table scan activity. Small table scans
are normally not a problem as Oracle can access data from small tables by quickly caching and scanning them. Large
tables are another story. Scans of large tables can be confirmed by examining the I/O access pattern metrics on the
Performance Analyst I/O home page. Information is also available on which sessions are causing the large table
scans.

Finally, one thing to watch is not only the actual waits, but also the wait time in seconds. If no wait time is observed
with each wait type, then performance is likely not endangered by the event.

Top Log File Waits
When viewing wait statistics, there are many levels of detail. The first level is the system view, which provides a
global, cumulative snapshot of all the waits that have occurred on a system. The Top Log File Waits section filters all
system wait events to present waits that are associated with log file activity. It ranks the top events by the count of
total waits.

Metrics
None.

Troubleshooting
Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in
Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based
event metrics.

The most common database file wait is a log file parallel write, which indicates how quickly Oracle can flush the log
buffer. Excessive wait times associated with this event count indicate a bottleneck at the log buffer level.

Finally, one thing to watch is not only the actual waits, but also the wait time in seconds. If no wait time is observed
with each wait type, then performance is likely not endangered by the event.

Hottest Database Files
The Hottest Database Files section lists the most active database files as indicated by physical I/O activity.
Metrics
Understanding the storage-level hot spots of a database is beneficial for a couple of reasons. First, you can get a feel for overworked physical disks by viewing I/O statistics at the tablespace and datafile levels. If a particular disk or set of disks is under too much strain, you can relocate the tablespaces to other less-used devices or create new tablespaces on different disks and move hot objects to them (assuming, of course, you have extra disks available). Second, if you have used standard DBA practice and placed indexes in their own tablespace, you can view the I/O statistics for that tablespace and see if the indexes are actually being used.

Troubleshooting
Here are some areas to consider when viewing Hottest Database Files:

- Seeing much activity in the SYSTEM tablespace and datafiles can indicate recursive calls (space management, etc.). The use of locally-managed tablespaces (Oracle8i and later) can help with space management as it relates to data dictionary references.
- Temporary tablespaces (devoted to sort activity) showing higher volumes of physical I/O could indicate a problem with excessive disk sorts.
- You might want to quickly review all the physical I/O for each drive/file system and get a feel for the overworked disks on your server. If you have under utilized disk drives (with their own controllers), consider relocating some tablespaces that exhibit high I/O characteristics to those drives.

I/O Detail
The following tabbed pages are available on the I/O Detail page:

<table>
<thead>
<tr>
<th>Access Pattern Detail</th>
<th>Datafile I/O</th>
<th>Job Queue Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log I/O - DBWR/LGWR</td>
<td>Rollback I/O</td>
<td>Sort Activity</td>
</tr>
<tr>
<td>Table I/O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I/O Tablespace Tab

Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. Other processes like the checkpoint and archive processes (CKPT and ARCH) may also be used.

The Tablespace I/O tab displays details concerning the physical I/O activity at the tablespace level. The table below lists the information available on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace Name</td>
<td>The name of the tablespace.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>The cumulative number of physical reads.</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>The cumulative number of physical writes.</td>
</tr>
<tr>
<td>Physical Block Reads</td>
<td>The cumulative number of physical block reads.</td>
</tr>
<tr>
<td>Physical Block Writes</td>
<td>The cumulative number of physical block writes.</td>
</tr>
<tr>
<td>Physical Read Time</td>
<td>The time spent reading from the tablespace (in hundredths of seconds).</td>
</tr>
<tr>
<td>Physical Write Time</td>
<td>The time spent writing to the tablespace (in hundredths of seconds).</td>
</tr>
</tbody>
</table>
Metrics
Generally, you want to see much more logical I/O activity than physical I/O, at least with respect to reads, although this in and of itself is a guarantee of good I/O performance. Seeing logical and physical reads keeping pace with one another is a sure sign that the Oracle SGA's buffer cache is sized too small or that needless, large table scans are occurring, which cause blocks of data to be continually read in and flushed out of the buffer cache.

Other telltale signs of trouble brewing are large amounts of activity visible in user's TEMP tablespaces. The normal interpretation of such a thing is that a large number of disk sorts are taking place (perhaps because the Init.ora/spfile parameter SORT_AREA_SIZE may be set too small).

For more information, see I/O Home Page.

Datafile I/O Tab

Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database against the physical datafile used to hold information The checkpoint and archive processes (CKPT and ARCH), also perform I/O work in the database.

The Datafile I/O tab displays details concerning the physical I/O activity at the datafile level. The table below lists the information available on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataFile Name</td>
<td>The name of the datafile.</td>
</tr>
<tr>
<td>Tablespace Name</td>
<td>The name of the tablespace.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>The cumulative number of physical reads.</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>The cumulative number of physical writes.</td>
</tr>
<tr>
<td>Block Reads</td>
<td>The cumulative number of physical block reads.</td>
</tr>
<tr>
<td>Block Writes</td>
<td>The cumulative number of physical block writes.</td>
</tr>
<tr>
<td>Read Time</td>
<td>The time spent reading from the tablespace (in hundredths of seconds).</td>
</tr>
<tr>
<td>Write Time</td>
<td>The time spent writing to the tablespace (in hundredths of seconds).</td>
</tr>
</tbody>
</table>

Metrics
Generally, you will want to see much more logical I/O activity than physical I/O, at least with respect to reads, although this in and of itself is a guarantee of good I/O performance. Seeing logical and physical reads keeping pace with one another is a sure sign that the Oracle SGA's buffer cache is sized too small or that needless, large table scans are occurring, which cause blocks of data to be continually read in and flushed out of the buffer cache.

Other telltale signs of trouble brewing are large amounts of activity visible in user's TEMP tablespaces. The normal interpretation of such a thing is that a large number of disk sorts are taking place (perhaps because the Init.ora/spfile parameter SORT_AREA_SIZE may be set too small).

This view also allows you to see the ‘hot’ files in terms of physical I/O activity. Seeing too much activity on one drive/file system may indicate a need for better file striping.

For more information, see I/O Home Page.
Rollback I/O Tab
The Rollback I/O tab includes the following sections:

- **Metrics**

To undo changes made to the Oracle database from within a transaction, Oracle writes data to individual rollback segments. Oracle also uses these segments to maintain read consistency for multiple users of data that is being modified. Because Oracle reads from and writes data to rollback segments, they can become very hot areas for I/O.

The Rollback I/O section presents everything necessary to view and troubleshoot rollback problems. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the rollback segment.</td>
</tr>
<tr>
<td>Size</td>
<td>The size of the rollback segment in KBs.</td>
</tr>
<tr>
<td>Shrinks</td>
<td>The number of times the rollback segment has decreased in size.</td>
</tr>
<tr>
<td>Extents</td>
<td>The number of times the rollback segment has increased in size.</td>
</tr>
<tr>
<td>Gets</td>
<td>The number of header gets (the segment has been used).</td>
</tr>
<tr>
<td>Waits</td>
<td>The number of header waits.</td>
</tr>
<tr>
<td>Writes</td>
<td>The number of bytes written to the rollback segment.</td>
</tr>
<tr>
<td>Active Trans</td>
<td>Indicates whether the rollback segment is active (non zero) or not (zero value).</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates the status of the rollback segment, with the two main results being OFFLINE (a segment is offline and unavailable for transactions) and ONLINE (a rollback segment is online and available for transactional use).</td>
</tr>
<tr>
<td>High Water Mark</td>
<td>The largest size that the rollback segment has ever grown to.</td>
</tr>
</tbody>
</table>

For more information, see [I/O Home Page](#).

**Metrics**

To properly tune rollback I/O, you must first make sure that you have enough segments to accommodate the workload of the database. Constantly seeing a count of active rollback segments equal to or near the number of rollbacks defined for the database is an indicator that you should create more. An overall rollback contention ratio of 1% or greater is an indicator of too few rollbacks. Seeing wait counts greater than zero for each rollback segment is further evidence that you should create more rollback segments. Oracle9i provides the UNDO tablespace to automatically generate and eliminate the 'correct' number of rollback segments for a system given a certain workload.

After ensuring that enough rollback segments exist in the database, you should then turn your attention to the question of sizing. Dynamic rollback extension can take a toll on performance when rollback segments are consistently enlarged to accommodate heavy transaction loads. Seeing rollback segments undergoing numerous extends and shrinks (as Oracle returns a segment back to its OPTIMAL setting), as well as rollback segments having current or high-water mark sizes greater than their OPTIMAL setting usually is a good indicator that they should be permanently enlarged. Again, Oracle9i’s automatic undo management can assist in this process.
Active Rollback Details
Rollback or undo segments are used to guarantee transactional integrity. When a transaction has not been committed or rolled back, a rollback segment will be in use with live transactional data. The Active Rollback Details section displays information concerning active rollback segments. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>The system ID of the session using the rollback segment.</td>
</tr>
<tr>
<td>Username</td>
<td>The account/schema name of the session.</td>
</tr>
<tr>
<td>Rollback</td>
<td>The rollback segment name containing the session’s transaction.</td>
</tr>
<tr>
<td>Start Time</td>
<td>The time the transaction was initiated.</td>
</tr>
<tr>
<td>Blocks Used</td>
<td>The number of rollback blocks used by the transaction.</td>
</tr>
</tbody>
</table>

Metrics
Seeing transactions with dated start times may indicate transactions that are the source of lock contention.

Session Rollback Activity
For all connected sessions, Oracle maintains statistics regarding each session’s counts of rollbacks and commits. The Session Rollback Activity section displays a transactional summary/history that helps identify sessions with excessive rollback activity. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>The system ID of the session.</td>
</tr>
<tr>
<td>Username</td>
<td>The account/schema name of the session.</td>
</tr>
<tr>
<td>Machine Name</td>
<td>The machine name where the session logged on.</td>
</tr>
<tr>
<td>Logon Time</td>
<td>The time the session logged into Oracle.</td>
</tr>
<tr>
<td>User Rollbacks</td>
<td>The number of times the user manually issued the ROLLBACK statement or an error occurred during the user's transaction.</td>
</tr>
<tr>
<td>Transactions Rolled Back</td>
<td>The number of transactions successfully rolled back.</td>
</tr>
<tr>
<td>User Commits</td>
<td>The number of times a user transaction was committed.</td>
</tr>
</tbody>
</table>

Metrics
Seeing high numbers of transactions rolled back might indicate an application error or other transactional problem.

DBWR/LGWR Tab
The DBWR/LGWR tab includes the following sections:

<table>
<thead>
<tr>
<th>Daily Archive Log Summary (Last 7 Days)</th>
<th>Database Writer Detail</th>
<th>Log Writer Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redo Wastage</td>
<td>Table I/O Tab</td>
<td></td>
</tr>
</tbody>
</table>

For more information, see I/O Home Page.
Database Writer Detail

Metrics

The database writer process (DBWR) handles the flow of information from Oracle's physical datafiles to and from the various memory structures in the system global area (SGA). On platforms that support it, you can configure and use multiple DBWR processes. The log writer (LGWR) process manages the information contained in Oracle's online redo log files and redo log buffer area. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistic</td>
<td>The specific metric for the database writer.</td>
</tr>
<tr>
<td>Amount</td>
<td>The value for the statistic.</td>
</tr>
</tbody>
</table>

Metrics

Of all the statistics presented for the DBWR process, the summed dirty queue length statistic deserves attention. Non-zero values typically indicate buffers left in the write queue after a write request and may indicate that the DBWR process is falling behind.

Log Writer Detail

Metrics

For the LGWR process, non-zero values seen for the redo log space requests and redo log space wait time statistics could be a cause for concern. Redo log space requests reflect the number of times a user process waited for space in the redo log buffer, while the redo log space wait time presents the total time waited in milliseconds. Both could indicate the presence of contention in the redo log buffer. Possible remedies include increasing the log_buffer size in the SGA.

Daily Archive Log Summary (Last 7 Days)

To allow for point-in-time recovery, Oracle writes copies of redo log information to disk. When a database is running in archive log mode, a DBA can (with proper backup techniques in place) recovery nicely from a database error and roll forward to almost any point in time needed, as long as the proper archive logs are in place.

The I/O needed to write these archive logs is handled by Oracle's ARCH process. The Daily Archive Log Summary shows the number of archive logs written each day for the past seven days.

Metrics

Heavy redo log activity can lead to many archive files being written out to disk. Batch jobs have the potential to move very fast - sometimes so fast that the online redo logs wrap back around before they have a chance to be archived. Messages of this nature will oftentimes show up in the Oracle alert log. If this is the case, you many want to think about increasing the size of the online redo log files, or increase the number of redo logs in general.

Seeing archive files written more than one per hour or half-hour may indicate a too small redo size (or above average data modification load).
If archive log protection is a concern (in other words, you do not want to lose an archive file that may be needed for recovery) and you are using Oracle8, you can now take advantage of the feature to write archive files to more than one destination on disk. It also allows multiple ARCH processes to now be invoked. Investigate the use of the Init.ora parameters log_archive_dest_n and log_archive_max_processes.

Always remember one thing with respect to archive files and running Oracle in archive log mode: running out of archive file space on the server is a sure fire way to halt all activity in a database. Make sure you have plenty of free space available on your archive drives. Also, implement a purge procedure for older archives in conjunction with a good backup routine.

Redo Wastage
Oracle's redo logs are hotbeds of I/O activity in databases that store heavily modified data. Redo log files are used to perform database recovery in the event of a system crash. Redo logs are written to in a cyclical fashion - each log file is filled up before Oracle moves on to the next file. The Redo Wastage section shows how many bytes were wasted by redo log buffers being written before they were full.

Metrics
Viewing a high percentage of wasted bytes to overall redo bytes can help you identify if redo wastage is a problem on your system.

Steps
Sometimes heavy redo wastage occurs when the log_checkpoint_interval parameter is set too high. If you think this is the case for your system, then

1. Edit the Init.ora file for the database.
2. Change the amount of log_checkpoint_interval to a lower value.
3. Cycle the Oracle server when possible to allow the new value to take effect.
4. Monitor new value to see if performance improves.

Table I/O Tab
I/O activity at the table level has historically been difficult to get with Oracle. Oracle9i and later contain better dictionary elements to help get a clearer picture of table I/O, but earlier versions of Oracle allow an estimation only. The table below lists the information available in the Estimated Physical I/O by table on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner Name</td>
<td>The owner of the table</td>
</tr>
<tr>
<td>Table Name</td>
<td>The name of the table or table partition.</td>
</tr>
<tr>
<td>Estimated Disk Reads</td>
<td>The estimated number of physical and physical direct reads for the table.</td>
</tr>
</tbody>
</table>

Metrics
Viewing high numbers of disk reads for a table could indicate excessive table scan activity. Such a table may benefit from better indexing or from being placed into a larger blocksize tablespace (Oracle9i and later).

Sort Activity Tab
The Sort Activity tab includes the following sections:

- Current Disk Sorts
Session Sort Detail
Excessive sort activity can degrade a user's or overall database performance. When a sort operation occurs, Oracle attempts to perform the sort in a memory space, assigned by the DBA, which exists at the operating system level. If the sort is too large to be contained within this space, it will continue the sort on disk - specifically, in the user's assigned temporary tablespace.

The Session Sort Detail section displays historical sort statistics for every logged on session. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>The unique identifier given the session by Oracle.</td>
</tr>
<tr>
<td>Username</td>
<td>The user account a session is using.</td>
</tr>
<tr>
<td>Machine Name</td>
<td>The machine name where the session logged on.</td>
</tr>
<tr>
<td>Logon Time</td>
<td>The time the session logged into Oracle.</td>
</tr>
<tr>
<td>Disk Sorts</td>
<td>The cumulative number of disk sorts for the session.</td>
</tr>
<tr>
<td>Memory Sorts</td>
<td>The cumulative number of memory sorts for the session.</td>
</tr>
<tr>
<td>Row Sorted</td>
<td>The cumulative number of rows sorted by the session.</td>
</tr>
<tr>
<td>Disk/Memory Sort Ratio</td>
<td>The percentage of times the session had to perform a disk sort.</td>
</tr>
</tbody>
</table>

Metrics
Techniques to include in your overall performance strategy are those that relate to minimizing the amount of sort activity overall and specifically sort activity that takes place on disk. A good place to start is by understanding things that cause sorts in the first place. A list of sort-related commands and SQL-related options include:

- CREATE INDEX, ALTER INDEX … REBUILD
- DISTINCT
- ORDER BY
- GROUP BY
- UNION
- INTERSECT
- MINUS
- IN, NOT IN
- Certain unindexed joins
- Certain correlated subqueries

All of these SQL commands have the potential to create a sort. As a DBA, you probably will not know which queries will have their sorts performed entirely in memory and which ones will be forced to go to disk.
There are times you simply cannot stop disk sort activity (such as in data warehousing environments). That being the case, you should ensure the following are true in your database:

1. Your users' TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.
2. The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.
3. The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.

If your overall memory sort ratio falls below 90%, you may want to increase the parameters devoted to memory sorts - sort_area_size and sort_area_retained_size. Keep in mind that individual users might have the ability to alter their own sessions and increase their sort_area_size assignments! As a DBA, you may want to restrict users that have the ALTER SESSION privilege.

In addition to increasing the amount of memory devoted to sorting, you should also hunt down inefficient SQL that cause needless sorts. For example, UNION ALL does not cause a sort whereas UNION does in an SQL query (to eliminate duplicate rows). DISTINCT oftentimes is coded inappropriately (especially by folks transferring from Microsoft Access, which used to use DISTINCT for nearly every SELECT query).

Current Disk Sorts

Excessive sort activity – especially disk sorts - can degrade a user's or overall database performance. When a sort operation occurs, Oracle attempts to perform the sort in a memory space, assigned by the DBA, which exists at the operating system level. If the sort is too large to be contained within this space, it will continue the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE.

The Current Disk Sorts section displays sort statistics for every session currently performing a disk sort. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Text</td>
<td>The SQL statement causing the disk sort.</td>
</tr>
<tr>
<td>SID</td>
<td>The unique identifier given the session by Oracle.</td>
</tr>
<tr>
<td>Username</td>
<td>The user account a session is using.</td>
</tr>
<tr>
<td>Machine Name</td>
<td>The machine name where the session logged on.</td>
</tr>
<tr>
<td>Tablespace</td>
<td>The tablespace containing the sort segments used by the sort.</td>
</tr>
<tr>
<td>Extents</td>
<td>The number of extents being used for the sort.</td>
</tr>
<tr>
<td>Blocks</td>
<td>The number of blocks being used for the sort.</td>
</tr>
</tbody>
</table>

Metrics

Techniques to include in your overall performance strategy are those that relate to minimizing the amount of sort activity overall and specifically sort activity that takes place on disk. A good place to start is by understanding things that cause sorts in the first place. A list of sort-related commands and SQL-related options include:

- CREATE INDEX, ALTER INDEX … REBUILD
- DISTINCT
- ORDER BY
- GROUP BY
- UNION
- INTERSECT
• MINUS
• IN, NOT IN
• Certain unindexed joins
• Certain correlated subqueries

All of these SQL commands have the potential to create a sort. As a DBA, you probably will not know which queries will have their sorts performed entirely in memory and which ones will be forced to go to disk.

There are times you simply cannot stop disk sort activity (such as in data warehousing environments). That being the case, you should ensure the following are true in your database:

1. Your users’ TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.
2. The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.
3. The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.

If your overall memory sort ratio falls below 90%, you may want to increase the parameters devoted to memory sorts - `sort_area_size` and `sort_area_retained_size`. Keep in mind that individual users might have the ability to alter their own sessions and increase their `sort_area_size` assignments! As a DBA, you may want to restrict users that have the ALTER SESSION privilege.

In addition to increasing the amount of memory devoted to sorting, you should also hunt down inefficient SQL that cause needless sorts. For example, UNION ALL does not cause a sort whereas UNION does in an SQL query (to eliminate duplicate rows). DISTINCT oftentimes is coded inappropriately (especially by folks transferring from Microsoft Access, which used to use DISTINCT for nearly every SELECT query).

Job Queue Activity Tab
The Job Queue Activity tab includes the following sections:

<table>
<thead>
<tr>
<th>Job Queue Summary</th>
<th>Problem Jobs</th>
<th>Running Jobs Detail</th>
</tr>
</thead>
</table>

For more information, see I/O Home Page.

Job Queue Summary
Oracle provides a way for Oracle developers to submit and run jobs through the job queue feature of the Oracle database. The table below lists the information available on this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Jobs</td>
<td>The number of jobs defined to the database.</td>
</tr>
<tr>
<td>Broken</td>
<td>The number of jobs broken (disabled).</td>
</tr>
<tr>
<td>Failures</td>
<td>The number of job failures.</td>
</tr>
<tr>
<td>Running</td>
<td>The number of jobs currently running.</td>
</tr>
</tbody>
</table>
Problem Jobs

Oracle provides a way for Oracle developers to submit and run jobs through the job queue feature of the Oracle database. Sometimes a job may fail or become disabled (broken). The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job ID</td>
<td>The job number for the job.</td>
</tr>
<tr>
<td>Login User</td>
<td>The user account used to run the job.</td>
</tr>
<tr>
<td>Last Date Successfully Run</td>
<td>The date/time of when the job last successfully ran.</td>
</tr>
<tr>
<td>Broken</td>
<td>Indicates if the job is disabled or not.</td>
</tr>
<tr>
<td>Failures</td>
<td>The number of failures the job has experienced.</td>
</tr>
</tbody>
</table>

Running Jobs Detail

Oracle provides a way for Oracle developers to submit and run jobs through the job queue feature of the Oracle database. The Running Jobs Detail section provides information about currently running jobs. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>The unique identified given the session by Oracle.</td>
</tr>
<tr>
<td>Job ID</td>
<td>The job number for the job.</td>
</tr>
<tr>
<td>Start Date/Time</td>
<td>Indicates when the job started.</td>
</tr>
<tr>
<td>Submitted by</td>
<td>Indicates the session that submitted the job.</td>
</tr>
<tr>
<td>Run As</td>
<td>Indicates which account is used to run the job.</td>
</tr>
<tr>
<td>Parse As</td>
<td>Indicates which account is used to parse the job.</td>
</tr>
<tr>
<td>Next Run Date/Time</td>
<td>Indicates when the job will next run.</td>
</tr>
<tr>
<td>Job Contents</td>
<td>Indicates the actual command used by the job.</td>
</tr>
</tbody>
</table>

Access Pattern Detail Tab

When a session submits a set of SQL or a transaction, there are many different combinations of methods the Oracle kernel will use to satisfy the query or transaction. The Access Pattern Detail tab displays all currently connected sessions, their demographic information, and a count of the various different methods that Oracle uses to handle incoming requests. The table below lists the information available on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>The unique identified given the session by Oracle.</td>
</tr>
<tr>
<td>Username</td>
<td>The user account a session is using.</td>
</tr>
<tr>
<td>Machine Name</td>
<td>The machine name where the session logged on.</td>
</tr>
<tr>
<td>Logon Time</td>
<td>The time the session logged into Oracle.</td>
</tr>
<tr>
<td>Large Table Scans</td>
<td>The cumulative count of 'large' table scans with 'large' normally being defined as any table over five blocks.</td>
</tr>
<tr>
<td>Small Table Scans</td>
<td>The cumulative count of 'small' table scans with 'small' normally being defined as any table under five blocks.</td>
</tr>
</tbody>
</table>
Space Page Statistics
The Space home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Redo Logs</th>
<th>Space Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Analysis</td>
<td>User Space Analysis - Top Space Hogs</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
Space Detail
Home Page Statistics
I/O Page Statistics
Memory Page Statistics
Objects Page Statistics
OS Page Statistics
Users Page Statistics

Space Analysis - Space
- Metrics
- Troubleshooting

The Oracle tablespace is the logical container for the physical storage needs of a database. The Space Analysis section displays details for all the tablespaces for a particular database, including their total, used, free space, and percent free, as well as if each can automatically extend to support incoming space requests.

TIP:
Metrics
If any tablespace’s free space percent amount goes below 10%, and at least one tablespace’s datafiles does not have AUTOEXTEND enabled (or the datafile has reached its extend limit), you should take action to ensure that the tablespace does not run out of available free space.

Troubleshooting
There are two things you can do to ensure that a tablespace does not run out of available free space:

1. First, you should look into the use of Oracle’s AUTOEXTEND feature. AUTOEXTEND lets you give an Oracle tablespace the ability to auto-grow when it has exhausted the free space contained within. You can let a tablespace grow in an unlimited fashion or put constraints on it to stop at a certain point. You can also dictate how much more free space the tablespace gets each time it needs more space than is available. However, AUTOEXTEND enabled for a tablespace does not mean that you cannot run out of space. Remember you still have the physical server limitations to contend with. Make sure you (or your sysadmin) keep a careful eye on the server drives that house your Oracle database files for available free space.

2. If the free space on a server drive nears its limit, disable AUTOEXTEND for the datafile(s) that are on the target drive, and use the ALTER TABLESPACE … ADD DATAFILE command to place a new datafile for the tablespace on another drive that has more free space to offer.

TIP: AUTOEXTEND is not a replacement for proactive space planning. When extra space is needed by the database, and AUTOEXTEND is activated by Oracle, performance slows as Oracle allocates more space for the tablespace. Avoiding AUTOEXTEND aids performance, albeit in a small way.

For more information, see Space Page Statistics.

Bottleneck Analysis - Space
When using bottleneck analysis to troubleshoot a database, you cannot rely only on the information contained in the wait event views that Oracle provides. For example, an object may attempt to extend into another extent of space in a tablespace and yet be denied if no such free space exists. Such a failure is not reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that you cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Space home page displays space-related bottlenecks that can decrease overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Space home page to succinctly communicate the general overall space situation of the monitored database:

- Autoextend Datafiles at Extend Limit
- Objects at Maximum Extents
- Objects with No Room to Expand
- Offline Tablespaces
- Top Fragmented Datafiles

For more information, see Space Page Statistics.
SQL Analysis - Space
Much of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

The SQL Analysis section of the Space home page shows a count of all SQL statements that have executed INSERT, UPDATE, DELETE, and TRUNCATE commands. Seeing these statistics should help you determine how much space-related SQL activity is occurring on your system.

Metrics
The statistics for this section can be used in many different ways. For example, large numbers of inserts could indicate a growing database and that you should keep a watchful eye on free space. Heavy update numbers could mean trouble for objects containing many indexes (indexes typically undergo splits and fragmentation when large volumes of updates are thrown their way). Many deletes might signal potential table high water mark problems.

Redo Logs
Oracle's redo logs are hotbeds of I/O activity in databases that store heavily modified data. Redo log files are used to perform database recovery in the event of a system crash. Redo logs are written to in a cyclical fashion - each log file is filled up before Oracle moves on to the next file. The redo bytes statistic reflects the total amount of redo generated in bytes since the last refresh. The redo wastage bytes indicate how many bytes were wasted by redo log buffers being written before they were full.

Metrics
Viewing a high percentage of wasted bytes to overall redo bytes can help you identify if redo wastage is a problem on your system.

Troubleshooting
Sometimes heavy redo wastage occurs when the log_checkpoint_interval parameter is set too high. To fix, do the following:

1. Edit the Init.ora file for the database.
2. Change the amount of log_checkpoint_interval to a lower value.
3. Cycle the Oracle server when possible to allow the new value to take effect.
4. Monitor new value to see if performance improves.

For Oracle9i and later, you can dynamically alter the log_checkpoint_interval parameter without stopping/starting the database.

For more information, see Space Page Statistics.

User Space Analysis - Top Space Hogs
The Top Space Hogs section details the schema in the database that own the most space. In databases where object owners can be many, it is oftentimes a good idea to take a quick look at which schema are leading the way in terms of space usage. This is not restricted to production environments only, but can be extended to dynamic development systems as well, especially if there are many developers in a database that have their own sets of objects. Developers sometimes have a bad habit of creating objects (or copies of objects) that they leave in a database even though they are not being used.

Metrics
None.
Troubleshooting
If a user owns many objects, or copies of objects, you can reclaim space in your database by contacting the user to see if those objects can be dropped.

For more information, see Space Page Statistics.

Top Fragmented Datafiles

• Metrics

• Troubleshooting

The Top Fragmented Datafiles section shows the tablespaces datafiles suffering from the highest levels of free space fragmentation in the database. Tablespaces are made up of object segments and space extents. Extents are either allocated to object segments or are free. When a tablespace is initially populated, all objects are neatly packed together in the front of the tablespace and all remaining free space is in one free chunk at the end. As objects grow (or extend) they are given new extents of space in the tablespace/datafile. If they are dropped, pockets of free space begin to appear throughout the tablespace. These pockets of space take one of two forms. The table below describes these forms:

<table>
<thead>
<tr>
<th>Free Space</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Honeycombs</td>
<td>Pockets of free space that are adjacent to one another.</td>
</tr>
<tr>
<td>Bubbles</td>
<td>Pockets of free space that are trapped between object extents in the tablespace.</td>
</tr>
</tbody>
</table>

Metrics
If you see a datafile that has many chunks of free space, determine if the tablespace is experiencing honeycomb or bubble fragmentation. You can handle honeycomb fragmentation quite easily, whereas bubble fragmentation is more difficult to solve.

Troubleshooting
You can eliminate honeycomb fragmentation with the ALTER TABLESPACE...COALESCE command. Issuing this command combines all pockets of adjacent free space into one extent. Each database maintenance plan should include a job that coalesces all the free honeycombs in a tablespace into one free chunk. Although Oracle is supposed to perform this operation automatically through the SMON process, it requires you to have the PCTINCREASE parameter of the tablespace set to a nonzero value. Having PCTINCREASE set to a value greater than zero encourages tablespace fragmentation through disparately sized extents. Plus, using SMON in this way is not efficient or entirely reliable.

You can temporarily solve bubble fragmentation by performing a total tablespace reorganization. A better long-term solution for Oracle databases 8.1.5 or later is to convert tablespaces over to locally managed tablespaces. With locally managed tablespaces you either specify the initial extent size and let Oracle automatically size all other extents, or specify a uniform extent size for everything. Problems caused by fragmentation then become a thing of the past.

To help stave off fragmentation problems:
• Set PCTINCREASE to zero for all tablespaces and objects to promote same-sized extents.
• Specify equal-sized allotments for your INITIAL and NEXT object storage parameters.
• Group objects with like growth and storage needs together in their own tablespaces.
• Considering converting any dictionary-managed tablespace to locally-managed.

You should also avoid fragmentation in the SYSTEM tablespaces. The best ways to do this include:
• Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
• Ensure no user has a quota set for SYSTEM.
• Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

Objects at Maximum Extents
When an object in Oracle (table, index, table partition, etc.) needs to expand, Oracle automatically allocates another extent of space to accommodate the incoming data. However, a table or index can reach its maximum extent limit, which is the maximum number of extents for an object.

NOTE: This statistic is available on the Objects home page and the Space home page.

Metrics
Unless an object is static in nature, and is not expected to grow in size, no object should ever get to its maximum extent limit.

Troubleshooting
If a maximum extent limit is reached for an object, you can take a number of actions:
• The object can be reorganized into a locally-managed tablespace where no maximum extent limit is applied to any object.
• The object can be altered to have unlimited extents.
• The object can be reorganized into one or a few extents.

The good news concerning locally-managed tablespaces is that unlimited extents are the rule, so if you choose to use these storage structures in your database, you will not have to worry about an object ever reaching a maximum extent limit. Some DBAs have expressed concerns over whether an object having hundreds or thousands of extents experiences performance problems when full table scans or similar operations are performed against them. While most Oracle experts agree that such I/O degradation might have been experienced in earlier Oracle versions, most feel that such is not the case any longer, especially when locally-managed tablespaces are used.

Objects with No Room to Expand
When an object in Oracle (table, index, table partition, etc.) needs to expand, Oracle automatically allocates another extent of space to accommodate the incoming data. However, an object may be prohibited from acquiring its next extent of space for a number of reasons:
• The tablespace does not have enough free space to accommodate the extent request, and cannot automatically extend.
• The tablespace has enough total free space, but the space is not contiguous in nature (and the tablespace cannot automatically extend) and therefore the object cannot extend.

NOTE: This statistic is available on the Objects home page and the Space home page.

Metrics
Unless an object is static in nature, and is not expected to grow in size, no object should ever be prohibited from acquiring its next extent of space.

Troubleshooting
If an object cannot extend, you can take a number of actions including:
• One or more of the tablespace’s datafiles can have its AUTOEXTEND property enabled, which lets the object to grow even though there is no space currently available in the tablespace.

• The object can be reorganized into a locally-managed tablespace all extents are uniform in nature.

• The object can be altered to have a reduced next extent size.

Autoextend Datafiles at Extend Limit
You can specify that one or more tablespace datafiles can automatically grow by enabling the datafile’s AUTOEXTEND property. You can also specify that the datafile not be permitted to grow beyond a certain limit, and once that limit is reached, the datafile and/or the tablespace may be permanently out of space.

Metrics
You should not see any positive indicator that any datafile has reached its extend limit. While it might be okay from the standpoint that a datafile cannot simply grow anymore (if the tablespace has other datafiles that can grow or that no more space is added to the tablespace), disable AUTOEXTEND on the datafile.

Troubleshooting
If a datafile has reached its extend limit, you can take a number of actions:

• Eliminate the extend limit for the datafile either by increasing the limit size or allowing to grow in an unlimited fashion.

• If the datafile is on a disk or file system that has no free space left, you can add another datafile to the tablespace to allow for space expansion and alter the problem datafile to have AUTOEXTEND disabled.

• If the datafile is on a disk or file system that has no free space left, the datafile can be physically relocated to another drive or file system with more free space.

Offline Tablespaces
A tablespace ONLINE status indicates that the tablespace is in normal read/write mode. Tablespaces marked read only displays a status of READ ONLY, while tablespaces engaged in backup mode indicates such in their status. An OFFLINE tablespace during normal processing hours is a cause for concern.

Metrics
No tablespace should have a status of OFFLINE during normal business hours, unless you are performing maintenance on it.

Troubleshooting
If a tablespace is viewed with a status of OFFLINE, you can alter it back ONLINE through the ALTER TABLESPACE command.

Space Detail
The following tabbed pages are available on the Space Detail page:

<table>
<thead>
<tr>
<th>Extent Analysis</th>
<th>Object Space</th>
<th>Fragmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace Detail</td>
<td>Tablespace Growth</td>
<td>User Space</td>
</tr>
</tbody>
</table>
Tablespace Detail Tab
The Tablespace Detail tab includes the following sections:

- Datafile Detail
- Tablespace Object Summary

For more information, see Space Home Page.

Tablespace Object Summary
Tablespaces are the logical containers for database objects inside Oracle. The Tablespace Object Summary section provides an overview of how much space is consumed in each tablespace by object type. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace Name</td>
<td>The name of the tablespace.</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates the status of the tablespace.</td>
</tr>
<tr>
<td>Autoextend</td>
<td>Indicates whether one or more of the tablespace’s datafiles has the ability to automatically grow.</td>
</tr>
<tr>
<td>Total Space (MB)</td>
<td>The total allocated space for the tablespace.</td>
</tr>
<tr>
<td>Table (MB)</td>
<td>The amount of used table and table partition space, in MB.</td>
</tr>
<tr>
<td>Index (MB)</td>
<td>The amount of used index and index partition space, in MB.</td>
</tr>
<tr>
<td>Rollback (MB)</td>
<td>The amount of used rollback and undo segment space, in MB.</td>
</tr>
<tr>
<td>Cluster (MB)</td>
<td>The amount of used cluster space, in MB.</td>
</tr>
<tr>
<td>LOB (MB)</td>
<td>The amount of used large object space, in MB.</td>
</tr>
</tbody>
</table>

Datafile Detail
Tablespaces are the logical containers for database objects inside Oracle, and datafiles are the actual physical files that make up a tablespace. The Datafile Detail section provides an overview of the size of each datafile and other demographic information. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>The name of the datafile.</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates the status of the tablespace.</td>
</tr>
<tr>
<td>Autoextend</td>
<td>Indicates whether the datafile has the ability to automatically grow.</td>
</tr>
<tr>
<td>Tablespace Name</td>
<td>The name of the tablespace.</td>
</tr>
<tr>
<td>Size (MB)</td>
<td>The total allocated space for the datafile.</td>
</tr>
<tr>
<td>Max Size (MB)</td>
<td>The maximum size the datafile can automatically grow to.</td>
</tr>
</tbody>
</table>
Object Space Tab
The Object Space tab displays a summary of space and extent information for all or selected objects in the database. The table below lists the information available on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>The owner of the object.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>Object Type</td>
<td>The type of object (table, index, etc.).</td>
</tr>
<tr>
<td>Space (KB)</td>
<td>The size of the object in KB.</td>
</tr>
<tr>
<td>Extents</td>
<td>The number of extents used by the object.</td>
</tr>
<tr>
<td>Max Extents</td>
<td>The maximum number of extents allowed for the object.</td>
</tr>
<tr>
<td>Initial Extent (KB)</td>
<td>The initial extent size for the object.</td>
</tr>
<tr>
<td>Next Extent (KB)</td>
<td>The next extent size for the object.</td>
</tr>
<tr>
<td>Tablespace</td>
<td>The tablespace where the object resides.</td>
</tr>
</tbody>
</table>

Metrics
One thing to keep an eye on are objects that are close to their maximum extent limit. Such objects can be altered to have more or unlimited extents if need be. The use of locally-managed tablespace eliminates the threat of an object ever encountering a maximum extent limit.

For more information, see Space Home Page.

Tablespace Growth Tab
• Metrics
Growing tablespaces do not have to spell problems for a DBA. Of course, up front planning is the key to sizing tablespaces correctly. Unfortunately, DBAs may not have all the information they need at tablespace creation time (or they may have the wrong information), so a tablespace can approach the end of its free space from time to time. The DBA can allow a tablespace to grow automatically (AUTOEXTEND) to prevent an out-of-space condition. Enabling AUTOEXTEND for a tablespace is quite reassuring for a DBA, but it introduces a new concept for the tablespace: monitoring data file growth. You should monitor your tablespaces that have AUTOEXTEND set so you can get an idea of the growth that is occurring in your database. Monitoring them lets you perform some mini-capacity planning and helps you get a jump-start on server upgrades.

The Tablespace Growth tab displays growth statistics for the database currently being monitored. The table below lists the information available on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace Name</td>
<td>The name of the tablespace.</td>
</tr>
<tr>
<td>Datafile Name</td>
<td>The name of the tablespace's datafile.</td>
</tr>
<tr>
<td>Autoextend</td>
<td>Indicates whether the tablespace has one or more datafiles that have the AUTOEXTEND feature set (allowing the datafile to automatically grow).</td>
</tr>
<tr>
<td>Created Size</td>
<td>The starting size for the physical datafile.</td>
</tr>
<tr>
<td>Current Size</td>
<td>The current size of the physical datafile.</td>
</tr>
<tr>
<td>Growth</td>
<td>The percentage of growth experienced for the datafile (if any).</td>
</tr>
</tbody>
</table>
Metrics
Using AUTOEXTEND can be especially important for tablespaces devoted to disk sorting. Large data warehouses often must endure large disk sort operations. Having AUTOEXTEND enabled for tablespaces used for temporary segments (sort activity) helps large sort queries to complete when they might otherwise fail due to running out of sort space.

Just having AUTOEXTEND enabled for a tablespace does not mean you cannot run out of space. Remember, you still have to contend with the physical server limitations. Make sure you (or your sysadmin) keep a careful eye on the server drives that house your Oracle database files for available free space. If the free space on a server drive nears its limit, disable AUTOEXTEND for the datafile(s) that are on that drive, and use the traditional ALTER TABLESPACE...ADD DATAFILE command to place a new datafile for the tablespace on another drive that has more free space to offer.

For more information, see Space Home Page.

Fragmentation Tab

• Metrics
• Troubleshooting

For more information, see Space Home Page.

Tablespaces are made up of object segments and space extents. Extents are either allocated to object segments or are free. When a tablespace is initially populated, all objects are neatly packed together in the front of the tablespace and all remaining free space are in one free chunk at the end. Unfortunately, this is not how things continue to be in a tablespace. As objects grow (or extend) they are given new extents of space in the tablespace. If you drop objects, pockets of free space begins to appear throughout the tablespace. These pockets of space are either honeycombs or bubbles. Honeycombs are not so difficult to deal with, but bubbles are another story.

The Fragmentation tab displays two different views of tablespace fragmentation. The first grid displays fragmentation at the tablespace level and the second grid displays fragmentation at the datafile level. The table below describes the information available in the Tablespace Fragmentation grid:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace Name</td>
<td>The tablespace name.</td>
</tr>
<tr>
<td>Free Space (MB)</td>
<td>The total amount of free space in MB for the tablespace.</td>
</tr>
<tr>
<td>Free Chunks</td>
<td>The total number of free chunks in the tablespace.</td>
</tr>
<tr>
<td>Largest Chunk (MB)</td>
<td>The largest free chunk (in MB) for the tablespace.</td>
</tr>
</tbody>
</table>

The table below describes the information available in the Datafile Fragmentation grid on the Fragmentation tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datafile</td>
<td>The name of the datafile.</td>
</tr>
<tr>
<td>Autoextend</td>
<td>Indicates whether the datafile can automatically grow in size.</td>
</tr>
<tr>
<td>Tablespace</td>
<td>The tablespace name.</td>
</tr>
<tr>
<td>Free Chunks</td>
<td>The number of free chunks in the datafile.</td>
</tr>
<tr>
<td>Largest Chunk</td>
<td>The largest free chunk (in MB) for the datafile.</td>
</tr>
</tbody>
</table>
Metrics
To spot and correct fragmentation in your tablespaces, you should periodically monitor the fragmentation levels of your tablespaces at a global level. Doing so helps you quickly spot tablespaces that are experiencing fragmentation issues. Seeing a tablespace with only one chunk of free space is a sign that a tablespace is not having fragmentation problems. Seeing a tablespace with a couple of free chunks may not be a big deal either, because the tablespace could be made up of more than one datafile. Each datafile has its own chunk or chunks of free space.

If you see a tablespace that has many chunks of free space, the next thing to do is drill down into it and find out if the tablespace is experiencing honeycomb or bubble fragmentation. Honeycomb fragmentation occurs when pockets of free space exist that are adjacent to one another. Bubbles are pockets of free space that are trapped between object segment extents.

You can eliminate honeycomb fragmentation with the ALTER TABLESPACE...COALESCE command. This command combines all pockets of adjacent free space into one extent. It is important to do this because when an object is newly introduced to a tablespace (or an existing object needs to extend), and a contiguous chunk of free space does not exist to accommodate an object's INITIAL or NEXT size allocation, Oracle must manually coalesce all available honeycombs to try and form a large enough free chunk. This is a performance hit. If possible, you should try to minimize performance hits.

Bubble fragmentation is a more serious matter and is normally only corrected through tablespace or database reorganization. The standard technique is to perform an export of all objects in the tablespace, drop the tablespace, and then import all the objects back into the tablespace. However, this technique may just treat the symptom and not the cause of bubble fragmentation. The real issue is to address the reuse of space extents within a tablespace so that bubble fragmentation does not occur in the first place.

Oracle8 and later offers the concept of locally-managed tablespaces, which can all but eliminate tablespace fragmentation. It totally does away with the storage parameters of MINEXTENTS, MAXEXENTS, PCTINCREASE, and NEXT. With locally managed tablespaces you either specify the initial extent size and let Oracle automatically size all other extents, or specify a uniform extent size for everything. Problems caused by fragmentation then become a thing of the past.

Troubleshooting
What can you do manually to help stave off fragmentation problems? A few suggestions include:

• Set PCTINCREASE to zero for all tablespaces and objects to promote same-sized extents.
• Specify equal-sized allotments for your INITIAL and NEXT object storage parameters.
• Group objects with like growth and storage needs together in their own tablespaces.

Of all your tablespaces, you want to avoid fragmentation problems in your SYSTEM tablespace the most as this is the major hotbed tablespace for Oracle activities. The easiest way to avoid this is to not allow any user (even the default DBA ID's SYS and SYSTEM) to have access to it. There are three ways to do this:

1 Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
2 Ensure no user has a quota set for SYSTEM.
3 Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

Extent Analysis Tab
Object fragmentation results when objects consume multiple extents of space in a tablespace rather than a single block of space. Although performance problems with respect to object fragmentation are not as severe as they were in previous versions of Oracle, some DBAs believe that response-time penalties can still be chalked up to this situation. When multiple extents exist for an object, the amount of time it takes Oracle to scan it can be longer than if the object was made up of only one extent. This typically holds true when extents are scattered on different parts of the physical server disk. In addition, a performance hit is taken each time an object must extend into another extent of space.
The Extent Analysis tab displays objects whose extent count has exceeded a user-suggested numerical limit. The table below lists the information available on this tab if any objects are found:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>The user account that owns the object.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>Object Type</td>
<td>The type of object (TABLE, INDEX, etc.).</td>
</tr>
<tr>
<td>Next Extent</td>
<td>The next extent amount of space the object will consume.</td>
</tr>
<tr>
<td>Total Extents</td>
<td>The number of extents the object is currently consuming.</td>
</tr>
<tr>
<td>Max Extents</td>
<td>The MAXEXTENTS limit imposed on the object by the DBA.</td>
</tr>
<tr>
<td>Largest Free Chunk (KB)</td>
<td>The size of the largest chunk.</td>
</tr>
<tr>
<td>Tablespace Name</td>
<td>The name of the tablespace.</td>
</tr>
</tbody>
</table>

### Metrics

When is object fragmentation a problem for an object? Opinions on this vary widely. As a general rule of thumb, if an object is exhibiting response time degradations, and no other factor can be attributed to the slowdown, examine the object to see how badly fragmented it is. If extent numbers are seen exceeding 100 or more, than you might want to think about reorganizing the object or using locally managed tablespaces. This rule of thumb does not apply to objects assigned to locally-managed tablespaces as extent fragmentation is expected and not a performance issue.

### Troubleshooting

The best preventative for this problem is specifying the correct allocation of space for the object in the first place, but what can you do if you have objects with high numbers of extents? For tables and indexes you have three options:

1. Use Oracle’s export/import utility to export, drop, and import the fragmented objects back into the database with the export parameter COMPRESS=Y. This will bring the objects back into the database with one large extent. Make sure that large enough chunks of available free space exist to accept the object back in, or you may experience space allocation errors.

2. With Oracle8, you can use the ALTER TABLE … MOVE command to reorganize a table back into one extent in a very safe and efficient manner.

3. Use ALTER INDEX … REBUILD to reorganize indexes that have moved into multiple extents.

Another thing to keep an eye on is the next extent size of an object vs. the largest free chunk of space in the tablespace. Seeing objects with larger next extents than free space may signal a problem unless the tablespace can automatically extend to accommodate the object’s next extent.

For more information, see [Space Home Page](#).

### User Space Tab

- **Metrics**

The User Space tab includes the following sections:

- **Top Space User Accounts**
- **User Space Detail**

For more information, see [Space Home Page](#).
Top Space User Accounts

It is vital that a DBA know how the various accounts assigned to a database consume space. This is true in either production or development databases, because even in development databases, space can become an issue that causes major headaches. If development accounts are creating and cloning many objects without removing them, the result can be a lot of wasted space.

The Top Space User Accounts statistic shows how much total allocated space the top twenty accounts possess in the database.

Metrics

Seeing owner account with larger than expected data volumes could warrant further investigation. In addition, seeing accounts that have objects in more tablespaces than they have quota’s for should also cause you to examine their account more closely to ensure the correct space privileges are in place.

User Space Detail

The User Space Detail section allows you to drill-down and see the specifics of the Top Space User accounts.

Objects Page Statistics

The Objects home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key Ratio Analysis</th>
<th>SQL Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Buffer Pool Placement</td>
<td>User Object Analysis</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

Objects Detail
- Home Page Statistics
- I/O Page Statistics
- Memory Page Statistics
- OS Page Statistics
- Space Page Statistics
- Users Page Statistics

Key Ratio Analysis - Objects

Object-related database activity can be examined using both ratio-based and wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios and statistical readings that can be used to indicate how active certain object types are. Performance ratios serve as roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis.

When using ratio-based analysis, there are some standards to adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that you examine come from the v$sysstat performance view. The performance view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impacts how a particular ratio is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.
A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database’s individual ‘personality’ with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead you to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Objects home page to succinctly communicate the general session-based performance levels of the monitored database:

<table>
<thead>
<tr>
<th>Active Rollback Ratio</th>
<th>Chained Row Fetch Ratio</th>
<th>Rollback Contention Ratio</th>
</tr>
</thead>
</table>

For more information, see Objects Page Statistics.

**Bottleneck Analysis - Objects**

When using bottleneck analysis to troubleshoot a database, you cannot rely only on the information contained in the wait event views that Oracle provides. For example, an object attempts to extend into another extent of space in a tablespace and yet be denied if no such free space exists. Such a failure is not reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that you cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Objects home page displays object-related bottlenecks decreases the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Objects home page to communicate the general overall conditions of objects in the monitored database:

<table>
<thead>
<tr>
<th>Chained Tables</th>
<th>Current Object Blocks</th>
<th>Enqueue Waits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free List Waits</td>
<td>Invalid Objects</td>
<td>Locked Objects</td>
</tr>
<tr>
<td>Objects at Maximum Extents</td>
<td>Objects with No Room to Expand</td>
<td>Table High Water Mark Problems</td>
</tr>
<tr>
<td>Tables/Indexes in Same Tablespace</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SQL Analysis - Objects**

Certain objects in an Oracle database are accessed more than others. These objects can become a source of contention given certain conditions. The SQL Analysis section of the Objects home page identifies the most accessed tables on the database.

**Metrics**

DML activity against tables can cause contention for space management objects like free lists. Oracle9i or later provides automatic segment management, which can remove problems with free lists and the like.

**Object/Buffer Pool Placement**

- **Metrics**
- **Troubleshooting**
Because data objects can be referenced with different types of usage patterns, Oracle8 offers the option to intelligently place objects into one of three buffer caches. The table below describes the buffer caches:

<table>
<thead>
<tr>
<th>Buffer Cache</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEEP</td>
<td>Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.</td>
</tr>
<tr>
<td>RECYCLE</td>
<td>Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>The traditional cache for all other data objects.</td>
</tr>
</tbody>
</table>

NOTE: Unless you specify the KEEP or RECYCLE buffer cache, Oracle automatically places objects into the DEFAULT buffer cache.

The Object/Buffer Pool Placement section identifies how many objects are currently assigned to the various caches.

Metrics
When looking at overall database I/O activity, you should keep an eye out for objects that you can place into a particular buffer cache. Consider using a KEEP cache for relatively small, frequently accessed tables that require fast response times. Large tables with random I/O activity and are scanned from beginning to end are good candidates for a RECYCLE cache.

Troubleshooting
Objects can be placed into different buffer pools at object creation time (using the STORAGE... BUFFER_POOL option in Oracle8 or later) or existing objects can be set into a different pool with the ALTER command.

NOTE: Tables, partitions, and indexes can be placed into the different caches.

If you just want to use the DEFAULT buffer pool and not enable any special caches, you can still encourage Oracle to keep certain objects in the cache as long as possible using the CACHE parameter. For example, issuing the command ALTER TABLE...CACHE specifies that the blocks retrieved for a table be placed at the most recently used end of the LRU list in the DEFAULT buffer cache when a full table scan is performed. You can also use the CACHE hint in SQL statements to cache a table, but used in this form, the blocks are only cached until the next time the database is shut down.

For more information, see Objects Page Statistics.

User Object Analysis- Objects
All objects are created inside of user accounts called schema. The User Object Analysis section of the Performance Analyst Objects home page shows schema that are ranked the worst in the several categories. The table below lists the categories:

<table>
<thead>
<tr>
<th>Worst Schema for</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chained Row Tables</td>
<td>The schema containing the most tables with chained rows is identified.</td>
</tr>
<tr>
<td>Invalid Objects</td>
<td>The schema containing the most invalid objects is identified.</td>
</tr>
<tr>
<td>Extent Problems</td>
<td>The schema containing the most objects with maximum extent limit or extent space deficit problems is identified.</td>
</tr>
</tbody>
</table>
Metrics
None.

Troubleshooting
To determine the degree and severity of each type of problem, go into detailed views that present specific information for each type of object problem.

For more information, see Objects Page Statistics.

Rollback Contention Ratio
Rollback segments are used by the Oracle database to hold data needed to rollback (or undo) any changes made through inserts, updates, or deletes to various Oracle objects. They also allow Oracle to have read consistency for long running queries, are used for recovery purposes, and play a role during exports of database information. In a heavy transaction processing environment, rollback segments are accessed continuously and therefore are subject to contention problems. The Rollback Contention Ratio helps identify contention occurring on the system relating to rollbacks.

Metrics
Overall, if the rollback contention ratio approaches 1% or more, consider creating more rollback segments. Also consider creating a specialized, larger, rollback segment to be used by long running transactions. Doing so alleviates dynamic rollback extensions and cuts down heavily on ORA-01555 snapshot too old errors.

Troubleshooting
If you are using Oracle8i or earlier, begin by creating new rollback segments and altering them to be online for use. Then monitor the overall contention ratio to see if it begins to drop.

For Oracle9i or later, consider the use of UNDO management, which is where Oracle itself automatically manages rollback segments in special UNDO tablespaces.

Active Rollback Ratio

Metrics

Troubleshooting

To undo changes made to the Oracle database from within a transaction, Oracle writes data to individual rollback segments. You can also use these to maintain read consistency for multiple users of modified data. Because Rollback Segments read and write data, they can become very hot areas for I/O. This statistic is a ratio that displays the percentage of rollback segments currently involved in work. Metrics

Seeing consistent ratios of 80% or higher could indicate the need for more rollback segments on the database.

Troubleshooting

If you are using Oracle8i or earlier, begin by creating new rollback segments and altering them to be online for use. Then monitor the overall active rollback ratio to see if it begins to drop.

For Oracle9i or later, consider the use of UNDO management, which is where Oracle itself automatically manages rollback segments in special UNDO tablespaces.

Chained Row Fetch Ratio

Metrics

Troubleshooting
In normal circumstances, a row of data should fit completely inside one Oracle block. Sometimes, however, this is not the case and the table suddenly contains chained or migrated rows (rows that span more than one data block).

Chaining occurs when a row is initially too large to fit inside one block. Two or more blocks are used by Oracle to hold the row. Migration deals with rows that have grown so much that they can no longer be contained within their original block. When this occurs, Oracle relocates the row out of its original block into another block, but leaves a pointer behind to indicate the relocation.

Both chaining and migration force Oracle to perform more than one I/O to retrieve data that could normally be obtained with a single I/O operation, with the end result being degraded performance.

The chain row fetch ratio statistic represents the percentage of rows fetched that were of either the chained or migrated row type.

**Metrics**
If the amount of chained rows in your tables exceeds 25-30%, you should take steps to eliminate the problem. Further, if the amount of chained rows accessed in your system vs. total rows accessed approaches 20-30%, you can start the process of eliminating the chained and migrated rows.

**Troubleshooting**
You can locate tables that contain chained rows. Once found, there are a couple of ways to reorganize tables to remove the chaining/migration problem. However, the best way to deal with chained and migrated rows is to prevent them from occurring. The table below describes two methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use a large block size</td>
<td>Because chaining and migrations occur when the Oracle block is too small to hold the rows in question, make sure you are using a large enough block size for each database you create. An 8KB block size or higher is normally recommended to help stave off chained and migrated rows. If you are using Oracle9i or later, you can create special tablespaces that have larger block sizes (16-32KB) than the overall database block size and place any table that is a candidate for chained/migrated rows into them.</td>
</tr>
<tr>
<td>Use proper values of PCTFREE for tables</td>
<td>The necessary amount of percent free in a table helps prevent row migrations from occurring. If you have a database that houses rows with the potential to grow substantially over their initially inserted size, provide a liberal amount of PCTFREE for each of your tables.</td>
</tr>
</tbody>
</table>

**Chained Tables**
A row of data should fit completely inside one Oracle block. Sometimes, a table suddenly contains chained or migrated rows, which are rows that span more than one data block.

Chaining occurs when a row is initially too large to fit inside one block. Two or more blocks are used by Oracle to hold the row. Migration deals with rows that have grown so much that they can no longer be contained within their original block. Oracle relocates the row out of its original block into another block, but leaves a pointer behind to indicate the relocation.

Both chaining and migration force Oracle to perform more than one I/O to retrieve data that could normally be obtained with a single I/O operation, with the end result being degraded performance.

The Chained Tables statistic identifies the number of tables in the database that contain chained or migrated rows.

**Metrics**
Seeing many chained/migrated tables could indicate a database whose block size is undersized. However, even one chained/migrated row table can be a problem if that table is an often-referenced or scanned table.
Troubleshooting
You can locate the actual tables that contain chained/migrated rows. Once found, there are a couple of ways to reorganize tables to remove the chaining/migration problem. However, the best way to deal with chained and migrated rows is to prevent them from occurring. The table below describes two methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use a large block size</td>
<td>Because chaining and migrations occur when the Oracle block is too small to hold the rows in question, make sure you are using a large enough block size for each database you create. An 8KB block size or higher is normally recommended to help stave off chained and migrated rows. If you are using Oracle9i or later, you can create special tablespaces that have larger block sizes (16-32KB) than the overall database block size and place any table that is a candidate for chained/migrated rows into them.</td>
</tr>
<tr>
<td>Use proper values of PCTFREE for tables</td>
<td>The necessary amount of percent free in a table helps prevent row migrations from occurring. If you have a database that houses rows with the potential to grow substantially over their initially inserted size, provide a liberal amount of PCTFREE for each of your tables.</td>
</tr>
</tbody>
</table>

For more information, see [Bottleneck Analysis](#).

**Objects at Maximum Extents**
When an object in Oracle (table, index, table partition, etc.) needs to expand, Oracle automatically allocates another extent of space to accommodate the incoming data. However, a table or index can reach its maximum extent limit, which is the maximum number of extents for an object.

**NOTE:** This statistic is available on the Objects home page and the Space home page.

**Metrics**
Unless an object is static in nature, and is not expected to grow in size, no object should ever get to its maximum extent limit.

**Troubleshooting**
If a maximum extent limit is reached for an object, you can take a number of actions:

- The object can be reorganized into a locally-managed tablespace where no maximum extent limit is applied to any object.
- The object can be altered to have unlimited extents.
- The object can be reorganized into one or a few extents.

The good news concerning locally-managed tablespaces is that unlimited extents are the rule, so if you choose to use these storage structures in your database, you will not have to worry about an object ever reaching a maximum extent limit. Some DBAs have expressed concerns over whether an object having hundreds or thousands of extents experiences performance problems when full table scans or similar operations are performed against them. While most Oracle experts agree that such I/O degradation might have been experienced in earlier Oracle versions, most feel that such is not the case any longer, especially when locally-managed tablespaces are used.

**Objects with No Room to Expand**
When an object in Oracle (table, index, table partition, etc.) needs to expand, Oracle automatically allocates another extent of space to accommodate the incoming data. However, an object may be prohibited from expanding for a number of reasons:
• The tablespace does not have enough free space to accommodate the extent request, and cannot automatically extend.

• The tablespace has enough total free space, but the space is not contiguous in nature (and the tablespace cannot automatically extend) and therefore the object cannot extend.

  **NOTE:** This statistic is available on the Objects home page and the Space home page.

**Metrics**

Unless an object is static in nature, and is not expected to grow in size, no object should ever be prohibited from acquiring its next extent of space.

**Troubleshooting**

If an object cannot extend, you can take a number of actions including:

• One or more of the tablespace’s datafiles can have its AUTOEXTEND property enabled, which lets the object to grow even though there is no space currently available in the tablespace.

• The object can be reorganized into a locally-managed tablespace all extents are uniform in nature.

• The object can be altered to have a reduced next extent size.

**Invalid/ Objects**

**Metrics**

Objects like procedures, packages, functions, triggers, and views can become invalidated for a variety of reasons. The main cause is generally a dependent object that has been altered or removed from the system. However, other objects, like indexes, can become invalid also due to scenarios like SQL*Loader problems. If an object that has become invalid is still referenced (through an application or SQL query tool), a variety of problems can result. Sometimes Oracle reports a clear error stating the problem, while other times seemingly odd behavior is exhibited by the database.

**Troubleshooting**

If code objects have become invalidated, you can issue an ALTER … COMPILE command to see if they compile properly and become valid once again. If they do not, then check the USER_ERRORS view for any resulting error messages. Indexes can be validated once more by using the ALTER INDEX … REBUILD command.

For more information, see [Bottleneck Analysis](#).
Table High Water Mark Problems
One form of object fragmentation revolves around what is called a table's high water mark. Tables that are the victim of much insert-delete activity can develop performance problems due to wasted space being present. A table's high-water mark equals the last block in the table that was used to hold data. The problem in Oracle is that this high-water mark is not reset by DELETE activity, so it is possible for a table to have absolutely no data in it but contain a high-water mark that is many blocks high. When such a table is scanned, Oracle will read up to the high-water mark even if no rows exist in the table at all. This can make for some unnecessarily large scan times. For a real world example, see Bottleneck Analysis - Users.

Metrics
Seeing non-zero numbers for this metric could indicate that table scans are taking longer than they should.

Troubleshooting
Performance Analyst does not currently provide diagnostics that identify the actual tables that contain artificially inflated high water marks. Embarcadero Space Analyst provides all the levels of detail regarding table high water marks plus a reorganization wizard that can reset the high water mark and thereby improve performance of table scans against tables suffering from this problem.

Tables/Indexes in Same Tablespace
One common Oracle practice is to physically separate tables and their child indexes onto distinct physical drives or file systems. Doing so reduces the potential for I/O contention when each object is accessed. The Tables/Indexes in Same Tablespace statistic displays a count of all tables whose index(es) reside in the same tablespace as they do.

Metrics
For development databases where performance is not an issue, seeing high counts of this metric is not an issue. Even on production systems, high counts of this metric may not be a problem, or cannot be avoided, if the server is configured with one large RAID5 disk configuration, each distinct drive or file system does not have its own controller, or there simply are not enough disks on the server to perform adequate object striping.

Troubleshooting
If the server is configured to allow for intelligent object striping, and you see a large count of tables and indexes that are housed in the same tablespace, you can:

• Create a separate tablespace for indexes on a drive different than that used for tables, and perform an ALTER INDEX … REBUILD operation on all indexes to move them to the new tablespace. Certain versions of Oracle even allow the rebuild operation to occur online (using the ONLINE keyword of the REBUILD command).
• Use the export/import utility to move indexes to a new tablespace.
• Use the ALTER TABLE ... MOVE command to move tables to a new tablespace in Oracle8i and later. Index-organized tables can be moved online.

For more information, see Bottleneck Analysis.

Locked Objects
• Metrics
Locked Objects is a count of all objects on the system that currently have some form of lock against them.
Metrics
None.

NOTE: Drilling down into the count of locked objects displays detail on each object that is locked, along with the user process holding the lock and the type of lock held.

For more information, see Bottleneck Analysis.

Session Blocks
A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

NOTE: This statistic is also called Current Object Blocks on the Home page and Sessions Blocked on the Users home page.

Metrics
Investigate any indicator above zero immediately before the situation has a chance to mushroom.

Troubleshooting
Once discovered, a blocking lock situation can normally be quickly remedied. You can issue a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. You can look at the Users Detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT … FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Free List Waits
Free lists are lists of Oracle data blocks that contain free space for an Oracle object. Every table has at least one free list. Free lists are used to locate free blocks of space when a request is made of a table for the insertion of a row. Free list contention can reduce the performance of applications when many processes are involved in the insertion of data to the same table.

NOTE: This statistic is available on the Home page and the Objects home page.

Metrics
If consistent numbers for free list waits continues to appear, add additional free lists to the most dynamic database objects through the use of the STORAGE parameter. Another indicator of insufficient free lists are consistent, non-zero numbers for the buffer busy wait event.
Troubleshooting
If free list waits are observed, add more free lists to tables and indexes with high insert rates. For Oracle9i or later, objects can be created inside of locally-managed tablespaces that use the automatic segment management feature, which entirely does away with free lists.

Enqueue Waits
An enqueue is an advanced locking device that allows multiple database processes to share certain resources. Enqueue waits typically occur when sessions wait to be granted a requested lock. Sometimes these locks are internal Oracle locks while other times they could be locks for rows of data in a table. Enqueues are issued implicitly by Oracle.

NOTE: This statistic is available on the Home page and the Objects home page.

Metrics
Investigate any enqueue waits that read consistently above one or more (delta statistics).

Troubleshooting
Removing contention for enqueues is almost always an application design issue. If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. Examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations.

Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).

Objects Detail
The following tabbed pages are available on the Objects Detail page:

<table>
<thead>
<tr>
<th>Hot Objects</th>
<th>Invalid Objects</th>
<th>Objects Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table and Index Contention</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hot Objects Tab
The Hot Objects tab includes the following sections:

- Hot Code
- Hot Tables

For more information, see Objects Page Statistics.

Hot Tables
Certain objects in an Oracle database are accessed more than others. These objects can become a source of contention given certain conditions. The Hot Tables section identifies tables that are being frequently accessed through various SQL statements. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>The owner of the table.</td>
</tr>
</tbody>
</table>
Metrics
DML activity against tables can cause contention for space management objects like free lists. Oracle9i and later provides automatic segment management, which can remove problems with free lists and the like.

Hot Code
Certain objects in an Oracle database are accessed more than others. Data objects can become a source of contention given certain conditions, while code objects rarely cause contention issues. The Hot Code section identifies code objects (procedure, functions, etc.) that are being frequently accessed through various SQL statements. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Owner</td>
<td>The owner of the object.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the objects.</td>
</tr>
<tr>
<td>Object Type</td>
<td>The type of object (package, etc.).</td>
</tr>
<tr>
<td>Executions</td>
<td>The number of estimated executions for the object.</td>
</tr>
<tr>
<td>Loads</td>
<td>The number of times the object was loaded into the shared pool.</td>
</tr>
<tr>
<td>Locks</td>
<td>The number of locks the object has experienced.</td>
</tr>
<tr>
<td>Pins</td>
<td>The number of times the object was pinned in the shared pool.</td>
</tr>
</tbody>
</table>

Metrics
Often referenced code objects should be pinned in the shared pool using the Oracle DBMS_SHARED_POOL package. This can also be done through DBArtisan’s pinned code utility. Objects with many executions and loads should be considered candidates for pinning.

Invalid Objects Tab
The Invalid Objects tab includes the following sections:

- Invalid Objects Detail
- Invalid Objects Summary

For more information, see Objects Page Statistics.
Objects like procedures, packages, functions, triggers, and views can become invalidated for a variety of reasons, with the main cause being a dependent object that has been altered or removed from the system. However, other objects, like indexes, can become invalid also due to scenarios like SQL*Loader problems. If an object that has become invalid is still referenced (through an application or SQL query tool), a variety of problems can result. Sometimes Oracle will report a clear error stating the problem, while other times seemingly quirky behavior will be exhibited by the database. In any event, as a DBA you should be on the lookout for objects in your database that have suddenly become invalid.

The Invalid Objects Summary section displays a count of the various objects in the database that are invalid.

**Metrics**

There is no reason to have invalid objects in a production database. If your production databases have invalid/unusable objects that are no longer needed, you should promptly remove them from each system. Any needed objects that are indicating an invalid status should quickly be fixed before access problems develop.

Correcting invalid objects like procedures and views often involves performing an ALTER...COMPILE operation. If the object status does not return to VALID, then further examination is warranted.

It is very normal for development databases to have invalid objects because developers will no doubt be creating, modifying, and compiling objects all the time. The only invalid object that really should not be present in either a development or production database is an invalid index.

### Invalid Objects Detail

Objects like procedures, packages, functions, triggers, and views can become invalidated for a variety of reasons, with the main cause being a dependent object that has been altered or removed from the system. However, other objects, like indexes, can become invalid also due to scenarios like SQL*Loader problems. If an object that has become invalid is still referenced (through an application or SQL query tool), a variety of problems can result. Sometimes Oracle will report a clear error stating the problem, while other times seemingly quirky behavior will be exhibited by the database. In any event, as a DBA you should be on the lookout for objects in your database that have suddenly become invalid.

The Invalid Objects Detail section displays information concerning invalid objects. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Account</td>
<td>The user account that owns the objects.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the invalid object.</td>
</tr>
<tr>
<td>Object Type</td>
<td>The type of object (procedure, view, etc.).</td>
</tr>
<tr>
<td>Created On</td>
<td>The timestamp of when the objects was created.</td>
</tr>
<tr>
<td>Last DDL Date</td>
<td>The last structural modification date for the object.</td>
</tr>
</tbody>
</table>

**Metrics**

There is no reason to have invalid objects in a production database. If your production databases have invalid objects that are no longer needed, you should promptly remove them from each system. Any needed objects that are indicating an invalid status should quickly be fixed before access problems develop.

Correcting invalid objects like procedures and views often involves performing an ALTER...COMPILE operation. If the object status does not return to VALID, then further examination is warranted.

It is very normal for development databases to have invalid objects because developers will no doubt be creating, modifying, and compiling objects all the time. The only invalid object that really should not be present in either a development or production database is an invalid index.
Objects Summary Tab

The Objects Summary tab includes the following sections:

- Global Summary
- User Summary

The table below describes these sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Summary</td>
<td>This section displays a count of all user objects in the database (minus SYS and SYSTEM objects).</td>
</tr>
<tr>
<td>User Summary</td>
<td>This section displays a count across user accounts of all the major object types in the database. <strong>NOTE</strong> SYS and SYSTEM accounts are not included.</td>
</tr>
</tbody>
</table>

For more information, see Objects Page Statistics.

Tables/Indexes in Same Tablespace Tab

To help avoid physical I/O contention, it is wise to separate a table and its indexes onto separate physical drives. This is normally accomplished by putting tables into one tablespace (whose datafiles exist on one drive) and their indexes into a different tablespace (whose datafiles exist on another drive).

The Tables/Indexes in Same Tablespace tab identifies indexes whose tables reside in the same tablespace as they do. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>The owner of the table.</td>
</tr>
<tr>
<td>Table Name</td>
<td>The name of the table.</td>
</tr>
<tr>
<td>Index Name</td>
<td>The name of the index that resides in the same tablespace as the table.</td>
</tr>
<tr>
<td>Tablespace Name</td>
<td>The name of the tablespace.</td>
</tr>
</tbody>
</table>

**Metrics**

If tables and indexes are found in the same tablespace, then either the table or index should be moved to a different tablespace if the server is set up in such a way that different drives and controllers are available that make the relocation worthwhile.

Indexes are traditionally easier to move than tables. They can be relocated through many reorganization tools (like Embarcadero Space Analyst) or through the manual use of the ALTER INDEX … REBUILD command.

**OS Page Statistics**

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Analyst lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX

**NOTE:** To view processor info and swap disk info on an HP-UX box, you need to login as ROOT in the OS login.
• Linux
• Solaris
• Unix
• Windows XP and 2000

**NOTE:** The statistics available on the OS page depend on the operating system platform.

**TIP:** If you magnify a graph to change back to the original size or minimize the size, close the OS Page and then reopen it again. Performance Analyst displays the graph in its original form and size.

The OS home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Disk Analysis</th>
<th>Key Resource Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Analysis</td>
<td>Workload Analysis - Top Resource Hogs</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topic**

**OS Detail**

**Key Resource Usage - OS**

The following ratios are used on the Performance Analyst OS home page to communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Disk Time</th>
<th>Load Average</th>
<th>Processor Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paged Memory Used (Windows)</td>
<td>Swap Memory Used (AIX, HP-UX, Linux, Solaris, Unix)</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the CPU tab, right-click Processor Time. To open the I/O tab, right-click Disk Time. To open the Memory tab, right-click Paged Memory Used.

**Bottleneck Analysis - OS**

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Average Disk Queue Length</th>
<th>Interrupts/Sec</th>
<th>Network Output Queue Length (Windows)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Queue Length (Solaris)</td>
<td>Page Faults/Sec</td>
<td>Processor Queue Length</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>Processor</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the I/O tab, right-click any Details menu item. To open the CPU tab, right-click the Item Processor Queues Length Details menu item. To open the Network tab, right-click Network Output Queues Length.

**NOTE:** The statistics available in this section depend on the platform of operating system.
Memory Analysis - OS
The following metrics are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Available Paged Memory (Windows)</th>
<th>Available Physical Memory</th>
<th>Available Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Paged Memory (Windows)</td>
<td>Total Physical Memory</td>
<td>Total Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)</td>
</tr>
</tbody>
</table>

**TIP:** To open the Memory tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Disk Analysis - OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Free Disk Space</th>
<th>Total Disk Space</th>
<th>Used Disk Space</th>
</tr>
</thead>
</table>

**TIP:** To open the Space tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Workload Analysis - Top Resource Hogs OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Number of Logins</th>
<th>Number of Processes</th>
<th>Top CPU Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top I/O Process</td>
<td>Top Memory Process</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the Processes tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Processor Time
The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

**Metrics**
If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.

**Troubleshooting**
Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.
Processor Speed
The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Processor
The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Disk Time
The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

Metrics
You should avoid consistently seeing values for this statistic greater then 90%.

Troubleshooting
Add more disk drives and partition the files among all of the drives.

Load Average
The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Metrics
High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Paged Memory Used
The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

NOTE: This statistic is available for the Windows platform.

Metrics
This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.

Troubleshooting
Increase the size of page file.

Swap Memory Used
The Swap Memory Used statistic is the percentage of swap space currently in use.
Metrics
If the percentage of swap memory used is above 90%, you may be running out of memory.

Troubleshooting
Increase the size of your swap files.

Average Disk Queue Length
The Average Disk Queue Length statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Metrics
This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.

Troubleshooting
Some things you can do if you have problems with this statistic include:

- Redistribute the data on the disk with the large average disk queue to other disks.
- Upgrade to faster disk(s).

Interrupts/Sec
Interrupts/Sec is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every 10 milliseconds, creating a background of interrupt activity. This statistic shows the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Metrics
A high value indicates possible excessive hardware interrupts; justification is dependent on device activity.

Page Faults/Sec
The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics
This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Troubleshooting
If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue Length
The Processor Queue Length statistic is the number of threads in the processor queue.

Metrics
Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Network Output Queue Length/Network Queue Length
The Network Output Queue Length statistic is the number of threads in the processor queue.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics
Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Available Physical Memory
The Available Physical Memory statistic represents the amount of RAM available to all processes.

Metrics
This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Troubleshooting
Some things you can do if you have problems with this statistic include:
- Check the running processes to see if there are any memory leaks.
- Stop any services that are not required.
- Install additional RAM.

Available Paged Memory
The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

**NOTE:** This statistic is available for the Windows platform.

**Metrics**
If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.

**Troubleshooting**
Increase the size of page file.

Available Swap Memory
The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

**Metrics**
If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.

**Troubleshooting**
Increase the size of swap files.

Total Physical Memory
The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

**Metrics**
This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Total Paged Memory/Total Swap Memory
The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

**NOTE:** The name of this statistic depends on the platform of the operating system.

**Metrics**
It is recommended that this be 1½ - 3 times the amount of RAM on the system.
Used Disk Space
The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Total Disk Space
Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two:

1. If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
2. You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Free Disk Space
The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Top Memory Process
Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics
If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.
Processes Overview
The Processes Overview of the OS Summary includes the following sections:

<table>
<thead>
<tr>
<th>Top CPU Process</th>
<th>Top I/O Process</th>
<th>Top Memory Process</th>
</tr>
</thead>
</table>

**Top CPU Process**
Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

**Metrics**
If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.

**Troubleshooting**
Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

**Top I/O Process**
The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

**Number of Logins**
This statistic displays the total number of logins on the server.

**Number of Processes**
This statistic displays the total number of processes on the server.

**OS Detail**
The following tabbed pages are available on the OS Detail page:

<table>
<thead>
<tr>
<th>CPU</th>
<th>I/O</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Processes</td>
<td>Space</td>
</tr>
</tbody>
</table>
CPU Tab
The CPU tab of the OS Detail includes the following sections:

<table>
<thead>
<tr>
<th>Context Switches/Sec</th>
<th>CPU Events</th>
<th>CPU Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrupts/Sec</td>
<td>Load Averages</td>
<td>Processor Queue Length</td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of the operating system.

CPU Utilization
The CPU Utilization section includes the following information:

- **% Privileged Time**
- **% User Time**

**% Privileged Time**
The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

**NOTE:** For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

**Metrics**
The ideal range should be 0-40% (less than 40% indicates excessive system activity).

**Troubleshooting**
If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

**% User Time**
The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

**Metrics**
If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.
CPU Events
The CPU Events section includes the following information:

<table>
<thead>
<tr>
<th>Interrupts/Sec</th>
<th>Context Switches/Sec</th>
<th>System Calls/Sec</th>
</tr>
</thead>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

**Interrupts/Sec**
The Interrupts/Sec statistic is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. It does not include deferred procedure calls (DPCs), which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every ten milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

**Metrics**
The ideal range should be 0-5000. A number greater than 5000 indicates possible excessive hardware interrupts; justification is dependent on device activity.

**Context Switches/Sec**
The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

**Metrics**
The ideal range should be between 0-10,000. A number greater than 10,000 may indicate too many threads contending for resources.

**System Calls/Sec**
System calls are functions which require the services of the operating system. Usually they involve access to some data that users must not be permitted to corrupt or even change.

**Troubleshooting**
Try to find a good ratio between the amount of time the operating system runs (fulfilling system calls and doing process scheduling), and the amount of time the application runs. Your goal should be running 60% to 75% of the time in application mode, and 25% to 40% of the time in operating system mode. If you find that the system is spending 50% of its time in each mode, then you should investigate to determine what is wrong.

**Processor Queue Length**
The Processor Queue Length statistic is the number of threads in the processor queue. There is a single queue for processor time even on computers with multiple processors.

**NOTE:** For Windows systems, unlike the disk counters, this counter shows ready threads only, not threads that are running.
Metrics
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Load Averages
The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Metrics
High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Processes Tab
The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>The name of the process.</td>
</tr>
<tr>
<td>User</td>
<td>The user of the process.</td>
</tr>
<tr>
<td>ID</td>
<td>The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.</td>
</tr>
<tr>
<td>CPU</td>
<td>The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.</td>
</tr>
<tr>
<td>User Mode</td>
<td>The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.</td>
</tr>
<tr>
<td>Memory</td>
<td>Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.</td>
</tr>
<tr>
<td>Memory (MB)</td>
<td>Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.</td>
</tr>
<tr>
<td>Active Memory</td>
<td>Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.</td>
</tr>
<tr>
<td>I/O Data</td>
<td>The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>The total elapsed time, in seconds, that this process has been running.</td>
</tr>
<tr>
<td>Thread Count</td>
<td>The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.</td>
</tr>
<tr>
<td>Handle Count</td>
<td>The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.</td>
</tr>
<tr>
<td>Priority</td>
<td>The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.</td>
</tr>
</tbody>
</table>
The I/O tab includes the following sections:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Disk Queue</td>
<td>Bytes per I/O Operation</td>
</tr>
<tr>
<td>Disk Transfers /Sec</td>
<td>I/O Details</td>
</tr>
<tr>
<td>Total Disk Queue</td>
<td>Transfer Rate</td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of operating system.

**I/O Details**

The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk</td>
<td>The disk number assignment.</td>
</tr>
<tr>
<td>Reading (KB/s)</td>
<td>The amount of bytes read from the device.</td>
</tr>
<tr>
<td>Writing (KB/s)</td>
<td>The amount of bytes written to the device.</td>
</tr>
<tr>
<td>Disk Read Time</td>
<td>Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.</td>
</tr>
<tr>
<td>Disk Write Time</td>
<td>Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.</td>
</tr>
<tr>
<td>Disk Time</td>
<td>Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.</td>
</tr>
<tr>
<td>Avg. Read Queue</td>
<td>Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.</td>
</tr>
<tr>
<td>Avg. Write Queue</td>
<td>Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.</td>
</tr>
<tr>
<td>Disk Reads/Sec</td>
<td>Disk Reads/Sec is the rate of read operations on the disk.</td>
</tr>
<tr>
<td>Disk Writes/Sec</td>
<td>Disk Writes/Sec is the rate of write operations on the disk.</td>
</tr>
</tbody>
</table>

**NOTE:** The columns available in this table depend on the platform of operating system.
Transfer Rate
The Transfer Rate section succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading</td>
<td>The average number of bytes transferred from the disk during read operations.</td>
<td>It is measured in number of KBs per second.</td>
</tr>
<tr>
<td>Writing</td>
<td>The average number of bytes transferred to the disk during write operations.</td>
<td>It is measured in number of KBs per second.</td>
</tr>
<tr>
<td>Transferred</td>
<td>The amount of time required to transfer the data between the device's cache and the actual device media.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Bytes per I/O Operation
The Bytes per I/O Operation section of the OS Detail succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading</td>
<td>The average number of bytes transferred from the disk during read operations.</td>
<td>It is measured in number of bytes per second.</td>
</tr>
<tr>
<td>Writing</td>
<td>The average number of bytes transferred to the disk during write operations.</td>
<td>It is measured in number of bytes per second.</td>
</tr>
</tbody>
</table>

Average Disk Queue/Total Disk Queue
The Average Disk Queue section of the OS Detail page succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Read Queue</td>
<td>Average Read Queue Length is the average number of read requests that were queued for a particular disk.</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system.</td>
</tr>
<tr>
<td>Average Write Queue</td>
<td>Average Write Queue is the average number of write requests that were queued for a particular disk.</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Windows System Administrator for assistance in troubleshooting these problems.</td>
</tr>
<tr>
<td>Average Queue</td>
<td>Average time in ms. to service I/O requests which for disks includes seek, rotational latency and data transfer time.</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Unix System Administrator for assistance in troubleshooting these problems.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Troubleshooting
Contact your Windows System Administrator for assistance in troubleshooting these problems.
Disk I/O Time
The Disk I/O Time section of the OS Detail page succinctly communicates the general overall performance levels of the disk’s I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Disk Read Time</td>
<td>% Disk Read Time is the percentage of time during the sample interval that the disk is busy servicing read requests.</td>
<td>Use this in conjunction with the % Disk Write Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
<tr>
<td>% Disk Write Time</td>
<td>% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.</td>
<td>Use this in conjunction with the % Disk Read Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
<tr>
<td>% Disk Time</td>
<td>% Disk Time is the percentage of time during the sample interval that the disk is busy servicing requests.</td>
<td>Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
</tbody>
</table>

NOTE: The statistics available in this section depend on the platform of operating system.

Disk Transfers/Sec
The Disk Transfers/Sec section of the OS Detail page succinctly communicates the general overall performance levels of the disk transfers. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads/Sec</td>
<td>The rate of read operations on the disk.</td>
</tr>
<tr>
<td>Writes/Sec</td>
<td>The rate of write operations on the disk.</td>
</tr>
<tr>
<td>Transfers/Sec</td>
<td>The rate of transfer operations per second.</td>
</tr>
</tbody>
</table>

NOTE: The statistics available in this section depend on the platform of operating system.

Memory Tab
The Memory tab of the OS Detail page includes the following sections:

<table>
<thead>
<tr>
<th>Buffer Size</th>
<th>Cache Efficiency</th>
<th>Cache Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Available</td>
<td>Paging Activity</td>
<td>Page Faults</td>
</tr>
<tr>
<td>Page Replacements</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE: The sections available on this tab depend on the platform of operating system.
Paging Activity
The Paging Activity section includes the following statistics:

<table>
<thead>
<tr>
<th>Blocks Input/Sec</th>
<th>Blocks Output/Sec</th>
<th>Paged In</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paged Out</td>
<td>Pages Input/Sec</td>
<td>Pages Output/Sec</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

**Blocks Input/Sec**
The Blocks Input/sec statistic is the number of blocks sent to a block device per second.

**Troubleshooting**
Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck:

1. Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
2. Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.

**Blocks Output/Sec**
The Blocks Output/sec statistic is the number of blocks received from a block device per second.

**Troubleshooting**
Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck:

1. Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
2. Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.

**Paged In**
The number of disk block paged into core from disk which occurs when the server is experiencing a shortage of RAM. Any nonzero value indicates excessive swap activity. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

**Troubleshooting**
High values indicate the possible need to increase the amount of RAM on the system.
Paged Out
The number of disk block paged out of core from disk. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Pages Input/Sec
The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Metrics
This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Pages Output/Sec
The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Metrics
Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.
Page Faults/Sec
The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics
This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Memory Available
The Memory Available section of the OS Detail page succinctly communicates the general overall performance levels of the server's memory. The statistics available in this section depend on the platform of operating system. The table below describes these statistics:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Memory</td>
<td>The Physical Memory statistic is the amount of committed virtual memory, in megabytes. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk.</td>
<td>This counter displays the last observed value only and not an average.</td>
<td>N/A</td>
</tr>
<tr>
<td>Physical</td>
<td>The Physical statistic is the total physical memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Paged Memory</td>
<td>The Paged Memory statistic is the amount of committed virtual memory, in bytes. Used Memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.</td>
<td>This counter displays the last observed value only; it is not an average.</td>
<td>N/A</td>
</tr>
<tr>
<td>Swap Memory</td>
<td>The Swap Memory statistic is the free, unreserved swap space.</td>
<td>N/A</td>
<td>Increase the size of your swap file or add additional files if you consistently see a shortage of free, unreserved swap space.</td>
</tr>
<tr>
<td>Swap</td>
<td>The Swap statistic is the total swap memory being used.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total Physical</td>
<td>The Total Physical statistic is the total physical memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total Swap</td>
<td>The Total Swap statistic is the total swap memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Swap-ins</td>
<td>The number of memory pages paged in from swapspace.</td>
<td>N/A</td>
<td>If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.</td>
</tr>
</tbody>
</table>
NOTE: The statistics available in this section depend on the platform of operating system.

Buffer Size
The Buffer Size statistic represents the amount of cache used by the kernel in kilobytes.

Cache Size
The Cache Size statistic represents the amount of cache used by the kernel in kilobytes.

Page Replacements
The following statistics are available in this section:

- Memory Freed (Pages/sec)
- Clock Algorithm Scans (Pages/sec)

Memory Freed (Pages/sec)
Freed or destroyed (Kb/s).

Clock Algorithm Scans (Pages/sec)
The VMM uses a technique known as the clock algorithm to select pages to be replaced. This technique takes advantage of a referenced bit for each page as an indication of what pages have been recently used (referenced). When the page-stealer routine is called, it cycles through the PFT, examining each page's referenced bit.
Cache Efficiency
The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

<table>
<thead>
<tr>
<th>Copy Read Hits%</th>
<th>Data Map Hits%</th>
<th>MDL Read Hits%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin Read Hits%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copy Read Hits %
The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

**Metrics**
When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

**Troubleshooting**
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Data Map Hits %
The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

**Metrics**
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

**Troubleshooting**
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %
The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

**Metrics**
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

**Troubleshooting**
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %
The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.
Metrics
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab
The Space tab of the OS Detail page includes the following sections:

- Device Summary
- Device Detail

Disk Space Free
The Disk Space Free metric displays the amount of free disk space in megabytes.

Metric
None.

Disk Space Detail
The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server’s disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>The drive letter of the disk.</td>
</tr>
<tr>
<td>Local Filesystem</td>
<td>The name of the file system.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of file system.</td>
</tr>
<tr>
<td>Total Space</td>
<td>Total size of the disk/device's capacity expressed in MBs.</td>
</tr>
<tr>
<td>Used Space</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free Space</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
<tr>
<td>Capacity</td>
<td>The percentage of space used on the device.</td>
</tr>
<tr>
<td>Mounted On</td>
<td>The mount point of the device.</td>
</tr>
</tbody>
</table>

Device Summary
The Device Summary section of the Space tab displays a bar chart for all devices. The Device Summary section succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
</tbody>
</table>
Device Detail
The Device Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>The drive letter of the disk.</td>
</tr>
<tr>
<td>Local Filesystem</td>
<td>The name of the file system.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of file system.</td>
</tr>
<tr>
<td>Total Space</td>
<td>Total size of the disk/device's capacity expressed in MBs.</td>
</tr>
<tr>
<td>Used Space</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free Space</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
<tr>
<td>Capacity</td>
<td>The percentage of space used on the device.</td>
</tr>
<tr>
<td>Mounted On</td>
<td>The mount point of the device.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Network Tab
The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The Network tab of the OS Detail page includes the following sections:

<table>
<thead>
<tr>
<th>Network Details</th>
<th>Packet Collisions</th>
<th>Packet Discards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Errors</td>
<td>Transmission Rate</td>
<td>Transmission Rate (Bytes)</td>
</tr>
<tr>
<td>Transmission Queue Length</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of operating system.

Network Details
The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Interface</td>
<td>The name of network interface.</td>
</tr>
<tr>
<td>INET Address/Address</td>
<td>The IP address assigned to the network interface.</td>
</tr>
<tr>
<td>Pkts Sent/Sec</td>
<td>The number of packets sent per second.</td>
</tr>
<tr>
<td>Pkts Received/Sec</td>
<td>The number of packets received per second.</td>
</tr>
<tr>
<td>Sent (KB/Sec)</td>
<td>The number of bytes sent per second.</td>
</tr>
<tr>
<td>Received (KB/Sec)</td>
<td>The number of bytes received per second.</td>
</tr>
<tr>
<td>Out Pkts Discarded</td>
<td>The number of outbound packets discarded.</td>
</tr>
<tr>
<td>In Pkts Discarded</td>
<td>The number of inbound packets discarded.</td>
</tr>
<tr>
<td>Out Pkt Errors</td>
<td>The number of outbound packet errors.</td>
</tr>
<tr>
<td>In Pkt Errors</td>
<td>The number of inbound packet errors.</td>
</tr>
</tbody>
</table>
Transmission Rate
The Transmission Rate section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

- **Packets Received/Sec**
- **Packets Sent/Sec**

Packets Sent/Sec
The Packets Sent/Sec statistic is the number of packets sent over each network adapter per second.

Packets Received/Sec
The Packets Received/Sec statistic is the number of packets received over each network adapter per second.

Transmission Rate (Bytes)
The Transmission Rate (Bytes) section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

- **Received (KB/Sec)**
- **Sent (KB/Sec)**

Sent (KB/Sec)
The Sent (KB/Sec) statistic is the rate at which bytes are sent over each network adapter including framing characters.

Received (KB/Sec)
The Received (KB/Sec) statistic is the rate at which bytes are received over each network adapter, including framing characters.

Transmission Queue Length
The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue

Packet Collisions
A collision occurs when two devices on the network sense that the network is idle and end up trying to send data at the same time. Some collisions are normal on a repeated network, but excess collisions can cause serious performance problems.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue Length</td>
<td>The queue length.</td>
</tr>
<tr>
<td>Collisions</td>
<td>The number of collisions.</td>
</tr>
<tr>
<td>Packets Discarded</td>
<td>The number of packets discarded.</td>
</tr>
</tbody>
</table>
Troubleshooting
Contact your network administrator if there is high rate of persistent Packet Collisions.

Packet Discards
The Packet Discards section of the Network tab succinctly communicates the packet discard rate. The following statistics are available in this section:

<table>
<thead>
<tr>
<th>Inbound Packets Discarded</th>
<th>Outbound Packets Discarded</th>
<th>Packet Discards</th>
</tr>
</thead>
</table>

**Outbound Packets Discarded**
The Outbound Packets Discarded statistic is the number of outbound packets that were discarded even though no errors had been detected to prevent from being transmitted.

**Troubleshooting**
A possible reason for discarding such a packet could be to free up buffer space.

**Inbound Packets Discarded**
Received Packets Discarded is the number of received packets that were discarded even though no errors had been detected to prevent from being transmitted.

**Troubleshooting**
A possible reason for discarding such a packet could be to free up buffer space.

**Packet Discards**
The Packet Discards statistic represents the number of network packets discarded.

**Troubleshooting**
Contact your network administrator if there is high rate of persistent packet discards.

Packet Errors
The Packet Errors section of the Network tab succinctly communicates the packet error rate. The following statistics are available in this section:

- Inbound Packet Errors
- Outbound Packet Errors

**Outbound Packet Errors**
The Outbound Packet Errors statistic is the outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
Inbound Packet Errors
The Packets Received Errors statistic is the received packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Top SQL
One or two bad queries can cause a lot of trouble for the remaining sessions in a database, and so can one or two bad queries. It is important to find them before they get into a production system, but sometimes a few sneak through.

By applying custom filters and performance-related thresholds, the Top SQL view locates inefficient SQL. By applying filters to certain I/O and statistical counters, you hope to isolate queries that far exceed their nearest competitors in the same area (like disk reads). When you find them, you should reduce the number of sorts a query performs. Or, for a query that returns only a few records, you should try to minimize the number of rows a query processes.

When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer:

1. What has been the worst SQL that’s historically been run in my database?
2. What is the worst SQL that’s running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient Transact SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

The Top SQL view displays requested SQLOracle datasources. The following tabs are available on the Top SQL page:

<table>
<thead>
<tr>
<th>Summary</th>
<th>Criteria</th>
<th>SQL Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recent History (Oracle 10g only)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Summary
The Summary tab of the Top SQL page includes the following sections:

- Top SQL Statements
- Top SQL Details

Top SQL Statements
A lot of a database’s overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what ‘bad SQL’ is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional’s satisfaction. The table below describes some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Response (Elapsed) Time</td>
<td>This is how much time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i and later.</td>
</tr>
</tbody>
</table>
Top SQL Details
A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The Top SQL Details section displays the top SQL statements across various performance categories. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Time</td>
<td>This is how much CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.</td>
</tr>
<tr>
<td>Physical I/O</td>
<td>Often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.</td>
</tr>
<tr>
<td>Logical I/O</td>
<td>This number of memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.</td>
</tr>
<tr>
<td>Repetition</td>
<td>This is a measure of how often the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute may still be causing a headache on your system if it is executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over.</td>
</tr>
</tbody>
</table>

Top SQL Details
A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The Top SQL Details section displays the top SQL statements across various performance categories. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top For</td>
<td>Indicates the performance category for the SQL statement.</td>
</tr>
<tr>
<td>Usage</td>
<td>Displays the percent usage for this SQL statement in this performance category.</td>
</tr>
<tr>
<td>Hash value</td>
<td>A unique identifier for the SQL statement assigned by Oracle.</td>
</tr>
<tr>
<td>SQL text</td>
<td>A snippet of the actual SQL statement.</td>
</tr>
<tr>
<td>User name</td>
<td>The username of the account that originally parsed the SQL.</td>
</tr>
<tr>
<td>Disk reads</td>
<td>The number of disk reads for the statement.</td>
</tr>
<tr>
<td>Buffer gets</td>
<td>The number of logical reads for the statement.</td>
</tr>
<tr>
<td>Parse calls</td>
<td>The number of parse calls for the statement.</td>
</tr>
<tr>
<td>Sorts</td>
<td>The number of sorts caused by the statement.</td>
</tr>
<tr>
<td>Executions</td>
<td>The number of times that statement has been executed.</td>
</tr>
<tr>
<td>Rows processed</td>
<td>The number of rows processed by the statement.</td>
</tr>
<tr>
<td>First load time</td>
<td>The date/time of when the statement was first loaded into the shared pool.</td>
</tr>
<tr>
<td>Sharable memory</td>
<td>The amount of sharable memory used by the statement.</td>
</tr>
<tr>
<td>Persistent memory</td>
<td>The amount of persistent memory used by the statement.</td>
</tr>
<tr>
<td>Runtime memory</td>
<td>The amount of runtime memory used by the statement.</td>
</tr>
</tbody>
</table>
Criteria
The Top SQL view displays all SQL running or executed in the current datasource. You can use the lists and boxes to reorder the display according to your specific needs. The Criteria tab of the Top SQL page includes the following sections that you can mix and match to give you the information you want to find.

- Demographic Criteria
- Statistical Criteria
- Sort Criteria

SQL Details
The SQL Details tab displays the following sections:

- Top SQL Results
- Selected SQL Text

Top SQL Results
A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The Top SQL results grid displays the top SQL statements across various performance categories. The table below describes the information available in this section

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>The address of the SQL statement.</td>
</tr>
<tr>
<td>SQL Text</td>
<td>A snippet of the full SQL statement. Clicking the statement will cause the full statement to be presented in the Selected SQL Text grid.</td>
</tr>
<tr>
<td>User name</td>
<td>The username of the account that originally parsed the SQL.</td>
</tr>
<tr>
<td>Disk reads per execution</td>
<td>The average number of disk reads caused by the statement each time it is executed.</td>
</tr>
<tr>
<td>Buffer gets per execution</td>
<td>The average number of logical reads caused by the statement each time it is executed.</td>
</tr>
<tr>
<td>Buffer gets</td>
<td>The number of logical reads for the statement.</td>
</tr>
<tr>
<td>Disk reads</td>
<td>The number of disk reads for the statement.</td>
</tr>
<tr>
<td>Parse calls</td>
<td>The number of parse calls for the statement.</td>
</tr>
<tr>
<td>Sorts</td>
<td>The number of sorts caused by the statement.</td>
</tr>
<tr>
<td>Executions</td>
<td>The number of times that statement has been executed.</td>
</tr>
<tr>
<td>Rows processed</td>
<td>The number of rows processed by the statement.</td>
</tr>
<tr>
<td>Hit Ratio</td>
<td>The cache hit ratio for the statement.</td>
</tr>
<tr>
<td>First load time</td>
<td>The date/time of when the statement was first loaded into the shared pool.</td>
</tr>
<tr>
<td>Sharable memory</td>
<td>The amount of sharable memory used by the statement.</td>
</tr>
</tbody>
</table>
Selected SQL Text
The Selected SQL Text window allows you to select any of the Top SQL result rows and get the complete SQL text. From here you can open an Explain Plan or Tune the SQL.

Recent History (Oracle 10g only)
The Recent History tab of the Top SQL view displays:

- SQL Activity (last 24 hours)
- SQL Details

SQL Activity (last 24 hours)
The SQL Activity graph shows you the resource usage of SQL statements over the last 24-hour period. You can see cumulative CPU, elapsed time, and wait time for all SQL statements that were executed during the last 24 hours.

You can also select a specific time period in the graph by drawing a box around part of the graph with your mouse. The graph will change to match your selection.

SQL Details
- Metrics

The SQL Details section of the Recent History view displays the SQL statements that correspond to the time period shown in the SQL Activity graph. The following columns are displayed:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistent memory</td>
<td>The amount of persistent memory used by the statement.</td>
</tr>
<tr>
<td>Runtime memory</td>
<td>The amount of runtime memory used by the statement.</td>
</tr>
<tr>
<td>Elapsed Time per Exec</td>
<td>For Oracle9i and later, this is the average elapsed response time for the statement.</td>
</tr>
<tr>
<td>CPU Time per Exe</td>
<td>For Oracle9i and later, this is the average CPU time for the statement.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>For Oracle9i and later, this is the total CPU time for the statement.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>For Oracle9i and later, this is the total elapsed time for the statement over all executions.</td>
</tr>
<tr>
<td>Hash value</td>
<td>A unique identifier for the SQL statement assigned by Oracle.</td>
</tr>
<tr>
<td>Address</td>
<td>The address of the SQL statement.</td>
</tr>
<tr>
<td>Persistent memory</td>
<td>The amount of persistent memory used by the statement.</td>
</tr>
<tr>
<td>Runtime memory</td>
<td>The amount of runtime memory used by the statement.</td>
</tr>
<tr>
<td>Elapsed Time per Exec</td>
<td>For Oracle9i and later, this is the average elapsed response time for the statement.</td>
</tr>
<tr>
<td>CPU Time per Exe</td>
<td>For Oracle9i and later, this is the average CPU time for the statement.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>For Oracle9i and later, this is the total CPU time for the statement.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>For Oracle9i and later, this is the total elapsed time for the statement over all executions.</td>
</tr>
<tr>
<td>Hash value</td>
<td>A unique identifier for the SQL statement assigned by Oracle.</td>
</tr>
<tr>
<td>Address</td>
<td>The address of the SQL statement.</td>
</tr>
</tbody>
</table>

Information | Description
---|---
Persistent memory | The amount of persistent memory used by the statement.
Runtime memory | The amount of runtime memory used by the statement.
Elapsed Time per Exec | For Oracle9i and later, this is the average elapsed response time for the statement.
CPU Time per Exe | For Oracle9i and later, this is the average CPU time for the statement.
CPU Time | For Oracle9i and later, this is the total CPU time for the statement.
Elapsed Time | For Oracle9i and later, this is the total elapsed time for the statement over all executions.
Hash value | A unique identifier for the SQL statement assigned by Oracle.
Address | The address of the SQL statement.

Selected SQL Text
The Selected SQL Text window allows you to select any of the Top SQL result rows and get the complete SQL text. From here you can open an Explain Plan or Tune the SQL.

Recent History (Oracle 10g only)
The Recent History tab of the Top SQL view displays:

- SQL Activity (last 24 hours)
- SQL Details

SQL Activity (last 24 hours)
The SQL Activity graph shows you the resource usage of SQL statements over the last 24-hour period. You can see cumulative CPU, elapsed time, and wait time for all SQL statements that were executed during the last 24 hours.

You can also select a specific time period in the graph by drawing a box around part of the graph with your mouse. The graph will change to match your selection.

SQL Details
- Metrics

The SQL Details section of the Recent History view displays the SQL statements that correspond to the time period shown in the SQL Activity graph. The following columns are displayed:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL ID</td>
<td>The unique identifier for the SQL statement</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The total number of CPU seconds used by the statement during the specified time period</td>
</tr>
<tr>
<td>CPU %</td>
<td>The percentage of CPU used by the statement during the specified time period</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>The total elapsed time for the statement (in seconds) during the specified time period</td>
</tr>
<tr>
<td>Elapsed %</td>
<td>The percentage of elapsed time for the statement during the specified time period</td>
</tr>
<tr>
<td>Wait Time</td>
<td>The total wait time for the statement (in seconds) during the specified time period</td>
</tr>
<tr>
<td>Wait %</td>
<td>The percentage of wait time for the statement during the specified time period</td>
</tr>
<tr>
<td>SQL Time</td>
<td>The total SQL execution time for the statement (in seconds) during the specified time period</td>
</tr>
<tr>
<td>PL/SQL Time</td>
<td>The total PL/SQL execution for the statement (in seconds) during the specified time period</td>
</tr>
</tbody>
</table>
**Java Time:** The total java execution time for the statement (in seconds) during the specified time period

**Executions:** The total number of executions for the statement (in seconds) during the specified time period

**Disk Reads:** The total number of disk reads for the statement (in seconds) during the specified time period

**Buffer Gets:** The total number of buffer gets for the statement (in seconds) during the specified time period

**Sorts:** The total number of sorts for the statement (in seconds) during the specified time period

**Parse Calls:** The total number of parse calls for the statement (in seconds) during the specified time period

**Metrics**
Potential SQL tuning targets will be those statements that have unusually large amounts of CPU time, elapsed time, wait time, or number of executions.

**Users Page Statistics**
The Users home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key User Analysis</th>
<th>SQL Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Analysis</td>
<td>Workload Activity</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**
- Users Detail
- Home Page Statistics
- I/O Page Statistics
- Memory Page Statistics
- Objects Page Statistics
- OS Page Statistics
- Space Page Statistics

**Key User Analysis - Users**
User database activity can be examined using both ratio-based and wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios and statistical readings that can be used to indicate how efficiently users are accessing the database. Performance ratios serve as roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis.

When using ratio-based analysis, there are some standards to adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that you examine come from the v$systat performance view. The performance view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impacts how a particular ratio is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.
A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database’s individual ‘personality’ with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead you to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Users home page to succinctly communicate the general session-based performance levels of the monitored database:

<table>
<thead>
<tr>
<th>Active Sessions</th>
<th>CPU/Parse Ratio</th>
<th>Inactive Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Sort Ratio</td>
<td>Open Cursors</td>
<td>Parse/Execute Ratio</td>
</tr>
</tbody>
</table>

For more information, see Users Page Statistics.

**Bottleneck Analysis - Users**

User database activity can be examined using both ratio-based and wait/bottleneck-based analysis. When connected to an Oracle database, every process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. You can use the Bottleneck Analysis section to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps you track where a database has been spending its time. If user latch contention or heavy table scan activity has been dragging a database’s performance down, you can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, you can oftentimes discover a fair amount of detail about which sessions and objects are causing the problem.

For wait event analysis to be properly carried out, it is imperative that the timed_statistics initialization parameter be set to TRUE. By default this parameter is set to FALSE, which disallows the collection of wait times for each wait event defined to the Oracle engine. For one to really understand the impact of wait events on database performance, you need to not only discover what the database is or has been waiting on, but the durations of the waits. Having both allows a complete picture to be formed regarding the magnitude of wait-initiated performance degradations. Almost all Oracle experts now agree that allowing the collection of timing statistics adds little if anything to database overhead, so setting timed_statistics to TRUE should not be a worry. The Performance Analyst Users home page identifies the current top session waits as well as the top session-related bottlenecks that have seen the most waits on the system.

When using bottleneck analysis, you cannot rely only on the information contained in the wait event views that Oracle provides. Other user-related bottlenecks include current disk sort activity. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system.

The following bottleneck indicators are used on the Performance Analyst Users home page to succinctly communicate the general overall performance level of the monitored database:

<table>
<thead>
<tr>
<th>Sessions Blocked</th>
<th>Sessions in Disk Sorts</th>
<th>Sessions with Enqueue Waits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sessions Waiting</td>
<td>Top Bottleneck Events</td>
<td>Top Session Bottlenecks</td>
</tr>
</tbody>
</table>

For more information, see Users Page Statistics.

**SQL Analysis - Users**

Much of a database’s overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times. SQL statements that use much memory can also cause a problem in a database.
Before you can identify problem SQL in your database, you have to ask the question, “What is bad SQL?” What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The SQL Analysis for Users shows what SQL statements have consumed the largest percentages of executions, parse calls, CPU time, and elapsed time. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Response (Elapsed) Time</td>
<td>The amount of time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i or later.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The amount of CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.</td>
</tr>
<tr>
<td>Physical I/O</td>
<td>This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.</td>
</tr>
<tr>
<td>Logical I/O</td>
<td>The number of memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.</td>
</tr>
<tr>
<td>Repetition</td>
<td>The number of times the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute may still be causing a headache on your system if it has executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over again.</td>
</tr>
</tbody>
</table>

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

Fortunately, Oracle records all the above measures (some only in 9i), which makes tracking the SQL that has been submitted against an Oracle database much easier.

**Metrics**

When you begin to look for inefficient SQL in a database, there are two primary questions you want answered:

1. What HAS been the worst SQL that has historically run in my database?
2. What IS the worst SQL that is running right now in my database?

**Troubleshooting**

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient PL/SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can go further into Performance Analyst's Top SQL view and, if you have Embarcadero SQL Tuner installed, you can port the SQL over to SQL Tuner to better optimize the statement.
Transaction Analysis - Users
On a transaction-oriented system, user transactions perform COMMITs to make permanent any data additions, changes, or deletions to the database. Or they ROLLBACK any modifications to undo any work that has encountered an error or other transaction-halting situation. The transaction analysis breakdown shows current commits and rollbacks and the percentage that each activity accounts for on the database.

Metrics
Large volumes of rollbacks can indicate a transaction breakdown on the system, that may need investigation at the application level.

For more information, see Users Page Statistics.

Workload Analysis - Users
When the database population as a whole experiences a system slowdown, it is not uncommon to find one or two users who are responsible for bringing the system to its knees. In the best of worlds, users have a pretty evenly divided amount of memory usage, disk I/O, CPU utilization, and parse activity. However, users submit large batch jobs during peak OLTP activity, or when sessions that are fire off untuned queries on a critical system.

If you are seeing a slowdown in your database, and cannot seem to find a root cause, one thing to examine is the resource consumption of the leading sessions on a system. A handful of users can overwhelm the I/O capabilities of Oracle (through untuned queries or runaway batch jobs) or hammer the CPU or memory structures of the database and server.

Performance Analyst makes it easy to pinpoint the top sessions by showing the leading processes at the physical I/O, logical I/O, memory, disk sort, table scan, and CPU usage levels.

Metrics
If any one session uses more than 50% of a total resource (CPU, memory, etc.) go into the session to find out what they are currently executing.

For more information, see Users Page Statistics.

CPU/Parse Ratio
CPU resources are obviously required to accomplish parse activities. The CPU/Parse Ratio determines the percentage of CPU spent on user-initiated parse tasks.

Metrics
Low numbers (0-5%) are desired with the CPU/parse ratio. Larger values can indicate a problem with excessive parse activity.

Troubleshooting
If you suspect an excessive parsing problem on your database, you can look into increasing the shared pool size (in hopes that SQL statements are not aged out and therefore reused) or investigate the usage of bind variable in SQL statements.

If an undersized shared pool is suspected on Oracle8i or earlier:

- Edit the Init.ora file for the database.
- Increase the amount of shared_pool_size to a higher value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).
• Cycle the Oracle server when possible to allow the new value to take effect.
• Monitor the new value to see if performance improves.

If an undersized shared pool is suspected on Oracle9i or later:
• Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).
• Monitor the new value to see if performance improves.
• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

If you determine that SQL literals are causing SQL to not be reused, you can (in Oracle 8.1.6 and later):
• Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.
• Monitor database to see if parse activity is reduced and library cache reloads shrink.
• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.

Parse/Execute Ratio
Each time a new SQL statement is submitted to Oracle, the kernel must ‘parse’ the statement, which involves syntax checks, security checks, and object validations. The Parse/Execute Ratio shows the percentage of SQL executed that did not incur a hard parse.

NOTE: This statistic is available on the Home page, Memory home page, and the Users home page.

Metrics
Seeing low values might indicate that users are firing SQL with many hard-coded literals instead of using bind variables within an application. High values (90% and greater) generally indicate Oracle is saving precious CPU by avoiding heavy parse tasks.

Troubleshooting
The best way to reduce unnecessary parse activity is to encourage SQL statement reuse. This can be done by promoting SQL execution through the use of stored procedures or applications where bind variables can be used. Oftentimes, literals in otherwise identical SQL statements can cause unneeded parse work for Oracle. The use of bind variables can counter that problem.

If you determine that SQL literals are causing SQL to not be reused, do the following (in Oracle 8.1.6 and later):
• Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.
• Monitor database to see if parse activity is reduced and library cache reloads shrink.
• If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.
Memory Sort Ratio

Oracle's SGA is not the only memory structure used by Oracle for database work. One of the other memory areas used by Oracle8i and earlier for normal activity is an area set aside for sort actions. When a sort operation occurs, Oracle attempts to perform the sort in a memory space that exists at the operating system level. If the sort is too large to be contained within this space, it continues the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE. Oracle records the overall number of sorts that are satisfied in memory as well as those that end up being finalized on disk. Using these numbers, you can calculate the percentage of memory sorts vs. disk sorts and get a feel for how fast your sort activity is being resolved. Obviously, memory sorts completes many times faster than sorts forced to use physical I/O to accomplish the task at hand.

Oracle9i or later now has the option of running automatic PGA memory management. Oracle has introduced a new Oracle parameter called pga_aggregate_target. When the pga_aggregate_target parameter is set and you are using dedicated Oracle connections, Oracle ignores all of the PGA parameters in the Oracle file, including sort_area_size, hash_area_size and sort_area_retained_size. Oracle recommends that the value of pga_aggregate_target be set to the amount of remaining memory (less a 10% overhead for other server tasks) on a server after the instance has been started.

NOTE: This statistic is available on the Home page, Memory home page, and the Users home page.

Metrics

If your memory sort ratio falls below 90%, and you are on Oracle8i or earlier, increase the parameters devoted to memory sorts - sort_area_size and sort_area_retained_size.

For Oracle9i or later, investigate the use of pga_aggregate_target. Once the pga_aggregate_target parameter has been set, Oracle automatically manages PGA memory allocation, based upon the individual needs of each Oracle connection. Oracle9i or later allows the pga_aggregate_target parameter to be modified at the instance level with the alter system command, thereby lets you dynamically adjust the total RAM region available to Oracle9i.

Oracle9i also introduced a new parameter called workarea_size_policy. When this parameter is set to automatic, all Oracle connections benefits from the shared PGA memory. When workarea_size_policy is set to manual, connections allocates memory according to the values for the sort_area_size parameter. Under the automatic mode, Oracle tries to maximize the number of work areas that are using optimal memory and uses one-pass memory for the others.

Troubleshooting

To fix, do the following:

• Edit the Init.ora or SPFILE file for the database.

• Increase the amount of sort_area_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value. EVERY user receives this amount for sorting.

• Cycle the Oracle server when possible to allow the new value to take effect.

• Monitor new value to see if performance improves.

In addition to increasing the amount of memory devoted to sorting, find inefficient SQL that cause needless sorts. For example, UNION ALL does not cause a sort whereas UNION does in an SQL query (to eliminate duplicate rows). DISTINCT is frequently misapplied to SQL statements and causes unnecessary sort actions.

There are times you simply cannot stop sort activity. This being the case, try to keep it in memory whenever possible. However, large data warehousing systems oftentimes simply exhaust RAM sort allotments, so if disk sorts must occur, try to ensure three things:

1 Your user's TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.

NOTE: For Oracle9i or later, you can specify a default tablespace other than SYSTEM for every user account that is created.
2. The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.

3. The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.

Active Sessions
The Active Sessions statistic is the total number of active and open threads currently reported in the database as well as the number of processes actively performing work.

**NOTE:** This statistic is also available on the Home page. See Active User Processes on the bar graph in the Workload Analysis section.

Metrics
None.

For more information, see Inactive Sessions.

Inactive Sessions

**Metrics**

For more information, see Inactive Sessions.

**Troubleshooting**

Double-click this statistic to open the Top Sessions tab of the Users Detail page. On this tab you can check sessions that have many seconds idle and/or that have been logged on for very long periods of time as indicated by the logon time column. After verifying that the session is no longer necessary, you can KILL the session.

For more information, see Active Sessions.

Open Cursors

**Metrics**

Open Cursors is the total number of all SQL open cursors that exist on the system. In some cases, Oracle cached cursors that have been open by PL/SQL procedures can be kept open for certain lengths of time, even though the actual activity has ceased.
Metrics
You should monitor sessions to make sure that they do not approach the Open Cursor limit (specified in the Init.ora file). The parameter, open_cursors, limits how many open cursors (context areas) a session can have open at one time.

Troubleshooting
If the total number of open cursors approaches the open_cursors limit on Oracle8i or earlier, then:

• Ensure that user processes are efficiently using cursors before editing the Init.ora file.
• Edit the Init.ora file for the database.
• Increase the amount of open_cursors to a higher value.
• Cycle the Oracle server when possible to allow the new value to take effect.

If the total number of open cursors approaches the open_cursors limit on Oracle9i or later then:

• Change the open_cursors parameter to FORCE by using the ALTER SYSTEM SET open_cursors=< new value > command.

Top Bottleneck Events
When viewing wait statistics, there are many levels of detail. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. The second level is the session view, which shows details on which events connected sessions have experienced. Viewing these numbers can help you determine which session-related wait events have caused the most commotion in a database thus far. The Top Bottleneck Events section identifies the top waits that have occurred on the Oracle database based on the amount of time waited per event.

Metrics
None.

Troubleshooting
Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. For example, a ‘db file scattered read’ event is typically indicative of table scan operations. If many of these events are observed, you should begin to see if large table scans are occurring in the database. Like the ‘db file scattered read’ event, each wait event has its own meaning and individual end-resolution diagnosis.

Sessions Waiting

• Metrics
• Troubleshooting

User connections that are waiting on a system generally occur for two reasons:

1. A process waits because a requested resource is not available.
2. A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not usually a concern. However the two wait causes mentioned above are the ones worth your time and investigation. The sessions waiting statistic is a count of all sessions that are currently waiting for one reason or another.
Metrics
To determine the actual wait causes currently experienced by user connections, you should drill down from the global count of users waiting, into the actual system and user wait details of a database.

Troubleshooting
If you find a problem, drill down into wait details to determine whether the waits are resource-related.

Sessions Blocked

• Metrics

• Troubleshooting

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up. User processes holding exclusive locks and not releasing them via a proper COMMIT generally cause most blocks.

NOTE: This statistic is also called Current Object Blocks on the Home page and Session Blocks on the Objects home page.

TIP: You should immediately investigate any indicator above zero for a blocking lock statistic before the situation has a chance to grow out of control.

Troubleshooting
Once discovered, a blocking lock situation can normally be quickly remedied. You can issue a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Embarcadero Performance Analyst makes it easier to discover the blocked lock situation, but the trick is to prevent the blocking lock situation in the first place. You can look at the Users Detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT ... FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Sessions in Disk Sorts

• Metrics

• Troubleshooting
Oracle's SGA is not the only memory structure used by Oracle for database work. One of the other memory areas used by Oracle8i and earlier for normal activity is an area set aside for sort actions. When a sort operation occurs, Oracle attempts to perform the sort in a memory space that exists at the operating system level. If the sort is too large to be contained within this space, it continues the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE. Oracle records the overall number of sorts that are satisfied in memory as well as those that end up being finalized on disk. Using these numbers, you can calculate the percentage of memory sorts vs. disk sorts and get a feel for how fast your sort activity is being resolved. Obviously, memory sorts completes many times faster than sorts forced to use physical I/O to accomplish the task at hand.

Oracle9i or later now has the option of running automatic PGA memory management. Oracle has introduced a new Oracle parameter called pga_aggregate_target. When the pga_aggregate_target parameter is set and you are using dedicated Oracle connections, Oracle ignores all of the PGA parameters in the Oracle file, including sort_area_size, hash_area_size and sort_area_retained_size. Oracle recommends that the value of pga_aggregate_target be set to the amount of remaining memory (less a 10% overhead for other server tasks) on a server after the instance has been started.

The sessions in disk sorts statistic displays a count of all sessions currently performing a disk sort.

Metrics
Consistently seeing non-zero numbers for this statistic (as well as low values observed for the memory sort ratio) can indicate excessive disk sort activity. If you are on Oracle8i or earlier, increase the parameters devoted to memory sorts - sort_area_size and sort_area_retained_size.

If you are using Oracle9i or later, investigate the use of pga_aggregate_target. Once the pga_aggregate_target has been set, Oracle automatically manages PGA memory allocation, based upon the individual needs of each Oracle connection. Oracle9i or later allows the pga_aggregate_target parameter to be modified at the instance level with the alter system command, thereby lets you dynamically adjust the total RAM region available to Oracle9i.

Oracle9i also introduces a new parameter called workarea_size_policy. When this parameter is set to automatic, all Oracle connections benefit from the shared PGA memory. When workarea_size_policy is set to manual, connections allocate memory according to the values for the sort_area_size parameter. Under the automatic mode, Oracle tries to maximize the number of work areas that are using optimal memory and uses one-pass memory for the others.

Troubleshooting
If you find a problem, do the following:

- Edit the Init.ora or SPFIL E file for the database.
- Increase the amount of sort_area_size to a higher value (take care not to not over-allocate; ensure enough free memory exists on server before increasing value). Realize that EVERY user receives this amount for sorting).
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

In addition to increasing the amount of memory devoted to sorting, you should also locate inefficient SQL that causes needless sorts. For example, in an SQL query (to eliminate duplicate rows) UNION ALL does not cause a sort whereas UNION does. People frequently code DISTINCT inappropriately (especially people transferring from Microsoft Access, which uses DISTINCT for most SELECT queries).

There are times you simply cannot stop sort activity. When this happens, you should try to keep it in memory whenever possible. However, large data warehousing systems frequently exhaust RAM sort allotments, so if disk sorts must occur, ensure three things:

- Your user's TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment. In Oracle9i or later, you can specify a default tablespace other than SYSTEM for every user account that is created.
- The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.
• The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.

Sessions with Enqueue Waits
An enqueue is an advanced locking device that allows multiple database processes to share certain resources. Enqueue waits typically occur when sessions wait to be granted a requested lock. Sometimes these locks are internal Oracle locks while other times they could be locks for rows of data in a table. Note that enqueues are issued implicitly by Oracle.

Metrics
Any enqueue waits that read consistently above 1 or more (delta statistics) should be investigated.

Troubleshooting
Removing contention for enqueues is almost always an application design issue. If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations.

Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).

Users Detail
The following tabbed pages are available on the Users Detail page:

<table>
<thead>
<tr>
<th>Locks/Blocking Locks</th>
<th>Session Waits</th>
<th>System Waits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Sessions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Top Sessions Tab
The Top Sessions tab includes the following sections:

<table>
<thead>
<tr>
<th>Top CPU Process</th>
<th>Top I/O Process</th>
<th>Top Memory Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Sessions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For more information, see Users Home Page.

Top I/O Process
The Top I/O Process section identifies the Oracle process that currently has caused the most I/O usage on the database.

Metrics
To obtain more details on the top I/O process, locate the SID in the Top Sessions grid and drill down to obtain more granular information.
Top Memory Process
The Top Memory Process section identifies the Oracle process that currently is using the highest percentage of memory in the database.

Metrics
To obtain more details on the top memory process, locate the SID in the Top Sessions grid and drill down to obtain more granular information.

Top CPU Process
The Top CPU Process section identifies the Oracle process that currently has the highest percentage of CPU usage in the database.

Metrics
To obtain more details on the top CPU process, locate the SID in the Top Sessions grid and drill down to obtain more granular information.

Top Sessions
When a system experiences heavy activity, sometimes you will find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are causing most of the activity. It may be that a large batch load or other typical process is running that is perfectly okay for your system. Or it may be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

The Top Sessions section displays information regarding all key metrics for all current sessions on a database. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>The system ID of the process.</td>
</tr>
<tr>
<td>User Name</td>
<td>The username of the process.</td>
</tr>
<tr>
<td>O/S ID</td>
<td>The operating system ID of the process.</td>
</tr>
<tr>
<td>Machine Name</td>
<td>The workstation where the process originated.</td>
</tr>
<tr>
<td>Logon Time</td>
<td>The date/time the process logged on to Oracle.</td>
</tr>
<tr>
<td>Tot Physical I/O</td>
<td>A summation of all physical reads and physical writes for the process (includes direct).</td>
</tr>
<tr>
<td>Tot Logical I/O</td>
<td>A summation of all logical I/O (buffer gets, consistent gets, etc.) for the process.</td>
</tr>
<tr>
<td>Hit Ratio</td>
<td>Shows how often object definitions are found in memory vs. having to read them in from disk.</td>
</tr>
<tr>
<td>Total Memory Usage</td>
<td>The total amount of memory used.</td>
</tr>
<tr>
<td>Parses</td>
<td>The total number of parses the process has caused.</td>
</tr>
<tr>
<td>Total CPU</td>
<td>The total CPU usage for the process.</td>
</tr>
<tr>
<td>Parse CPU</td>
<td>The amount of process CPU caused by parse activity.</td>
</tr>
<tr>
<td>Recursive CPU</td>
<td>The amount of process CPU caused by recursive calls.</td>
</tr>
<tr>
<td>Other CPU</td>
<td>Total CPU usage minus parse and recursive activity. This can be a negative number if the Oracle kernel is not properly reporting total CPU usage.</td>
</tr>
<tr>
<td>Disk Sorts</td>
<td>The total number of disk sorts caused by the process.</td>
</tr>
<tr>
<td>Memory Sorts</td>
<td>The total number of memory sorts caused by the process.</td>
</tr>
<tr>
<td>Rows Sorted</td>
<td>The total number of rows sorted by the process.</td>
</tr>
</tbody>
</table>
Metrics
To obtain more details on any process, drill down to obtain more granular information.

System Waits Tab
The System Waits tab includes the following sections:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commits</td>
<td>The total number of commits executed by the process.</td>
</tr>
<tr>
<td>Rollbacks</td>
<td>The total number of rollbacks executed by the process.</td>
</tr>
<tr>
<td>Executions</td>
<td>The total number of executions.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>The total number of physical reads produced by the process.</td>
</tr>
<tr>
<td>DB Block Gets</td>
<td>The total number of block gets produced by the process.</td>
</tr>
<tr>
<td>Consistent Gets</td>
<td>The total number of consistent gets produced by the process.</td>
</tr>
<tr>
<td>Consistent Changes</td>
<td>The total number of consistent changes produced by the process.</td>
</tr>
<tr>
<td>Last Wait</td>
<td>Event Defines the last wait event experienced by the session</td>
</tr>
<tr>
<td>Wait Class</td>
<td>The wait class of the last wait event</td>
</tr>
<tr>
<td>Wait time</td>
<td>The sessions last wait time</td>
</tr>
<tr>
<td>P1text</td>
<td>Text that describes the first wait parameter</td>
</tr>
<tr>
<td>P2text</td>
<td>Text that describes the second wait parameter</td>
</tr>
<tr>
<td>P2 r</td>
<td>The second wait parameter</td>
</tr>
<tr>
<td>P2Raw</td>
<td>The data for the second parameter</td>
</tr>
<tr>
<td>P3text</td>
<td>Text that describes the third wait parameter</td>
</tr>
<tr>
<td>P3</td>
<td>The third wait parameter</td>
</tr>
<tr>
<td>P3Raw</td>
<td>The data for the third parameter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System Waits</th>
<th>Wait Percent by Time Waited</th>
<th>Wait Percent by Total Waits</th>
</tr>
</thead>
</table>

For more information, see Users Home Page.

System Waits
Waits on a system generally occur for three reasons:

1. A process waits because it has no work to do.
2. A process waits because a requested resource is not available.
3. A process waits for Oracle to perform a prerequisite task for its given operation.
Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The system Waits section presents all the various system waits that have occurred on the system since startup. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>The name of the wait event.</td>
</tr>
<tr>
<td>Class</td>
<td>Wait classes contain a collection of related wait events (like User I/O, System I/O, etc.)</td>
</tr>
<tr>
<td>Total Waits</td>
<td>The total number of waits for the event.</td>
</tr>
<tr>
<td>Percent of Total</td>
<td>The percentage of all waits that this event makes up.</td>
</tr>
<tr>
<td>Time Waited (Secs)</td>
<td>The total amount of time waited for the event, in seconds.</td>
</tr>
<tr>
<td>Percent Time Waited</td>
<td>The percentage of time that this event makes up.</td>
</tr>
<tr>
<td>Total Timeouts</td>
<td>The total number of timeouts for the event.</td>
</tr>
<tr>
<td>Percent Total Timeouts</td>
<td>The percentage of timeouts that this event makes up.</td>
</tr>
<tr>
<td>Average Waits (Secs)</td>
<td>The average amount of time waited for the event, in seconds.</td>
</tr>
</tbody>
</table>

**Metrics**

Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include:

- SQL*Net more data from client
- SQL*Net message from client
- client message
- rdbms ipc message
- pipe get
- pmon timer
- smon timer
- Null event

**Troubleshooting**

The table below describes wait events that deserve attention:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enqueue waits</td>
<td>If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).</td>
</tr>
<tr>
<td>Buffer busy waits</td>
<td>Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.</td>
</tr>
</tbody>
</table>
The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database. To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

### Wait Percent by Total Waits
Waits on a system generally occur for three reasons:

- A process waits because it has no work to do.
- A process waits because a requested resource is not available.
- A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Wait Percent by Total Waits section displays the percentage of waits for all top non-idle wait events.

### Metrics
Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include:

- SQL*Net more data from client
- SQL*Net message from client
- client message
- rdbms ipc message
- pipe get
- pmon timer
- smon timer
- Null event

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db file scattered read waits</td>
<td>An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.</td>
</tr>
<tr>
<td>Latch free waits</td>
<td>Latch free waits indicate contention for latches.</td>
</tr>
</tbody>
</table>

Latch free waits indicate contention for latches.
Troubleshooting

The table below describes wait events that deserve attention:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enqueue waits</td>
<td>If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).</td>
</tr>
<tr>
<td>Buffer busy waits</td>
<td>Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.</td>
</tr>
<tr>
<td>db file scattered read waits</td>
<td>An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.</td>
</tr>
<tr>
<td>Latch free waits</td>
<td>Latch free waits indicate contention for latches.</td>
</tr>
</tbody>
</table>

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

Wait Percent by Time Waited

Waits on a system generally occur for three reasons:

1. A process waits because it has no work to do.
2. A process waits because a requested resource is not available.
3. A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Wait Percent by Time Waited section displays the percentage of waits for all top non-idle wait events.

Metrics

The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database.

Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include:

- SQL*Net more data from client
- SQL*Net message from client
- client message
- rdbms ipc message
- pipe get
- pmon timer
- smon timer
- Null event
Troubleshooting
The table below describes wait events that deserve attention:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enqueue waits</td>
<td>If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).</td>
</tr>
<tr>
<td>Buffer busy waits</td>
<td>Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.</td>
</tr>
<tr>
<td>db file scattered read waits</td>
<td>An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.</td>
</tr>
<tr>
<td>Latch free waits</td>
<td>Latch free waits indicate contention for latches.</td>
</tr>
</tbody>
</table>

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your Init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Session Waits Tab
The Session Waits tab includes the following sections:

- Current Waits
- Historical Waits

For more information, see Users Home Page.

Historical Waits
Waits on a system generally occur for three reasons:

1. A process waits because it has no work to do.
2. A process waits because a requested resource is not available.
3. A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Historical Waits section presents all the various waits that have occurred for all currently connected oracle sessions. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>The name of the wait event.</td>
</tr>
<tr>
<td>Wait Class</td>
<td>Wait classes contain a collection of related wait events (like User I/O, System I/O, etc.)</td>
</tr>
<tr>
<td>Total Waits</td>
<td>The number of waits for the event.</td>
</tr>
</tbody>
</table>
Metrics

Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include:

- SQL*Net more data from client
- SQL*Net message from client
- client message
- rdbms ipc message
- pipe get
- pmon timer
- smon timer
- Null event

Troubleshooting

The table below describes wait events that deserve attention:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enqueue waits</td>
<td>If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).</td>
</tr>
<tr>
<td>Buffer busy waits</td>
<td>Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.</td>
</tr>
<tr>
<td>db file scattered read waits</td>
<td>An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.</td>
</tr>
<tr>
<td>Latch free waits</td>
<td>Latch free waits indicate contention for latches.</td>
</tr>
</tbody>
</table>

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database. To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.
Current Waits

Waits on a system generally occur for three reasons:

- A process waits because it has no work to do.
- A process waits because a requested resource is not available.
- A process waits for Oracle to perform a prerequisite task for its given operation.

Session contention is merely a subset of contention that is viewable at the global database level. Oftentimes, it takes analysis at the session level to pinpoint the exact source of contention that is occurring globally. So you need to become accustomed to viewing contention statistics and waits at the user process level.

When monitoring waits with respect to user sessions, there are two areas of interest:

1. What HAS the user session been waiting on?
2. What IS the user session waiting on?

Oracle records both sets of wait statistics for you. In reviewing previous waits for a session, you can see which types of things have caused the session to be bottlenecked. The Current Waits section displays information to help you troubleshoot session contention. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>The name of the wait event.</td>
</tr>
<tr>
<td>Wait (Secs)</td>
<td>The number of seconds in the current wait.</td>
</tr>
<tr>
<td>Last Wait Time</td>
<td>A non-zero value is the session's last wait time. A zero value means the session is currently waiting. A -2 value indicates the platform cannot support a fast timing mechanism or TIMED_STATISTICS is not set for the system.</td>
</tr>
<tr>
<td>State</td>
<td>Indicates the possible values. They includes the following:</td>
</tr>
<tr>
<td></td>
<td>WAITING - the session is waiting.</td>
</tr>
<tr>
<td></td>
<td>WAITED UNKNOWN TIME - duration of last wait is not known.</td>
</tr>
<tr>
<td></td>
<td>WAITED SHORT TIME - last wait was less than 1/100th of a second.</td>
</tr>
<tr>
<td></td>
<td>WAITED KNOWN TIME - the wait is equal to the time of the last wait.</td>
</tr>
<tr>
<td>P1 - P3</td>
<td>Any additional wait parameters.</td>
</tr>
<tr>
<td>P1TEXT - P3TEXT</td>
<td>Any descriptions of additional parameter.</td>
</tr>
<tr>
<td>P1RAW - P3RAW</td>
<td>Any additional wait parameters.</td>
</tr>
</tbody>
</table>

Metrics

Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include:

- SQL*Net more data from client
- SQL*Net message from client
- client message
- rdbms ipc message
- pipe get
- pmon timer
- smon timer
null event

Troubleshooting
The table below describes wait events that deserve attention:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enqueue waits</td>
<td>If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).</td>
</tr>
<tr>
<td>Buffer busy waits</td>
<td>Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.</td>
</tr>
<tr>
<td>db file scattered read waits</td>
<td>An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.</td>
</tr>
<tr>
<td>Latch free waits</td>
<td>Latch free waits indicate contention for latches.</td>
</tr>
</tbody>
</table>

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database. To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Locks/Blocking Locks Tab
The Locks/Blocking Locks tab includes the following sections:

- All Locks
- Blocking Locks

For more information, see Users Home Page.

All Locks
To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, Oracle itself will issue lock requests to carry out its internal duties. The All Locks section displays information about all locks currently on a system. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
<td>The workstation of the Oracle process.</td>
</tr>
<tr>
<td>Serial #</td>
<td>The serial number of the session holding the lock.</td>
</tr>
<tr>
<td>Request</td>
<td>The lock requests that Oracle issues to carry out its internal duties.</td>
</tr>
<tr>
<td>Table</td>
<td>The table name of the locked table.</td>
</tr>
<tr>
<td>Mode</td>
<td>The lock mode (EXCLUSIVE, SHARE, etc.).</td>
</tr>
</tbody>
</table>
Locks that are held for unusually long periods of time may be candidates for further investigation. The application logic may be inefficient or perhaps the program is not issuing frequent enough COMMITs.

Blocking Locks
Without a doubt, blocking lock situations can give the appearance of a frozen database almost more than anything else (except, perhaps, for a stuck archiver process). A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up – sometimes frequently.

The Blocking Locks section contains information relating to user accounts that are currently blocked and the sessions that are blocking them. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocked User</td>
<td>The user account of the session waiting for the lock.</td>
</tr>
<tr>
<td>Blocking User</td>
<td>The user account of the session holding the offending lock.</td>
</tr>
<tr>
<td>Waiting SID</td>
<td>The session identifier of the session waiting for the lock.</td>
</tr>
<tr>
<td>Holding SID</td>
<td>The session identifier of the session holding the offending lock.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of lock (TRANSACTION, DML, etc.).</td>
</tr>
<tr>
<td>Mode</td>
<td>The lock mode (EXCLUSIVE, SHARE, etc.).</td>
</tr>
<tr>
<td>Request</td>
<td>The type of lock being requested by the session.</td>
</tr>
<tr>
<td>Row</td>
<td>The rowid of the row being held.</td>
</tr>
<tr>
<td>ID 1</td>
<td>Lock identifier #1 (depends on type).</td>
</tr>
<tr>
<td>ID 2</td>
<td>Lock identifier #2 (depends on type).</td>
</tr>
<tr>
<td>Time Waited (Secs)</td>
<td>The amount of time, in seconds, the blocked user has been waiting.</td>
</tr>
<tr>
<td>Locked Object</td>
<td>The name of the object being locked.</td>
</tr>
</tbody>
</table>

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT … FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Data warehouses (whose data is mostly read) can benefit from tablespaces set in READ ONLY mode. READ ONLY signals to the other database that exclusive locks need not be used for the data contained within the tablespace. This is especially helpful in Oracle Parallel Server environments and drastically reduces ping activity.
RAC Detail View
The Performance Analyst for Oracle has been upgraded to support for the monitoring of Oracle Real Application Clustering (RAC) environments (Oracle 9i and higher).

Oracle RAC allows a single physical Oracle database to be accessed by simultaneous instances of Oracle running across several CPUs. All statistics you see are listed by instance ID.

The RAC Detail view is divided into the following tabs:

<table>
<thead>
<tr>
<th>Overview</th>
<th>Memory</th>
<th>Ping Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locks</td>
<td>Sessions</td>
<td></td>
</tr>
</tbody>
</table>

Overview
The Overview tab is of the RAC Detail view divided into the following sections:

<table>
<thead>
<tr>
<th>Key Ratio Analysis</th>
<th>Instance Analysis</th>
<th>Bottleneck Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload Analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Key Ratio Analysis
Database performance analysts typically use one of two methods for examining the performance levels of a database: ratio-based or wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and can’t spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the v$sysstat performance view. This view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impact how a particular ratio that a DBA may be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database’s individual “personality” with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst RAC Overview page to succinctly communicate the general overall performance levels of the monitored database:

- Cross Instance Cache Flush ratio
- Ping Ratio
- Local Buffer Access Percent
- Local Read Percent
- Global Buffer Busy Wait Percent
Instance Analysis
As a DBA managing an Oracle RAC, you will oftentimes want a quick performance snapshot of the efficiency and activity across all your Oracle RAC nodes. The Instance Analysis section provides a fast picture of which nodes of your Oracle RAC appear overworked and those that are underutilized.

The following statistics are provided:

- **ID**: The Oracle RAC instance ID
- **Name**: The instance name
- **Sessions**: The total number of sessions logged onto a RAC node
- **Physical I/O**: A summation of all physical reads and physical writes for the RAC node
- **Logical I/O**: A summation of all logical I/O (buffer gets, consistent gets, etc.) for the RAC node
- **Hit Ratio**: An indicator of how often user requests for data are satisfied.
- **Memory Usage**: The session's total memory usage against all session memory usage on the database.
- **Parses**: The total number of parses experienced by the RAC node
- **Total CPU**: The total CPU usage for the RAC node
- **Parse CPU**: The amount of process CPU for the RAC node
- **Recursive CPU**: The amount of process CPU caused by recursive calls on the RAC node
- **Other CPU**: Total CPU usage minus parse and recursive activity. This can be a negative number if the Oracle kernel is not properly reporting total CPU usage
- **Disk Sorts**: The total number of disk sorts for the RAC node
- **Memory Sorts**: The total number of memory sorts for the RAC node
- **Rows Sorts**: The total number of rows sorted on the RAC node
- **Commits**: The total number of commits executed for the RAC node
- **Rollbacks**: The total number of rollbacks executed for the RAC node
- **Executions**: The total number of calls made on the RAC node
- **Physical Reads**: The total number of physical reads produced on the RAC node
- **DB Block Gets**: The total number of block gets produced on the RAC node
- **Consistent Gets**: The total number of consistent gets produced on the RAC node
- **Consistent Changes**: The total number of consistent changes produced on the RAC node

Bottleneck Analysis
When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting can mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs used this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.
Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If latch contention or heavy table scan activity has been dragging a database's performance down, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about what sessions and objects are causing the problem.

For wait event analysis to be properly carried out, it is imperative that the `timed_statistics` initialization parameter be set to TRUE. By default this parameter is set to FALSE, which disallows the collection of wait times for each wait event defined to the Oracle engine. For one to really understand the impact of wait events on database performance, the DBA not only needs to discover what the database is or has been waiting on, but the durations of the waits. Having both measures in hand allows a complete picture to be formed regarding the magnitude of wait-initiated performance degradations. Almost all Oracle experts now agree that allowing the collection of timing statistics adds little if anything to database overhead, so setting `timed_statistics` to TRUE should not be a worry. The Performance Analyst home page identifies the top system and session waits that are currently a cause of lessened performance.

When using bottleneck analysis, a DBA cannot rely only on the information contained in the wait event views that Oracle provides. For example, an object may attempt to extend into another extent of space in a tablespace and yet be denied if no such free space exists. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system.

For an Oracle RAC, Performance Analyst highlights the top RAC-related waits as well as conflicts that may be slowing down response times. Here you see statistics for Top Rac-Related Waits and Conflicts. Within each section, you find:

### Top Rac-Related Waits
When viewing wait statistics, there are several levels of detail that a DBA can drill down into. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. Viewing these numbers can help a DBA determine what wait events have caused the most commotion in a database thus far. The Top RAC-related bottlenecks section display identifies the top waits that have occurred on the node that Performance Analyst is currently connected to.

Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. For example, a "db file scattered read" event is typically indicative of table scan operations. If many of these events are observed, then the DBA should begin to see if large table scans are occurring in the database. Like the "db file scattered read" event, each wait event has its own meaning and individual end-resolution diagnosis.

After looking at system-level wait activity, a DBA can drill down further to discover what current connections may be responsible for any reported waits that are being observed at the system level. Performance Analyst reports on historical and current wait events at the session level, making this investigation easy to accomplish.

### Conflicts
There are a number of conflicts that can occur over an Oracle RAC that can reduce response times. The Conflicts section highlights some of the more common conflicts and provides a count of how many times each has occurred.

The conflicts listed include:

**Library Cache**: How many library cache invalidations have occurred on the node that Performance Analyst is connected to.

**Data Dictionary Cache**: How many data dictionary cache invalidations have occurred on the node that Performance Analyst is connected to.

**Forced Writes (Oracle 9.2 and above)**: Number of times a block was written to cache because an instance had used the block, but another instance had requested the lock on the block in a conflicting mode.

**Pings**: How many cross instance writes occurred
Cache Timeouts: The number of global cache convert timeouts that have occurred.

Workload Analysis
When your phone starts ringing with complaints of performance slowdowns, one of the first things you should get a handle on is:

• Who is currently connected to the database
• What resources are they using
• What are they currently executing

The Workload Activity Analysis section of the Performance Analyst home page provides insight into the leading resource hogs across all instances of an Oracle RAC. Drill down’s are available so you can easily get detailed information into what each leading session is currently involved with.

If you are seeing a slowdown in your database, and can’t seem to find a root cause, one thing to examine is the resource consumption of the leading sessions on a system. Often you will find one or a handful of users making life miserable for everyone else. They accomplish this by overwhelming the I/O capabilities of Oracle (through untuned queries or runaway batch jobs) or hammering the CPU or memory structures of the database and server.

The Workload Analysis section presents:

• Top Physical I/O Process
• Top Logical I/O Process
• Top Memory Process
• Top CPU Process

For each of these, you see the SID, Instance ID, Username, %Used.

If any one session appears to be using more than 50% of a total resource (CPU, memory, etc.), then you should drill down into the session to find out what it is currently executing.

Memory
The Memory tab of the RAC Detail view is divided into the following sections:

<table>
<thead>
<tr>
<th>Key Ratio Analysis</th>
<th>Memory Analysis</th>
<th>Library Cache Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Dictionary Cache Analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Key Ratio Analysis
Database performance analysts typically use one of two methods for examining the performance levels of a database: Ratio-based or wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and can’t spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.
While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA examines come from the v$sysstat performance view. This view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA may be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database’s individual “personality” with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used in the Memory Key Ratio Analysis section to succinctly communicate the general overall performance levels of the monitored database:

- **Buffer Cache Hit Ratio**: How often requested information is found in memory versus on disk.
- **Library Cache Hit Ratio**: Shows how often SQL code is being reused by other database users versus the number of times a SQL statement is broken down, parsed, and then loaded (or reloaded) into the shared pool.
- **Latch Miss Ratio**: Defines the number of times a process obtained a willing-to-wait latch versus missing the attempt.

**Memory Analysis**

The Memory Analysis section displays the memory configurations for each node currently up in the RAC. The following information is shown:

**Name**: The name of the instance.

**DB Buffers (MB)**: The default memory cache that maintains data blocks when they are read from the database. If the DBA doesn't specifically place objects in another data cache (which will be covered next), then any data requested by clients from the database will be placed into this cache. This memory area is controlled by the `db_block_buffers` parameter in Oracle8i and below and `db_cache_size` in Oracle9i and above.

**Log Buffer Size (MB)**: This area buffers modifications that are made to the database before they are physically written to the redo log files. The `log_buffer` configuration parameter controls this memory area.

**Shared Pool Size (MB)**: This familiar area holds object structure as well as code definitions, and other metadata. Setting the proper amount of memory in the shared pool assists a great deal in improving overall performance with respect to code execution.

**DB Caches (MB)**: Beginning in Oracle9i, a DBA can create tablespaces whose blocksize differs from the overall database blocksize. When data is read into the SGA from these tablespaces, their data has to be placed into memory regions that can accommodate their special block size. Oracle9i and above has memory settings for 2K, 4K, 8K, 16K, and 32K caches. The configuration parameter names are in the pattern of `db_nk_cache_size`. For each RAC node, you see the following statistics:

- Buffer Cache Hit Ratio
- Library Cache Hit Ratio
- Latch Miss Ratio
Library Cache Analysis
The Library Cache Analysis section provides a drill down view into invalidations that have occurred across the cluster. The following information is shown:

**ID**: The instance ID.

**Name**: The instance name.

**Namespace**: The area of the library cache.

**DLM Lock Requests**: The number of get requests for instance locks.

**DLM Pin Requests**: The number of pin requests for instance locks.

**DLM Pin Releases**: The number of release requests for pin instance locks.

**DLM Invalidation Requests**: The number of get requests for invalidation instance locks.

**DLM Invalidations**: The number of invalidation pings received from other RAC instances.

Data Dictionary Cache Analysis
The Data Dictionary Cache Analysis section provides a drill-down view into invalidations and conflicts that have occurred across the cluster. The following information is shown:

**ID**: The instance ID.

**Name**: The instance name.

**Parameter**: The specific object area of the dictionary cache.

**DLM Requests**: The number of lock manager requests.

**DLM Conflicts**: The number of lock manager conflicts encountered.

**DLM Releases**: The number of lock manager releases.

Ping Activity
Pinging is the process where one Oracle Instance requests another to write a set of blocks from its SGA to disk so it can obtain it in exclusive mode. To move a data block from one instance's SGA to another is a slow process. The Ping Activity tab of the RAC Detail view displays several levels of specificity, starting from general to specific. The Ping Activity tab is divided into the following sections:

<table>
<thead>
<tr>
<th>Ping by Instance</th>
<th>Ping by Datafile</th>
<th>Ping by Object</th>
</tr>
</thead>
</table>

**Ping by Instance**
The Ping by Instance view shows how often pinging has occurred across all the monitored instances.

**Ping by Datafile**
The Ping by Datafile view shows the number of blocks pinged from the shared set of RAC datafiles.

**Ping by Object**
The Ping by Object provides a drill down view into the amount of specific ping activity across the shared objects and datafiles of a RAC:

- **Owner**
• Object Name
• Partition Name
• Block Number
• Segment Type
• Kind
• Forced Reads
• Forced Writes
• Datafile

Locks
The Locks tab of the RAC Detail view is divided into the following sections:

<table>
<thead>
<tr>
<th>Lock Convert Summary</th>
<th>DLM Lock Activity</th>
<th>PCM Lock Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latch Detail</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lock Convert Summary
The Lock Convert Summary view provides efficiency statistics regarding the speed at which the RAC acquires and converts locks across the instances. You see the average get and convert times for each RAC node.

DLM Lock Activity
The DLM (Distributed Lock Manager) Lock Activity view provides detailed information on the number of lock conversions that have occurred across the RAC.

PCM Lock Summary
The PCM (Parallel Cache Management) Lock Summary provides a summary count of PCM locks that are releasing, acquiring, or are invalid.

Latch Detail
Protecting the memory structures in Oracle's SGA are latches. Latches ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you don't and are forced to continually retry.

The latch detail tab of the contention detail view presents a detailed view of latch activity. Information presented includes:

• **Instance Name**: The name of the instance
• **Name**: The name of the latch
• **Gets**: The total number of times the latch was requested by a process
• **Misses**: The total number of failed attempts to acquire the latch on the first attempt
• **Immediate Gets**: The total number of no-wait requests for a latch
• **Immediate Misses**: The total number of failed no-wait attempts to acquire the latch on the first attempt
• **Immediate Sleeps**: The total number of requests that "paused" while waiting for a latch
Common indicators of latch contention are a latch miss ratio (which records willing-to-wait mode latch requests) and a latch immediate miss ratio (which records no-wait mode latch requests). These statistics reflect the overall health of how often latch requests were made and satisfied without waiting.

Here are a few latch contention situations that you should recognize and get to the bottom of quickly:

- **Cache buffer chain latch**: This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the `db_block_buffers` parameter) or by adding more LRU latch chain latches with the `db_block_lru_latches` parameter.

- **Library cache latches**: Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.

Other routine latch contention problems used to include the redo allocation and redo copy latches, but these have pretty much been made obsolete in Oracle 8.1.5 and higher.

**Sessions**

The Sessions tab of the RAC Detail view is divided into the following sections:

<table>
<thead>
<tr>
<th>User Session by Instance</th>
<th>Blocks by Instance</th>
<th>User Waits by Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Top Sessions</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**User Session by Instance**

The User Session by Instance section provides a graphical count of the number of connected sessions for each node in the RAC.

**Blocks by Instance**

The Blocks by Instance section provides a graphical count of the number of blocked sessions for each node in the RAC.

**User Waits by Instance**

The User Waits by Instance section provides a graphical count of the number of historical session waits (non-idle) for each node in the RAC.

**Top Sessions**

When a system experiences heavy activity, sometimes you’ll find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are causing most of the activity. It may be that a large batch load or other typical process is running that is perfectly OK for your system. Or it may be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

The top sessions control displays information regarding all key metrics for all current sessions on a database. Information presented includes:

- **SID**: The system ID of the process
- **Instance**: The instance the session is connected to
- **Username**: The username of the process
- **O/S ID**: The operating system ID of the process
- **Machine Name**: The workstation where the process originated
- **Logon Time**: The date/time the process logged on to Oracle
Tot Physical I/O: A summation of all physical reads and physical writes for the process (includes direct)

Tot Physical I/O: A summation of all logical I/O (buffer gets, consistent gets, etc.) for the process

 Parses:  The total number of parses the process has caused

 Total CPU:  The total CPU usage for the process

 Parse CPU:  The amount of process CPU caused by parse activity

 Recursive CPU:  The amount of process CPU caused by recursive calls

 Other CPU:  Total CPU usage minus parse and recursive activity.  This can be a negative number if the Oracle kernel is not properly reporting total CPU usage

 Disk sorts:  The total number of disk sorts caused by the process

 Memory sorts:  The total number of memory sorts caused by the process

 Rows sorted:  The total number of rows sorted by the process

 Commits:  The total number of commits executed by the process

 Rollbacks:  The total number of rollbacks executed by the process

 Physical reads:  The total number of physical reads produced by the process

 DB block gets:  The total number of block gets produced by the process

 Consistent gets:  The total number of consistent gets produced by the process

 Consistent changes:  The total number of consistent changes produced by the process

Response Time Analysis View (Oracle 10G Only)
The Response Time Analysis View displays statistics that are a part of the Oracle 10g wait enhancements. By analyzing response times, you can review historical data and pinpoint who was on a database and what they were doing during a specific period of time. This allows you to drill deep into the root causes, if any of response time problems, up to 24 hours after the fact.

The main Response Time Analysis View displays the following tabbed pages:

<table>
<thead>
<tr>
<th>Response Time</th>
<th>Active Time</th>
<th>System Bottlenecks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Bottlenecks</td>
<td>Wait Histogram</td>
<td>Historical Session Analysis</td>
</tr>
</tbody>
</table>

Response Time
The Response Time tab of the Response Time Analysis View displays the following statistics:

<table>
<thead>
<tr>
<th>Current Workload</th>
<th>Current Response Times</th>
<th>Last Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Hour Detail</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Current Workload
- Metrics
- Troubleshooting
The Current Workload section provides a breakdown of active and idle time for the Oracle database engine in terms of percentage used.

**Metrics**
Seeing consistently high percentages for the Database CPU Time (greater than 75-90%) should indicate a possibly overworked server.

**Troubleshooting**
If you see high numbers for the Database CPU Time, then begin looking into these statistics:
- SQL Statements with high elapsed times and CPU times (use the Top SQL view to find these).
- Sessions with high wait times (especially in the User I/O wait class).

The Current Response Times section provides a quick view into the current average response time that a transaction experiences on the system along with the current average SQL service time. Both statistics are expressed in seconds.

**Metrics**
Defining ‘good’ and ‘bad’ metrics for these two measurements can only be accomplished by viewing these statistics over time and determining what ‘normal’ is for your system. You can get help by checking the Last Hour section, which details the minimum, maximum, and average values for these statistics as well as other key metrics that provide insight into how well your database is running.

**Last Hour**
- **Metrics**
- **Troubleshooting**

The Last Hour section provides a window into various key performance metrics. Each metric's minimum, maximum, and average value is displayed.

**Metrics**
Key metrics to keep an eye on include:
- CPU usage per Txn
- Database CPU Time Ratio
- Response Time Per Txn
- SQL Service Response Time

**Troubleshooting**
If you see high numbers for the Database CPU Time, then begin looking into these statistics:
- SQL Statements with high elapsed times and CPU times (use the Top SQL view to find these).
- Sessions with high wait times (especially in the User I/O wait class).
Last Hour Detail
The Last Hour Detail section provides a graphical look back over the last hour of a selected performance metric. Oracle 10g and higher automatically maintains a running history of selected metrics so you can observe trends and patterns in performance.

Active Time
The Active Time tab of the Response Time Analysis detail view displays the following statistics:

### System Time Summary

The System Time Summary section allows you to see where, in broad terms, the database engine has been spending the largest percentages of its time.

### Metrics
The metrics shown here include (note, this table is largely from Oracle 10g documentation):

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Time</td>
<td>You see the amount of elapsed time (in microseconds) spent performing Database user-level calls. Instance background processes such as PMON are not included.</td>
</tr>
<tr>
<td>DB CPU</td>
<td>Amount of CPU time (in microseconds) spent on database user-level calls. Instance background processes such as PMON are not included.</td>
</tr>
<tr>
<td>background cpu time</td>
<td>Amount of CPU time (in microseconds) taken up by database background processes.</td>
</tr>
<tr>
<td>sequence load elapsed time</td>
<td>Amount of elapsed time spent getting the next sequence number from the data dictionary. If a sequence is cached, then this is the amount of time spent replenishing the cache when it runs out. No time is charged when a sequence number is found in the cache. For non-cached sequences, some time is charged for every nextval call.</td>
</tr>
<tr>
<td>parse time elapsed</td>
<td>Amount of elapsed time spent parsing SQL statements. Both soft and hard parse times are included.</td>
</tr>
<tr>
<td>hard parse elapsed time</td>
<td>Amount of elapsed time spent hard parsing SQL statements.</td>
</tr>
<tr>
<td>sql execute elapsed time</td>
<td>Amount of elapsed time SQL statements are executing. NOTE: For select statements this also includes the amount of time spent performing fetches of query results.</td>
</tr>
<tr>
<td>connection management call elapsed time</td>
<td>Amount of elapsed time spent performing session connect and disconnect calls.</td>
</tr>
<tr>
<td>failed parse elapsed time</td>
<td>Amount of time spent performing SQL parses that ultimately fail with some parse error.</td>
</tr>
<tr>
<td>hard parse (sharing criteria) elapsed time</td>
<td>Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from not being able to share an existing cursor in the SQL cache.</td>
</tr>
<tr>
<td>hard parse (bind mismatch) elapsed time</td>
<td>Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from bind type or bind size mismatch with an existing cursor in the SQL cache.</td>
</tr>
<tr>
<td>PL/SQL execution elapsed time</td>
<td>Amount of elapsed time spent running the PL/SQL interpreter. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing the Java VM.</td>
</tr>
</tbody>
</table>
The System Time Detail section allows you to see where the database engine has been spending its time. Both raw time and percentage of total metrics are included.

The metrics shown here include (*note*, this table is largely from Oracle 10g documentation):

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PL/SQL compilation elapsed time</strong></td>
<td>Amount of elapsed time spent running the PL/SQL compiler.</td>
</tr>
<tr>
<td><strong>inbound PL/SQL rpc elapsed time</strong></td>
<td>Time inbound PL/SQL remote procedure calls have spent executing. It includes all time spent recursively executing SQL and JAVA, and therefore is not easily related to &quot;PL/SQL execution elapsed time&quot;.</td>
</tr>
<tr>
<td><strong>Java execution elapsed time</strong></td>
<td>Amount of elapsed time spent running the Java VM. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing PL/SQL.</td>
</tr>
<tr>
<td><strong>DB Time</strong></td>
<td>Amount of elapsed time (in microseconds) spent performing Database user-level calls. This does not include the time spent on instance background processes such as PMON.</td>
</tr>
<tr>
<td><strong>DB CPU</strong></td>
<td>Amount of CPU time (in microseconds) spent on database user-level calls. This does not include the CPU time spent on instance background processes such as PMON.</td>
</tr>
<tr>
<td><strong>background cpu time</strong></td>
<td>Amount of CPU time (in microseconds) consumed by database background processes.</td>
</tr>
<tr>
<td><strong>sequence load elapsed time</strong></td>
<td>Amount of elapsed time spent getting the next sequence number from the data dictionary. If a sequence is cached, then this is the amount of time spent replenishing the cache when it runs out. No time is charged when a sequence number is found in the cache. For non-cached sequences, some time will be charged for every nextval call.</td>
</tr>
<tr>
<td><strong>parse time elapsed</strong></td>
<td>Amount of elapsed time spent parsing SQL statements. It includes both soft and hard parse time.</td>
</tr>
<tr>
<td><strong>hard parse elapsed time</strong></td>
<td>Amount of elapsed time spent hard parsing SQL statements.</td>
</tr>
<tr>
<td><strong>sql execute elapsed time</strong></td>
<td>Amount of elapsed time SQL statements are executing. Note that for select statements this also includes the amount of time spent performing fetches of query results.</td>
</tr>
<tr>
<td><strong>connection management call elapsed time</strong></td>
<td>Amount of elapsed time spent performing session connect and disconnect calls.</td>
</tr>
<tr>
<td><strong>failed parse elapsed time</strong></td>
<td>Amount of time spent performing SQL parses which ultimately fail with some parse error.</td>
</tr>
<tr>
<td><strong>hard parse (sharing criteria) elapsed time</strong></td>
<td>Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from not being able to share an existing cursor in the SQL cache.</td>
</tr>
<tr>
<td><strong>hard parse (bind mismatch) elapsed time</strong></td>
<td>Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from bind type or bind size mismatch with an existing cursor in the SQL cache.</td>
</tr>
<tr>
<td><strong>PL/SQL execution elapsed time</strong></td>
<td>Amount of elapsed time spent running the PL/SQL interpreter. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing the Java VM.</td>
</tr>
<tr>
<td><strong>PL/SQL compilation elapsed time</strong></td>
<td>Amount of elapsed time spent running the PL/SQL compiler.</td>
</tr>
</tbody>
</table>
**Session Time Detail**

The Session Time Detail section allows you to see where sessions that are currently logged on are spending their time.

**Metrics**

The metrics shown here include (note, this table is largely from Oracle 10g documentation):

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inbound PL/SQL rpc elapsed time</td>
<td>Time inbound PL/SQL remote procedure calls have spent executing. It includes all time spent recursively executing SQL and JAVA, and therefore is not easily related to &quot;PL/SQL execution elapsed time&quot;.</td>
</tr>
<tr>
<td>Java execution elapsed time</td>
<td>Amount of elapsed time spent running the Java VM. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing PL/SQL.</td>
</tr>
<tr>
<td>DB Time</td>
<td>Amount of elapsed time (in microseconds) spent performing Database user-level calls. This does not include the time spent on instance background processes such as PMON.</td>
</tr>
<tr>
<td>DB CPU</td>
<td>Amount of CPU time (in microseconds) spent on database user-level calls. This does not include the CPU time spent on instance background processes such as PMON.</td>
</tr>
<tr>
<td>background cpu time</td>
<td>Amount of CPU time (in microseconds) consumed by database background processes.</td>
</tr>
<tr>
<td>sequence load elapsed time</td>
<td>Amount of elapsed time spent getting the next sequence number from the data dictionary. If a sequence is cached, then this is the amount of time spent replenishing the cache when it runs out. No time is charged when a sequence number is found in the cache. For non-cached sequences, some time will be charged for every nextval call.</td>
</tr>
<tr>
<td>parse time elapsed</td>
<td>Amount of elapsed time spent parsing SQL statements. It includes both soft and hard parse time.</td>
</tr>
<tr>
<td>hard parse elapsed time</td>
<td>Amount of elapsed time spent hard parsing SQL statements.</td>
</tr>
<tr>
<td>sql execute elapsed time</td>
<td>Amount of elapsed time SQL statements are executing. Note that for select statements this also includes the amount of time spent performing fetches of query results.</td>
</tr>
<tr>
<td>connection management call elapsed time</td>
<td>Amount of elapsed time spent performing session connect and disconnect calls.</td>
</tr>
<tr>
<td>failed parse elapsed time</td>
<td>Amount of time spent performing SQL parses which ultimately fail with some parse error.</td>
</tr>
<tr>
<td>hard parse (sharing criteria) elapsed time</td>
<td>Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from not being able to share an existing cursor in the SQL cache.</td>
</tr>
<tr>
<td>hard parse (bind mismatch) elapsed time</td>
<td>Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from bind type or bind size mismatch with an existing cursor in the SQL cache.</td>
</tr>
<tr>
<td>PL/SQL execution elapsed time</td>
<td>Amount of elapsed time spent running the PL/SQL interpreter. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing the Java VM.</td>
</tr>
<tr>
<td>PL/SQL compilation elapsed time</td>
<td>Amount of elapsed time spent running the PL/SQL compiler.</td>
</tr>
<tr>
<td>inbound PL/SQL rpc elapsed time</td>
<td>Time inbound PL/SQL remote procedure calls have spent executing. It includes all time spent recursively executing SQL and JAVA, and therefore is not easily related to &quot;PL/SQL execution elapsed time&quot;.</td>
</tr>
</tbody>
</table>
System Bottlenecks

The System Bottlenecks tab of the Response Time Analyst detail view displays the following information:

<table>
<thead>
<tr>
<th>Bottleneck Summary</th>
<th>Bottleneck Detail</th>
<th>Bottleneck History (One Hour)</th>
</tr>
</thead>
</table>

**Bottleneck Summary**

- **Metrics**

Waits on a system generally occur for three reasons:

1. A process waits because it has no work to do.
2. A process waits because a requested resource is not available.
3. A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Bottleneck Summary section presents the broad categories of waits (called classes) so you can see if, for example, User I/O is causing high wait times. The percentage of each wait class is shown in the pie chart.

**Metrics**

Details can be viewed for the wait classes by drilling down into the System Waits view.

**Bottleneck Detail**

- **Metrics**

Waits on a system generally occur for one of three reasons:

1. A process waits because it has no work to do.
2. A process waits because a requested resource is not available.
3. A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Bottleneck Detail section presents the broad categories of waits (called classes) so you can see if, for example, User I/O is causing high wait times. The wait class, total waits, percentage of total waits, wait time (in seconds) and percentage of wait time is shown. Note that all these metrics are cumulative since the instance was started.

**Metrics**

Details can be viewed for the wait classes by drilling down into the System Waits view.
Bottleneck History (One Hour)
Waits on a system generally occur for three reasons:

1. A process waits because it has no work to do.
2. A process waits because a requested resource is not available.
3. A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention.

The Bottleneck History section allows you to view wait counts, wait time, or a count of user sessions waiting over the past hour for all the specific wait classes (User I/O, System I/O, etc.)

Session Bottlenecks
The Session Bottlenecks tab of the Response Time Analysis detail view displays the following information:

<table>
<thead>
<tr>
<th>Session Bottleneck Summary</th>
<th>Session Bottleneck Detail</th>
<th>Session Wait History (One Hour)</th>
</tr>
</thead>
</table>

Session Bottleneck Summary

Waits on a system generally occur for three reasons:

1. A process waits because it has no work to do.
2. A process waits because a requested resource is not available.
3. A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention.

The Session Bottleneck Summary section shows where currently logged on sessions have been spending the largest percentages of their time (over broad wait classes).

Metrics
To view the actual wait events that comprise the highest percentages of time viewed in each wait class, drill down into the Session Waits view.

Session Bottleneck Detail

Waits on a system generally occur for three reasons:

1. A process waits because it has no work to do.
2. A process waits because a requested resource is not available.
3. A process waits for Oracle to perform a prerequisite task for its given operation.
Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention.

The Session Bottleneck Detail section shows where currently logged on sessions have been spending the largest percentages of their time (over broad wait classes). Each session is shown along with its SID, user name, wait class, total waits, time waited (in seconds), and percent of total time.

**Metrics**

To view the actual wait events that comprise the highest wait times/percentages of time viewed in each wait class, drill down into the Session Waits view.

**Session Wait History (One Hour)**

The Session Wait History allows you to look back over the last hour to see when sessions where waiting, how many waits occurred, and the amount of wait time associated during a specific period of time. Analyzing such data over time will give you an idea of how often waits are occurring and their overall impact on system performance.

**Wait Histogram**

The Wait Histogram view allows you to view a histogram of wait times and wait counts for a variety of wait events. Using the wait histogram view, you can see, for example, that a db file scattered read event had 20 waits that took an average of 1 second and 10 waits that took an average of 2 seconds, and so on. This allows you to get a better idea of the actual impact of various wait events than viewing broad percentages or averages.

**Historical Session Analysis**

The Historical Session Analysis tab of the Response Time Analysis detail view shows data for:

- **Historical Session Analysis**

- **Historical Session Analysis Detail - All Waits**

**Historical Session Analysis**

Oracle10g and above allow you to review various historical performance metrics for your database. The Historical Analysis graph allows you to input a time period and review wait times to see when your server experienced high volumes of wait activity. You can choose to look at global wait classes or specific waits events for wait classes.

You can also select a specific time period in the graph by drawing a box around part of the graph with your mouse. The graph will change to match your selection.

**Historical Session Analysis Detail - All Waits**

Oracle10g and above allows you to review various historical performance metrics for your database. The Historical Analysis Detail section allows you to review specific details regarding times when your database server experienced unacceptable wait times. You can review historical response time data for sessions, SQL statements, files, and objects.

This section allows you, for example, to see what wait events long running SQL statements experienced and what objects caused excessive wait times.
Session Details
The following tabbed pages are available on the Session Details page:

<table>
<thead>
<tr>
<th>Current Work</th>
<th>Locks</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL</td>
<td>Waits</td>
<td></td>
</tr>
</tbody>
</table>

Overview Tab
Several sections are available on the Overview tab of the Session Details page. The table below describes the sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Usage</td>
<td>The Memory Usage section defines the session's total memory usage against all session memory usage on the database.</td>
</tr>
<tr>
<td>I/O Activity</td>
<td>The I/O section defines the session's total I/O consumption against all I/O activity on the database. <strong>NOTE</strong> Bugs in some versions of the Oracle data dictionary cause negative numbers to display if the database has experienced large amounts of I/O activity.</td>
</tr>
<tr>
<td>CPU Activity</td>
<td>The CPU Activity section defines the session's total CPU consumption against all CPU usage on the database. <strong>NOTE</strong> Bugs in some versions of the Oracle data dictionary cause the total CPU usage for a session to not be calculated correctly.</td>
</tr>
<tr>
<td>Statistical Detail</td>
<td>The Statistical Detail section details every performance metric available for the session. <strong>NOTE</strong> Each measure continue to increment as long as the session stays connected to Oracle.</td>
</tr>
</tbody>
</table>

Current Work Tab
The Current Work tab of the Session Details page includes the following sections:
- Open Transactions
- Open Cursors

Open Transactions
This Open Transactions section displays details regarding any open transaction that the session currently has. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Text</td>
<td>The SQL that contains the transaction.</td>
</tr>
<tr>
<td>Start Time</td>
<td>The time the statement was issued.</td>
</tr>
<tr>
<td>Logical I/O</td>
<td>The amount of memory I/O the transaction has used.</td>
</tr>
<tr>
<td>Physical I/O</td>
<td>The amount of disk I/O the transaction has used.</td>
</tr>
<tr>
<td>Gets</td>
<td>A subset of logical I/O.</td>
</tr>
<tr>
<td>Changes</td>
<td>Any data modifications made by the transaction.</td>
</tr>
</tbody>
</table>

Metrics
Seeing a long running, persistent transaction might indicate a session caught in a loop or other coding snafu. Watching a transaction sit idle could clue you into a lock contention problem that exists.
Open Cursors
Programs opening cursors consume Oracle resources and have the potential to degrade performance, especially if the SQL code being used in the cursor is inefficient. The Open Cursors tab lets you quickly spot user accounts that have many cursors opened as well as the actual performance statistics for each opened cursor.

The Open Cursor section displays pertinent performance statistics for each opened cursor in the system. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Text</td>
<td>The SQL text being used for the cursor.</td>
</tr>
<tr>
<td>Disk Reads</td>
<td>The number of physical I/O reads.</td>
</tr>
<tr>
<td>Buffer Gets</td>
<td>The number of buffer gets.</td>
</tr>
<tr>
<td>Rows Processed</td>
<td>The number of rows returned for the statement.</td>
</tr>
<tr>
<td>Parse Calls</td>
<td>The total of all parse calls to all the child cursors under this parent.</td>
</tr>
<tr>
<td>Sharable Memory</td>
<td>The sum of all sharable memory, in bytes, of all the child cursors under this parent.</td>
</tr>
<tr>
<td>Persistent Memory</td>
<td>The sum of all persistent memory, in bytes, of all the child cursors under this parent.</td>
</tr>
<tr>
<td>Runtime Memory</td>
<td>The sum of all the ephemeral frame sizes of all the children.</td>
</tr>
<tr>
<td>Sorts</td>
<td>The sum of the number of sorts that were done for all the children of the parent code.</td>
</tr>
<tr>
<td>Loaded Versions</td>
<td>The number of children that are present in the cache and have their context heap loaded.</td>
</tr>
<tr>
<td>Loads</td>
<td>The number of times the object was loaded or reloaded.</td>
</tr>
<tr>
<td>Execs</td>
<td>The number of times the code was executed since being brought into the shared pool.</td>
</tr>
<tr>
<td>First Load Time</td>
<td>The creation timestamp of the parent code.</td>
</tr>
</tbody>
</table>

Metrics
The Init.ora parameter OPEN_CURSORS controls the maximum number of open cursors (context areas) a session can have at once. Seeing individual sessions approaching this limit should be a cause of concern. If you want to avoid cursor allocation errors, you should set it to a very high number. The Init.ora file used to have a 255 cursor limit, but that has been exceeded in Oracle8 and later.

With respect to individual cursor performance statistics, be on the lookout for cursors with abnormally high physical I/O counts (which may indicate inefficient SQL or indexing schemes), and cursors with high load counts (which may indicate an undersized library cache).

Waits Tab
The Waits tab of the Session Details page includes the following sections:

<table>
<thead>
<tr>
<th>Current Waits</th>
<th>Historical Waits</th>
<th>Time Summary (Oracle 10g only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Ten Waits (Oracle 10g only)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Historical Waits
Waits on a system generally occur for three reasons:

1. A process waits because it has no work to do.
2. A process waits because a requested resource is not available.
3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Historical Waits section presents all the various waits that have occurred for all currently connected oracle sessions. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>The name of the wait event.</td>
</tr>
<tr>
<td>Total Waits</td>
<td>The number of waits for the event.</td>
</tr>
<tr>
<td>Percent of Total Waits</td>
<td>The percentage of all waits that this event makes up.</td>
</tr>
<tr>
<td>Time Waited (Secs)</td>
<td>The total amount of time waited for the event, in seconds.</td>
</tr>
<tr>
<td>Average Wait (Secs)</td>
<td>The average amount of time waited for the event, in seconds.</td>
</tr>
<tr>
<td>Max Wait (Secs)</td>
<td>The maximum amount of time waited for the event, in seconds.</td>
</tr>
<tr>
<td>Total Timeouts</td>
<td>The number of timeouts for the event.</td>
</tr>
</tbody>
</table>

**Metrics**

Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include:

- SQL*Net more data from client
- SQL*Net message from client
- client message
- rdbms ipc message
- pipe get
- pmon timer
- smon timer
- Null event

**Troubleshooting**

The table below describes wait events that deserve attention:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enqueue waits</td>
<td>If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).</td>
</tr>
<tr>
<td>Buffer busy waits</td>
<td>Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.</td>
</tr>
<tr>
<td>db file scattered read waits</td>
<td>An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.</td>
</tr>
</tbody>
</table>
The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database. To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Current Waits
Waits on a system generally occur for three reasons:

- A process waits because it has no work to do.
- A process waits because a requested resource is not available.
- A process waits for Oracle to perform a prerequisite task for its given operation.

Session contention is merely a subset of contention that is viewable at the global database level. Oftentimes, it takes analysis at the session level to pinpoint the exact source of contention that is occurring globally. So you need to become accustomed to viewing contention statistics and waits at the user process level.

When monitoring waits with respect to user sessions, there are two areas of interest:

1. What HAS the user session been waiting on?
2. What IS the user session waiting on?

Oracle records both sets of wait statistics for you. In reviewing previous waits for a session, you can see which types of things have caused the session to be bottlenecked. The Current Waits section displays information to help you troubleshoot session contention. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>The name of the wait event.</td>
</tr>
<tr>
<td>Wait (Secs)</td>
<td>The number of seconds in the current wait.</td>
</tr>
<tr>
<td>Last Wait Time</td>
<td>A non-zero value is the session's last wait time. A zero value means the session is currently waiting. A -2 value indicates the platform cannot support a fast timing mechanism or TIMED_STATISTICS is not set for the system.</td>
</tr>
<tr>
<td>State</td>
<td>Indicates the possible values. They includes the following:</td>
</tr>
<tr>
<td></td>
<td>WAITING - the session is waiting.</td>
</tr>
<tr>
<td></td>
<td>WAITED UNKNOWN TIME - duration of last wait is not known.</td>
</tr>
<tr>
<td></td>
<td>WAITED SHORT TIME - last wait was less than 1/100th of a second.</td>
</tr>
<tr>
<td></td>
<td>WAITED KNOWN TIME - the wait is equal to the time of the last wait.</td>
</tr>
<tr>
<td>P1 - P3</td>
<td>Any additional wait parameters.</td>
</tr>
<tr>
<td>P1TEXT - P3TEXT</td>
<td>Any descriptions of additional parameter.</td>
</tr>
<tr>
<td>P1RAW - P3RAW</td>
<td>Any additional wait parameters.</td>
</tr>
</tbody>
</table>

Metrics
Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latch free waits</td>
<td>Latch free waits indicate contention for latches.</td>
</tr>
</tbody>
</table>
• SQL*Net more data from client
• SQL*Net message from client
• client message
• rdbms ipc message
• pipe get
• pmon timer
• smon timer
• Null event

Troubleshooting
The table below describes wait events that deserve attention:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enqueue waits</td>
<td>If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).</td>
</tr>
<tr>
<td>Buffer busy waits</td>
<td>Buffer busy waits normally center around contention for rollback segments, too small an INITTRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITTRANS to allow for more transactions per data block, and more free lists.</td>
</tr>
<tr>
<td>db file scattered read waits</td>
<td>An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.</td>
</tr>
<tr>
<td>Latch free waits</td>
<td>Latch free waits indicate contention for latches.</td>
</tr>
</tbody>
</table>

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database. To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Time Summary (Oracle 10g only)
The Time Summary section shows you where the session has been spending its time. Using the Time Summary section, you can easily determine if a session has been bottlenecked by SQL calls, PL/SQL work, or other activities.

Last Ten Waits (Oracle 10g only)
The Last Ten Waits section shows you, in order, the last 10 wait events experienced by the session, along with other important metrics like the wait time, and so forth.
Locks Tab
The Locks tab of the Session Details page includes the following sections:

- All Locks
- Blocking Locks

All Locks
To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, Oracle itself will issue lock requests to carry out its internal duties. The All Locks section displays information about all locks currently on a system. The table below lists the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
<td>The workstation of the Oracle process.</td>
</tr>
<tr>
<td>Serial #</td>
<td>The serial number of the session holding the lock.</td>
</tr>
<tr>
<td>Table</td>
<td>The table name of the locked table.</td>
</tr>
<tr>
<td>Mode</td>
<td>The lock mode (EXCLUSIVE, SHARE, etc.).</td>
</tr>
<tr>
<td>Request</td>
<td>The lock requests that Oracle issues to carry out its internal duties.</td>
</tr>
<tr>
<td>Blocking?</td>
<td>Indicates if the lock is a blocking lock.</td>
</tr>
</tbody>
</table>

Metrics
Locks that are held for unusually long periods of time may be candidates for further investigation. The application logic may be inefficient or perhaps the program is not issuing frequent enough COMMITs.

Blocking Locks
Without a doubt, blocking lock situations can give the appearance of a frozen database almost more than anything else (except, perhaps, for a stuck archiver process). A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up - sometimes frequently.

The Blocking Locks section contains information relating to user accounts that are currently blocked and the sessions that are blocking them. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocked User</td>
<td>The user account of the session waiting for the lock.</td>
</tr>
<tr>
<td>Blocking User</td>
<td>The user account of the session holding the offending lock.</td>
</tr>
<tr>
<td>Waiting SID</td>
<td>The session identifier of the session waiting for the lock.</td>
</tr>
<tr>
<td>Holding SID</td>
<td>The session identifier of the session holding the offending lock.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of lock (TRANSACTION, DML, etc.).</td>
</tr>
<tr>
<td>Mode</td>
<td>The lock mode (EXCLUSIVE, SHARE, etc.).</td>
</tr>
<tr>
<td>Request</td>
<td>The type of lock being requested by the session.</td>
</tr>
<tr>
<td>Row</td>
<td>The rowid of the row being held.</td>
</tr>
<tr>
<td>Object</td>
<td>The name of the object being locked.</td>
</tr>
<tr>
<td>ID 1</td>
<td>Lock identifier #1 (depends on type).</td>
</tr>
</tbody>
</table>
Metrics

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT … FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Data warehouses (whose data is mostly read) can benefit from tablespaces set in READ ONLY mode. READ ONLY signals to the other database that exclusive locks need not be used for the data contained within the tablespace. This is especially helpful in Oracle Parallel Server environments and drastically reduces ping activity.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID 2</td>
<td>Lock identifier #2 (depends on type).</td>
</tr>
<tr>
<td>Time Waited (Secs)</td>
<td>The amount of time, in seconds, the blocked user has been waiting.</td>
</tr>
</tbody>
</table>

SQL Tab

The SQL tab of the Session Details page includes the following sections:

- **Current SQL**
- **Current SQL Statistics**
- **Other SQL**

**Current SQL**

The Current SQL section displays the most current SQL issued by the session.

**NOTE:** You can run an EXPLAIN PLAN on the SQL or (if you have Embarcadero SQL Tuner installed) tune the SQL for better performance.

**Current SQL Statistics**

The Current SQL Statistics section displays the performance execution metrics for the current SQL statement in graphical format. Physical and logical I/O is displayed along with rows processed and number of executions.
Other SQL
The Other SQL section displays other SQL that has been parsed by the Oracle username. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Text</td>
<td>A snippet of the SQL code issued.</td>
</tr>
<tr>
<td>Disk Reads</td>
<td>The number of total disk reads for the statement.</td>
</tr>
<tr>
<td>Buffer Gets</td>
<td>The number of total buffer gets for the statement.</td>
</tr>
<tr>
<td>Rows Processed</td>
<td>The number of total rows processed for the statement.</td>
</tr>
<tr>
<td>Executions</td>
<td>The number of times the statement has been executed.</td>
</tr>
<tr>
<td>Sorts</td>
<td>The number of sorts for the statement.</td>
</tr>
</tbody>
</table>

**NOTE:** Bugs in some versions of Oracle will cause negative numbers to display for various execution metrics.
Performance Analyst for Microsoft SQL Server Tutorial

Embarcadero Performance Analyst provides key insight into the complex world of database performance. Running inside the Embarcadero DBArtisan console, or standalone in a Web browser, Performance Analyst helps you pinpoint performance problems that threaten a database’s availability or response times. The powerful analysis facilities will satisfy advanced experts, yet novices will find Performance Analyst’s interface easy to use and the expert help a welcome aid to troubleshooting difficult performance issues.

About This Performance Analyst for Microsoft Tutorial

This tutorial is intended to help you get started with Embarcadero Performance Analyst.

After completing this tutorial, you will have the foundation you need to explore the many features and benefits of Performance Analyst. You will have learned how to accurately monitor your database using the most popular performance methodologies in use today. You will know how to drill down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. Finally, you’ll understand how to fix performance problems using the integration that exists between Performance Analyst and Embarcadero DBArtisan.

This tutorial is divided into 8 sections.

Session 1: Getting Started with Performance Analyst
Session 2: Using Ratio-based Performance Analysis
Session 3: Using Bottleneck/Wait-based Performance Analysis
Session 4: Using Workload Performance Analysis
Session 5: Using SQL Performance Analysis
Session 6: Using Operating System Performance Analysis
Session 7: Identifying and Getting Help with Performance Problems
Session 8: Fixing Identified Problems Using Performance Analyst for Microsoft

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief guide.

For more information, see Additional Evaluation Resources.

Session 1: Getting Started with Performance Analyst for Microsoft

Download and Install

You can obtain the latest version of Performance Analyst by downloading the DBArtisan software from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Performance Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Performance Analyst (although Performance Analyst can also run in standalone fashion within a Microsoft Internet Explorer Web browser).

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.
Overview of Performance Analyst for Microsoft

When DBArtisan is connected to a Microsoft SQL Server database, you can open Performance Analyst two ways:

1. On Utilities menu, select Performance Analyst.

For Performance Analyst to work, several procedures need to be installed by the SA account. When Performance Analyst is first invoked, it will check for the presence of this package and if it has not been installed, the product will prompt you to install it. Once created, any account with access to these procedures may monitor databases using Performance Analyst. No other objects need to be installed on a monitored database.

Performance Analyst for Microsoft Product Navigation

Navigating through Performance Analyst is easily accomplished through two main mechanisms:

1. The main Performance Analyst toolbar contains buttons that invoke the major performance analysis views within the product as well as frequently used actions like Refresh and Performance Threshold Customizations.

2. Positioning the mouse pointer on any area within Performance Analyst and clicking the right mouse button will bring up a navigation menu that allows you to easily move between any of the main Performance Analyst home pages and drill down views. The right mouse menu also contains context sensitive options like drilling down further into a particular statistic, copying the display to a text file, and more.

Session 2: Using Ratio-Based Performance Analysis in Performance Analyst for Microsoft

Ratio-based analysis has been around for many years, and used to be the only technique database administrators used when they were called on to diagnose the cause of a database slowdown. Performance ratios are very good roll-up mechanisms for busy DBAs, making possible the analysis-at-a-glance approach, however ratios must be computed correctly for them to be meaningful. Key ratios in Performance Analyst are calculated using delta statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. Performance Analyst makes it easy to see the ratios that are computed with delta statistics vs. cumulative. A triangle symbol denotes a delta-based statistic, while square symbols represent a cumulative-based measure.

Ratio-based analysis is available for all performance categories (memory, etc.) where ratios can assist in determining the performance level of a database. For example, if you would like to see memory-specific ratios, on the Performance Analyst toolbar click Memory. Performance Analyst displays the Memory home page that showcases all important memory-related ratios.

Obtaining Detail Information for a Ratio Using Performance Analyst for Microsoft

A ratio gives you a global picture for a particular performance measure. However, you often need more information regarding a statistical measure so you can make an informed decision. Performance Analyst has easy-to-use drill down features that make obtaining detailed information for various ratios a breeze.

For example, to view more information for the Buffer Cache Hit Ratio statistic, do the following:

1. On either the Main Home page or the Memory Home page, locate the Buffer Cache Hit Ratio statistic.

2. Position the pointer over the statistic.

   The pointer changes to a hand symbol.
3 Double-click the left mouse button.

Performance Analyst opens the Buffer Cache Tab of the memory drill down view.

OR

4 Right-click the statistic, select Details.

OR

5 Select Navigate to, Memory, Buffer Cache.

Performance Analyst opens the Buffer Cache Tab of the memory drill down view.

Session 3: Using Bottleneck/Wait-Based Performance Analysis in Performance Analyst for Microsoft

When a Microsoft SQL Server database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things, or it can be an indicator that a database bottleneck exists. Bottleneck analysis is a valid method of measuring performance because it helps you track where a database and user sessions have been spending their time. If latch contention or heavy table-scan activity has been dragging a database's performance down, you can use bottleneck analysis to confirm the actual root cause.

Performance Analyst uses two broad areas of metrics to form its bottleneck analysis. The first is the Microsoft SQL Server wait event interface, which displays where the database and user sessions have been bottlenecked. But, when using bottleneck analysis you can not rely only on the information contained in the wait event views that Microsoft SQL Server provides. For example, an object may attempt to extend into another extent of space in a tablespace and yet, be denied if no such free space exists. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. For this reason, Performance Analyst also shows non-wait event based bottlenecks that represent a very real threat to a database's availability and performance.

Performance Analyst prominently displays bottleneck analysis on every performance home page. The Main home page, displays a summary of all key bottlenecks threatening the database. You can also see specific bottlenecks for memory, I/O, space, objects, and users by navigating to each home page and viewing the Bottleneck Analysis section for that home page. For example, to see bottlenecks that specifically relate to I/O, do the following

1 On the toolbar, click the I/O button.

2 Locate the Bottleneck Analysis section at the lower-left portion of the screen.

The I/O bottleneck analysis section displays the Top I/O Wait Events by wait count and wait time, along with the Hottest Database and Log Files.

Obtaining Detail on System Bottlenecks Using Performance Analyst for Microsoft

You can easily obtain more information regarding global bottlenecks that are plaguing a database by drilling down into a particular bottleneck metric. For example, if you would like to see more data regarding the current top system bottlenecks, do the following:

1 Open the Main home page.

2 Double-click the Top System Bottlenecks area.

Performance Analyst opens the System Waits view.

3 Or right-click anywhere in Performance Analyst, point to Users, and then click System Waits.

Performance Analyst opens the System Waits view.
The System Waits view displays critical details regarding all the wait events that Microsoft SQL Server has recorded. You can easily see what events that occurred the most often as well as the events that have accumulated the most wait time.

**TIP:** If you would like to see an expanded view of the System waits grid shown on the top of the System Waits view, click the Expand button located at the top right corner of the section. All section views in Performance Analyst can be maximized/minimized using this control.

**Viewing Detail for Session Bottlenecks Using Performance Analyst for Microsoft**

Not only can you obtain detail information for system bottlenecks, but you can also use Performance Analyst to get drill down information regarding session-level bottlenecks. There are several layers to session-level details you can view.

To see more information regarding session-level waits, do the following:

1. Open the **Main** home page.
2. Double-click the **Top Session Bottlenecks** area.
3. **OR**
   - Right-click anywhere in Performance Analyst, point to **Users**, and then click **Session Waits**.

   Performance Analyst opens the Session Waits drill down view.

When you are in the Session Waits drill down view, to drill further into a session’s details, do the following:

1. In the **Historical Waits** or **Current Waits** grid, select a session that appears to be experiencing critical waits.
2. Double-click the session or right-click, and then select **Details**.

   Performance Analyst opens a drill down view that shows key statistics for the selected session, including its current work activities, wait events, and SQL statements.

**Session 4: Using Workload Performance Analysis in Performance Analyst for Microsoft**

Key ratios give you a global perspective on database activity. Bottleneck analysis gives you insight into things that are holding up user activity and throughput. But another technique is necessary if a database professional is to really get a handle on what’s occurring inside a badly performing database.

Workload analysis involves the investigation of two critical areas of a database’s performance:

- Session resource consumption and activity
- SQL execution

Without looking at these two key performance categories, you can miss a lot of what could be responsible for perceived performance problems. When performance on a database takes a sudden nosedive, it is not uncommon to find one or two sessions that are causing the bulk of the workload. Finding these processes can be easily accomplished by viewing session metadata coupled with resource consumption and statistical execution statistics.
Pinpointing Top Resource Consumers Using Performance Analyst for Microsoft

Performance Analyst identifies top resource consumers in a couple of different ways. On the main home page, Performance Analyst highlights the top resource sessions across I/O, memory usage, and CPU consumption. The percentage used across all statistical categories are displayed so a DBA can immediately pinpoint a session that is using all or most of a particular resource. The top resource sessions are also displayed on the memory, I/O, and other home pages as well with a listing for the top sessions for that category (for example, the top memory users appear on the memory home page, etc.)

Getting Workload Details Using Performance Analyst for Microsoft

Performance Analyst offers several details on top resource sessions in addition to the summaries provided on each home page. A detailed and flexible top sessions view lets you view a summary of all the major resource sessions on the database as well as every critical detail regarding each session connected to the database.

Further drill downs are available that highlights specific details regarding each session’s activities and submitted SQL. To see detailed information regarding a session identified in one of Performance Analyst’s Workload Analysis sections, do the following:

1. In a Workload Analysis display, position the pointer over one of the processes.
2. Double-click the process or right-click, and then select Details.

If you want like to see all key workload analysis metadata for sessions, do the following:

1. On the toolbar, click the Top Sessions button.
   OR
2. Right-click anywhere in Performance Analyst, point to Users, and then click Session Waits.

Session 5: Using SQL Performance Analysis in Performance Analyst for Microsoft

SQL-based performance analysis is really a subset of Workload Analysis. Understanding current and historical SQL execution patterns lets you to have the second set of data points necessary to properly perform workload analysis. You may find that optimizing SQL code produces some of the best performance-enhancing boosts available for a database.

Uncovering Resource-Intensive SQL Using Performance Analyst for Microsoft

Performance Analyst offers plenty of insight into SQL performance and execution patterns. Each one of the home pages depicts the most costly SQL statements as determined by a variety of different metrics for that performance category (most physical, logical I/O, etc.) For example, Performance Analyst might show that a single SQL statement is responsible for almost 60 percent of all the CPU consumption on the database.
Getting Details on Top SQL with Performance Analyst for Microsoft

Getting details on identified SQL statements is easy in Performance Analyst. For example, if you would like to see further information on a SQL statement identified on the Performance Analyst Home page, do the following:

1. In the **SQL Analysis** section, position the pointer over **Hash Values**.
2. Double-click or right-click, and then select **Details**.

   Performance Analyst opens the Top SQL summary page that shows a breakdown of all top SQL statements along with their execution statistics.

If you would like to see the full SQL text of a SQL statement, do the following

1. In the **Top SQL Details** section, double-click a statement.

   Performance Analyst opens the SQL Details Tab of the Top SQL view to display the entire SQL statement and the execution metrics. While on this pane, you can also perform an EXPLAIN of the SQL statement and examine its access path information.

To automatically tune a problem SQL statement if you have Embarcadero SQL Tuner installed on your workstation, do the following:

1. On the **SQL Details** Tab, select a problem statement, and then click the **Tune SQL** button.

   SQL Tuner opens and you can begin a tuning session.

Session 6: Using Operating System Performance Analysis in Performance Analyst for Microsoft

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O or Memory), a slow CPU, insufficient I/O devices, and insufficient memory. Performance Analyst allows the DBA to examine the relevant operating system metrics for Windows Platforms (NT/2000/XP/2003) as they relate to SQL Server.

To view the operating system metrics of the machine hosting the SQL Server you are monitoring, do the following:

1. Click the **OS** Performance Analyst toolbar button.

   OR

2. Navigate to the OS Home Page by going to **OS**, and then **Home**.

   **NOTE:** The first time you invoke the OS feature you will be prompted for an operating system login. Please specify the correct operating system, host name, username & password and protocol.

The Operating System home page has been designed in a similar fashion as the datasource home page. On this page you can see the pertinent operating system metrics displayed in five quadrants broken down into the following categories:

- Resource Usage
- Memory Analysis
- Disk Analysis
- Workload Analysis
- Bottleneck Analysis
Obtaining Details on the Operating System Using Performance Analyst for Microsoft

You can easily obtain more information regarding operating system problems that are degrading database performance by drilling down into a particular analysis quadrant. For example, if you'd like to see more information regarding the processes running on the host machine, do the following:

1. Navigate to the Performance Analyst Operating System home page.
2. Double-click the Top Memory Processes in the Workload Analysis quadrant.
   
   Performance Analyst opens the Processes Drilldown view.

   OR

3. Right-click anywhere in Performance Analyst and select the Navigate to, OS, and then Processes.

   The Processes detail tab shows all the processes on the host machine. From this view, you can easily determine which processes are showing high marks in I/O or CPU, for example, by simply double clicking on the column headers and sorting the processes in ascending or descending order.

Session 7: Identifying and Getting Help with Performance Problems Using Performance Analyst for Microsoft

You do not have to be an advanced expert in performance tuning to spot trouble in a database if you use Performance Analyst. There are a number of mechanisms in the tool that help you quickly locate areas of concern within a database.

Using the Alarm Log in Performance Analyst for Microsoft

Performance Analyst has a central and flexible alarm log that lists every performance violation found in all monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst lists all alarms for each database you are monitoring in one central log so you can see every problem in one place. To view the Performance Analyst Alarm log, do the following:

1. On the toolbar, click the Alarms button.

   Once in the Alarm log, you can perform a number of actions including:

   • Selecting an alarm and viewing the application page that shows details regarding the metric in violation.
   • Filtering the types of alarms shown through the Options dialog.
   • Clearing all current alarms.
   • Copying or saving the Alarm log into a different format.

To select any of these options, do the following:

1. Position the pointer over any area of the Alarm Log.
2. Right-click and in the shortcut menu, select an option.

Recognizing Performance Violations Using Performance Analyst for Microsoft

Performance Analyst makes it easy to visually pinpoint performance violations in a database. Any statistic that does not conform to your predefined performance standards is colorized in any view that it appears so you can easily see potential troublespots.
The colors used to highlight a particular performance violation depend on the severity of the problem. Performance Analyst uses several layers of performance standards/threshold violations with each layer being represented by a different color. You can control everything regarding threshold violations, including how thresholds are determined, their color representation in Performance Analyst’s display, and more by using the Threshold Template feature of Performance Analyst.

How can Performance Analyst inform you of performance violations in your database if you minimize the product to your desktop and are working on other things? Even if Performance Analyst is not right in front of you, it can still let you know that a problem has been found in your database through the systray alarm icon. When Performance Analyst detects a problem in one of your databases, one of the things the product does is enable an alarm icon in your workstation’s systray area that communicates the presence of alarms in your database. So even if the Performance Analyst interface is not in sight, the product can let you know that something in your database requires attention.

Creating Threshold Templates for Performance Analyst for Microsoft
How does Performance Analyst know what to signal as a problem in a database? It uses predefined performance standards that you define and enforce. All of this is accomplished by using Performance Analyst’s Threshold templates. A template is a set of performance metrics and their thresholds that can be reused to enforce performance standards across any number of databases. Embarcadero provides a default performance template that contains nearly all of the most-watched performance statistics so you can begin monitoring with confidence right out of the box. However, you can create your own Threshold templates so you only are alerted to performance violations that are important to you and your particular environment.

To view or create Threshold templates, do the following:

1. On the toolbar, click the Thresholds button.

   Performance Analyst opens the Threshold Template Properties dialog box.

   The Threshold Template Properties dialog box lets you select the options to add or edit templates, as well as other maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your own needs. You can control what statistics are sampled for performance violations as well as the levels of severity you would like reported. Once you are satisfied with your template, you can save it and then apply it to one or more datasources in one action to immediately begin enforcing your performance standards.

Getting Help with Performance Problems Using Performance Analyst for Microsoft
As databases become more complex, so do the problems that plague them. Knowing what to do for a particular performance problem can be hard in some cases, but Performance Analyst helps by providing a complete set of expert help that you can use as a guide when performance problems occur. To obtain expert help for any statistic shown in Performance Analyst, do the following:

1. Position the pointer over a statistic that you need help.

2. Right-click and select Help.

   The Expert Help gives you a general description of the statistic, the range of values that are considered optimal, and some suggestions on how to fix a problem situation. Expert help can also be invoked for any of the Ratio, Bottleneck, SQL, or Workload Analysis sections shown on every home page by clicking on the help icon for that section, which is located at the top right corner of the section.

Session 8: Fixing Identified Problems Using Performance Analyst for
Microsoft
Finding performance problems is one thing, but fixing them is another. Many performance monitors point out a particular problem in a database, but then offer no way to actually correct the situation. Such is not the case with Performance Analyst. Because of its tight integration with DBArtisan, certain problems found in Performance Analyst can be visually corrected by DBArtisan.

A Space Problem Correction Example
If an Microsoft SQL Server tablespace is about to run out of free space, Performance Analyst can issue an alarm that a space shortage exists. Once the space problem has been identified, do the following:

1. On the Space home page toolbar, click the Space button.
2. In the Space Analyst grid, right-click a tablespace, and then select Manage with DBArtisan.

   DBArtisan opens an editor where you can perform corrective actions, such as enabling Autoextend for the tablespace, resizing the tablespace’s existing datafile(s), or adding a new datafile for the tablespace.
Performance Analyst for Microsoft SQL Server Expert Guide

This section includes expert help for all Microsoft SQL Server statistics in the Embarcadero Performance Analyst views and pages:

<table>
<thead>
<tr>
<th>Home Page Statistics</th>
<th>Database Page Statistics</th>
<th>I/O Page Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Page Statistics</td>
<td>OS Page Statistics</td>
<td>Space Page Statistics</td>
</tr>
<tr>
<td>Users Page Statistics</td>
<td>Top SQL Views</td>
<td></td>
</tr>
</tbody>
</table>

Home Page Statistics

The Embarcadero Performance Analyst Home page lets you review availability and overall performance of all monitored databases from a single window. Statistics on the Home view are organized into the following categories:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key Ratio Analysis</th>
<th>SQL Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Analysis</td>
<td>Workload Analysis</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

- Database Page Statistics
- I/O Page Statistics
- Memory Page Statistics
- OS Page Statistics
- Space Page Statistics
- Users Page Statistics

Key Ratio Analysis - Home

Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBA’s to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the master.sysperfinfo performance table. This table maintains a count of various statistics for the server and particular databases since the server was brought up. For servers that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA might be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server’s individual ‘personality’ with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.
The following ratios are used on the Performance Analyst Home page to succinctly communicate the general overall performance levels of the monitored database:

<table>
<thead>
<tr>
<th>Ad Hoc SQL Hit Ratio</th>
<th>Buffer Cache Hit Ratio</th>
<th>CPU Busy</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Busy</td>
<td>Log Cache Hit Ratio</td>
<td>Procedure Hit Ratio</td>
</tr>
<tr>
<td>Server Idle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Buffer Cache Hit Ratio**

- **Metrics**
- **Troubleshooting**

Data read from memory will produce end-user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the purposes of the SQL Server buffer/data cache.

The buffer cache hit ratio is a terrific indicator of how often user requests for data are satisfied through memory vs. being physically read from disk.

**NOTE:** This statistic is also available on the Memory home page.

**Metrics**

To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, you should be aware that every server has its own ‘personality’ and might exhibit excellent performance with below average readings for the cache hit ratio. You should also be aware that excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

Consistently viewed low readings (60% or lower) might require tuning attention on the part of the DBA.

**Troubleshooting**

Ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server memory configuration option. Also, consider increasing your SQL Server Min. Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Often a user process is taking a large amount of memory due to an inordinate amount of I/O.

**Procedure Plan Hit Ratio**

- **Metrics**
- **Troubleshooting**

The SQL Server procedure cache is used to hold the execution plans for all Transact-SQL statements currently executing in the server. When a user executes a Transact-SQL statement, SQL Server looks in the procedure cache for a query plan to use.

This statistic is the percentage of query plan requests generated by stored procedures that are found in the procedure cache area. The percentage of times that a statement's plan and definition can be referenced in memory, the better the procedure execution time.

**NOTE:** This statistic is also available on the Memory home page.
Metrics
A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for code reference. Note that when a database is first started, the procedure cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

Troubleshooting
First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Ad Hoc SQL Hit Ratio

Metrics
A high ad hoc hit rate is desirable, but is harder to maintain at a high level than something like a procedure cache hit rate. Therefore, an 80% or greater ad hoc cache hit rate is a good performance benchmark for code reference. Note that when a database is first started, the ad hoc cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the ad hoc cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

Troubleshooting
First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (To obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Log Cache Hit Ratio

Metrics
The Log Cache Hit Ratio represents the percentage of log cache reads satisfied from the log cache. This statistic is also available on the Memory home page.

Troubleshooting
First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (To obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.
Troubleshooting
A low percentage on this statistic is not necessarily a bad sign, as it is possible that the information needed from the log will not be readily available in memory.

CPU Busy
• Metrics
• Troubleshooting
The CPU Busy statistic represents the percentage of time that the CPU has spent working since the last refresh.

NOTE: This statistic is also available on the Database home page.

Metrics
Seeing consistently high numbers might signal an overworked server.

Troubleshooting
If you consistently see high numbers for CPU busy, then drill down into session activity to see what processes are consuming the most CPU and look at their activity.

I/O Busy
The I/O Busy statistic represents the percentage of time that SQL Server has spent performing input and output operations since the last refresh.

NOTE: This statistic is also available on the Database home page.

Metrics
None.

Server Idle
The Server Idle statistic represents the percentage of time that SQL Server has spent idle since the last refresh.

Metrics
None.

Bottleneck Analysis - Home
When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem. The Performance Analyst Home page identifies the top system and session waits that might be currently a cause of lessened performance.
When using bottleneck analysis, a DBA cannot rely only on the information contained in the wait events that SQL Server provides. For example, a database might be suspect or offline. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Home page displays space-related bottlenecks as well as other sources of contention/bottlenecks that might be dragging down the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

<table>
<thead>
<tr>
<th>Current Object Blocks</th>
<th>DB’s Low on Space</th>
<th>Deadlocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Errors in Current Log</td>
<td>Logs Low on Space</td>
<td>Offline Databases</td>
</tr>
<tr>
<td>Suspect Databases</td>
<td>Table Lock Escalations</td>
<td>Top System Bottlenecks</td>
</tr>
</tbody>
</table>

Top System Bottlenecks
The Top System Bottlenecks display identifies the top waits that have occurred on SQL Server based on the wait time.

**Metrics**
None.

**Troubleshooting**
Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

Current Object Blocks
- **Metrics**
- **Troubleshooting**

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely.

**Metrics**
You should immediately investigate any indicator above zero, before the situation has a chance to mushroom.

**Troubleshooting**
Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Then other user processes, almost always, complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into Users Detail and see all current blocking locks, learning exactly which sessions are holding the currently restrictive locks.
The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the `SET LOCK_TIMEOUT` command, which limits the number of seconds that a process will wait for a lock before timing out.

**Deadlocks**

The Deadlocks statistic represents the number of deadlocks per second detected by SQL Server. Deadlocks occur when processes cannot proceed because they are waiting on a set of resources held by each other or held by other processes.

**Metrics**

Consistently seeing deadlock counts greater than zero will indicate that some user processes are experiencing delays in completing their work. When SQL Server identifies a deadlock, it resolves the situation by choosing the process that can break the deadlock. This process is termed the deadlock victim. SQL Server rolls back the deadlock victim's transaction, and then notifies the process’ application by returning an error message. It also cancels the process’ request and allows the transactions of the remaining processes to continue.

SQL Server always attempts to choose the least expensive thread running the transaction as the deadlock victim.

**Troubleshooting**

Because SQL Server automatically resolves deadlock situations, you should do proactive work to prevent them in the first place.

The culprit of most blocking lock and deadlock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

You can change default deadlock behavior by using the `SET DEADLOCK_PRIORITY` command, which reprioritizes a process’ position in a deadlock situation.

**Table Lock Escalations**

The Table Lock Escalations statistic represents the number of times locks on a table were escalated.

**Metrics**

Many table lock escalations could indicate contention problems. If increasing numbers of table lock escalations are viewed at the same time as blocking or deadlock problems, then the application design might be at fault.

**Suspect Databases**

The Suspect Databases statistic represents the number of databases SQL Server has marked as suspect. Databases are marked suspect by SQL Server if they fail during automatic recovery, which is performed at server startup. If serious damage is experienced by a database during regular uptime, SQL server will also mark a database suspect.

**NOTE:** This statistic is also available on the Database home page.

**Metrics**

You should not have any suspect databases found on any production server. You should immediately investigate any non-zero numbers for this statistic.
Troubleshooting
The steps to handling a suspect database will vary from one installation to another. However, there are some general guidelines you can use to troubleshoot a suspect database.

• Begin by examining the SQL Server error log for clues as to what caused the database to be marked as suspect.

• It is not unusual for a server to run out of physical disk space on drives used by SQL Server. When this happens, recovery for databases can sometimes fail with the end result being SQL Server marking a database as suspect. To remedy this situation, you should free up space on the identified drives or add files to the newly marked suspect database. For SQL Server 2000, this can be accomplished by utilizing the two procedures - sp_add_data_file_recover_suspect_db and sp_add_log_file_recover_suspect_db stored. For version 7.0 of SQL Server, you must use the sp_resetstatus stored procedure to reset the suspect status flag for the database in question, use the alter database command to add new datafiles to the database, and then stop/start the SQL Server.

• Many times, suspect databases are caused by SQL Server not being able to access a database or log file. This happens if a particular physical hard drive has become unavailable, but also can occur if another operating system process has obtained exclusive access to a file. If this scenario proves to be true, once you have ensured that the file(s) are available once again to the operating system, you can use the sp_resetstatus stored procedure to reset the suspect status flag for the database and then stop/start the SQL Server.

If none of these solutions are possible, you will likely have to restore your database using the last full and transaction log backups.

Offline Databases
The Offline Databases statistic represents the number of databases SQL Server has offline, meaning that no database modifications can occur.

   NOTE: This statistic is also available on the Database home page.

Metrics
You should not have any offline databases found on any production server. You should immediately investigate any non-zero numbers for this statistic.

Troubleshooting
Should an offline database be found by Performance Analyst, you can easily place it back online by utilizing either the sp_dboption stored procedure or the alter database command.

Errors in Current Log
SQL Server records various system events in its system or error log. The majority of messages placed into the log are informational in nature, however since some contain critical messages, you should immediately review them.

Performance Analyst indicates the number of actual error messages in the most recent error log so you know if there are potential events that require your attention.

Metrics
None.

Troubleshooting
Drill down into the Error Log Detail tab within Performance Analyst to view the actual error messages.
DB’s Low on Space
A SQL server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The Databases Low on Space statistic indicates databases that have fallen below a recommended percentage of free space.

**NOTE:** This statistic is also available on the Database home page.

**Metrics**
This might or might not be a problem. Some databases are not dynamic in nature (meaning they are not expected to grow in size) and are sized so very little free space is present. However, growing databases are another situation and might require DBA intervention if their free space amounts run low.

**Troubleshooting**
If the percent used amount of a database is approaching problematic levels, there are three ways a DBA can rectify the situation:

1. The DBA can resize the current file(s) used by the database via an ALTER DATABASE … MODIFY FILE command.
2. The DBA can add a new file to the database via the ALTER DATABASE … ADD FILE command.
3. The DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

Of course, the DBA should also ensure that enough physical space exists on the server to accommodate additional database space.

Logs Low on Space
Each database in SQL Server has a transaction log, which is a serial record of all modifications that have occurred in a database as well as the transactions that caused each change. The Logs Low on Space statistic indicates transaction logs that have fallen below a recommended percentage of free space.

**NOTE:** This statistic is also available on the Database home page.

**Metrics**
If any log’s used space exceeds the Performance Analyst recommended thresholds, then the DBA should take action to ensure that the log does not run out of available free space.

**Troubleshooting**
There are several things a DBA can do to ensure that a database’s log does not run out of available free space:

1. First, most transactional-oriented databases should have their logs assigned to a separate physical drive than the database. Reasons for doing so include:
   - It prevents competition for space between the log and the database itself.
   - It allows the log to be monitored for space more effectively.
   - It improves performance.
2. If the database is not critical in nature, you can set the truncate log on checkpoint option trunc log on chkpt, which will eliminate any non-active space in the log when a database checkpoint occurs.
3. Critical databases needing higher levels of recovery should have schedules established that regular perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.
4 If a critical transaction log becomes full, it might be impossible to use standard procedures to dump transactions and reclaim space. The dump operation will likely have to incorporate the no log or truncate only options.

5 If a transaction log continuously approaches dangerously low levels of free space, then the DBA should allow the underlying file(s) of the log to automatically grow to meet the demand. This can be accomplished by using the ALTER DATABASE ... MODIFY FILE ... FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

The DBA should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log since only non-active space in the transaction log is removed from log dumps or truncation operations.

SQL Analysis - Home

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured server down in terms of end-user response times.

Before you can identify problem SQL in your database, you have to ask the question - “What is bad SQL?” What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads</td>
<td>This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.</td>
</tr>
<tr>
<td>Writes</td>
<td>These can be caused by DML activity (INSERTs, etc.), but writes can also be indicative of heavy disk sort activity.</td>
</tr>
<tr>
<td>CPU</td>
<td>This is how much CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>This is how much time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server.</td>
</tr>
</tbody>
</table>

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

**NOTE:** SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

**Metrics**

When you begin to look for inefficient SQL in a database, there are two primary questions you should answer:

- What has been the worst SQL that has historically been run in my database?
• What is the worst SQL that is running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It might be that the query is in an inefficient Transaction SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can drill down further into the Top SQL view and begin working with the query through EXPLAIN plan analysis and other techniques inside Embarcadero’s DBArtisan.

Storage Analysis - Home

While DBAs focus on memory settings and tuning SQL, they frequently forget how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

• The ‘hit-the-wall’ variety that can bring things to a complete standstill.

• The ‘performance vampire’ kind that slowly drains the performance of a database over time.

Storage problems have the capability to bring the curtain down on a database very quickly, as in the case of a transaction log running out of free space. But storage problems can also silently work behind the scenes to slowly, but surely rob a database of its performance. For example, a hub table in a busy database might be accessed very quickly when an application is first given life, but over time, if it develops a heavy forwarded row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page displays the Total Used and Free Space of all tablespaces in the database. It also provides a count of the number of databases, files, and file groups that exist on the server. For more detailed information on database space, you can use the Performance Analyst Space home page or the Space Detail.

Total Used/Total Free Space

These statistics represent the total used and free space available in all databases on the server. Although good to know, a more detailed listing by database is needed to determine where any actual space shortages exist on the server. This information can be viewed in the Performance Analyst Space home page.

Metrics

If any one database or transaction log begins to approach the Performance Analyst recommended thresholds (and the objects contained within it are dynamic and growing as opposed to static), then the DBA should take action to prevent any future space allocation errors.

Troubleshooting

Here are a few things a DBA can do to prevent free space shortages:

For Databases:

• The DBA can resize the current file(s) used by the database via an ALTER DATABASE … MODIFY FILE command.

• The DBA can add a new file to the database via the ALTER DATABASE … ADD FILE command.

• The DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.
For Transaction Logs:

- First, most transactional-oriented databases should have their logs assigned to a separate physical drive than the database. Reasons for doing so include:
  - It prevents competition for space between the log and the database itself.
  - It allows the log to be monitored for space more effectively.
  - It improves performance.
- If the database is not critical in nature, you can set the truncate log on checkpoint option (trunc log on chkpt), which will eliminate any non-active space in the log when a database checkpoint occurs.
- Critical databases needing higher levels of recovery should have schedules established that regular perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.
- If a critical transaction log becomes full, it might be impossible to use standard procedures to dump transactions and reclaim space. The dump operation will likely have to incorporate the no log or truncate only options.
- If a transaction log continuously approaches dangerously low levels of free space, then the DBA should allow the underlying file(s) of the log to automatically grow to meet the demand. This can be accomplished by using the ALTER DATABASE ... MODIFY FILE ... FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

The DBA should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log since only non-active space in the transaction log is removed from log dumps or truncation operations.

Workload Analysis - Home

When your phone starts ringing with complaints of performance slowdowns, some of the first questions you should answer are:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Home page provides insight into the leading resource hogs of a server, as well as a count of total active and inactive processes. The Users Detail lets you easily get detailed information into what each leading session is currently doing.

Workload Analysis statistics include:

- **Active User Processes**
- **Inactive User Processes**

**Metrics**

If any one session appears to be using more than 50% of a total resource (CPU, memory, etc.), then you should drill down into the session to find out what they are currently executing.

**Active User Processes**

The Active User Processes statistic represents the total number of active and open threads reported on the server. This number displays the number of processes actively performing work.
Metrics
None.

Inactive User Processes
The Inactive Sessions statistic represents the total number of threads logged on to the server that are idle at the current time.

NOTE: This statistic is also available on the Users home page.

Metrics
A large number of inactive users could indicate user sessions that have mistakenly been left logged on. Since each user thread consumes a portion of memory on the server, to reduce resource usage, you should sever any sessions not needing a connection.

Troubleshooting
Drill down into the Session Waits tab and check sessions that have many seconds idle and/or that have been logged on for very long periods of time, as indicated by the logon time column. After verifying that a session is no longer necessary, you can KILL it.

Memory Page Statistics
The Memory page includes the following sections and statistics:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key Ratio Analysis</th>
<th>SQL Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Analysis</td>
<td>Workload Analysis - Top Memory Hogs</td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
Memory Detail
Home Page Statistics
Database Page Statistics
I/O Page Statistics
OS Page Statistics
Space Page Statistics
Users Page Statistics

Key Ratio Analysis - Memory
Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBA's to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.
While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the master.sysperfinfo performance table. This table maintains a count of various statistics for the server and particular databases since the server was brought up. For servers that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA might be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server’s individual ‘personality’ with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following memory ratios are used on the Performance Analyst Home page to succinctly communicate the general overall memory performance levels of the monitored database:

<table>
<thead>
<tr>
<th>Ad Hoc SQL Hit Ratio</th>
<th>Buffer Cache Hit Ratio</th>
<th>Cache Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Cache Hit Ratio</td>
<td>Procedure Cache Used</td>
<td>Procedure Hit Ratio</td>
</tr>
<tr>
<td>Used Cache %</td>
<td>Used Procedure Cache %</td>
<td></td>
</tr>
</tbody>
</table>

**Bottleneck Analysis - Memory**

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem.

Memory bottlenecks can definitely cause performance degradation in an otherwise well-running database. Typically, these bottlenecks center around the SQL Server latches, which are lightweight locks used to protect certain resources in memory. To help you identify such problems, the following statistics are presented on the Performance Analyst Memory home page:

<table>
<thead>
<tr>
<th>Latch Waits/Sec</th>
<th>Total Latch Wait Time</th>
<th>Top Latch Waits</th>
</tr>
</thead>
</table>

**SQL Analysis - Memory**

When SQL Server executes a set of SQL (ad hoc, procedure, trigger, etc.), it places the code into its procedure cache. The SQL Analysis section of the Performance Analyst Memory page provides a summary of the total memory pages currently in use by the various SQL types (procedure, ad hoc, etc.)
The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

**NOTE:** SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

### Memory Analysis

SQL Server uses various memory regions to accomplish various tasks (like retain frequently used data, procedures, etc.) and improve overall performance. At start up, SQL Server is allocated memory that will fall between the range of two configurable parameters: Min Memory and Max Memory. These statistics show you the amount of SQL Server memory allocated at present. In a normal SQL Server system, this number fluctuates as SQL Server uses more memory and releases unused memory back to the operating system. SQL Server also works in conjunction with the Windows Memory Manager, deallocating space when it detects that the operating system is unable to satisfy memory requests of other operating system processes. Depending on the version of SQL Server you are using, expect to deal with different memory regions. The table below describes some of the SQL Server memory regions:

<table>
<thead>
<tr>
<th>Memory Scan</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer Cache</td>
<td>The buffer cache is a memory pool of buffer pages into which SQL Server reads data pages for quick access.</td>
</tr>
<tr>
<td>Procedure Cache</td>
<td>Beginning with SQL Server version 7.0, you can place other SQL-related code objects in memory for reuse. Once SQL Server parses through and places a set of SQL of program code in memory, response time can be increased for subsequent calls to the same set of SQL or SQL code objects.</td>
</tr>
<tr>
<td>Connection Memory</td>
<td>The total amount of dynamic memory the server is using for maintaining connections.</td>
</tr>
<tr>
<td>Granted Workspace Memory</td>
<td>The total amount of memory granted to executing processes such as sort, hash, bulk copy operations, and so forth.</td>
</tr>
<tr>
<td>Lock Memory</td>
<td>The total amount of dynamic memory the server is using for locks.</td>
</tr>
<tr>
<td>Optimizer Memory</td>
<td>The total amount of dynamic memory the server is using for query optimization.</td>
</tr>
<tr>
<td>SQL Cache Memory</td>
<td>The total amount of memory used to retrieve stored SQL caches.</td>
</tr>
</tbody>
</table>

### Workload Analysis - Top Memory Hogs

It is not uncommon for one or two users to cause the majority of runtime problems that plague a server. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation.

If your database server does not have an overabundance of memory, then you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 25-50% of the total memory usage, then you should further investigate the sessions to see what activities they are performing.

### Buffer Cache Hit Ratio

- **Metrics**
- **Troubleshooting**
Data read from memory will produce end-user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the purposes of the SQL Server buffer/data cache.

The buffer cache hit ratio is a terrific indicator of how often user requests for data are satisfied through memory vs. being physically read from disk.

**Metrics**

To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, you should be aware that every server has its own ‘personality’ and might exhibit excellent performance with below average readings for the cache hit ratio. You should also be aware that excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

Consistently viewed low readings (60% or lower) might require tuning attention on the part of the DBA.

**Troubleshooting**

Ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server memory configuration option. Also, consider increasing your SQL Server Min. Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Often a user process is taking a large amount of memory due to an inordinate amount of I/O.

---

**Procedure Plan Hit Ratio**

- **Metrics**

  A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for code reference. Note that when a database is first started, the procedure cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

- **Troubleshooting**

  First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

---

**Ad Hoc SQL Hit Ratio**

- **Metrics**

  Check for any large objects that are pinned in memory that could possibly be removed.
• Troubleshooting

When an ad hoc SQL statement is issued, the query plan is then stored in the SQL Server procedure cache area. If the identical ad hoc statement is launched in the future, SQL Server uses the query plan already stored in the procedure cache if it is still there. This statistic defines the percentage of times that a query plan for an ad hoc SQL statement is found in the procedure cache.

Metrics

A high ad hoc hit rate is desirable, but is harder to maintain at a high level than something like a procedure cache hit rate. Therefore, an 80% or greater ad hoc cache hit rate is a good performance benchmark for code reference. Note that when a database is first started, the ad hoc cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the ad hoc cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

Troubleshooting

First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (To obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Log Cache Hit Ratio

• Metrics

• Troubleshooting

The Log Cache Hit Ratio represents the percentage of log cache reads satisfied from the log cache.

Metrics

None.

Troubleshooting

A low percentage on this statistic is not necessarily a bad sign, as it is possible that the information needed from the log will not be readily available in memory.

Used Cache %

• Metrics

• Troubleshooting

The Used Buffer Cache Percent statistic represents the total percentage of memory that is in use in the SQL Server buffer cache. Each instance of SQL Server has its own buffer cache where it stores recently used data pages to reduce physical I/O. The goal is to make the buffer cache large enough to maximize the ratio of logical reads to physical reads, but not so large that excessive memory swapping starts generating physical I/O to the pagefile. (Instances of SQL Server 2000 do this automatically under the default configuration settings.)

Metrics

A percentage used consistently remaining close to 100% indicates a deficient amount of memory available to SQL Server.
Troubleshooting
First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Used Procedure Cache %

- **Metrics**

The total percentage of memory that SQL Server is using in its SQL/procedure cache. Microsoft has begun to transition from the term “procedure cache” to “SQL cache” to define this area of memory. The reason being that in SQL Server's past, this area was devoted exclusively to holding query plans for stored procedures only.

The SQL Cache (procedure cache) is the part of the SQL Server memory pool that is used to store execution plans for Transact-SQL batches, stored procedures, and triggers. Execution plans record the steps that SQL Server must take to produce the results specified by the Transact-SQL statements contained in the batches, stored procedures, or triggers.

- **Metrics**

None.

Cache Used

- **Metrics**

- **Troubleshooting**

The total MB of space that SQL Server is using in the buffer cache. Each instance of SQL Server has its own buffer cache where it stores recently used data pages to reduce physical I/O. The goal is to make the buffer cache large enough to maximize the ratio of logical reads to physical reads, but not so large that excessive memory swapping starts generating physical I/O to the pagefile. (Instances of SQL Server 2000 do this automatically under the default configuration settings.)

- **Metrics**

A percentage used consistently remaining close to 100% indicates a deficient amount of memory available to SQL Server.

Troubleshooting
First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Procedure Cache Used

- **Metrics**

The total MB of space that SQL Server is using in its SQL/procedure cache. Microsoft has begun to transition from the term “procedure cache” to “SQL cache” to define this area of memory. The reason being that in SQL Server's past, this area was devoted exclusively to holding query plans for stored procedures only.
The SQL Cache (procedure cache) is the part of the SQL Server memory pool that is used to store execution plans for Transact-SQL batches, stored procedures, and triggers. Execution plans record the steps that SQL Server must take to produce the results specified by the Transact-SQL statements contained in the batches, stored procedures, or triggers.

**Metrics**
None.

### Top Latch Waits

**Metrics**

**Troubleshooting**

The Top Latch Waits graph depicts the top latch waits by wait time.

**Metrics**

Latch waits rarely impact performance, however seeing latch waits with high wait time might indicate an area that needs further investigation.

**Troubleshooting**

Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

[http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455)

### Latch Waits/Sec

**Metrics**

The Latch Waits statistic represents the number of latches per second that could not be satisfied immediately by SQL Server. Latches can be thought of as lightweight, mini-locks that are used to protect actions and resources used inside transactions.

**Metrics**

Unless accompanied by long wait times, latch waits should not pose too much of a performance problem in normal SQL Server installations.

If you have a high number of latch waits per second, then it is likely that SQL Server is also sporting a low buffer cache hit ratio and is being forced to perform extra physical I/O.

### Total Latch Wait Time

**Metrics**

Latches can be thought of as lightweight, mini-locks that are used to protect actions and resources used inside transactions. The total latch wait time statistic represents the sum total of all latch wait time that has occurred on the system.
Metrics
Unless accompanied by long wait times, latch waits should not pose too much of a performance problem in normal SQL Server installations.

If you have a high number of latch waits per second, then it is likely that SQL Server is also sporting a low buffer cache hit ratio and is being forced to perform extra physical I/O.

Memory Detail
The Memory Detail includes the following tabbed pages:

<table>
<thead>
<tr>
<th>Buffer Cache</th>
<th>Log Cache</th>
<th>Procedure Cache</th>
</tr>
</thead>
</table>

Buffer Cache Tab
The Buffer Cache Hit Ratio tab includes the following sections:

<table>
<thead>
<tr>
<th>Buffer Cache Hit Ratio</th>
<th>Page Activity</th>
<th>Total vs. Target Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Pages</td>
<td>Page Life Expectancy</td>
<td></td>
</tr>
</tbody>
</table>

Buffer Cache Hit Ratio
- Metrics
- Troubleshooting

Data read from memory will produce end-user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the purposes of the SQL Server buffer/data cache.

The buffer cache hit ratio is a terrific indicator of how often user requests for data are satisfied through memory vs. being physically read from disk.

Metrics
To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, you should be aware that every server has its own ‘personality’ and might exhibit excellent performance with below average readings for the cache hit ratio. You should also be aware that excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

Consistently viewed low readings (60% or lower) might require tuning attention on the part of the DBA.

Troubleshooting
Ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server memory configuration option. Also, consider increasing your SQL Server min memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Often a user process is taking a large amount of memory due to an inordinate amount of I/O. This information can be found in the Session Waits tab of the Users Detail.

Related Topics
Log Cache Tab
Procedure Cache Tab
Page Activity

Metrics

SQL Server uses various memory regions to accomplish various tasks (like retain frequently used data, procedures, etc.) and improve overall performance. At start up, SQL Server is allocated memory that falls between the range of two configurable parameters: Min Memory and Max Memory. The Page Activity chart shows you the amount of SQL Server memory allocated at present. In a normal SQL Server system, this number fluctuates as SQL Server uses more memory and releases unused memory back to the operating system. SQL Server also works in conjunction with the Windows Memory Manager, deallocating space when it detects that the operating system is unable to satisfy memory requests of other operating system processes. Depending on the version of SQL Server you are using, there are several memory regions you should expect to deal with. The table below describes the SQL Server memory regions:

<table>
<thead>
<tr>
<th>Memory Scan</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Buffer Cache</td>
<td>The buffer cache is a memory pool of buffer pages into which SQL Server reads data pages for quick access.</td>
</tr>
<tr>
<td>Procedure/SQL Cache</td>
<td>Beginning with SQL Server version 7.0, you can place other SQL-related code objects in memory for reuse. Once SQL Server parses through and places a set of SQL of program code in memory, response time can be increased for subsequent calls to the same set of SQL or SQL code objects.</td>
</tr>
<tr>
<td>Connection Memory</td>
<td>The total amount of dynamic memory the server is using for maintaining connections.</td>
</tr>
<tr>
<td>Lock Memory</td>
<td>The total amount of dynamic memory the server is using for locks.</td>
</tr>
<tr>
<td>Optimizer Memory</td>
<td>The total amount of dynamic memory the server is using for query optimization.</td>
</tr>
</tbody>
</table>

The Active and Free Page chart in Performance Analyst displays a count of active memory pages (those being used by SQL Server) and free pages (pages not being used).

Metrics

Seeing consistent high volumes of free pages might indicate that SQL Server is using more memory than necessary.

Total vs. Target Memory

Metrics

Total memory is the amount of memory that's currently in use by SQL Server. Target memory is the amount of memory that could be used. SQL Server dynamically adjusts its memory allocations depending either on workload up to the maximum memory configuration set by the DBA or the amount it thinks it should give other (non-SQL) server operations.

Metrics

When total memory approaches or reaches the maximum amount of target (defined) memory, then SQL server is experiencing memory pressure. You can try adding more RAM to the machine or adjusting memory ceiling limits imposed at the SQL server configuration level.

Database Pages

Metrics

The database pages metric gives the amount of pages in memory devoted to database content compared to all other content.
**Metrics**

If you see that a large percentage of memory is devoted to database pages, the server is database page centric and not procedure plan centric. This can indicate that too many table scans are taking place, which might be flooding the buffer cache with infrequently used pages. You can determine if this is the case by looking at the page life expectancy metric. Page life expectancy readings of 300 seconds or less can indicate excessive table scan activity.

**Page Life Expectancy**

**Metrics**

The page life expectancy metric indicates how long SQL server believes a page will stay in the buffer cache. Pages served from memory result in much shorter response times than pages read from disk into the cache. It's best if frequently used data is pinned in the buffer cache.

**Metrics**

Readings of 300 seconds or less frequently mean that too many table scans are occurring and flooding the buffer cache with pages that are used rarely. Check the I/O access patterns on the I/O home page or the User I/O drill-down to confirm if this is the case.

**Procedure Cache Tab**

The Procedure Cache tab includes the following sections:

<table>
<thead>
<tr>
<th>Procedure Cache Hit Ratio</th>
<th>Procedure/SQL Cache Details</th>
<th>Procedure/SQL Cache Hit Summary</th>
</tr>
</thead>
</table>

**Related Topics**

Buffer Cache Hit Ratio Tab

Log Cache Tab

**Procedure Cache Hit Ratio**

**Metrics**

The SQL Server procedure cache holds the execution plans for all Transact-SQL statements currently executing in the server. When a user executes a Transact-SQL statement, SQL Server looks in the procedure cache for a query plan to use.

The Procedure Cache Hit Ratio statistic is the percentage of query plan requests generated by stored procedures that are found in the procedure cache area. The greater the percentage of times that a statement's plan and definition can be referenced in memory, the better the procedure execution time.

**Metrics**

A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for code reference. Note that when a database first starts, the procedure cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate does not increase to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.
Troubleshooting
First, ensure SQL Server is configured to use as much physical memory as possible by checking the max server memory configuration option. Also, consider increasing your SQL Server min memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Procedure/SQL Cache Summary

Metrics
The SQL Server procedure/SQL cache holds often-referenced code object definitions and ad-hoc SQL. The Procedure/SQL Cache Summary lists each type of code object, the amount of memory each code type (trigger, ad-hoc SQL, etc.) consumes, and a total count of each type of code object.

Metrics
Seeing a lot of memory devoted to ad-hoc SQL plans might indicate an environment where sessions are submitting a lot of SQL requests. Many SQL Server DBAs like to control code submissions through stored procedures.

Procedure/SQL Cache Detail

Metrics
Beginning with SQL Server version 7.0, other SQL-related code objects might be placed in memory for reuse. Once SQL Server parses through and places a set of SQL of program code in memory, response time can be increased for subsequent calls to the same set of SQL or SQL code object. The Procedure/SQL Cache Details section displays the top 1000 objects in the procedure/SQL cache. The table below describes the information available in the Procedure/SQL Cache Details section on the Procedure Cache tab of the Memory Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Type</td>
<td>The type of object (ad hoc SQL, etc.)</td>
</tr>
<tr>
<td>Cache Type</td>
<td>The category of SQL code inside the cache (executable plan, etc.)</td>
</tr>
<tr>
<td>Object ID</td>
<td>The numeric ID of the object.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the object (if applicable).</td>
</tr>
<tr>
<td>Database ID</td>
<td>The numeric ID of the database.</td>
</tr>
<tr>
<td>Database</td>
<td>The database where the code originated.</td>
</tr>
<tr>
<td>User ID</td>
<td>The numeric ID of the user.</td>
</tr>
<tr>
<td>User</td>
<td>The owner of the object or code.</td>
</tr>
<tr>
<td>Use Count</td>
<td>The number of uses for the object.</td>
</tr>
<tr>
<td>SQL Bytes</td>
<td>The amount of SQL bytes used by the code object.</td>
</tr>
<tr>
<td>Size (KB)</td>
<td>The size used by the object in kilobytes.</td>
</tr>
<tr>
<td>SQL</td>
<td>The actual SQL statement or code being executed.</td>
</tr>
</tbody>
</table>

Metrics
Seeing many objects of the ad hoc SQL type might indicate an environment where sessions are submitting a lot of SQL requests. Many SQL Server DBAs like to control code submissions through stored procedures.
Log Cache Tab
The Log Cache tab includes the following sections:

<table>
<thead>
<tr>
<th>Database Log Cache Ratios</th>
<th>Log Cache Details</th>
<th>Log Cache Hit Ratio</th>
</tr>
</thead>
</table>

Related Topics
Buffer Cache Hit Ratio Tab
Procedure Cache Tab

Log Cache Hit Ratio
- **Metrics**
- **Troubleshooting**

The Log Cache Hit Ratio statistic represents the percentage of log cache reads satisfied from the log cache.

**Metrics**
None.

**Troubleshooting**
A low percentage on this statistic is not necessarily a bad sign, as it is possible that the information needed from the log will not be readily available in memory.

Database Log Cache Ratios
- **Metrics**
- **Troubleshooting**

The Log Cache Hit Ratio statistic represents the percentage of log cache reads satisfied from the log cache. The database log cache ratios display breaks down log cache hit ratios by database. Note that some databases with inactive logs will not display a log cache hit ratio.

**Metrics**
None.

**Troubleshooting**
A low percentage on this statistic is not necessarily a bad sign, as it is possible that the information needed from the log will not be readily available in memory.

Log Cache Details
- **Metrics**
- **Troubleshooting**
Before ever writing transactions to disk, the log manager of SQL Server formats everything in memory. This area of memory is known as the log cache. The log writer of SQL Server moves through the log caches when transactions are committed (as well as other events) and flushes each cache out to disk. SQL Server also reads from the cache when log records are needed. The table below describes the information available in the Log Cache Details section on the Log Cache tab of the Memory Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The name of the SQL Server database.</td>
</tr>
<tr>
<td>Log Cache Reads</td>
<td>The number of reads from the log cache.</td>
</tr>
<tr>
<td>Log Flushes</td>
<td>The number of times data was flushed from the log to disk.</td>
</tr>
<tr>
<td>Log Flush Waits</td>
<td>The number of times the log had to wait before flushing data to disk.</td>
</tr>
<tr>
<td>Log Flush Wait Time</td>
<td>The amount of time the log waited before flushing data to disk in milliseconds.</td>
</tr>
<tr>
<td>Log Growths</td>
<td>The number of times the log had to physically grow in size to meet the need for more space.</td>
</tr>
<tr>
<td>Log Shrinks</td>
<td>The number of times the log contracted in physical size.</td>
</tr>
</tbody>
</table>

**Metrics**

Seeing high amounts of wait time for log flushes could indicate a bottleneck at the disk level. A log that shows high number of growths likely indicates an undersized log. While automatic growth can alleviate out-of-space conditions, many growth operations can slow down overall operations. It is better to have a properly sized transaction log that allows SQL Server to continually enlarge it in size when needed.

**Troubleshooting**

Consider relocating logs showing high amounts of wait time to faster disks. For logs showing high numbers of growths, permanently enlarge the log(s) via an ALTER DATABASE command that will resize the log files.

**I/O Page Statistics**

The I/O page includes the following sections and statistics:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>I/O Access Pattern</th>
<th>Key System Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Analysis</td>
<td>Workload Analysis - Top I/O Hogs</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- I/O Detail
- Home Page Statistics
- Database Page Statistics
- Memory Page Statistics
- OS Page Statistics
- Space Page Statistics
- Users Page Statistics
Key System Analysis - I/O

When complaints begin to surface about your database's performance, oftentimes the root cause can be traced to one or more issues with I/O. The thing to keep in mind when you begin to monitor the I/O of your database is that you are actually reviewing the success of your physical design model.

All the physical storage characteristics and placements, the table and index designs, and the speed that it all works are on display when I/O is monitored. Because a database's main index of performance is measured by how fast I/O needs are satisfied, it is your responsibility to quickly interrogate SQL Server to determine if a reported database slowdown is I/O related.

The following key I/O values are reported in this section:

<table>
<thead>
<tr>
<th>Checkpoint Pages</th>
<th>Lazy Writes</th>
<th>Log Cache Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Flushes</td>
<td>Page Reads</td>
<td>Page Writes</td>
</tr>
<tr>
<td>Read Ahead Pages</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bottleneck Analysis - I/O

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. The Performance Analyst I/O home page identifies the top I/O related waits that might be currently a cause of lessened performance. Also included are the current hottest database/log files and a summary of I/O activity.

Bottleneck Analysis includes the following information:

<table>
<thead>
<tr>
<th>I/O Busy</th>
<th>I/O Errors</th>
<th>Hottest Database/Log Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total I/O</td>
<td>Top I/O Waits</td>
<td></td>
</tr>
</tbody>
</table>

SQL Analysis - I/O

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured server down in terms of end-user response times.

Before you can identify problem SQL in your database, you have to ask the question - "What is bad SQL?" What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads (Physical)</td>
<td>This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user’s request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements’ logical I/O as well.</td>
</tr>
</tbody>
</table>
There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian
joins and the like), but more often than not, these measures are reflected in the criteria listed above.

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed
Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you
are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see
that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90%
of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the
Top SQL details views.

**NOTE:** SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL
traffic. If you do not want to collect SQL execution information, use the options inside of
Performance Analyst to disable SQL profiling.

### Metrics
When you begin to look for inefficient SQL in a database, there are two primary questions you should answer:

- What has been the worst SQL that has historically been run in my database?
- What is the worst SQL that is running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that
is significantly larger than any other query on the system. It might be that the query is in an inefficient Transaction SQL
loop, or other problematic programming construct. Only by bringing the query to the attention of the application
developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can drill down further into
the Performance Analyst Top SQL view and begin working with the query through EXPLAIN plan analysis and other
techniques inside Embarcadero’s DBArtisan.

### I/O Access Patterns

<table>
<thead>
<tr>
<th>Access Style</th>
<th>Description</th>
</tr>
</thead>
</table>
| Full Scans   | Full Scans are the total number of full table or index scans. Full scans occur if a table is
|              | inadequately indexed or if SQL Server truly needs to access all rows in a table or index to
|              | satisfy a query. Other operations that can cause full scans include UPDATE STATISTICS
|              | calls. Unnecessary scans on large tables is something to avoid, and can be a signal to you as
|              | a DBA to investigate the use of more indexes and to review SQL access through EXPLAIN
|              | plans. Small table scans are actually a good thing because SQL Server can often cache the
|              | entire table in a single I/O operation. Large numbers of index scans are normally desirable too,
|              | since it typically indicates the fastest possible resolution to data access requests. |

Depending on the version of SQL Server you are using, expect to deal with different access styles. The table below
describes some of the SQL Server access styles:
Troubleshooting
Actions to take after observing the ways SQL Server is accessing data depend on the actual situation you witness.

If consistent numbers are present for Forward Record Fetches, you might want to examine your databases to see which tables have forwarded records. You can easily do this by utilizing the Embarcadero Space Analyst to view forwarded record data for tables.

If you do not have Space Analyst, then to see the total count of forwarded records in a table, you can enable trace flag 2509, and then execute the DBCC CHECKTABLE command. The output should display the number of forwarded records in that table. Tables with many forwarded records could be candidates for table reorganization.

Here are some methods you can use to avoid unnecessary full scans:

- Try not to use SQL statements that include the NOT IN, NOT LIKE, <>, IS NULL operators since they typically suppress the use of indexes.
- When referencing concatenated indexes with queries, be sure the leading column in the index is used. If it is not, the index will not be used at all.
- Avoid using functions in WHERE predicates.

Workload Analysis - Top I/O Hogs

- Metrics

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. The Top I/O Hogs display showcases the top sessions that have caused the most physical I/O on the system (both reads and writes).

Metrics

If you see one or two users who have caused more than 25-50% of total I/O, then you should drill down further into the sessions to see what activities they are performing.
Page Reads

• **Metrics**

  The Page Reads statistic represents that number of physical database page reads that are issued per second by SQL Server.

**Metrics**

Page reads are to be expected, especially after initial server start up. This is because SQL Server must first satisfy requests for data and meta-data by reading information in from physical disk. Numerous page reads can also be expected if the physical server does not contain an adequate amount of memory to hold repetitively requested blocks of information.

No hard-and-fast rules exist for how many page reads per second is too much. You can cross-reference this statistic with the physical server disk statistics to see if physical page reads and accompanying physical disk I/O is approaching the server’s premium capacity levels. And because logical I/O is always many times faster than physical I/O, you should also evaluate the buffer cache hit ratio to determine overall memory vs. physical read efficiency.

**Troubleshooting**

If you find that the server is becoming overworked from a physical I/O standpoint, there are several courses of action you can take:

• Examine index usage to ensure that unnecessary table scans are not occurring.

• Check the physical database design to see if table objects have been over-normalized.

• Ensure that SQL Server is configured to use sufficient amounts of memory. Examine the min server memory (MB) and max server memory (MB) parameters to see if SQL Server is constrained on either end of the memory spectrum.

• Check for large pinned table objects that could be using excessive amounts of space in the buffer cache.

• Last, but not least, investigate the possibility of adding more RAM to the physical server.

Page Writes

• **Metrics**

  The Page Writes statistic represents the number of physical database page writes issued by SQL Server. Page Writes take place during operations such as checkpoints, lazywriter writes, index creations, and BCP routines.

**Metrics**

Page Writes can give you an idea of overall physical write activity, however there are a number of statistics that pertain specifically to certain write activities like checkpoints, etc., that you can examine to determine the amount of physical writes caused by distinct SQL Server processes.

With respect to performance, response times experienced by SQL Server users are normally not impacted by write operations unless the writes are synchronous in nature. These are typically BCPs, database recovery operations, and index creations.

Read Ahead Pages

• **Metrics**
The Read Ahead Pages statistic represents the number of physical database page read in anticipation of use by SQL Server. If SQL Server senses that database pages are being read in a sequential manner, it will institute a pre-fetch mechanism that moves pages into the buffer cache before they are actually needed by a process.

**Metrics**
If data is accessed sequentially (for example, through the use of a clustered index), the read ahead mechanism of SQL Server can increase performance by needed database pages already in the buffer cache before they are actually requested.

However, because the read ahead mechanism is typically triggered by full table or index range scans, if the read ahead pages are actually required to satisfy a user’s query, then performance might actually suffer. In these cases, the judicious use of indexes is a better route to take.

**Log Flushes**

- **Metrics**
- **Troubleshooting**

The Log Flushes statistic represents the total number of log pages for all databases written to disk by the log writer process. A log flush occurs when SQL Server writes all changes from the database’s log cache out to the database’s log files on disk.

**Metrics**
Increasing numbers observed for log flushes should not cause concern unless the I/O subsystem of the server appears overwhelmed. In addition, to minimize I/O contention between a database and its accompanying log, it is wise to place database files and log files on separate disks.

**Troubleshooting**
If you have placed a very active database on the same physical file as its log, you can look into moving the log to a separate physical device by adding new log files to a new drive and subsequently removing the old log files when they are not being used.

**Log Cache Reads**

- **Metrics**

The Log Cache Reads statistic represents the reads performed per second through the log manager cache. Before ever writing transactions to disk, the log manager of SQL Server formats them in memory. This area of memory is known as the log cache and only contains log records for SQL Server 2000 and later. The log writer of SQL Server moves through the log caches when transactions are committed (as well as other events) and flushes each cache out to disk.

**Metrics**
None.

**Checkpoint Pages**

- **Metrics**
- **Troubleshooting**
The Checkpoint Pages statistic represents the number of pages flushed to disk per second by a checkpoint or other operation that require all dirty (modified) pages to be flushed.

**Metrics**
Checkpoint operations are used by SQL Server to minimize the amount of work the server must perform when databases are recovered during system startup. Checkpoints periodically write out modified pages that are found in the buffer cache to disk. Afterwards, SQL Server records the operation in the log to signify that the operation succeeded.

Checkpoints can be explicitly performed by a database owner issuing the checkpoint command. SQL Server also performs checkpoints automatically for databases that have the trunc log on chkpt option set.

Large SQL Servers have the potential to generate lots of checkpoint write operations. Although SQL Server will do what it can to minimize checkpoint activity, you can also set the recovery interval server parameter to influence how often checkpoints should run.

**Troubleshooting**
If you believe excessive checkpoint activity is occurring, you can take the following steps:

- Set the recovery interval server parameter to a larger value with `sp_configure`.
- Restart SQL Server so the change will take effect or use the RECONFIGURE option to make the change immediately.

**Lazy Writes**
The lazy writer is a SQL Server system process that flushes out batches of dirty and aged buffers (buffers that contain changes that must be written back to disk before the buffer can be reused for a different page) and make them available to user processes. The Lazy Writes statistic represents the number of buffers written by the buffer manager's lazy writer process.

**Metrics**
High values might indicate a thrashing situation with data scanned into the buffer cache (but referenced seldom) and then immediately moved back out.

**Total I/O**
The Total I/O statistic represents the total number of physical reads and writes.

**I/O Busy**
The I/O Busy statistic represents the time in milliseconds that SQL Server has spent performing input and output operations since the last refresh.

**NOTE:** This statistic is also available on the Database home page.

**Metrics**
None.
I/O Errors

- **Metrics**
- **Troubleshooting**

I/O Error Rate reflects total number of I/O errors (errors during read and write operations) encountered by the server since the last refresh inside Performance Analyst Center. I/O Error Rate is a percentage based on Total I/O (the sum of the physical reads and writes).

**Metrics**
You should observe few, if any errors.

**Troubleshooting**
If you notice any errors, you should check the SQL Server error log for details.

Top I/O Waits

- **Metrics**
- **Troubleshooting**

The Top I/O Waits statistics rank the top I/O related waits by wait time.

**Metrics**
None.

**Troubleshooting**
Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

http://support.microsoft.com/default.aspx?scid=kb:en-us;Q244455

Hottest Database/Log Files

- **Metrics**
- **Troubleshooting**

The Hottest Database/Log Files display presents the most active database files as indicated by physical I/O activity (reads and writes).

**Metrics**
Understanding the storage-level hot spots of a database is beneficial for a couple of reasons.

1. First, you can get a feel for overworked physical disks by viewing I/O statistics at the tablespace and datafile levels. If a particular disk or set of disks is under too much strain, you attempt to relocate the databases to other less-used devices or create new file groups on different disks and move hot objects to them (assuming, of course, you have extra disks available).

2. Second, if you have used standard DBA practice and placed indexes in their own file group, you can view the I/O statistics for that file group and see if the indexes are actually being used.
Troubleshooting
Some areas to consider when viewing the hottest database files display are:

- Seeing a lot of activity in the master database and datafiles might indicate a lot of recursive calls.
- The tempdb database (devoted to sort activity) shows higher volumes of physical I/O which could indicate a problem with excessive disk sorts.
- You might want to quickly review all the physical I/O for each drive/file system and get a idea of the overworked disks on your server. If you have under-utilized disk drives (with their own controllers), you should consider relocating some databases or filegroups that exhibit high I/O characteristics to those drives.
- To minimize contention at the physical disk level, and improve performance overall, it is generally recommended that a SQL Server database have its database and log files physically separated onto different drives.

I/O Detail
The I/O Detail includes the following tabbed pages:

<table>
<thead>
<tr>
<th>Database I/O</th>
<th>File</th>
<th>System I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>User I/O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

System I/O Tab
The System I/O tab of the I/O Detail includes the following sections:

<table>
<thead>
<tr>
<th>AWE I/O</th>
<th>I/O Activity</th>
<th>Physical I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space I/O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
- Database I/O Tab
- File Tab
- System I/O Tab
- User I/O Tab

Physical I/O
- Metrics
- Troubleshooting

SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The System I/O tab of the I/O Detail displays statistics that track various system-related I/O operations. The Physical I/O section details I/O tasks that involve physical disk activity. The table below describes the information available in the Physical I/O section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Reads</td>
<td>The number of physical database page reads that are issued per second by SQL Server. It is collective in nature, meaning that it represents the total page reads per second across all databases that exist on the target SQL Server.</td>
</tr>
</tbody>
</table>
Metrics

Page reads are to be expected, especially after initial server start up. This is because SQL Server must first satisfy requests for data and metadata by reading information in from physical disk. Numerous page reads can also be expected if the physical server does not contain an adequate amount of memory to hold repetitively requested blocks of information.

No hard and fast rules exist for how many page reads per second is too much. You can cross reference this statistic with the physical server disk statistics to see if physical page reads and accompanying physical disk I/O is approaching the server's premium capacity levels. And because logical I/O is always many times faster than physical I/O, you should also evaluate the buffer cache hit ratio to determine overall memory vs. physical read efficiency.

Page Writes can give you an idea of overall physical write activity. There are a number of statistics, however, that pertain specifically to certain write activities like checkpoints, that can be examined to determine the amount of physical writes caused by distinct SQL Server processes. With respect to performance, response times experienced by SQL Server users are normally not impacted by write operations unless the writes are synchronous in nature. These are typically BCPs, database recovery operations, and index creations.

With respect to read ahead pages, if data is accessed sequentially (for example, through the use of a clustered index), the read ahead mechanism of SQL Server can increase performance by needed database pages already in the buffer cache before they are actually requested. However, because the read ahead mechanism is typically triggered by full table or index range scans, if the read ahead pages are actually required to satisfy a user's query, performance can actually suffer. In these cases, the judicious use of indexes is a better route to take.

Checkpoint operations are used by SQL Server to minimize the amount of work the server must perform when databases are recovered during system startup. Checkpoints periodically write out modified pages that are found in the buffer cache to disk. Afterward, SQL Server records the operation in the log to signify that the operation succeeded. A database owner issuing the checkpoint command can explicitly perform checkpoints. SQL Server also performs checkpoints automatically for databases that have the trunc log on chkpt option set. Large SQL Servers have the potential to generate lots of checkpoint write operations. Although SQL Server minimizes checkpoint activity, you can also set the recovery interval server parameter to influence how often checkpoints can run.

Troubleshooting

If you find that the server is becoming overworked from a physical I/O standpoint, here are a few things you can do:

- Examine index usage to ensure that unnecessary table scans are not occurring.
- Check the physical database design to see if table objects have been over-normalized.
- Ensure that SQL Server is configured to use sufficient amounts of memory. Examine the min server memory (MB) and max server memory (MB) parameters to see if SQL Server is constrained on either end of the memory spectrum.

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Writes</td>
<td>The number of physical database page writes issued by SQL Server. Page Writes take place during operations such as checkpoints, lazy writer writes, index creations, and BCP routines.</td>
</tr>
<tr>
<td>Checkpoint Pages</td>
<td>The number of pages flushed to disk per second by a checkpoint or other operation that require all dirty (modified) pages to be flushed.</td>
</tr>
<tr>
<td>Lazy Writes</td>
<td>The number of buffers written by the buffer manager's lazy writer. The lazy writer is a system process that flushes out batches of dirty and aged buffers and makes them available to SQL Server processes. The lazy writer eliminates the need to perform frequent checkpoints in order to create available buffers.</td>
</tr>
<tr>
<td>Read Ahead Pages</td>
<td>The number of physical database pages read in anticipation of use by SQL Server. If SQL Server senses that database pages are being read in a sequential manner, it institutes a pre-fetch mechanism that moves pages into the buffer cache before they are actually needed by a process.</td>
</tr>
<tr>
<td>Page Lookups</td>
<td>The number of requests to find a page in the buffer pool.</td>
</tr>
</tbody>
</table>
• Check for large pinned table objects that could be using excessive amounts of space in the buffer cache.

• Last, but not least, investigate the possibility of adding more RAM to the physical server.

If you believe excessive checkpoint activity is occurring, you can take the following steps:

• Set the recovery interval server parameter to a larger value with `sp_configure`.

• Restart SQL Server or use the `RECONFIGURE` option to make the change immediately.

**Space I/O**

- **Metrics**
- **Troubleshooting**

SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The Space I/O section of the I/O Detail details space-related I/O operations. The table below describes the information available in the Space I/O section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extents Allocated</td>
<td>The number of space extents that SQL Server allocated. Rapidly increasing numbers for these statistics indicates that SQL Server is receiving large volumes of incoming data and is allocating space to make room.</td>
</tr>
<tr>
<td>Extent Deallocations</td>
<td>This indicates that SQL Server is reclaiming space from database objects due to shrinking database volumes.</td>
</tr>
<tr>
<td>Freespace Page Fetches</td>
<td>The number of pages by free space scans used to satisfy requests to insert record fragments.</td>
</tr>
<tr>
<td>Freespace Scans</td>
<td>The number of scans performed by SQL Server to locate free space for an incoming record.</td>
</tr>
<tr>
<td>Mixed Page Allocations</td>
<td>The number of pages allocated from mixed extents. These are used for storing the first eight pages that are allocated to an index or table.</td>
</tr>
<tr>
<td>Page Deallocations</td>
<td>This indicates that SQL Server is reclaiming space from database objects due to shrinking database volumes.</td>
</tr>
<tr>
<td>Page Splits</td>
<td>When data is inserted or updated in a table, SQL Server might reorganize the storage of the data in the table's index pages. When an index page becomes full, but a DML operation demands room on that page, SQL Server moves about half the rows to a new page to accommodate the request. This reorganization is known as a page split. Performance for DML actions can be impaired from page split operations. In addition, more index pages can make for longer index scan times.</td>
</tr>
</tbody>
</table>

**Metrics**

Increasing numbers for extent and page allocation, and freespace operations likely indicates aggressive volumes of data being inserted or modified in SQL Server.

Page splits cause additional overhead in the form of CPU usage and I/O. Observing large numbers of page splits can signal a resource bottleneck in your server.

**Troubleshooting**

To avoid page splits, you can look into tuning the FILLFACTOR property of an index, which controls the percentage of the index page that is filled during creation. The default, 100, tells SQL Server to completely fill each page, whereas lower numbers tell SQL Server to leave room for additional index rows or updates to existing rows.
I/O Activity
SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The I/O Activity section of the I/O Detail details maintenance-related I/O operations. The table below describes the information available in the I/O Activity section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCC Logical Scans</td>
<td>The number of logical read scan bytes per second caused by DBCC operations.</td>
</tr>
<tr>
<td>Bulk Copy Rows</td>
<td>The number of rows copied either into or out of the database via the BCP utility.</td>
</tr>
<tr>
<td>Bulk Copy Throughput</td>
<td>The amount of data (in KB) copied via BCP operations.</td>
</tr>
<tr>
<td>Transactions</td>
<td>The number of transactions that have occurred within the database.</td>
</tr>
<tr>
<td>Log Flushes</td>
<td>The number of log flushes for the server.</td>
</tr>
<tr>
<td>Backup/Restore T-Put</td>
<td>Defines the read/write throughput for backup and restore operations.</td>
</tr>
</tbody>
</table>

AWE I/O
SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The AWE I/O section of the I/O Detail details I/O relating to Address Windowing Extensions (the SQL Server mechanism for supporting very large amounts of RAM). The table below describes the information available in the AWE I/O section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Maps</td>
<td>The number of times that a database page was requested by the server, found in the buffer pool, and mapped. When it is mapped, it is made a part of the server's virtual address space.</td>
</tr>
<tr>
<td>Stolen Maps</td>
<td>The number of times that a buffer was taken from the free list and mapped.</td>
</tr>
<tr>
<td>Unmap Cells</td>
<td>The number of calls to unmap buffers. When a buffer is unmapped, it is excluded from the virtual server address space. One or more buffers might be unmapped on each call.</td>
</tr>
<tr>
<td>Unmap Pages</td>
<td>The number of SQL Server buffers that are unmapped.</td>
</tr>
<tr>
<td>Write Maps</td>
<td>The number of times that it is necessary to map in a dirty buffer so it can be written to disk.</td>
</tr>
</tbody>
</table>

User I/O Tab
The User I/O tab of the I/O Detail includes the following sections:

<table>
<thead>
<tr>
<th>Lock Activity</th>
<th>SQL Activity</th>
<th>Temporary Object Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Object Activity</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
Database I/O Tab
File Tab
System I/O Tab

User Object Activity

- Metrics
- Troubleshooting
The User Object Activity section of the I/O Detail displays statistics that track various user-related I/O operations. The I/O function along with its counter value is presented. The User Object Activity section details performance statistics that reflect how SQL Server is performing object access operations. The table below describes the information available in the User Object Activity section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwarded Records</td>
<td>The number of records per second fetched through forwarded record pointers. At times forwarded records can reduce performance because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself is read.</td>
</tr>
<tr>
<td>Full Scans</td>
<td>Full scans of moderately sized indexes or tables are generally okay. SQL Server can scan and cache a small table much faster than using its index to navigate to any requested data. Full, unrestricted, large table scans, however, are typically not good and degrade overall system performance and response time.</td>
</tr>
<tr>
<td>Index Searches</td>
<td>The total number of index searches per second. Index searches are normally used to start range scans, for single index record fetches and can be used to reposition an index.</td>
</tr>
<tr>
<td>Probe Scans</td>
<td>The total number of probe scans per second. Probe scans are used in SQL Server to directly find rows in an index or base table.</td>
</tr>
<tr>
<td>Range Scans</td>
<td>The total number of qualified range scans through indexes per second.</td>
</tr>
<tr>
<td>Skipped Ghosted Records</td>
<td>The number of ghosted records per second skipped during scans.</td>
</tr>
</tbody>
</table>

**Metrics**

Full scans occur if a table is inadequately indexed or if SQL Server truly needs to access all rows in a table or index to satisfy a query. UPDATE STATISTICS calls can also cause full scans.

Unnecessary scans on large tables is something to avoid, and can be a signal to you as a DBA to investigate the use of more indexes and to review SQL access through EXPLAIN plans. Small table scans are actually a good thing because SQL Server can often cache the entire table in a single I/O operation. Large numbers of index scans are normally desirable too, because this typically indicates the fastest possible resolution to data access requests.

When SQL Server creates a forwarding pointer, it remains in place unless one of two things happens. The first is when a row shrinks enough to move back to its original location. The second is when the entire database shrinks. When a database file shrinks, SQL Server reassigns the row identifiers, which are used as the row locators, so the shrink process never generates forwarded rows. Forwarded records can reduce performance at times because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself is read.

Large numbers of index searches and probe scans are normally desirable because they typically indicate the fastest possible resolution to data access requests is being taken.

**Troubleshooting**

Here are some methods you can use to avoid unnecessary large table scans:

- Try not to use SQL statements that include the NOT IN, NOT LIKE, <>, IS NULL operators because they typically suppress the use of indexes.
- When referencing concatenated indexes with queries, be sure the leading column in the index is used. If it is not, the index will not be used at all.
- Avoid using functions in WHERE predicates.

If consistent numbers are present for Forward Record Fetches, examine your databases to see which tables have forwarded records. This can easily be done with the Embarcadero Space Analyst component. If you do not have Space Analyst, then to see the total count of forwarded records in a table, enable trace flag 2509, and then execute the DBCC CHECKTABLE command. The output should display the number of forwarded records in that table. Tables with many forwarded records could be candidates for table reorganization.
SQL Activity

- Metrics
- Troubleshooting

The SQL Activity section of the User I/O tab of the I/O Detail displays statistics that track various user-related I/O operations. The I/O function along with its counter value is presented. The SQL Activity section details performance statistics that reflect SQL I/O-related operations. The table below describes the information available in the SQL Activity section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-Param Attempts</td>
<td>Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Auto-param Attempts statistic shows the number of auto-parameterization attempts per second and includes failed, safe, and unsafe auto-parameterizations.</td>
</tr>
<tr>
<td>Batch Requests</td>
<td>A batch is a collection of one or more SQL statements sent in one unit by the client. Each batch is compiled into a single execution plan. If the batch contains multiple SQL statements, all of the optimized steps needed to perform all the statements are built into a single execution plan. The Batch Requests statistic shows the number of batch requests per second that are processed by the SQL Server instance.</td>
</tr>
<tr>
<td>Failed Auto-Params</td>
<td>Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Failed Auto-params statistic shows the number of failed auto-parameterization attempts per second.</td>
</tr>
<tr>
<td>Safe Auto-Params</td>
<td>Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Safe auto-params statistic shows the number of auto-parameterization attempts per second and includes only safe auto-parameterizations (ones where the cached plan can be shared).</td>
</tr>
<tr>
<td>Scan Point Revalidations</td>
<td>The number of times per second that the scan point had to be revalidated to continue the scan.</td>
</tr>
<tr>
<td>SQL Compilations</td>
<td>The number of SQL compilations performed, indicating the number of times the compile code path is entered. This also includes compiles due to recompiles. When SQL Server user activity levels become stable, this value reaches a steady state.</td>
</tr>
<tr>
<td>SQL Re-Compilations</td>
<td>The total number of recompiles triggered per second in a SQL Server instance. Recompiles occur when SQL Server determines that the currently defined execution plan for an executing stored procedure might no longer be the best possible plan. SQL Server pauses the query execution and recompiles the stored procedure.</td>
</tr>
<tr>
<td>Unsafe Auto-Params</td>
<td>Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Unsafe auto-params statistic shows the number of auto-parameterization attempts per second and includes only safe auto-parameterizations (ones where the cached plan cannot be shared).</td>
</tr>
</tbody>
</table>

Metrics

SQL Server's ability to match new SQL statements with existing, unused execution plans is increased when parameters or parameter markers are used in Transact-SQL statements. If an SQL statement is executed without parameters, SQL Server parameterizes the statement internally to increase the possibility of matching it against an existing execution plan. A high number for auto-param attempts shows that SQL Server is efficiently reusing existing cached plans. A small number for failed auto-param attempts shows that SQL Server is efficiently reusing existing cached plans.
Because compilation is a significant part of a query's turnaround time, you should strive to have as many compilations stored in the cache as possible. If this number does not stabilize in direct proportion to user activity stabilizing, you should investigate your SQL Cache to see if it has adequate memory assigned to it. Recompiles slow down the process that is executing the procedure and increases the load on the CPU. By extension, the more recompiles that are occurring on your system, the more overall load increases resulting in poor performance. In general, you want to keep the number of recompiles low. The most common reasons SQL Server would issue a recompile are:

- Running sp_recompile against any table referenced in the stored procedure.
- Significant data changes in a referenced table.
- Schema changes to referenced objects.
- The use of the WITH RECOMPILE clause in the CREATE PROCEDURE or EXECUTE statement.
- A plan no longer available in the system cache.

**Troubleshooting**

For failed auto-param attempts, you can increase the ability of the relational engine to match complex SQL statements to existing, unused execution plans, by explicitly specify the parameters using either sp_executesql or parameter markers in your T-SQL code.

For high numbers of SQL re-compilations, try to practice coding standards that eliminate the most frequent causes detailed above. Also, try to:

- Use temporary tables only in the stored procedure that created them.
- Minimize creating temporary tables in control block structures.
- Use the KEEP PLAN option on references to static temporary tables.
- Issue the CREATE TABLE statement before any other references to the created table.
- Minimize the use of temporary tables.

**Lock Activity**

**Metrics**

The Lock Activity section of the User I/O tab of the I/O Detail displays statistics that track various user-related I/O operations. The I/O function along with its counter value is presented. The Lock Activity section details performance statistics that reflect how SQL Server is handling lock operations. The table below describes the information available in the Lock Activity section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Lock Escalations</td>
<td>The number of times locks on a table were escalated.</td>
</tr>
<tr>
<td>Locks</td>
<td>The total number of locks acquired on the server.</td>
</tr>
<tr>
<td>Blocked Users</td>
<td>The users currently being blocked by other processes.</td>
</tr>
<tr>
<td>Deadlocks</td>
<td>The number of deadlocks detected by SQL Server. Page Deadlocks occur when processes cannot proceed because they are waiting on a set of resources held by each other or held by other processes.</td>
</tr>
</tbody>
</table>

**Metrics**

Many table lock escalations could indicate contention problems. If increasing numbers of table lock escalations are viewed at the same time as blocking or deadlock problems, the application design could be at fault.
Consistently seeing positive numbers for the blocked statistic should also clue you into the fact that a bottleneck exists for some processes. You can easily drill down and discover the exact process(es) holding locks that are blocking out other user activity. Another situation to look for with respect to locking, is when the total number of acquired locks reaches the maximum lock limit currently set on SQL Server.

Consistently seeing page deadlock counts greater than zero indicates that some user processes are experiencing delays completing their work. When SQL Server identifies a page deadlock, it resolves the situation by choosing the process that can break the deadlock. This process is termed the deadlock victim. SQL Server rolls back the deadlock victim's transaction, and then notifies the process' application by returning an error message. It also cancels the process' request and allows the transactions of the remaining processes to continue. SQL Server always attempts to choose the least expensive thread running the transaction as the deadlock victim.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the stranglehold on the objects the user was accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Embarcadero Performance Analyst, but preventing the blocking lock situation in the first place is tricky. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

Because SQL Server automatically resolves deadlock situations, you should work proactively to prevent them in the first place. You can change default deadlock behavior by using the SET DEADLOCK_PRIORITY command, which reprioritizes a process' position in a deadlock situation.

Temporary Object Activity

The Temporary Object Activity section of the User I/O tab of the Users Detail displays statistics that track various user-related I/O operations. The I/O function along with its counter value is presented. The Temporary Object Activity section details performance statistics that reflect how SQL Server is handling the creation of temporary objects, such as temporary tables used in stored procedures. The table below describes the information available in the Temporary Object Activity section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workfiles Created</td>
<td>The number of workfiles created by SQL Server.</td>
</tr>
<tr>
<td>Worktables Created</td>
<td>The total number of work tables created. Worktables are used many times by SQL Server to perform a logical operation specified in an end-user SQL statement. GROUP BY, ORDER BY, or UNION queries can cause worktables to be created as can specific CREATE statements used in Transact SQL processing. Worktables are built in the tempdb database and are dropped automatically at the end of the statement or procedure run.</td>
</tr>
</tbody>
</table>

Metrics

The only thing to keep in mind with respect to worktables is that the tempdb database should be large enough to hold large worktables.
Database I/O Tab
The Database I/O tab of the I/O Detail includes the following sections:

<table>
<thead>
<tr>
<th>Database Bytes Read/Write Summary</th>
<th>Database I/O Detail</th>
<th>Database Read/Write Summary</th>
</tr>
</thead>
</table>

Related Topics
File Tab
System I/O Tab
User I/O Tab

Database Read/Write Summary

**Metrics**
The Reads statistic represents that number of physical database page reads that are issued per second by SQL Server. The Writes statistic represents the number of physical database page writes issued by SQL Server. Writes take place during operations such as checkpoints, lazywriter writes, index creations, and BCP routines. Both statistics are collective in nature, meaning that they represent the total page reads and writes across all databases that exist on the target SQL Server.

Page reads are to be expected, especially after initial server start up. This is because SQL Server must first satisfy requests for data and metadata by reading information in from physical disk. Numerous page reads can also be expected if the physical server does not contain an adequate amount of memory to hold repetitively requested blocks of information. No hard-and-fast rules exist for how many page reads per second is too much. You can cross reference this statistic with the physical server disk statistics to see if physical page reads and accompanying physical disk I/O is approaching the server's premium capacity levels. And because logical I/O is always many times faster than physical I/O, you should also evaluate the buffer cache hit ratio to determine overall memory vs. physical read efficiency.

Page Writes can give you an idea of overall physical write activity. There are a number of statistics, however, that pertain specifically to certain write activities like checkpoints, that can be examined to determine the amount of physical writes caused by distinct SQL Server processes.

With respect to performance, response times experienced by SQL Server users are normally not impacted by write operations unless the writes are synchronous in nature. These are typically BCPs, database recovery operations, and index creations.

Database Bytes Read/Write Summary

**Metrics**
The Database Bytes Read/Write Summary statistics represent the number of bytes read and written by SQL Server.

None.

Database I/O Detail

**Metrics**
The Database I/O section of the Database I/O tab of the I/O Detail summarizes I/O activity for each database, letting you quickly spot the “hot” databases on your server. The table below describes the information available in the Database I/O Detail section for SQL Server 2000 and later:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Reads</td>
<td>The number of reads issued against the database.</td>
</tr>
<tr>
<td>Writes</td>
<td>The number of writes issued against the database.</td>
</tr>
<tr>
<td>Bytes Read</td>
<td>The total number of bytes read for the database.</td>
</tr>
<tr>
<td>Bytes Written</td>
<td>The total number of bytes written for the database.</td>
</tr>
<tr>
<td>I/O Stall</td>
<td>The total amount of time that processes have waited for I/O operations to complete, in milliseconds.</td>
</tr>
</tbody>
</table>

**Metrics**

Consider moving databases with lots of I/O activity and wait time onto separate drives/devices.

**File Tab**

The File tab of I/O Detail includes the following sections:

- **File Bytes Read/Write Summary**
- **File I/O Detail**
- **File Read/Write Summary**

**Related Topics**

- **Database I/O Tab**
- **System I/O Tab**
- **User I/O Tab**

**File Read/Write Summary**

- **Metrics**

The Reads statistic represents that number of physical database page reads that are issued per second by SQL Server. The Writes statistic represents the number of physical database page writes issued by SQL Server. Writes take place during operations such as checkpoints, lazywriter writes, index creations, and BCP routines. Both statistics are collective in nature, meaning that they represent the total reads and writes across all databases that exist on the target SQL Server.

**Metrics**

Page reads are to be expected, especially after initial server start up. This is because SQL Server must first satisfy requests for data and metadata by reading information in from physical disk. Numerous page reads can also be expected if the physical server does not contain an adequate amount of memory to hold repetitively requested blocks of information. No hard and fast rules exist for how many page reads per second is too much. You can cross reference this statistic with the physical server disk statistics to see if physical page reads and accompanying physical disk I/O is approaching the server’s premium capacity levels. And because logical I/O is always many times faster than physical I/O, you should also evaluate the buffer cache hit ratio to determine overall memory vs. physical read efficiency.

Page Writes can give you an idea of overall physical write activity. There are a number of statistics, however, that pertain specifically to certain write activities like checkpoints, that can be examined to determine the amount of physical writes caused by distinct SQL Server processes.
With respect to performance, response times experienced by SQL Server users are normally not impacted by write operations unless the writes are synchronous in nature. These are typically BCPs, database recovery operations, and index creations.

File Bytes Read/Write Summary

- **Metrics**

The File Bytes Read/Written statistics communicate the number of bytes read and written by SQL Server.

**Metrics**

None.

File I/O Detail

- **Metrics**

The File I/O Detail section of the File tab of the I/O Detail summarizes I/O activity for each database file, letting you quickly spot the “hot” databases and files on your server. The table below describes the information available in the File I/O Detail section for SQL Server 2000 and later:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The database name.</td>
</tr>
<tr>
<td>File ID</td>
<td>The file identifier for the target file.</td>
</tr>
<tr>
<td>Logical Name</td>
<td>The name given the file by the DBA.</td>
</tr>
<tr>
<td>File Name</td>
<td>The physical file name of the file.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>The internal time stamp of when the data was obtained.</td>
</tr>
<tr>
<td>Reads</td>
<td>The number of reads issued against the database file.</td>
</tr>
<tr>
<td>Writes</td>
<td>The number of writes issued against the database file.</td>
</tr>
<tr>
<td>Bytes Read</td>
<td>The total number of bytes read for the database file.</td>
</tr>
<tr>
<td>Bytes Written</td>
<td>The total number of bytes written for the database file.</td>
</tr>
<tr>
<td>I/O Stall</td>
<td>The total amount of time that processes have waited for I/O operations to complete, in milliseconds.</td>
</tr>
</tbody>
</table>

**NOTE:** Data is only available for SQL Server 2000 and later.

**Metrics**

Consider moving databases and/or files with lots of I/O activity and wait time onto separate drives/devices.

Space Page Statistics

The Space page includes the following sections and statistics:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Disk Analysis</th>
<th>Space Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A SQL server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The Key Space Analysis section of the Space home page displays details about the space situation for each database in a SQL server, including their total, used, and free space. The percent used amount for each database is also shown. In addition, each database’s transaction log space is analyzed and presented.

**Metrics**

If a database or transaction log’s free space goes below the Performance Analyst recommended threshold, (and either the database or transaction log does not have its automatic growth property enabled or the files have reached their growth limit) then the DBA should take action to ensure that the database or transaction log does not run out of available free space.

**Troubleshooting**

If the percent used amount of a database is approaching problematic levels, there are three ways a DBA can rectify the situation:

1. The DBA can resize the current file(s) used by the database via an ALTER DATABASE … MODIFY FILE command.
2. The DBA can add a new file to the database via the ALTER DATABASE … ADD FILE command.
3. The DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting of each file is set appropriately.

Of course, the DBA should also ensure that enough physical space exists on the server to accommodate additional database space.

There are several things a DBA can do to ensure that a database’s log does not run out of available free space:

1. First, most transactional-oriented databases should have their logs assigned to separate physical drives other than the database. Reasons for doing this include:
   - It prevents competition for space between the log and the database itself.
   - It allows the log to be monitored for space more effectively.
   - It improves performance.
2. If the database is not critical in nature, you can set the truncate log on checkpoint option (trunc log on chkpt), which eliminates any non-active space in the log when a database checkpoint occurs.

3. Critical databases needing higher levels of recovery should have schedules established that regular perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.

4. If a critical transaction log becomes full, it might be impossible to use standard procedures to dump transactions and reclaim space. The dump operation will likely have to incorporate the no log or truncate only options.

5. If a transaction log continuously approaches dangerously low levels of free space, then the DBA should allow the underlying file(s) of the log to automatically grow to meet the demand. This can be accomplished by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting of each file is set appropriately.

The DBA should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log since only non-active space in the transaction log is removed from log dumps or truncation operations.

**Bottleneck Analysis - Space**

When using bottleneck analysis to troubleshoot a database, a DBA cannot rely only on the information contained in the wait event views that SQL Server provides. For example, a database might attempt to allocate another segment of space and yet be denied if no such free space exists on the server or if the underlying file has reached its growth limit. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Space home page displays space-related bottlenecks that might be threatening the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Space home page to succinctly communicate the general overall space situation of the monitored database:

<table>
<thead>
<tr>
<th>Database Files Near Max Size</th>
<th>Databases Low on Space</th>
<th>Databases on Same Disk as Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases Without Auto-Grow</td>
<td>Databases With Auto-Shrink</td>
<td>Log Files Near Max Size</td>
</tr>
<tr>
<td>Logs Low on Space</td>
<td>Logs Without Auto-Grow</td>
<td></td>
</tr>
</tbody>
</table>

**SQL Analysis - Space**

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end-user response times.

The SQL Analysis section of the Space home page shows a count of all SQL statements that have executed INSERT, UPDATE, DELETE, and TRUNCATE commands. Seeing these statistics should help you determine how much space-related SQL activity is occurring on your system.
The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

**NOTE:** SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

**Metrics**

The statistics in this section can be used in many different ways. For example, large numbers of inserts could indicate a growing database and that a DBA should keep a watchful eye on free space. Heavy update numbers could mean trouble for objects containing many indexes (indexes typically undergo splits and fragmentation when large volumes of updates are thrown their way). Many deletes might signal potential wasted space problems.

**Disk Analysis**

- **Metrics**
- **Troubleshooting**

The Disk Analysis section of the Performance Analyst Space home page provides a summary of the total database and log space used per server disk drive, as well as a summary of free disk space for each server drive.

**Metrics**

If any database or transaction log file has been set up to automatically grow, the DBA should ensure there is enough server disk space to accommodate any new, additional requests for space.

**Troubleshooting**

If you see any drive that has reached zero free space (or is close), you might want to add new files, on other disks with abundant free space, to any databases or transaction logs so that no out-of-space errors result.

**Databases Low on Space**

- **Metrics**
- **Troubleshooting**

A SQL server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The Databases Low on Space statistic represents a count of databases that have fallen below the Performance Analyst recommended free space limit.

**Metrics**

If a database’s free space goes below the Performance Analyst recommended threshold, (and either the database or transaction log does not have their automatic growth property enabled or the files have reached their growth limit) then the DBA should take action to ensure that the database or transaction log does not run out of available free space.
Troubleshooting
If the percent used amount of a database is approaching problematic levels, then there are three ways a DBA can rectify the situation:

1. The DBA can resize the current file(s) used by the database via an ALTER DATABASE … MODIFY FILE command.
2. The DBA can add a new file to the database via the ALTER DATABASE … ADD FILE command.
3. The DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting of each file is set appropriately.

Of course, the DBA should also ensure that enough physical space exists on the server to accommodate additional database space.

Databases Without Auto-Grow

**Metrics**

In Microsoft SQL Server, a DBA has the ability to tell SQL Server to automatically grow a database in size when more space is required. This feature can save a critical transaction or other database request from failing due to a lack of free space in the database. It is recommended that critical databases have this feature enabled.

The Databases Without Auto-Grow statistic provides a count of databases that do not have their automatic growth property enabled.

**Metrics**

Static databases (those not expected to grow in size) will likely not need their auto-growth property enabled. Growing, dynamic databases should almost always be allowed to automatically grow when needed.

**Troubleshooting**

If any critical, dynamic database is found to not have its auto-growth feature enabled, then the DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting of each file is set appropriately.

Databases With Auto-Shrink

**Metrics**

A database can be set up to automatically shrink in size if SQL Server finds the database with an abundance of unused free space. Setting a database up so that this occurs can help reduce wasted storage on a database server. The Databases With Auto-Shrink statistic provides a count of databases that are allowed to automatically shrink in size.

**Metrics**

Dynamic databases that have the potential to balloon in size and then have large amounts of that same data removed should be considered candidates for auto-shrink.
Troubleshooting
If a DBA wants to configure a database to automatically shrink, they can issue a command similar to the following:

```
EXEC sp_dboption 'mydatabase', 'autoshrink', true
```

Database Files Near Max Size

A DBA can specify that one or more files of a SQL Server database can automatically grow by enabling the file’s auto-growth property. A DBA can also specify that the file not be permitted to grow beyond a certain limit, and once that limit is reached, the file and/or the database might be permanently out of space. The Database Files Near Max Size statistic provides a count of how many database files are near their maximum file size.

Metrics
A DBA should not see any positive indicator that any file has reached its growth limit. While it might be okay from the standpoint that a file cannot simply grow anymore (if the database has other files that can grow or that no more space will be added to the database), the DBA should disable the auto-growth property of the file.

Troubleshooting
If a file has reached its growth limit, a DBA can take a number of actions, including:

- Eliminate the growth limit for the file either by increasing the limit size or allowing to grow in an unlimited fashion.
- If the file is on a disk or file system that has no free space left, the DBA can add another file to the database to allow for space expansion and alter the problem file to have auto-growth property disabled.

Databases on Same Disk as Log

To minimize I/O contention, a database should have its database files located on different physical disks than its log files. The Databases on Same Disk as Log statistic provides a count of how many databases have files that are on the same physical disk as their transaction log.

Metrics
For critical production systems, all databases should have their database and log files on different physical disks.

Troubleshooting
If a DBA has database and log files on the same physical disk, they can create new database filegroups on different disks, relocate objects to the new filegroups, and then delete the old files/filegroups.

Logs Low on Space

Troubleshooting
A SQL server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The Logs Low on Space statistic represents a count of database transaction logs that have fallen below the Performance Analyst recommended free space limit.

**Metrics**

If a database's transaction log free space goes below the Performance Analyst recommended threshold, (and the transaction log does not have their automatic growth property enabled or the files have reached their growth limit) then the DBA should take action to ensure that the transaction log does not run out of available free space.

**Troubleshooting**

There are several things a DBA can do to ensure that a database's log does not run out of available free space:

1. First, most transactional-oriented databases should have their logs assigned to a separate physical drive than the database. Reasons for doing so include:
   - It prevents competition for space between the log and the database itself.
   - It allows the log to be monitored for space more effectively.
   - It improves performance.

2. If the database is not critical in nature, you can set the truncate log on checkpoint option (trunc log on chkpt), which will eliminate any non-active space in the log when a database checkpoint occurs.

3. Critical databases needing higher levels of recovery should have schedules established that regular perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.

4. If a critical transaction log becomes full, it might be impossible to use standard procedures to dump transactions and reclaim space. The dump operation will likely have to incorporate the no log or truncate only options.

5. If a transaction log continuously approaches dangerously low levels of free space, then the DBA should allow the underlying file(s) of the log to automatically grow to meet the demand. This can be accomplished by using the ALTER DATABASE ... MODIFY FILE ... FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

The DBA should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log since only non-active space in the transaction log is removed from log dumps or truncation operations.

**Logs Without Auto-Grow**

- **Metrics**
- **Troubleshooting**

In Microsoft SQL Server, a DBA has the ability to tell SQL Server to automatically grow a database or transaction log in size when more space is required. This feature can save a critical transaction or other database request from failing due to a lack of free space in the database or transaction log. It is recommended that critical databases and their transaction logs have this feature enabled.

The Logs Without Auto-Grow statistic provides a count of transaction logs that do not have their automatic growth property enabled.

**Metrics**

Static databases (those not expected to grow in size) will likely not need their transaction log's auto-growth property enabled. Growing, dynamic databases should almost always have their transaction log be set to automatically grow when needed.
Troubleshooting
If any critical, dynamic database is found to not have their transaction log auto-growth feature enabled, then the DBA can modify the file(s) used by the database’s transaction log to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

Log Files Near Max Size
- **Metrics**
- **Troubleshooting**

A DBA can specify that one or more transaction log files of a SQL Server database can automatically grow by enabling the file’s auto-growth property. A DBA can also specify that the file not be permitted to grow beyond a certain limit, and once that limit is reached, the file and/or the transaction log might be permanently out of space. The Log Files Near Max Size statistic provides a count of how many transaction log files are near their maximum file size.

**Metrics**
A DBA should not see any positive indicator that any file has reached its growth limit. While it might be okay from the standpoint that a file cannot simply grow anymore (if the database transaction log has other files that can grow or that no more transactions are expected to occur within the database), the DBA should disable the auto-growth property of the file.

**Troubleshooting**
If a file has reached its growth limit, a DBA can take a number of actions:

- Eliminate the growth limit for the file either by increasing the limit size or allowing to grow in an unlimited fashion.
- If the file is on a disk or file system that has no free space left, the DBA can add another file to the database’s transaction log to allow for space expansion and alter the problem file to have auto-growth property disabled.

Space Detail
The Space Detail includes the following tabbed pages:

<table>
<thead>
<tr>
<th>File Groups Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Metrics</strong></td>
</tr>
<tr>
<td><strong>Troubleshooting</strong></td>
</tr>
</tbody>
</table>

SQL Server manages physical storage space through files and file groups. The file groups tab displays detailed information regarding storage usage in each of the SQL Server file groups. The table below describes the information available in the File Groups Summary section of the File Groups tab:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>File Group ID</td>
<td>The numeric ID of the file group.</td>
</tr>
</tbody>
</table>
Metrics
Unless space is tight on a server, it is normally wise practice to allow your files to automatically grow to meet demand for more incoming data. It is also smart to physically separate your database and log files onto separate physical drives.

Troubleshooting
To let your files automatically grow until out of space, you can easily do this by setting the file's growth option in Embarcadero’s DBArtisan or Embarcadero’s Rapid SQL, or by using the ALTER DATABASE... MODIFY FILE command.

Files Tab
SQL Server manages physical storage space through files and file groups. The Files tab displays detailed information regarding storage usage in each of the SQL Server files. The table below describes the information available in the Files Summary section of the Files tab:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Group</td>
<td>The name of the file group.</td>
</tr>
<tr>
<td>Can Grow</td>
<td>This indicates if one of more of the files making up the file group is allowed to grow automatically.</td>
</tr>
<tr>
<td>File Count</td>
<td>The number of files that make up the file group.</td>
</tr>
<tr>
<td>Size (MB)</td>
<td>The total physical size of the file group.</td>
</tr>
<tr>
<td>Table Res</td>
<td>The amount of reserved space consumed by tables.</td>
</tr>
<tr>
<td>Index Res</td>
<td>The amount of reserved space consumed by indexes.</td>
</tr>
<tr>
<td>Free (MB)</td>
<td>The amount of free space for the file group.</td>
</tr>
<tr>
<td>Free Percent</td>
<td>The percentage of free space for the file group.</td>
</tr>
</tbody>
</table>

Metrics
Unless space is tight on a server, it is normally wise practice to allow your files to automatically grow to meet demand for more incoming data. It is also smart to physically separate your database and log files onto separate physical drives.
Troubleshooting
To let your files automatically grow until out of space, you can easily do this by setting the file’s growth option in Embarcadero’s DBArtisan or Embarcadero’s Rapid SQL, or by using the ALTER DATABASE... MODIFY FILE command.

Virtual Log Files Tab

The Virtual Log Files tab of the Space Detail displays an internal structure of sorts for each database's log. The presented information is helpful when trying to shrink a database's log because you can see how much of the log is active and exactly where the active portion resides. The table below describes the information available on the Virtual Log Files tab of the Space Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of database.</td>
</tr>
<tr>
<td>File Name</td>
<td>The name of the log file.</td>
</tr>
<tr>
<td>Status</td>
<td>This indicates if this portion of the log is active or inactive (not being used).</td>
</tr>
<tr>
<td>Size</td>
<td>The size of this portion of the log in MBs.</td>
</tr>
</tbody>
</table>

Metrics
None.

Objects Tab
The Objects tab of the Space/Databases Detail includes the following sections:

- **Database Object Detail**
- **Database Object Summary**

Database Object Summary

The Database Object Summary grid displays summary information regarding table and index storage for all databases. The table below describes the information available in the Database Object Summary section of the Objects tab of the Space Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Table Count</td>
<td>The count of all tables in the database.</td>
</tr>
<tr>
<td>Index Count</td>
<td>The count of all indexes in the database.</td>
</tr>
<tr>
<td>Table Reserved (MB)</td>
<td>The amount of space (in MB) reserved by tables.</td>
</tr>
<tr>
<td>Index Reserved (MB)</td>
<td>The amount of space (in MB) reserved by indexes.</td>
</tr>
</tbody>
</table>
Metrics

Negative numbers viewed for index space information can be caused by inaccuracies contained in the SQL Server data dictionary. Frequently, running a DBCC UPDATEUSAGE command against the database resolves the problem. However, there are bugs in SQL Server that sometimes caused erroneous numbers to the reported for the reserved space amounts used by tables and indexes.

Database Object Detail

Tables and indexes consume the storage in all databases. The Database Object Detail grid displays object space details for the database selected in the Database Object Summary grid.

The table below describes the information available in the Database Object Detail section of the Objects tab of the Space Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Owner</td>
<td>The owner of the object.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>Clustered</td>
<td>Indicates if the index is clustered.</td>
</tr>
<tr>
<td>File Group</td>
<td>The file group where the object resides.</td>
</tr>
<tr>
<td>Reserved (KB)</td>
<td>The amount of space reserved by the object in kilobytes.</td>
</tr>
<tr>
<td>Used (KB)</td>
<td>The amount of space used by the object in kilobytes.</td>
</tr>
<tr>
<td>Free (KB)</td>
<td>The amount of free space used by the object in kilobytes.</td>
</tr>
<tr>
<td>Percent Used</td>
<td>The percentage of space used by the object.</td>
</tr>
</tbody>
</table>

Metrics

Negative numbers viewed for index space information can be caused by inaccuracies contained in the SQL Server data dictionary. Frequently, running a DBCC UPDATEUSAGE command against the database resolves the problem. However, there are bugs in SQL Server that sometimes caused erroneous numbers to the reported for the reserved space amounts used by tables and indexes.

Server Space Tab

The Server Space tab of the Space Detail displays the amounts of space used by SQL Server per physical drive and by database per physical drive. The first section displays summary information and the second section contains space information per database.

The table below describes the information available in the Disk Summary By Space section on the Server Space tab of the Space Detail:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Data (MB)</td>
<td>The amount of reserved database space on the drive.</td>
</tr>
<tr>
<td>SQL Log (MB)</td>
<td>The amount of reserved log space on the drive.</td>
</tr>
<tr>
<td>Free</td>
<td>The total amount of free space that remains on the drive.</td>
</tr>
</tbody>
</table>
The table below describes the information available in the Disk Summary By Database section on the Server Space tab of the Space Detail:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Drive</td>
<td>The physical drive letter.</td>
</tr>
<tr>
<td>Database</td>
<td>The database name.</td>
</tr>
<tr>
<td>SQL Data (MB)</td>
<td>The amount of reserved database space on the drive.</td>
</tr>
<tr>
<td>SQL Log (MB)</td>
<td>The amount of reserved log space on the drive.</td>
</tr>
</tbody>
</table>

**Metrics**

If you allow your database and/or log files to automatically grow, and you see little or no free space left on their physical drives, an option is to add new files to the database or log on different drives to avoid any out of space errors.

It is also smart to physically separate your database and log files onto separate physical drives.

**Troubleshooting**

If you need to add new files to your databases or logs, you can do so easily by using the `ALTER DATABASE... ADD FILE` and `ALTER DATABASE... ADD LOG FILE` commands.

**Users Page Statistics**

The Users page includes the following sections and statistics:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key User Analysis</th>
<th>Login Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Analysis</td>
<td>Workload Analysis</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topics**

- Users Detail
- Home Page Statistics
- Database Page Statistics
- I/O Page Statistics
- Memory Page Statistics
- OS Page Statistics
- Space Page Statistics

**Key User Analysis - Users**

User database activity can be examined using key performance statistics and wait/bottleneck-based analysis.

The following key statistics are used on the Performance Analyst Users home page to succinctly communicate the general session-based activity levels of the monitored database:
Bottleneck Analysis - Users

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem. The Performance Analyst Home page identifies the top system and session waits that might be currently a cause of lessened performance.

When using bottleneck analysis, a DBA cannot rely only on the information contained in the wait events that SQL Server provides. For example, a database might be suspect or offline. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment.

Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Home page displays space-related bottlenecks as well as other sources of contention/bottlenecks that might be dragging down the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Users home page to succinctly communicate the general overall performance level of the monitored database:

<table>
<thead>
<tr>
<th>Current Blocked Processes</th>
<th>Failed Auto-Param Attempts</th>
<th>Lock Timeouts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock Waits</td>
<td>Lock Wait Time</td>
<td>Scan Point Revalidations</td>
</tr>
<tr>
<td>SQL Re-Compilations</td>
<td>Table Lock Escalations</td>
<td>Top Bottlenecked User Processes</td>
</tr>
</tbody>
</table>

SQL Analysis - Users

- Metrics

A lot of a database’s overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured server down in terms of end-user response times.

Before you can identify problem SQL in your database, you have to ask the question - “What is bad SQL?” What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL, so there is no way to sufficiently answer this question to every SQL Server professional’s satisfaction. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads (Physical)</td>
<td>This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user’s request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements’ logical I/O as well.</td>
</tr>
<tr>
<td>Writes</td>
<td>These can be caused by DML activity (INSERTs, etc.), but writes can also be indicative of heavy disk sort activity.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>This is how much CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.</td>
</tr>
</tbody>
</table>
There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

**NOTE:** SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

### Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you should answer:

- What has been the worst SQL that has historically been run in my database?
- What is the worst SQL that is running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It might be that the query is in an inefficient Transaction SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can drill down further into the Performance Analyst Top SQL view and begin working with the query through EXPLAIN plan analysis and other techniques inside Embarcadero’s DBArtisan.

### Login Analysis

- **Metrics**

The Login Analysis section displays the number of successful login and logout operations.

- **Metrics**

  None.

### Workload Analysis - Users

- **Metrics**

When the database population as a whole experiences a system slowdown, it is not uncommon to find one or two users who are responsible for bringing the system to its knees. In the best of worlds, users will have a pretty evenly divided amount of memory usage, disk I/O, CPU utilization, and parse activity. Unfortunately, this usually is not the case, and many times you will find users submitting large batch jobs during peak OLTP activity, or sessions that are firing off untuned queries on a critical system.
If you are seeing a slowdown in your database, and cannot seem to find a root cause, one thing to examine is the resource consumption of the leading sessions on a system. Oftentimes you will find one or a handful of users making things miserable for everyone else. They accomplish this by overwhelming the I/O capabilities of SQL Server (through untuned queries or runaway batch jobs) or hammering the CPU or memory structures of the database and server.

Performance Analyst makes it easy to pinpoint the top sessions by showing the leading processes at the physical I/O, memory, CPU, and transaction submission usage levels.

**Metrics**

If any one session appears to be using more than 50% of a total resource (CPU, memory, etc.), then you should drill down into the session to find out what they are currently executing.

**Transaction/Sec**

The transaction/sec statistic refers to the number of database transactions processed per second by SQL Server.

**Metrics**

None.

**Active Transactions**

The active transactions statistic represents a count of the number of in-process transactions for SQL Server.

**Metrics**

None.

**T-SQL Batches**

The T-SQL batches statistic refers to the number of transact SQL batches processed by SQL Server.

**Metrics**

None.

**Active Processes**

The Active Processes statistic represents the total number of active and open threads reported on the server. This number displays the number of processes actively performing work.

**Metrics**

None.
Inactive Processes

- **Metrics**

The Inactive Processes statistic represents the total number of threads logged on to the server that are idle at the current time.

**Metrics**

None.

System Processes

- **Metrics**

The System Processes statistic represents the total number of threads logged on to the server that are SQL Server internal processes.

**Metrics**

None.

Top Bottlenecked User Processes

- **Metrics**

When viewing wait statistics, there are several levels of detail that a DBA can drill down into. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. The second level is the session view, which shows details on what events connected sessions are experiencing. Viewing these numbers can help a DBA determine which session-related wait events have caused the most commotion in a database thus far. The Top Bottlenecked User Processes display identifies the processes with the most current wait time along with the event causing their current wait.

**Metrics**

None.

SQL Re-Compilations

- **Metrics**

The SQL re-compilations statistic represents the total number of recompiles triggered per second in a SQL Server instance. Recompiles occur when SQL Server determines that the currently defined execution plan for an executing stored procedure might no longer be the best possible plan. SQL Server pauses the query execution and recompiles the stored procedure.

**Metrics**

Recompiles slow down the process that is executing the procedure and increase the load on the CPU. By extension, the more recompiles that are occurring on your system, the more overall load increases, resulting in poor performance. In general, you want to keep the number of recompiles low. The most common reasons SQL Server would issue a recompile are: Running sp_recompile against any table referenced in the stored procedure, significant data changes in a referenced table, schema changes to referenced objects, the use of the WITH RECOMPILE clause in the CREATE PROCEDURE or EXECUTE statement, and a plan no longer available in the system cache.
Troubleshooting
Try to practice coding standards that eliminate the most frequent causes detailed above. Also, try to:

- Use temporary tables only in the stored procedure that created them.
- Minimize creating temporary tables in control block structures.
- Use the KEEP PLAN option on references to static temporary tables.
- Issue the CREATE TABLE statement before any other references to the created table.
- Minimize the use of temporary tables.

Table Lock Escalations

- **Metrics**

The Table Lock Escalations statistic represents the number of times locks on a table were escalated.

**Metrics**
Many table lock escalations could indicate contention problems. If increasing numbers of table lock escalations are viewed at the same time as blocking or deadlock problems, then the application design might be at fault.

Scan Point Revalidations

- **Metrics**

The Scan Point Revalidations statistic represents the number of times per second that the scan point had to be revalidated to continue the scan.

**Metrics**
None.

Failed Auto-Param Attempts

- **Metrics**

Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Failed Auto-Param Attempts statistic shows the number of failed auto-parameterization attempts per second.

**Metrics**
SQL Server's ability to match new SQL statements with existing, unused execution plans increases when parameters or parameter markers are used in Transact-SQL statements. If an SQL statement is executed without parameters, SQL Server parameterizes the statement internally to increase the possibility of matching it against an existing execution plan. A small number for this statistic shows that SQL Server is efficiently reusing existing cached plans.

**Troubleshooting**
You can increase the ability of the relational engine to match complex SQL statements to existing, unused execution plans, by explicitly specify the parameters using either `sp_executesql` or parameter markers in your T-SQL code. Doing so helps lower this number.
Current Blocked Processes

- **Metrics**

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely.

**Metrics**

You should immediately investigate any indicator above zero, before the situation has a chance to mushroom.

**Troubleshooting**

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Lock Timeouts

- **Metrics**

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely. The lock timeouts represent the number of locks that have timed out due to the issuing of the aforementioned command.

**Metrics**

Any indicator above zero might indicate the presence of lock contention on the server.

**Troubleshooting**

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.
The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Lock Waits

• **Metrics**

• **Troubleshooting**

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely. The Lock Waits statistic represents the number of lock requests that required a process to wait.

**Metrics**
Any indicator above zero might indicate the presence of lock contention on the server.

**Troubleshooting**

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Lock Wait Time

• **Metrics**

• **Troubleshooting**

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely. The lock wait time statistic represents the wait time (in milliseconds) that processes have had to wait to obtain a lock request.

**Metrics**
Any indicator above zero might indicate the presence of lock contention on the server.
Troubleshooting
Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Users Detail
The Users Detail includes the following tabbed pages:

<table>
<thead>
<tr>
<th>Locks</th>
<th>Session Waits</th>
<th>System Waits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Sessions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Top Sessions Tab
The Top Sessions tab of the Users Detail includes the following processes:

<table>
<thead>
<tr>
<th>Top CPU Process</th>
<th>Top I/O Process</th>
<th>Top Memory Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Sessions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related Topics
- Locks Tab
- Session Waits Tab
- System Waits Tab

Top I/O Process

- Metrics

The Top I/O Process statistic identifies the SQL Server process that currently has caused the most I/O usage on the database.

Metrics
None.

Top Memory Process

- Metrics
The Top Memory Process statistic identifies the SQL Server process that currently is using the highest percentage of memory in the database.

**Metrics**

None.

**Top CPU Process**

**Metrics**

The Top CPU Process statistic identifies the SQL Server process that currently has the highest percentage of CPU usage in the database.

**Metrics**

None.

**Top Sessions**

When a system experiences heavy activity, sometimes you will find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are causing most of the activity. It might be that a large batch load or other typical process is running that is perfectly okay for your system. Or it might be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

The Top Sessions section displays information regarding all key statistics for all current sessions on a SQL Server. The table below describes the information available in the Top Sessions section on the Top Sessions tab of the Users Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The unique identifier for the process.</td>
</tr>
<tr>
<td>Login</td>
<td>The login name for the process.</td>
</tr>
<tr>
<td>Windows User</td>
<td>The operating system name of the process.</td>
</tr>
<tr>
<td>Database</td>
<td>The name of the database the process is working in.</td>
</tr>
<tr>
<td>Status</td>
<td>The current status of the process.</td>
</tr>
<tr>
<td>Program</td>
<td>This identifies the program being run by the process.</td>
</tr>
<tr>
<td>Memory</td>
<td>The number of pages in the procedure/SQL cache allocated to the process.</td>
</tr>
<tr>
<td>CPU</td>
<td>The cumulative CPU time for the process.</td>
</tr>
<tr>
<td>Physical I/O</td>
<td>The cumulative total of all physical reads and writes for the process.</td>
</tr>
<tr>
<td>Blocked</td>
<td>This indicates if the process is blocked by another process.</td>
</tr>
<tr>
<td>Host</td>
<td>This identifies the workstation of the process.</td>
</tr>
<tr>
<td>Open Transaction</td>
<td>This indicates if the process has an open transaction.</td>
</tr>
<tr>
<td>Command</td>
<td>The command being currently issued by the process.</td>
</tr>
<tr>
<td>Login Time</td>
<td>This identifies when the process logged into SQL Server.</td>
</tr>
<tr>
<td>Last Batch</td>
<td>This indicates the date/time the process executed a remote stored procedure or issued an EXEC command.</td>
</tr>
<tr>
<td>Host Process</td>
<td>The process identifier on the SQL Server machine.</td>
</tr>
<tr>
<td>Net Address</td>
<td>The network address of the process.</td>
</tr>
</tbody>
</table>
Current SQL
This view shows a snippet of the SQL associated with the user in question.

System Waits Tab
The System Waits tab of the Users Detail includes the following sections:

<table>
<thead>
<tr>
<th>System Waits</th>
<th>Wait Percent by Time Waited</th>
<th>Wait Percent by Total Waits</th>
</tr>
</thead>
</table>

Related Topics
Locks Tab
Session Waits Tab
Top Sessions Tab

System Waits
- Metrics
- Troubleshooting

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The System Waits section identifies the waits that have occurred on SQL Server. The System Waits section identifies the waits that have occurred on SQL Server. The table below describes the information available in the System Waits section on the System Waits tab of the Users Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait Type</td>
<td>The actual wait event name.</td>
</tr>
<tr>
<td>Requests</td>
<td>The number of waits.</td>
</tr>
<tr>
<td>Percent of Total</td>
<td>The percentage of the occurrences of this wait to all wait requests.</td>
</tr>
<tr>
<td>Time Waited</td>
<td>The number of seconds SQL Server spent waiting for this wait.</td>
</tr>
<tr>
<td>Percent Time Waited</td>
<td>The percentage of time SQL Server spent waiting on this wait.</td>
</tr>
<tr>
<td>Signal Wait Time</td>
<td>The signal wait time for the wait.</td>
</tr>
<tr>
<td>Percent Signal Wait Time</td>
<td>The percentage of signal wait time SQL Server spent waiting on this wait.</td>
</tr>
</tbody>
</table>

Metrics
Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

Troubleshooting

The table below describes some possible solutions to some of the most common wait events:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURSOR</td>
<td>This indicates SQL is waiting to sync up with asynchronous cursors and can point to the excessive use of cursors.</td>
</tr>
<tr>
<td>CMEMTHREAD</td>
<td>This indicates waits for memory to be freed up for use.</td>
</tr>
<tr>
<td>CXPACKET</td>
<td>This relates to SQL Server parallelism. Oftentimes the cost of SQL Server using parallelism for a query is high, with the end result being these types of waits. If high, you can adjust SQL Server and turn parallelism off for the server or for just low cost queries.</td>
</tr>
<tr>
<td>EXCHANGE</td>
<td>Related to CXPACKET. See information above.</td>
</tr>
<tr>
<td>IO_COMPLETION</td>
<td>This means certain I/O related actions are being delayed. Bulk insert operations, growth of database or log files, high physical I/O SQL, page splitting, and server paging are potential causes. You can check Performance Analyst’s SQL Analysis section to ferret out heavy I/O SQL and the Database and File I/O Detail pages (SQL Server 2000 and later) to examine disk and database hot spots with respect to I/O. The I/O Stall column is especially useful in determining which files are experiencing the most delays.</td>
</tr>
<tr>
<td>LOGMGR</td>
<td>This refers to waits for the Log Writer to start writing a transaction. High waits here might warrant transfer transaction logs to faster devices or breaking up long running DML transactions.</td>
</tr>
<tr>
<td>OLEDB</td>
<td>This indicates waits for an OLE DB operation to act on its requests. Slow connection speeds or very high transaction rates can cause these.</td>
</tr>
<tr>
<td>PAGEIOLATCH</td>
<td>Related to IO_COMPLETION. See information above.</td>
</tr>
<tr>
<td>PAGELATCH</td>
<td>Related to IO_COMPLETION. See information above.</td>
</tr>
<tr>
<td>PAGESUPP</td>
<td>Related to CXPACKET. See information above.</td>
</tr>
<tr>
<td>PSS_CHILD</td>
<td>This is SQL waiting for a child thread within an asynchronous cursor and can point to the excessive use of cursors.</td>
</tr>
<tr>
<td>TEMPOBJ</td>
<td>This wait occurs when temp tables and the like are dropped.</td>
</tr>
<tr>
<td>WAITFOR</td>
<td>Waits caused by the WAITFOR T-SQL command. Not a cause for concern.</td>
</tr>
<tr>
<td>WRITELOG</td>
<td>Related to LOGMGR. Refers to waits for writes to disk of transactions. See information above.</td>
</tr>
<tr>
<td>XACTLOCKINFO</td>
<td>Involves the release of escalation of locks during bulk operations.</td>
</tr>
</tbody>
</table>

Wait Percent by Total Waits

- Metrics
- Troubleshooting

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Wait Percent by Total Waits section graphically depicts the top waits by their percentage to all waits.

Metrics

None.
Troubleshooting
Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

Wait Percent by Time Waited

• Metrics
• Troubleshooting

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Wait Percent by Time Waited section graphically depicts the top waits by their percentage of time waited to total time waited.

Metrics
None.

Troubleshooting
Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

Session Waits Tab
The Session Waits tab of the Users Detail includes the following sections:

<table>
<thead>
<tr>
<th>Session Wait Percent by Time Waited</th>
<th>Session Wait Percent by Wait Type</th>
<th>Session Waits</th>
</tr>
</thead>
</table>

Related Topics
Locks Tab
System Waits Tab
Top Sessions Tab

Session Waits

• Metrics
• Troubleshooting

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Session Waits section shows processes that are currently waiting on SQL Server. The table below describes the information available in the Session Waits section on the Session Waits tab of the Users Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The unique identifier for the process.</td>
</tr>
</tbody>
</table>
### Metrics
Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

[http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455)

### Troubleshooting
The table below describes some possible solutions to some of the most common wait events:

<table>
<thead>
<tr>
<th>Wait Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURSOR</td>
<td>This indicates SQL is waiting to sync up with asynchronous cursors and can point to the excessive use of cursors.</td>
</tr>
<tr>
<td>CMEMTHREAD</td>
<td>This indicates waits for memory to be freed up for use.</td>
</tr>
<tr>
<td>CXPACKET</td>
<td>This relates to SQL Server parallelism. Oftentimes the cost of SQL Server using parallelism for a query is high, with the end result being these types of waits. If high, you can adjust SQL Server and turn parallelism off for the server or for just low cost queries.</td>
</tr>
<tr>
<td>EXCHANGE</td>
<td>Related to CXPACKET. See information above.</td>
</tr>
<tr>
<td>IO_COMPLETION</td>
<td>This means certain I/O related actions are being delayed. Bulk insert operations, growth of database or log files, high physical I/O SQL, page splitting, and server paging are potential causes. You can check Performance Analyst's SQL Analysis section to ferret out heavy I/O SQL and the Database and File I/O Detail pages (SQL Server 2000 and later) to examine disk and database hot spots with respect to I/O. The I/O Stall column is especially useful in determining which files are experiencing the most delays.</td>
</tr>
<tr>
<td>LOGMGR</td>
<td>This refers to waits for the Log Writer to start writing a transaction. High waits here might warrant transfer transaction logs to faster devices or breaking up long running DML transactions.</td>
</tr>
<tr>
<td>OLEDB</td>
<td>This indicates waits for an OLE DB operation to act on its requests. Slow connection speeds or very high transaction rates can cause these.</td>
</tr>
<tr>
<td>PAGEIOLATCH</td>
<td>Related to IO_COMPLETION. See information above.</td>
</tr>
<tr>
<td>PAGELATCH</td>
<td>Related to IO_COMPLETION. See information above.</td>
</tr>
<tr>
<td>PAGESUPP</td>
<td>Related to CXPACKET. See information above.</td>
</tr>
<tr>
<td>PSS_CHILD</td>
<td>This is SQL waiting for a child thread within an asynchronous cursor and can point to the excessive use of cursors.</td>
</tr>
<tr>
<td>TEMPOBJ</td>
<td>This wait occurs when temp tables and the like are dropped.</td>
</tr>
<tr>
<td>WAITFOR</td>
<td>Waits caused by the WAITFOR T-SQL command. Not a cause for concern.</td>
</tr>
</tbody>
</table>
Session Wait Percent by Wait Type

- **Metrics**
- **Troubleshooting**

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Session Wait Percent by Wait Type section graphically depicts the top waits by wait type.

**Metrics**

None.

**Troubleshooting**

Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

[http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455)

Session Wait Percent by Time Waited

- **Metrics**
- **Troubleshooting**

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Session Wait Percent by Time Waited section graphically depicts the top session waits by their percentage of time waited to total time waited.

**Metrics**

None.

**Troubleshooting**

Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

[http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455)

**Locks Tab**

The Locks tab of the Users Detail includes the following sections:

<table>
<thead>
<tr>
<th>Locks</th>
<th>Locks by Lock Type</th>
<th>Locks by Database</th>
</tr>
</thead>
</table>
To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, SQL Server itself issues lock requests to carry out its internal duties. The Locks section gives information the locks currently on the system and also indicates if any blocking situations are occurring. The table below describes the information available in the Locks section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The process id of the process holding the lock.</td>
</tr>
<tr>
<td>Login</td>
<td>The login name of the process.</td>
</tr>
<tr>
<td>Windows User</td>
<td>The operating system name of the process.</td>
</tr>
<tr>
<td>Database</td>
<td>The database in which the process is running.</td>
</tr>
<tr>
<td>Table Name</td>
<td>The name of the table involved in a lock. This will be NULL for non-table</td>
</tr>
<tr>
<td></td>
<td>locks or table locks that take place in the tempdb database.</td>
</tr>
<tr>
<td>Index ID</td>
<td>The index ID involved in the lock.</td>
</tr>
<tr>
<td>Lock Type</td>
<td>The type of the lock (database, table, row id, etc.).</td>
</tr>
<tr>
<td>Lock Mode</td>
<td>The lock's mode (shared, exclusive, etc.).</td>
</tr>
<tr>
<td>Lock Status</td>
<td>The lock's status (waiting or granted).</td>
</tr>
<tr>
<td>Lock Owner Type</td>
<td>Whether the lock came from a regular session or a transaction.</td>
</tr>
<tr>
<td>User Program</td>
<td>The executable the process is using against the server.</td>
</tr>
<tr>
<td>Blocking SPID</td>
<td>If zero, the process is not being blocked. If non-zero, this column</td>
</tr>
<tr>
<td></td>
<td>represents the process ID of the process blocking the requested lock.</td>
</tr>
<tr>
<td>Wait Time</td>
<td>The current amount of wait time for the process, in milliseconds.</td>
</tr>
<tr>
<td>SPID Status</td>
<td>Indicates if the process is actively performing work, is idle, blocked by</td>
</tr>
<tr>
<td></td>
<td>another process, etc.</td>
</tr>
<tr>
<td>SPID Command</td>
<td>The command the process is currently issuing.</td>
</tr>
<tr>
<td>NT Domain</td>
<td>The name of Windows 2000/NT domain.</td>
</tr>
</tbody>
</table>

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the stranglehold on the objects the user was accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Embarcadero Performance Analyst, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.
Locks by Lock Type

Metrics

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, SQL Server itself issues lock requests to carry out its internal duties. The Locks by Lock Type section of the Users Detail provides a graphical percentage breakdown of the different locks types being issued across all databases.

Metrics
None.

Locks by Database

The Locks by Database section of the Lock tab provides a graphical representation of the percentage breakdown of locks by database.

Database Page Statistics

The Database page includes the following sections and statistics:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Database Analysis</th>
<th>SQL Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload Analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related Topics

Database Detail
Home Page Statistics
I/O Page Statistics
Memory Page Statistics
OS Page Statistics
Space Page Statistics
Users Page Statistics

Database Analysis

SQL Server is broken up into many different databases, all of which have their own levels of activity. The Database Analysis section displays a summary of activities that have occurred within each database. The table below describes the information available in the Database Analysis section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>DBCC Logical Scans</td>
<td>The number of DBCC scans that have occurred within the database.</td>
</tr>
<tr>
<td>Transactions</td>
<td>The number of transactions that have occurred within the database.</td>
</tr>
<tr>
<td>Active Transactions</td>
<td>The number of transactions currently running in the database.</td>
</tr>
<tr>
<td>Bulk Copy Rows</td>
<td>The number of rows inserted per second into the database via BCP operations.</td>
</tr>
</tbody>
</table>
Bottleneck Analysis - Database

When using bottleneck analysis to troubleshoot a database, a DBA cannot rely only on the information contained in the wait event views that SQL Server provides. For example, a critical database might be offline or be marked suspect. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Database home page displays database-related bottlenecks that might be dragging down the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Database home page to communicate the general overall conditions of databases on the monitored server:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulk Copy Throughput</td>
<td>The amount of data bulk (KB) copied into the database per second.</td>
</tr>
<tr>
<td>Log Cache Reads</td>
<td>The number of reads per second that have gone through the log manager cache.</td>
</tr>
<tr>
<td>Log Flushes</td>
<td>The number of transaction log flushes per second. A log flush occurs when SQL Server writes all changes from the database’s log cache out to the database’s log files on disk.</td>
</tr>
<tr>
<td>Log Growths</td>
<td>The number of times the transaction log has grown in size.</td>
</tr>
<tr>
<td>Log Shrinks</td>
<td>The number of times the transaction log has reduced itself in size.</td>
</tr>
</tbody>
</table>

SQL Analysis - Database

Certain databases on a SQL Server will experience more SQL activity than others. The SQL Analysis section displays a count of all statement executions for each database.

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

**NOTE:** SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

Error Log Analysis

- Metrics
- Troubleshooting
SQL Server records various system events in its system or error log. The majority of messages placed into the log are informational in nature, however since some contain critical messages, you should immediately review them. This section indicates the number of actual error messages in the most recent error log so you know if there are potential events that require your attention.

**NOTE:** This information is also available on the Home page view.

**Metrics**
You should investigate any positive values for errors.

**Troubleshooting**
If non-zero values are found for this statistic, you should drill down into the current error log and examine the detail found that accompanies each error issued by SQL Server.

**Workload Analysis - Database**
SQL Server is broken up into many different databases, all of which have their own levels of activity. The Workload Analysis section displays a summary of resource consumption across all databases. The table below describes the information available in the Workload Analysis section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Connections</td>
<td>The number of processes that are connected to the database.</td>
</tr>
<tr>
<td>Blocked Users</td>
<td>The number of processes in the database experiencing blocks from lock activity.</td>
</tr>
<tr>
<td>Total Memory</td>
<td>The total number of memory pages in use by processes for each database.</td>
</tr>
<tr>
<td>Total I/O</td>
<td>The total amount of physical I/O being experienced by each database.</td>
</tr>
<tr>
<td>Total CPU</td>
<td>The total amount of cumulative CPU for all processes connected to the database.</td>
</tr>
<tr>
<td>Total Wait Time</td>
<td>The total wait time (in milliseconds) that all processes connected to the database have experienced.</td>
</tr>
<tr>
<td>DBCC Running Count</td>
<td>This indicates if any DBCC operations are currently running against the database.</td>
</tr>
<tr>
<td>BCP Running</td>
<td>This indicates if any BCP operations are currently running against the database.</td>
</tr>
<tr>
<td>Backup Restore Running</td>
<td>This indicates if any backup or restore operations are currently running against the database.</td>
</tr>
</tbody>
</table>

**Suspect Databases**

- **Metrics**
- **Troubleshooting**

The Suspect Databases statistic represents the number of databases SQL Server has marked as suspect. Databases are marked suspect by SQL Server if they fail during automatic recovery, which is performed at server startup. If serious damage is experienced by a database during regular uptime, SQL server will also mark a database as suspect.

**Metrics**
You should not find any suspect databases on any production server. You should immediately investigate any non-zero numbers, for this statistic.
Troubleshooting

The steps to handling a suspect database will vary from one installation to another. However, here are some general guidelines you can use to troubleshoot a suspect database:

• Begin by examining the SQL Server error log for clues as to what caused the database to be marked as suspect.

• It is not unusual for a server to run out of physical disk space on drives used by SQL Server. When this happens, recovery for databases can sometimes fail with the end result being SQL Server marking a database as suspect. To remedy this situation, you should free up space on the identified drives or add files to the newly marked suspect database. For SQL Server 2000, this can be accomplished by utilizing the following stored procedures: sp_add_data_file_recover_suspect_db and sp_add_log_file_recover_suspect_db. For version 7.0 of SQL Server, you will need to use the sp_resetstatus stored procedure to reset the suspect status flag for the database in question, use the alter database command to add new datafiles to the database, and then stop/start the SQL Server.

• Many times, suspect databases are caused by SQL Server not being able to access a database or log file. This happens if a particular physical hard drive has become unavailable, but also can occur if another operating system process has obtained exclusive access to a file. If this scenario proves to be true, once you have ensured that the file(s) are available once again to the operating system, you can use the sp_resetstatus stored procedure to reset the suspect status flag for the database and then stop/start the SQL Server.

If none of these solutions are possible, you will likely have to restore your database using the last full and transaction log backups.

Offline Databases

• Metrics

• Troubleshooting

The Offline Databases statistic represents the number of databases SQL Server has offline, meaning that no database modifications can occur.

Metrics

You should not find any offline databases on any production server. You should immediately investigate any non-zero numbers, for this statistic.

Troubleshooting

Should an offline database be found by Performance Analyst, you can easily place it back online by utilizing either the sp_dboption stored procedure or the alter database command.

Databases Requiring Backup

• Metrics

• Troubleshooting

The Databases Requiring Backup statistic represents the number of databases in SQL Server that have not been backed up for more than seven days. This statistic excludes the pubs, tempdb, Northwind, msdb, and model databases.

Metrics

To ensure proper protection for most databases, it is recommended that even the most static databases be backed up at least once a week. You should frequently backup critical, dynamic databases and include transaction log backups to enable point-in-time recovery ability.
Troubleshooting
Any critical databases found with obsolete or no backups should immediately be backed up. Moreover, to ensure proper data protection for each database, you should institute a planned backup schedule. The timing and repetition of the backups depend on the critical recovery needs of each database.

Databases Without Auto-Update Stats

• Metrics
• Troubleshooting

The Databases Without Auto-Update Statistics statistic represents the total number of databases defined on SQL Server that do not have the AUTO_UPDATE_STATISTICS option enabled.

Metrics
When the AUTO_UPDATE_STATISTICS option is enabled, SQL Server automatically updates existing statistics when the statistics become out-of-date because data in the tables has changed enough to affect the optimizer's decision-making process.

Troubleshooting
If possible, a DBA should keep databases in AUTO_UPDATE_STATISTICS mode. If a database is found without this option set, you can easily change its auto-update statistics to true by using the command:

EXEC sp_dboption '(database name)',

Databases Without Auto-Create Stats

• Metrics
• Troubleshooting

The Databases Without Auto-Create Stats statistic represents the total number of databases defined on SQL Server that do not have the AUTO_CREATE_STATISTICS option enabled.

Metrics
When the AUTO_CREATE_STATISTICS option is enabled, statistics are automatically created on columns used in a SQL query predicate. Keeping object statistics fresh in the SQL Server data dictionary improves query performance because the optimizer can better determine how to evaluate a query and return the requested data. If the statistics are not used, SQL Server should automatically delete them.

Troubleshooting
If possible, a DBA should keep their databases in AUTO_CREATE_STATISTICS mode. If a database is found without this option set, you can easily change it by using the command:

EXEC sp_dboption '<database name>','auto create statistics',true

Page Splits

• Metrics
• Troubleshooting
When data is added or modified for a table that has indexes, the indexes must be updated as well. As index pages fill up, free space needed to keep index keys in their proper order can oftentimes run very low. If an index update needs to be made to a page, and no free space exists on that page, SQL Server will perform a page split where it moves approximately half the rows on a page to a new page. The Page Splits statistic provides a count of how many page splits are occurring in SQL Server.

**Metrics**

Page splits cause additional overhead in the form of CPU usage and I/O. Observing large numbers of page splits can signal a resource bottleneck in your server.

**Troubleshooting**

To avoid page splits, you can look into tuning the FILLFACTOR property of an index, which controls the percentage of the index page that is filled during creation. The default, 100, tells SQL Server to completely fill each page, whereas lower numbers tell SQL Server to leave room for additional index rows or updates to existing rows.

---

### Extents Allocated

**Metrics**

SQL Server groups eight 8KB pages together to form an extent of space (64KB). Space is allocated to database objects (tables, indexes) in the form of extents. The Extents Allocated statistic provides a count of the number of extents allocated to tables and indexes.

**Metrics**

None. However, be aware that excessive counts of allocated extents might signal a large load job occurring in the database.

---

### Freespace Scans

**Metrics**

The Freespace Scans statistic provides a count of the number of scans SQL Server initiated to search for free space that will accommodate requests to insert new data rows.

**Metrics**

None. However, be aware that excessive counts of freespace scans might signal INSERT problems being performed on large tables.

---

### Database Detail

The Database Detail includes the following tabbed pages:

<table>
<thead>
<tr>
<th>Backups</th>
<th>Configuration</th>
<th>Error Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>Replication</td>
<td>SQL Agent</td>
</tr>
<tr>
<td>Objects</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview Tab
The Overview tab of the Database Detail includes the following sections:

<table>
<thead>
<tr>
<th>Database Object Analysis</th>
<th>Database Size Analysis</th>
<th>Database Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database User Analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Database Size Analysis
- **Metrics**
The Database Size Analysis chart displays each database in terms of its total percentage of size to the total size of all databases on the SQL Server.

**Metrics**
None.

Database Object Analysis
- **Metrics**
The Database Object Analysis chart displays each database in terms of its total percentage of object to the total objects for all databases on the SQL Server.

**Metrics**
None.

Database User Analysis
- **Metrics**
The Database User Analysis chart displays each database in terms of its total percentage of user accounts to the total user accounts of all databases on the SQL Server.

**Metrics**
None.

Database Summary
- **Metrics**
- **Troubleshooting**
The Database Summary Analysis section displays summarized metadata for all defined databases on SQL Server. The table below describes the information available in the Database Summary section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Created</td>
<td>The date/time when the database was created.</td>
</tr>
<tr>
<td>Status</td>
<td>The status of the database (online, offline, suspect, etc.)</td>
</tr>
<tr>
<td>Database Size (MB)</td>
<td>The total size of the database in MB.</td>
</tr>
</tbody>
</table>
### Metrics

The following are things to take note of:

- Any critical database that shows a last full backup date that is older than the database’s required backup needs.
- Any database that shows a status of offline or suspect.
- Any growing database that does not have its database or log files set to automatically grow in size.
- Any dynamic database that does not have its object statistics set to automatically update.

### Troubleshooting

Depending on the situation, you should take the following actions:

For databases that require a full backup, perform a full backup of the database when appropriate.

For suspect databases, the steps to handling will vary from one installation to another. However, here are some general guidelines you can use to troubleshoot a suspect database:

- Begin by examining the SQL Server error log for clues as to what caused the database to be marked as suspect.
- It is not unusual for a server to run out of physical disk space on drives used by SQL Server. When this happens, recovery for databases can sometimes fail with the end result being SQL Server marking a database as suspect. To remedy this situation, you should free up space on the identified drives or add files to the newly marked suspect database. For SQL Server 2000, this can be accomplished by utilizing the following stored procedures: `sp_add_data_file_recover_suspect_db` and `sp_add_log_file_recover_suspect_db`. For SQL Server version 7.0, you will need to use the `sp_resetstatus` stored procedure to reset the suspect status flag for the database in question, use the `alter database` command to add new datafiles to the database, and then stop/start SQL Server.
- Many times, suspect databases are caused by SQL Server not being able to access a database or log file. This happens if a particular physical hard drive has become unavailable, but also can occur if another operating system process has obtained exclusive access to a file. If this scenario proves to be true, once you have ensured that the file(s) are available once again to the operating system, you can use the `sp_resetstatus` stored procedure to reset the suspect status flag for the database and then stop/start SQL Server.
Should an offline database be found by Performance Analyst, you can easily place it back online by utilizing either the sp_dboption stored procedure or the alter database command.

If any critical, dynamic database or log is found to not have their auto-growth feature enabled, then the DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE ... MODIFY FILE ... FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

If possible, a DBA should keep their databases in AUTO_CREATE_STATISTICS and AUTO_UPDATE_STATISTICS mode. If a database is found without this option set, you can easily change it by using the following commands:

```
EXEC sp_dboption '<database name>','auto create statistics',true
EXEC sp_dboption '<database name>','auto update statistics',true
```

**Error Log Tab**
The Error Log tab of the Database Detail includes the following sections:

<table>
<thead>
<tr>
<th>SQL Server Error Log</th>
<th>Error Log Content Analysis</th>
<th>Error Log Volume Analysis</th>
</tr>
</thead>
</table>

**SQL Server Error Log**

- **Metrics**

SQL Server logs events pertinent to its operations to the SQL Server error log and the operating system application log. Errors encountered by SQL Server are clearly visible in the log, however large logs can be difficult to navigate. The SQL Server Error Log section displays all events for a selected log or just errors.

**Metrics**

You should investigate any positive values reported for errors.

**Error Log Volume Analysis**

- **Metrics**

SQL Server logs events pertinent to its operations to the SQL Server error log and the operating system application log. Errors encountered by SQL Server as well as informational message are recorded in the log. The Error Log Volume Analysis section displays a graph that shows the volume amount of messages/errors over the last five days.

**Metrics**

Seeing a disproportionate percentage of messages for one day might indicate that something has, or is occurring that needs investigation.

**Error Log Content Analysis**

- **Metrics**

SQL Server logs events pertinent to its operations to the SQL Server error log and the operating system application log. Errors encountered by SQL Server as well as informational message are recorded in the log. The Error Log Content Analysis section displays a graph that shows percentage of informational messages to errors in the current log.

**Metrics**

You should investigate any indicated errors.
Objects Tab
The Objects tab of the Database Detail includes the following sections:

- Database Object Summary
- Database Object Detail

Database Object Summary
The table below describes the information available in the Database Object Summary section of the Objects tab of the Database Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Table Count</td>
<td>The count of all tables in the database.</td>
</tr>
<tr>
<td>Index Count</td>
<td>The count of all indexes in the database.</td>
</tr>
<tr>
<td>Table Reserved (MB)</td>
<td>The amount of space (in MB) reserved by tables.</td>
</tr>
<tr>
<td>Index Reserved (MB)</td>
<td>The amount of space (in MB) reserved by indexes.</td>
</tr>
</tbody>
</table>

Database Object Detail
The table below describes the information available in the Database Object Detail section of the Objects tab of the Database Detail:

**NOTE:** The information you see and the order in which you see it varies between the table and index views.

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Owner</td>
<td>The owner of the object.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>Clustered</td>
<td>Indicates if the index is clustered.</td>
</tr>
<tr>
<td>File Group</td>
<td>The file group where the object resides.</td>
</tr>
<tr>
<td>Pinned</td>
<td>Yes or No</td>
</tr>
<tr>
<td>Rows</td>
<td>Number of Rows</td>
</tr>
<tr>
<td>Indexes</td>
<td>Number of Indexes</td>
</tr>
<tr>
<td>Foreign Key Constraints</td>
<td>Yes or No</td>
</tr>
<tr>
<td>Check Constraints</td>
<td>Yes or No</td>
</tr>
<tr>
<td>Has Identity</td>
<td>Yes or No</td>
</tr>
<tr>
<td>Unique</td>
<td>Yes or No</td>
</tr>
<tr>
<td>Depth</td>
<td>Index depth</td>
</tr>
<tr>
<td>Stats Recompute</td>
<td>Yes or No</td>
</tr>
<tr>
<td>Max Nonleaf Index Row Size</td>
<td>Value</td>
</tr>
<tr>
<td>Max Row Size</td>
<td>Value</td>
</tr>
<tr>
<td>Reserved (KB)</td>
<td>The amount of space reserved by the object in kilobytes.</td>
</tr>
<tr>
<td>Used (KB)</td>
<td>The amount of space used by the object in kilobytes.</td>
</tr>
</tbody>
</table>
Metrics
Negative numbers viewed for index space information can be caused by inaccuracies contained in the SQL Server data dictionary. Frequently, running a DBCC UPDATEUSAGE command against the database resolves the problem. However, there are bugs in SQL Server that sometimes caused erroneous numbers to the reported for the reserved space amounts used by tables and indexes.

SQL Agent Tab
The SQL Agent tab of the Database Detail includes the following sections:

<table>
<thead>
<tr>
<th>Job/Alert Detail</th>
<th>SQL Server Alert Summary</th>
<th>SQL Server Job Summary</th>
</tr>
</thead>
</table>

Job/Alert Detail
SQL Server provides the ability to submit and run jobs as well as be notified about certain SQL Server-related events. The Job/Alert Detail section displays information regarding the status of the server’s SQL Agent as well as details for all jobs and alerts that are defined to the system.

SQL Server Job Summary

<table>
<thead>
<tr>
<th>Metrics</th>
</tr>
</thead>
</table>

The SQL Server Job Summary section graphically displays the outcome of all jobs for the last two days, as well as current activity.

Metrics
You should investigate any counts noted for failed jobs.

SQL Server Alert Summary

<table>
<thead>
<tr>
<th>Metrics</th>
</tr>
</thead>
</table>

The SQL Server alert summary section graphically displays the number of times that performance and event alerts have fired.

Metrics
You should investigate any counts noted for any event.

Backups Tab
The Backups tab of the Database Detail includes the following:

<table>
<thead>
<tr>
<th>Backup Detail</th>
<th>Database Backup Summary</th>
<th>Log Summary Backup Detail</th>
</tr>
</thead>
</table>

Information | Description
---|---
Free (KB) | The amount of free space used by the object in kilobytes.
Percent Used | The percentage of space used by the object.
Backup Detail

The Backup section displays the most recent 25 backups for a selected database. The table below describes the information available on the Backups tab of the Database Detail:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Backup Start</td>
<td>The time stamp when the backup began.</td>
</tr>
<tr>
<td>Backup Finish</td>
<td>The time stamp when the backup finished.</td>
</tr>
<tr>
<td>Backup Type</td>
<td>The type of backup (FULL, INCREMENTAL, etc.)</td>
</tr>
<tr>
<td>Backup Size (KB)</td>
<td>The size of the backup, in kilobytes.</td>
</tr>
<tr>
<td>Expiration Date</td>
<td>The expiration date for the backup, if any.</td>
</tr>
</tbody>
</table>

Metrics

Although the needs of an application determine the frequency and type of backup, it is generally recommended that most dynamic databases have a solid plan in place for full and differential backups. For databases requiring point-in-time recovery, a backup plan should also include log backups.

Database Backup Summary

The Database Backup Summary section graphically displays the backup successes and failures for database backups over the last week.

Metrics

You should investigate any backup failures.

Log Backup Summary

The Log Backup Summary section graphically displays the backup successes and failures for database log backups over the last week.

Metrics

You should investigate any backup failures.

Configuration Tab

The Configuration tab of the Database Detail displays metadata regarding the hardware characteristics of the SQL Server as well as the SQL Server configuration parameters. The table below describes the information available on the Configuration tab:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>The parameter name.</td>
</tr>
<tr>
<td>Value</td>
<td>The value set for the parameter.</td>
</tr>
</tbody>
</table>
Replication Tab
The Replication tab of the Database Detail provides basic information regarding replication activities that are occurring on the monitored SQL Server.

Replication Agent Details
The Replication Agent Details section shows information regarding replication. The table below describes the information available on the Replication tab:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>The agent performing the replication work.</td>
</tr>
<tr>
<td>Publication</td>
<td>The object being published to a subscribing database.</td>
</tr>
<tr>
<td>Publisher</td>
<td>The SQL Server providing the source material.</td>
</tr>
<tr>
<td>Publisher Database</td>
<td>The database of the SQL Server providing the source material.</td>
</tr>
<tr>
<td>Status</td>
<td>The current status of the replication agent's task.</td>
</tr>
<tr>
<td>Subscriber</td>
<td>The SQL Server requesting the source material.</td>
</tr>
<tr>
<td>Subscriber Database</td>
<td>The database of the SQL Server requesting the source material.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of replication activity (push, pull, etc.)</td>
</tr>
<tr>
<td>Start Time</td>
<td>The start time for the replication task.</td>
</tr>
<tr>
<td>Duration</td>
<td>The duration of the replication task.</td>
</tr>
<tr>
<td>Last Action</td>
<td>The output message for the replication task.</td>
</tr>
</tbody>
</table>

Replication Throughput Details
The Replication Throughput Details section of the Replication tab displays:

- Replication Type
- Measure
- Instance Name
- Value

OS Page Statistics
In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Analyst lets you examine operating system metrics for the following platforms:
• AIX
• HP-UX

**NOTE:** To view processor info and swap disk info on an HP-UX box, you need to login as ROOT in the OS login.

• Linux
• Solaris
• Unix
• Windows XP and 2000

**NOTE:** The statistics available on the OS page depend on the operating system platform.

**TIP:** If you magnify a graph to change back to the original size or minimize the size, close the OS Page and then reopen it again. Performance Analyst displays the graph in its original form and size.

The OS home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Disk Analysis</th>
<th>Key Resource Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Analysis</td>
<td>Workload Analysis - Top Resource Hogs</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topic**

**OS Detail**

**Key Resource Usage - OS**

The following ratios are used on the Performance Analyst OS home page to communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Disk Time</th>
<th>Load Average</th>
<th>Processor Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paged Memory Used (Windows)</td>
<td>Swap Memory Used (AIX, HP-UX, Linux, Solaris, Unix)</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the CPU tab, right-click Processor Time. To open the I/O tab, right-click Disk Time. To open the Memory tab, right-click Paged Memory Used.
Bottleneck Analysis - OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Unit</th>
<th>Metric</th>
<th>Unit</th>
<th>Metric</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Disk Queue Length</td>
<td></td>
<td>Interrupts/Sec</td>
<td></td>
<td>Network Output Queue Length (Windows)</td>
<td></td>
</tr>
<tr>
<td>Network Queue Length (Solaris)</td>
<td></td>
<td>Page Faults/Sec</td>
<td></td>
<td>Processor Queue Length</td>
<td></td>
</tr>
<tr>
<td>Processor Speed</td>
<td>Processor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the I/O tab, right-click any Details menu item. To open the CPU tab, right-click the Item Processor Queues Length Details menu item. To open the Network tab, right-click Network Output Queues Length.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Memory Analysis - OS
The following metrics are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Unit</th>
<th>Metric</th>
<th>Unit</th>
<th>Metric</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Paged Memory (Windows)</td>
<td></td>
<td>Available Physical Memory</td>
<td></td>
<td>Available Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)</td>
<td></td>
</tr>
<tr>
<td>Total Paged Memory (Windows)</td>
<td></td>
<td>Total Physical Memory</td>
<td></td>
<td>Total Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the Memory tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Disk Analysis - OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Unit</th>
<th>Metric</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free Disk Space</td>
<td></td>
<td>Total Disk Space</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used Disk Space</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the Space tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.
Workload Analysis - Top Resource Hogs OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Number of Logins</th>
<th>Number of Processes</th>
<th>Top CPU Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top I/O Process</td>
<td>Top Memory Process</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the Processes tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.

**Processor Time**
The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

**Metrics**
If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.

**Troubleshooting**
Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

**Processor Speed**
The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

**Processor**
The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

**Disk Time**
The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

**Metrics**
You should avoid consistently seeing values for this statistic greater then 90%.

**Troubleshooting**
Add more disk drives and partition the files among all of the drives.

**Load Average**
The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.
Metrics
High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Paged Memory Used
The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

NOTE: This statistic is available for the Windows platform.

Metrics
This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.

Troubleshooting
Increase the size of page file.

Swap Memory Used
The Swap Memory Used statistic is the percentage of swap space currently in use.

Metrics
If the percentage of swap memory used is above 90%, you may be running out of memory.

Troubleshooting
Increase the size of your swap files.

Average Disk Queue Length
The Average Disk Queue Length statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Metrics
This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queue lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.

Troubleshooting
Some things you can do if you have problems with this statistic include:

- Redistribute the data on the disk with the large average disk queue to other disks.
- Upgrade to faster disk(s).
Interrupts/Sec
Interrupts/Sec is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every 10 milliseconds, creating a background of interrupt activity. This statistic shows the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Metrics
A high value indicates possible excessive hardware interrupts; justification is dependent on device activity.

Page Faults/Sec
The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics
This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue Length
The Processor Queue Length statistic is the number of threads in the processor queue.

Metrics
Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Network Output Queue Length/Network Queue Length
The Network Output Queue Length statistic is the number of threads in the processor queue.

**NOTE:** The name of this statistic depends on the platform of the operating system.
Metrics
Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Available Physical Memory
The Available Physical Memory statistic represents the amount of RAM available to all processes.

Metrics
This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Troubleshooting
Some things you can do if you have problems with this statistic include:

- Check the running processes to see if there are any memory leaks.
- Stop any services that are not required.
- Install additional RAM.

Available Paged Memory
The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

**NOTE:** This statistic is available for the Windows platform.

Metrics
If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.

Troubleshooting
Increase the size of page file.

Available Swap Memory
The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Metrics
If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.
Troubleshooting
Increase the size of swap files.

Total Physical Memory
The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Metrics
This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Total Paged Memory/Total Swap Memory
The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics
It is recommended that this be 1½ - 3 times the amount of RAM on the system.

Used Disk Space
The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Total Disk Space
Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two:

1. If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
2. You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.
Free Disk Space
The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Top Memory Process
Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics
If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Processes Overview
The Processes Overview of the OS Summary includes the following sections:

<table>
<thead>
<tr>
<th>Top CPU Process</th>
<th>Top I/O Process</th>
<th>Top Memory Process</th>
</tr>
</thead>
</table>

Top CPU Process
Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics
If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.

Troubleshooting
Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process
The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.
Number of Logins
This statistic displays the total number of logins on the server.

Number of Processes
This statistic displays the total number of processes on the server.

OS Detail
The following tabbed pages are available on the OS Detail page:

<table>
<thead>
<tr>
<th>CPU</th>
<th>I/O</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Processes</td>
<td>Space</td>
</tr>
</tbody>
</table>

CPU Tab
The CPU tab of the OS Detail includes the following sections:

<table>
<thead>
<tr>
<th>Context Switches/Sec</th>
<th>CPU Events</th>
<th>CPU Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrupts/Sec</td>
<td>Load Averages</td>
<td>Processor Queue Length</td>
</tr>
</tbody>
</table>

NOTE: The sections available on this tab depend on the platform of operating system.

CPU Utilization
The CPU Utilization section includes the following information:

- % Privileged Time
- % User Time

% Privileged Time
The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

NOTE: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

Metrics
The ideal range should be 0-40% (less than 40% indicates excessive system activity).

Troubleshooting
If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).
% User Time
The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Metrics
If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

CPU Events
The CPU Events section includes the following information:

<table>
<thead>
<tr>
<th>Interrupts/Sec</th>
<th>Context Switches/Sec</th>
<th>System Calls/Sec</th>
</tr>
</thead>
</table>

NOTE: The statistics available in this section depend on the platform of operating system.

Interrupts/Sec
The Interrupts/Sec statistic is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. It does not include deferred procedure calls (DPCs), which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every ten milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Metrics
The ideal range should be 0-5000. A number greater then 5000 indicates possible excessive hardware interrupts; justification is dependent on device activity.

Context Switches/Sec
The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Metrics
The ideal range should be between 0-10,000. GA number greater then 10,000 may indicate too many threads contending for resources.

System Calls/Sec
System calls are functions which require the services of the operating system. Usually they involve access to some data that users must not be permitted to corrupt or even change.
Troubleshooting
Try to find a good ratio between the amount of time the operating system runs (fulfilling system calls and doing process scheduling), and the amount of time the application runs. Your goal should be running 60% to 75% of the time in application mode, and 25% to 40% of the time in operating system mode. If you find that the system is spending 50% of its time in each mode, then you should investigate to determine what is wrong.

Processor Queue Length
The Processor Queue Length statistic is the number of threads in the processor queue. There is a single queue for processor time even on computers with multiple processors.

**NOTE:** For Windows systems, unlike the disk counters, this counter shows ready threads only, not threads that are running.

Metrics
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Load Averages
The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

**Metrics**
High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Processes Tab
The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>The name of the process.</td>
</tr>
<tr>
<td>User</td>
<td>The user of the process.</td>
</tr>
<tr>
<td>ID</td>
<td>The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.</td>
</tr>
<tr>
<td>CPU</td>
<td>The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.</td>
</tr>
<tr>
<td>User Mode</td>
<td>The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.</td>
</tr>
<tr>
<td>Memory WINDOWS ONLY</td>
<td>Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.</td>
</tr>
<tr>
<td>Memory (MB)</td>
<td>Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.</td>
</tr>
<tr>
<td>Memory</td>
<td>Memory is the percentage of the memory used of the total memory.</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Active Memory</td>
<td>Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.</td>
</tr>
<tr>
<td>I/O Data</td>
<td>The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>The total elapsed time, in seconds, that this process has been running.</td>
</tr>
<tr>
<td>Thread Count</td>
<td>The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.</td>
</tr>
<tr>
<td>Handle Count</td>
<td>The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.</td>
</tr>
<tr>
<td>Priority</td>
<td>The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.</td>
</tr>
<tr>
<td>Creating Proc ID</td>
<td>The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.</td>
</tr>
<tr>
<td>Page Faults/Sec</td>
<td>Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.</td>
</tr>
<tr>
<td>Page File</td>
<td>Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.</td>
</tr>
<tr>
<td>Private</td>
<td>Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.</td>
</tr>
</tbody>
</table>

**I/O Tab**

The I/O tab includes the following sections:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Disk Queue</td>
<td>Bytes per I/O Operation</td>
</tr>
<tr>
<td>Disk Transfers /Sec</td>
<td>I/O Details</td>
</tr>
<tr>
<td>Total Disk Queue</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of operating system.
I/O Details
The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk</td>
<td>The disk number assignment.</td>
</tr>
<tr>
<td>Reading (KB/s)</td>
<td>The amount of bytes read from the device.</td>
</tr>
<tr>
<td>Writing (KB/s)</td>
<td>The amount of bytes written to the device.</td>
</tr>
<tr>
<td>Disk Read Time</td>
<td>Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.</td>
</tr>
<tr>
<td>Disk Write Time</td>
<td>Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.</td>
</tr>
<tr>
<td>Disk Time</td>
<td>Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.</td>
</tr>
<tr>
<td>Avg. Read Queue</td>
<td>Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.</td>
</tr>
<tr>
<td>Avg. Write Queue</td>
<td>Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.</td>
</tr>
<tr>
<td>Disk Reads/Sec</td>
<td>Disk Reads/Sec is the rate of read operations on the disk.</td>
</tr>
<tr>
<td>Disk Writes/Sec</td>
<td>Disk Writes/Sec is the rate of write operations on the disk.</td>
</tr>
</tbody>
</table>

**NOTE:** The columns available in this table depend on the platform of operating system.

Transfer Rate
The Transfer Rate section succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading</td>
<td>The average number of bytes transferred from the disk during read operations.</td>
<td>It is measured in number of KBs per second.</td>
</tr>
<tr>
<td>Writing</td>
<td>The average number of bytes transferred to the disk during write operations.</td>
<td>It is measured in number of KBs per second.</td>
</tr>
<tr>
<td>Transferred</td>
<td>The amount of time required to transfer the data between the device's cache and the actual device media.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Bytes per I/O Operation
The Bytes per I/O Operation section of the OS Detail succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading</td>
<td>The average number of bytes transferred from the disk during read operations.</td>
<td>It is measured in number of bytes per second.</td>
</tr>
<tr>
<td>Writing</td>
<td>The average number of bytes transferred to the disk during write operations.</td>
<td>It is measured in number of bytes per second.</td>
</tr>
</tbody>
</table>
Average Disk Queue/Total Disk Queue
The Average Disk Queue section of the OS Detail page succinctly communicates the general overall performance levels of the server’s I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Read Queue</td>
<td>Average Read Queue Length is the average number of read requests that were queued for a particular disk.</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system.</td>
</tr>
<tr>
<td>Average Write Queue</td>
<td>Average Write Queue is the average number of write requests that were queued for a particular disk.</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Windows System Administrator for assistance in troubleshooting these problems.</td>
</tr>
<tr>
<td>Average Queue</td>
<td>Average time in ms. to service I/O requests which for disks includes seek, rotational latency and data transfer time).</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Unix System Administrator for assistance in troubleshooting these problems.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Troubleshooting
Contact your Windows System Administrator for assistance in troubleshooting these problems.

Disk I/O Time
The Disk I/O Time section of the OS Detail page succinctly communicates the general overall performance levels of the disk’s I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Disk Read Time</td>
<td>% Disk Read Time is the percentage of time during the sample interval that the disk is busy servicing read requests.</td>
<td>Use this in conjunction with the % Disk Write Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
<tr>
<td>% Disk Write Time</td>
<td>% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.</td>
<td>Use this in conjunction with the % Disk Read Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
<tr>
<td>% Disk Time</td>
<td>% Disk Time is the percentage of time during the sample interval that the disk is busy servicing requests.</td>
<td>Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.
Disk Transfers/Sec
The Disk Transfers/Sec section of the OS Detail page succinctly communicates the general overall performance levels of the disk transfers. The table below describes the statistics in this section.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads/Sec</td>
<td>The rate of read operations on the disk.</td>
</tr>
<tr>
<td>Writes/Sec</td>
<td>The rate of write operations on the disk.</td>
</tr>
<tr>
<td>Transfers/Sec</td>
<td>The rate of transfer operations per second.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Memory Tab
The Memory tab of the OS Detail page includes the following sections:

<table>
<thead>
<tr>
<th>Buffer Size</th>
<th>Cache Efficiency</th>
<th>Cache Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Available</td>
<td>Paging Activity</td>
<td>Page Faults</td>
</tr>
<tr>
<td>Page Replacements</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of operating system.

Paging Activity
The Paging Activity section includes the following statistics:

<table>
<thead>
<tr>
<th>Blocks Input/Sec</th>
<th>Blocks Output/Sec</th>
<th>Paged In</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paged Out</td>
<td>Pages Input/Sec</td>
<td>Pages Output/Sec</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Blocks Input/Sec
The Blocks Input/sec statistic is the number of blocks sent to a block device per second.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck:

1. Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
2. Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.

Blocks Output/Sec
The Blocks Output/sec statistic is the number of blocks received from a block device per second.)
Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck.

1. Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.

2. Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.

Paged In
The number of disk block paged into core from disk which occurs when the server is experiencing a shortage of RAM. Any nonzero value indicates excessive swap activity. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that’s reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server’s operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Troubleshooting
High values indicate the possible need to increase the amount of RAM on the system.

Paged Out
The number of disk block paged out of core from disk. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that’s reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server’s operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Pages Input/Sec
The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Metrics
This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.

- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.
Pages Output/Sec
The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Metrics
Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Page Faults/Sec
The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics
This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Memory Available
The Memory Available section of the OS Detail page succinctly communicates the general overall performance levels of the server's memory. The statistics available in this section depend on the platform of operating system. The table below describes these statistics:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Memory</td>
<td>The Physical Memory statistic is the amount of committed virtual memory, in megabytes. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk.</td>
<td>This counter displays the last observed value only and not an average.</td>
<td>N/A</td>
</tr>
<tr>
<td>Physical</td>
<td>The Physical statistic is the total physical memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Statistic</td>
<td>Description</td>
<td>Metrics</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Paged Memory</td>
<td>The Paged Memory statistic is the amount of committed virtual memory, in bytes. Used Memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.</td>
<td>This counter displays the last observed value only; it is not an average.</td>
<td>N/A</td>
</tr>
<tr>
<td>Swap Memory</td>
<td>The Swap Memory statistic is the free, unreserved swap space.</td>
<td>N/A</td>
<td>Increase the size of your swap file or add additional files if you consistently see a shortage of free, unreserved swap space.</td>
</tr>
<tr>
<td>Swap</td>
<td>The Swap statistic is the total swap memory being used.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total Physical</td>
<td>The Total Physical statistic is the total physical memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total Swap</td>
<td>The Total Swap statistic is the total swap memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Swap-ins</td>
<td>The number of memory pages paged in from swapspace.</td>
<td>N/A</td>
<td>If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.</td>
</tr>
<tr>
<td>Swap-outs</td>
<td>The number of memory pages paged in from swapspace.</td>
<td>N/A</td>
<td>If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.</td>
</tr>
<tr>
<td>Buffers</td>
<td>The Buffers statistic represents the amount of buffers used by the kernel in kilobytes.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
NOTE: The statistics available in this section depend on the platform of operating system.

Buffer Size
The Buffer Size statistic represents the amount of cache used by the kernel in kilobytes.

Cache Size
The Cache Size statistic represents the amount of cache used by the kernel in kilobytes.

Page Replacements
The following statistics are available in this section:

- Memory Freed (Pages/sec)
- Clock Algorithm Scans (Pages/sec)

Memory Freed (Pages/sec)
Freed or destroyed (Kb/s).

Clock Algorithm Scans (Pages/sec)
The VMM uses a technique known as the clock algorithm to select pages to be replaced. This technique takes advantage of a referenced bit for each page as an indication of what pages have been recently used (referenced). When the page-stealer routine is called, it cycles through the PFT, examining each page’s referenced bit.

Cache Efficiency
The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server’s memory. The following statistics are available in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Faults/Sec</td>
<td>The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.</td>
<td>This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.</td>
<td>If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.</td>
</tr>
<tr>
<td>Copy Read Hits%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pin Read Hits%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Copy Read Hits %
The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

Metrics
When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Data Map Hits %
The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Metrics
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %
The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Metrics
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %
The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Metrics
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.
Space Tab
The Space tab of the OS Detail page includes the following sections:

- Device Summary
- Device Detail

Disk Space Free
The Disk Space Free metric displays the amount of free disk space in megabytes.

Metric
None.

Disk Space Detail
The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server’s disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>The drive letter of the disk.</td>
</tr>
<tr>
<td>Local Filesystem</td>
<td>The name of the file system.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of file system.</td>
</tr>
<tr>
<td>Total Space</td>
<td>Total size of the disk/device's capacity expressed in MBs.</td>
</tr>
<tr>
<td>Used Space</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free Space</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
<tr>
<td>Capacity</td>
<td>The percentage of space used on the device.</td>
</tr>
<tr>
<td>Mounted On</td>
<td>The mount point of the device.</td>
</tr>
</tbody>
</table>

Device Summary
The Device Summary section of the Space tab displays a bar chart for all devices. The Device Summary section succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
</tbody>
</table>
Device Detail
The Device Detail section of the Space tab succinctly communicates the general overall performance levels of the server’s disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>The drive letter of the disk.</td>
</tr>
<tr>
<td>Local Filesystem</td>
<td>The name of the file system.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of file system.</td>
</tr>
<tr>
<td>Total Space</td>
<td>Total size of the disk/device's capacity expressed in MBs.</td>
</tr>
<tr>
<td>Used Space</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free Space</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
<tr>
<td>Capacity</td>
<td>The percentage of space used on the device.</td>
</tr>
<tr>
<td>Mounted On</td>
<td>The mount point of the device.</td>
</tr>
</tbody>
</table>

NOTE: The statistics available in this section depend on the platform of operating system.

Network Tab
The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server’s networking. The Network tab of the OS Detail page includes the following sections:

<table>
<thead>
<tr>
<th>Network Details</th>
<th>Packet Collisions</th>
<th>Packet Discards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Errors</td>
<td>Transmission Rate</td>
<td>Transmission Rate (Bytes)</td>
</tr>
<tr>
<td>Transmission Queue Length</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE: The sections available on this tab depend on the platform of operating system.

Network Details
The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Interface</td>
<td>The name of network interface.</td>
</tr>
<tr>
<td>INET Address/Address</td>
<td>The IP address assigned to the network interface.</td>
</tr>
<tr>
<td>Pkts Sent/Sec</td>
<td>The number of packets sent per second.</td>
</tr>
<tr>
<td>Pkts Received/Sec</td>
<td>The number of packets received per second.</td>
</tr>
<tr>
<td>Sent (KB/Sec)</td>
<td>The number of bytes sent per second.</td>
</tr>
<tr>
<td>Received (KB/Sec)</td>
<td>The number of bytes received per second.</td>
</tr>
<tr>
<td>Out Pkts Discarded</td>
<td>The number of outbound packets discarded.</td>
</tr>
<tr>
<td>In Pkts Discarded</td>
<td>The number of inbound packets discarded.</td>
</tr>
<tr>
<td>Out Pkt Errors</td>
<td>The number of outbound packet errors.</td>
</tr>
<tr>
<td>In Pkt Errors</td>
<td>The number of inbound packet errors.</td>
</tr>
<tr>
<td>Queue Length</td>
<td>The queue length.</td>
</tr>
</tbody>
</table>
Transmission Rate
The Transmission Rate section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

- **Packets Received/Sec**
- **Packets Sent/Sec**

Packets Sent/Sec
The Packets Sent/Sec statistic is the number of packets sent over each network adapter per second.

Packets Received/Sec
The Packets Received/Sec statistic is the number of packets received over each network adapter per second.

Transmission Rate (Bytes)
The Transmission Rate (Bytes) section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

- **Received (KB/Sec)**
- **Sent (KB/Sec)**

Sent (KB/Sec)
The Sent (KB/Sec) statistic is the rate at which bytes are sent over each network adapter including framing characters.

Received (KB/Sec)
The Received (KB/Sec) statistic is the rate at which bytes are received over each network adapter, including framing characters.

Transmission Queue Length
The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue.

Packet Collisions
A collision occurs when two devices on the network sense that the network is idle and end up trying to send data at the same time. Some collisions are normal on a repeated network, but excess collisions can cause serious performance problems.

Troubleshooting
Contact your network administrator if there is high rate of persistent Packet Collisions.
Packet Discards
The Packet Discards section of the Network tab succinctly communicates the packet discard rate. The following statistics are available in this section:

| Inbound Packets Discarded | Outbound Packets Discarded | Packet Discards |

Outbound Packets Discarded
The Outbound Packets Discarded statistic is the number of outbound packets that were discarded even though no errors had been detected to prevent from being transmitted.

Troubleshooting
A possible reason for discarding such a packet could be to free up buffer space.

Inbound Packets Discarded
Received Packets Discarded is the number of received packets that were discarded even though no errors had been detected to prevent from being transmitted.

Troubleshooting
A possible reason for discarding such a packet could be to free up buffer space.

Packet Discards
The Packet Discards statistic represents the number of network packets discarded.

Troubleshooting
Contact your network administrator if there is high rate of persistent packet discards.

Packet Errors
The Packet Errors section of the Network tab succinctly communicates the packet error rate. The following statistics are available in this section:

- Inbound Packet Errors
- Outbound Packet Errors

Outbound Packet Errors
The Outbound Packet Errors statistic is the outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Inbound Packet Errors
The Packets Received Errors statistic is the received packets that contained errors preventing them from being deliverable to a higher-layer protocol.
Top SQL Page Statistics

- Metrics

One or two bad queries can cause a lot of trouble for the remaining sessions in a database, and so can one or two bad queries. It is important to find them before they get into a production system, but sometimes a few sneak through.

By applying custom filters and performance-related thresholds, the Top SQL view locates inefficient SQL. By applying filters to certain I/O and statistical counters, you hope to isolate queries that far exceed their nearest competitors in the same area (like disk reads). When you find them, you should reduce the number of sorts a query performs. Or, for a query that returns only a few records, you should try to minimize the number of rows a query processes.

The Top SQL view displays requested SQL for SQL Server datasources. The following tabs are available on the Top SQL page:

<table>
<thead>
<tr>
<th>Summary</th>
<th>Criteria</th>
<th>SQL Details</th>
</tr>
</thead>
</table>

Summary Tab

The Summary tab of the Top SQL page includes the following sections:

- Top SQL Statements
- Top SQL Details

Top SQL Statements - SQL Server

A lot of a database’s overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what ‘bad SQL’ is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional’s satisfaction. The table below describes some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Response (Elapsed) Time</td>
<td>This is how much time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>This is how much CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.</td>
</tr>
<tr>
<td>Physical I/O</td>
<td>Often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user’s request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at statements’ logical I/O as well.</td>
</tr>
<tr>
<td>Logical I/O</td>
<td>This is a measure of how many memory reads the query took to satisfy the user’s request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.</td>
</tr>
<tr>
<td>Repetition</td>
<td>This is a measure of how often the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute may still be causing a headache on your system if it is executed erroneously over and over.</td>
</tr>
</tbody>
</table>
There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

Fortunately, Microsoft records some of these statistics and Performance Analyst computes the rest, which makes tracking the SQL that has been submitted against a SQL Server much easier.

**Metrics**

When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer:

1. What has been the worst SQL that’s historically been run in my database?
2. What is the worst SQL that’s running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient Transact SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

**Top SQL Details - SQL Server**

A lot of a database’s overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what ‘bad SQL’ is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional’s satisfaction. The Top SQL Details Section displays the top SQL statements across various performance categories. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top For</td>
<td>Indicates the performance category for the SQL statement.</td>
</tr>
<tr>
<td>Usage</td>
<td>This displays the percent usage for this SQL statement in this performance</td>
</tr>
<tr>
<td></td>
<td>category.</td>
</tr>
<tr>
<td>SQL text</td>
<td>A snippet of the actual SQL statement or code object (procedure, etc.)</td>
</tr>
<tr>
<td>Total Reads</td>
<td>The total physical reads caused by the statement since monitoring began.</td>
</tr>
<tr>
<td>Total Writes</td>
<td>The total physical writes caused by the statement since monitoring began.</td>
</tr>
<tr>
<td>Executions</td>
<td>The total number of times the statement or code object has been executed</td>
</tr>
<tr>
<td></td>
<td>since monitoring began.</td>
</tr>
<tr>
<td>Avg CPU Time</td>
<td>The average CPU time used by the statement each time it is executed.</td>
</tr>
<tr>
<td>Avg Elapsed Time</td>
<td>The average elapsed response time experienced each time the statement is</td>
</tr>
<tr>
<td></td>
<td>executed.</td>
</tr>
</tbody>
</table>

**Metrics**

When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer:

1. What has been the worst SQL that’s historically been run in my database?
2. What is the worst SQL that’s running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient Transact SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.
Criteria Tab
The Top SQL view displays all SQL running or executed in the current datasource. You can use the lists and boxes to reorder the display according to your specific needs. The Criteria tab of the Top SQL page includes the following sections that you can mix and match to give you the information you want to find.

- Demographic Criteria
- Statistical Criteria
- Sort Criteria

SQL Details Tab
The SQL Details tab of the Top SQL page includes the following sections:

- Top SQL Results
- Selected SQL Text

Top SQL Results
Overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements or stored procedures can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what ‘bad SQL’ is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional’s satisfaction. The Top SQL results grid displays the top SQL statements across various performance categories. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Text</td>
<td>A snippet of the full SQL statement. Clicking the statement will cause the full statement to be presented in the Selected SQL Text grid.</td>
</tr>
<tr>
<td>Database</td>
<td>The database where the statement or code object’s execution originated.</td>
</tr>
<tr>
<td>Avg Reads</td>
<td>The average physical reads caused by the statement each time it is executed.</td>
</tr>
<tr>
<td>Total Reads</td>
<td>The total physical reads caused by the statement since monitoring began.</td>
</tr>
<tr>
<td>Avg Writes</td>
<td>The average physical writes caused by the statement each time it is executed.</td>
</tr>
<tr>
<td>Total Writes</td>
<td>The total physical writes caused by the statement since monitoring began.</td>
</tr>
<tr>
<td>Executions</td>
<td>The total number of times the statement or code object has been executed since monitoring began.</td>
</tr>
<tr>
<td>Avg Elapsed Time</td>
<td>The average elapsed response time experienced each time the statement is executed.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>The total elapsed response time over all executions.</td>
</tr>
<tr>
<td>Avg CPU Time</td>
<td>The average CPU time used by the statement each time it is executed.</td>
</tr>
<tr>
<td>Total CPU Time</td>
<td>The total CPU time over all executions.</td>
</tr>
<tr>
<td>First Start Time</td>
<td>The date/time when the statement was first executed.</td>
</tr>
</tbody>
</table>

Selected SQL Text
The Selected SQL Text window allows you to select any of the Top SQL result rows and get the complete SQL text. From here you can open an Explain Plan or Tune the SQL.
Performance Analyst for Sybase Tutorial

Embarcadero Performance Analyst provides key insight into the complex world of database performance. Running inside the Embarcadero DBArtisan console, or standalone in a Web browser, Performance Analyst helps you pinpoint performance problems that threaten a database’s availability or response times. The powerful analysis facilities will satisfy advanced experts, yet novices will find Performance Analyst’s interface easy to use and the expert help a welcome aid to troubleshooting difficult performance issues.

About This Performance Analyst for Sybase Tutorial

This tutorial is intended to help you get started with Embarcadero Performance Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Performance Analyst. You will have learned how to accurately monitor your database using the most popular performance methodologies in use today. You will know how to drill down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. Finally, you’ll understand how to fix performance problems using the integration that exists between Performance Analyst and Embarcadero DBArtisan.

This tutorial is divided into 7 sections.

**Session 1: Getting Started with Performance Analyst**

**Session 2: Using Ratio-based Performance Analysis**

**Session 3: Using Bottleneck/Wait-based Performance Analysis**

**Session 4: Using Workload Performance Analysis**

**Session 5: Using Operating System Performance Analysis**

**Session 6: Identifying and Getting Help with Performance Problems**

**Session 7: Fixing Identified Problems Using Performance Analyst for Sybase**

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief guide.

For more information, see Additional Evaluation Resources.

---

Session 1: Getting Started with Performance Analyst for Sybase

**Download and Install**

You can obtain the latest version of Performance Analyst by downloading the DBArtisan software from the Embarcadero Web site at [http://www.embarcadero.com/downloads/downloaddbartisan.asp](http://www.embarcadero.com/downloads/downloaddbartisan.asp). Performance Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Performance Analyst (although Performance Analyst can also run in standalone fashion within a Microsoft Internet Explorer Web browser).

Click **Download**, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.
Overview of Performance Analyst for Sybase

When DBArtisan is connected to a Sybase database, you can open Performance Analyst two ways:

1. On the Analyst toolbar click the Performance Analyst icon.

OR

2. On Utilities menu, select Performance Analyst.

For Performance Analyst to work, several procedures need to be installed by the SA account. When Performance Analyst is first invoked, it will check for the presence of this package and if it has not been installed, the product will prompt you to install it. Once created, any account with access to these procedures can monitor databases using Performance Analyst. No other objects need to be installed on a monitored database.

Performance Analyst for Sybase Product Navigation

Navigating through Performance Analyst is easily accomplished through two main mechanisms:

1. The main Performance Analyst toolbar contains buttons that invoke the major performance analysis views within the product as well as frequently used actions like Refresh and Performance Threshold Customizations.

2. Positioning the mouse pointer on any area within Performance Analyst and clicking the right mouse button will bring up a navigation menu that allows you to easily move between any of the main Performance Analyst home pages and drill down views. The right mouse menu also contains context sensitive options like drilling down further into a particular statistic, copying the display to a text file, and more.

Session 2: Using Ratio-Based Performance Analysis in Performance Analyst for Sybase

Ratio-based analysis has been around for many years, and used to be the only technique database administrators used when they were called on to diagnose the cause of a database slowdown. Performance ratios are very good roll-up mechanisms for busy DBAs, making possible the analysis-at-a-glance approach, however ratios must be computed correctly for them to be meaningful. Key ratios in Performance Analyst are calculated using delta statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. Performance Analyst makes it easy to see the ratios that are computed with delta statistics vs. cumulative. A triangle symbol denotes a delta-based statistic, while square symbols represent a cumulative-based measure.

Ratio-based analysis is available for all performance categories (memory, etc.) where ratios can assist in determining the performance level of a database. For example, if you would like to see memory-specific ratios, on the Performance Analyst toolbar click Memory. Performance Analyst displays the Memory home page that showcases all important memory-related ratios.

Obtaining Detail Information for a Ratio Using Performance Analyst for Sybase

A ratio gives you a global picture for a particular performance measure. However, you often need more information regarding a statistical measure so you can make an informed decision. Performance Analyst has easy-to-use drill down features that make obtaining detailed information for various ratios a breeze.

For example, to view more information for the Data Cache Hit Ratio statistic, do the following:

1. On either the Main Home page or the Memory Home page, locate the Data Cache Hit Ratio statistic.

2. Position the pointer over the statistic.

   The pointer changes to a hand symbol.
3 Double-click the left mouse button.

   Performance Analyst opens the Data Cache Tab of the memory drill down view.

   OR

4 Right-click the statistic, select Details.

   OR

5 Select Navigate to, Memory, Data Cache.

   Performance Analyst opens the Data Cache Tab of the memory drill-down view.

Session 3: Using Bottleneck/Wait-Based Performance Analysis in Performance Analyst for Sybase

When a Sybase database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things, or it can be an indicator that a database bottleneck exists. Bottleneck analysis is a valid method of measuring performance because it helps you track where a database and user sessions have been spending their time. If latch contention or heavy table-scan activity has been dragging a database’s performance down, you can use bottleneck analysis to confirm the actual root cause.

Performance Analyst uses three broad areas of metrics to form its bottleneck analysis. The first are contentions and conflicts as reported in Sybase’s monitoring `sp_sysmon` procedure, as well as overall lock statistics. The second is the Sybase wait event interface (available only in Sybase 12.5.03 and higher), which displays where the database and user sessions have been bottlenecked. A third area of bottleneck information has to do with space related bottlenecks like databases and transaction logs that are running low on available free space.

Performance Analyst prominently displays bottleneck analysis on every performance home page. The Main home page, displays a summary of all key bottlenecks threatening the database. You can also see specific bottlenecks for memory, I/O, space, objects, and users by navigating to each home page and viewing the Bottleneck Analysis section for that home page. For example, to see bottlenecks that specifically relate to I/O, do the following

1 On the toolbar, click the I/O button.

2 Locate the Bottleneck Analysis section at the lower-left portion of the screen.

   The I/O bottleneck analysis section shows the I/O related headaches, along with the most active devices.

Obtaining Detail on System Bottlenecks Using Performance Analyst for Sybase

You can easily obtain more information regarding global bottlenecks that are plaguing a database by drilling down into a particular bottleneck metric. For example, if you would like to see more data regarding the current top system bottlenecks, do the following:

1 Right-click anywhere in Performance Analyst, point to Users, and then click System Waits.

   Performance Analyst opens the System Waits view.

The System Waits view (available in Sybase 12.5.03 and higher with the MDA monitoring views installed) displays critical details regarding all the wait events that Sybase has recorded. You can easily see what events that occurred the most often as well as the events that have accumulated the most wait time.

   **TIP:** If you would like to see an expanded view of the System Waits grid shown on the top of the System Waits view, click the Expand button located at the top right corner of the section. All section views in Performance Analyst can be maximized/minimized using this control.
Viewing Detail for Session Bottlenecks Using Performance Analyst for Sybase

Not only can you obtain detail information for system bottlenecks, but you can also use Performance Analyst to get drill down information regarding session-level bottlenecks. There are several layers to session-level details you can view.

To see more information regarding session-level waits, do the following:

1. Right-click anywhere in Performance Analyst, point to Users, and then click Session Waits.

Performance Analyst opens the Session Waits drill-down view.

When you are in the Session Waits drill down view, to drill further into a session's details, do the following:

1. In the Historical Waits or Current Waits grid, select a session that appears to be experiencing critical waits.
2. Double-click the session or right-click, and then select Details.

Performance Analyst opens a drill-down view that shows key statistics for the selected session, including its current work activities, wait events, and SQL statements.

Session 4: Using Workload Performance Analysis in Performance Analyst for Sybase

Key ratios give you a global perspective on database activity. Bottleneck analysis gives you insight into things that are holding up user activity and throughput. But another technique is necessary if a database professional is to really get a handle on what’s occurring inside a badly performing database.

Workload analysis involves the investigation of two critical areas of a database's performance:

- Session resource consumption and activity
- SQL execution (if available)

Without looking at these two key performance categories, you can miss a lot of what could be responsible for perceived performance problems. When performance on a database takes a sudden nosedive, it is not uncommon to find one or two sessions that are causing the bulk of the workload. Finding these processes can be easily accomplished by viewing session metadata coupled with resource consumption and statistical execution statistics.

Pinpointing Top Resource Consumers

Performance Analyst identifies top resource consumers in a couple of different ways. On the main home page, Performance Analyst highlights the top resource sessions across I/O, memory usage, and CPU consumption. The percentage used across all statistical categories are displayed so a DBA can immediately pinpoint a session that is using all or most of a particular resource. The top resource sessions are also displayed on the memory, I/O, and other home pages as well with a listing for the top sessions for that category (for example, the top memory users appear on the memory home page, etc.)

Getting Workload Details

Performance Analyst offers several details on top resource sessions in addition to the summaries provided on each home page. A detailed and flexible top sessions view lets you view a summary of all the major resource sessions on the database as well as every critical detail regarding each session connected to the database.
Further drill downs are available that highlight specific details regarding each session's activities and submitted SQL. To see detailed information regarding a session identified in one of Performance Analyst's Workload Analysis sections, do the following:

1. In a **Workload Analysis** display, position the pointer over one of the processes.
2. Double-click the process or right-click, and then select **Details**.

If you want to see all key workload analysis metadata for sessions, do the following:

1. On the toolbar, click the **Top Sessions** button.

   OR

2. Right-click anywhere in Performance Analyst, point to **Users**, and then click **Session Waits**.

**Session 5: Using Operating System Performance Analysis in Performance Analyst for Sybase**

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O, or Memory), a slow CPU, insufficient I/O devices, and insufficient memory. Performance Analyst allows you to examine the relevant operating system metrics for Windows Platforms (NT, 2000, XP, 2003) as they relate to Sybase.

To view the operating system metrics of the machine hosting the Sybase server you are monitoring, do the following:

1. Click the **OS** Performance Analyst toolbar button.

   OR

2. Navigate to the OS Home Page by going to **OS**, and then **Home**.

   **NOTE:** The first time you invoke the OS feature you will be prompted for an operating system login. Please specify the correct operating system, host name, username and password, and protocol.

The Operating System home page has been designed in a similar fashion as the datasource home page. On this page you can see the pertinent operating system metrics displayed in five quadrants broken down into the following categories:

- Resource Usage
- Memory Analysis
- Disk Analysis
- Workload Analysis
- Bottleneck Analysis
Obtaining Details on the Operating System Using Performance Analyst for Sybase

You can easily obtain more information regarding operating system problems that are degrading database performance by drilling down into a particular analysis quadrant. For example if you’d like to see more information regarding the processes running on the host machine, do the following:

1. Navigate to the Performance Analyst Operating System home page.
2. Double-click the Top Memory Process in the Workload Analysis quadrant.
   - Performance Analyst opens the Processes detail view.
   
   OR

3. Right-click anywhere in Performance Analyst and select the Navigate to, OS, and then Processes.

The Processes detail tab shows all the processes on the host machine. From this view you can easily determine which processes are showing high marks in I/O or CPU, for example, by simply double clicking on the column headers and sorting the processes in ascending or descending order.


You do not have to be an advanced expert in performance tuning to spot trouble in a database if you use Performance Analyst. There are a number of mechanisms in the tool that help you quickly locate areas of concern within a database.

Using the Alarm Log in Performance Analyst for Sybase

Performance Analyst has a central and flexible alarm log that lists every performance violation found in all monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst lists all alarms for each database you are monitoring in one central log so you can see every problem in one place. To view the Performance Analyst Alarm log, do the following:

1. On the toolbar, click the Alarms button.

Once in the Alarm log, you can perform a number of actions including:

- Selecting an alarm and viewing the application page that shows details regarding the metric in violation.
- Filtering the types of alarms shown through the Options dialog.
- Clearing all current alarms.
- Copying or saving the Alarm log into a different format.

To select any of these options, do the following:

1. Position the pointer over any area of the Alarm Log.
2. Right-click and in the shortcut menu, select an option.

Recognizing Performance Violations

Performance Analyst makes it easy to visually pinpoint performance violations in a database. Any statistic that does not conform to your predefined performance standards is colorized in any view that it appears so you can easily see potential troublespots.
The colors used to highlight a particular performance violation depend on the severity of the problem. Performance Analyst uses several layers of performance standards/threshold violations with each layer being represented by a different color. You can control everything regarding threshold violations, including how thresholds are determined, their color representation in Performance Analyst’s display, and more by using the Threshold Template feature of Performance Analyst.

How can Performance Analyst inform you of performance violations in your database if you minimize the product to your desktop and are working on other things? Even if Performance Analyst is not right in front of you, it can still let you know that a problem has been found in your database through the systray alarm icon. When Performance Analyst detects a problem in one of your databases, one of the things the product does is enable an alarm icon in your workstation’s systray area that communicates the presence of alarms in your database. So even if the Performance Analyst interface is not in sight, the product can let you know that something in your database requires attention.

Creating Threshold Templates

How does Performance Analyst know what to signal as a problem in a database? It uses predefined performance standards that you define and enforce. All of this is accomplished by using Performance Analyst’s Threshold templates. A template is a set of performance metrics and their thresholds that can be reused to enforce performance standards across any number of databases. Embarcadero provides a default performance template that contains nearly all of the most-watched performance statistics so you can begin monitoring with confidence right out of the box. However, you can create your own Threshold templates so you only are alerted to performance violations that are important to you and your particular environment.

To view or create Threshold templates, do the following:

1. On the toolbar, click the Thresholds button.

   Performance Analyst opens the Threshold Template Properties dialog box.

The Threshold Template Properties dialog box lets you select the options to add or edit templates, as well as other maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your own needs. You can control what statistics are sampled for performance violations as well as the levels of severity you would like reported. Once you are satisfied with your template, you can save it and then apply it to one or more datasources in one action to immediately begin enforcing your performance standards.

Getting Help with Performance Problems

As databases become more complex, so do the problems that plague them. Knowing what to do for a particular performance problem can be hard in some cases, but Performance Analyst helps by providing a complete set of expert help that you can use as a guide when performance problems occur. To obtain expert help for any statistic shown in Performance Analyst, do the following:

1. Position the pointer over a statistic that you need help.

2. Right-click and select Help.

The Expert Help gives you a general description of the statistic, the range of values that are considered optimal, and some suggestions on how to fix a problem situation. Expert help can also be invoked for any of the Ratio, Bottleneck, SQL, or Workload Analysis sections shown on every home page by clicking on the help icon for that section, which is located at the top right corner of the section.
Session 7: Fixing Identified Problems
Finding performance problems is one thing, but fixing them is another. Many performance monitors point out a particular problem in a database, but then offer no way to actually correct the situation. Such is not the case with Performance Analyst. Because of its tight integration with DBArtisan, certain problems found in Performance Analyst can be visually corrected by DBArtisan.

A Space Problem Correction Example
If a Sybase Server tablescape is about to run out of free space, Performance Analyst can issue an alarm that a space shortage exists. Once the space problem has been identified, do the following:

1. On the Space home page toolbar, click the Space button.
2. In the Space Analyst grid, right-click a tablescape, and then select Manage with DBArtisan.

DBArtisan opens an editor where you can perform corrective actions, such as enabling Autoextend for the tablescape, resizing the tablescape’s existing datafile(s), or adding a new datafile for the tablescape.
Performance Analyst for Sybase ASE Expert Guide

This section includes expert help for all Sybase ASE categories and statistics in the Embarcadero Performance Analyst views and pages. This help is divided into the following sections:

<table>
<thead>
<tr>
<th>Home View</th>
<th>Memory Statistics</th>
<th>I/O Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Statistics</td>
<td>Database Statistics</td>
<td>Users Statistics</td>
</tr>
<tr>
<td>OS Page Statistics</td>
<td>Top SQL Statistics</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** Sybase 12.5.0.3 users should make sure their MDA tables are properly configured so they can see all available statistics. Please refer to the instructions for enabling MDA tables.

Enabling MDA Tables

In ASE 12.5.0.3, a new feature called “MDA tables” was introduced. These proxy tables are mapped to RPCs which provide access to low-level monitoring information about the server.


DBArtisan’s Performance Analyst for Sybase requires these tables be installed to enable several statistics, for example the Top SQL view.

Please follow the instructions below to install and configure these tables.

1. Ensure the ‘enable cis” configuration parameter is set to 1.
   
   ```sql
   sp_configure 'enable cis'
   go
   ```

2. Add the ‘loopback’ server name alias.
   
   ```sql
   if not exists (select 1 from master..sysservers where srvname = "loopback")
   BEGIN
   exec sp_addserver "loopback", null, <enter your server name>
   END
   ```

3. Install the MDA tables.
   
   The script can be found in the following location:
   
   `$SYBASE/$SYBASE_ASE/scripts/installmontables`

4. Assign the ‘mon_role’ to logins allowed MDA access.

   **NOTE:** The ‘sa’ login requires this grant as well.

   ```sql
   use master
   go
   grant role mon_role to sa
   go
   ```
Enable the following configuration parameters.

- `exec sp_configure "enable monitoring", 1`
- `exec sp_configure "sql text pipe active", 1`
- `exec sp_configure "sql text pipe max messages", 2000`
- `exec sp_configure "plan text pipe active", 1`
- `exec sp_configure "plan text pipe max messages", 2000`
- `exec sp_configure "statement pipe active", 1`
- `exec sp_configure "statement pipe max messages", 2000`
- `exec sp_configure "errorlog pipe active", 1`
- `exec sp_configure "errorlog pipe max messages", 2000`
- `exec sp_configure "deadlock pipe active", 1`
- `exec sp_configure "deadlock pipe max messages", 2000`
- `exec sp_configure "wait event timing", 1`
- `exec sp_configure "process wait events", 1`
- `exec sp_configure "object lockwait timing", 1`
- `exec sp_configure "SQL batch capture", 1`
- `exec sp_configure "statement statistics active", 1`
- `exec sp_configure "per object statistics active", 1`
- `exec sp_configure "max SQL text monitored", 2000`

Home Page Statistics - Sybase
The Embarcadero Performance Analyst Home view lets you review availability and overall performance of all monitored databases from a single window. Statistics on the Home view are organized into the following categories:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key Ratio Analysis</th>
<th>I/O Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Analysis</td>
<td>Workload Analysis</td>
<td></td>
</tr>
</tbody>
</table>

Key Ratio Analysis - Home
Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.
While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the master.sysperfinfo performance table. This table maintains a count of various statistics for the server and particular databases since the server was brought up. For servers that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA might be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Home page to succinctly communicate the general overall performance levels of the monitored database:

<table>
<thead>
<tr>
<th>Data Cache Hit Ratio</th>
<th>Procedure Plan Hit Ratio</th>
<th>Large I/O Hit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean Buffer Grab Rate</td>
<td>CPU Busy</td>
<td>I/O Busy</td>
</tr>
<tr>
<td>Server Idle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Data Cache Hit Ratio

- **Metrics**

- **Troubleshooting**

Data read from memory produces user response times many times faster than when that same data is read from disk. The Sybase data cache assists with keeping physical I/Os to an absolute minimum.

The data cache hit ratio is an excellent indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. The table below describes the three key counters in Sybase used to arrive at this statistic:

<table>
<thead>
<tr>
<th>Counter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGICAL READS</td>
<td>Data read from memory for user requests.</td>
</tr>
<tr>
<td>PAGES PER I/O</td>
<td>The number of pages retrieved in a single I/O operation.</td>
</tr>
<tr>
<td>PHYSICAL READS</td>
<td>Data read physically from disk.</td>
</tr>
</tbody>
</table>

**NOTE:** This statistic is also available on the Memory home page.

### Metrics

To help ensure excellent performance, keep your cache hit rate in the neighborhood of 90% or greater. Lower amounts can be okay for user ad hoc databases where sporadic, large table scan operations occur. However, anything below this general threshold for normal databases can require tuning attention, and the adjustment of the Sybase memory tuning parameters.

If you are using named caches, you can drill down into the cache hit rates for each named cache. This helps you understand which objects/operations are depressing the overall cache hit rate for the server.
Troubleshooting
If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter or reduce the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when reducing the procedure cache, as this could reduce performance in the server as it relates to reading procedures in from disk.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the data cache (and any named caches), but in 12.5, if you want to reduce the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the data cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary and examination of unnecessary large table scan operations. Also, keep a careful eye on the actual machine’s memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Procedure Plan Hit Ratio

- Metrics
- Troubleshooting

The Sybase procedure cache is used to hold the definitions and query plans of stored procedures and triggers. It is used for short-term memory needs like statistics and query plans for parallel queries. When a user executes a stored procedure, Sybase looks in the procedure cache for a query plan to use. If a query plan is available, Sybase places it on the most recently used (MRU) end of the memory chain and the procedure begins to execute. If no execution plan is in memory, or if all copies of the plan are currently being used, the query tree for the procedure is read in again from the data dictionary, optimized, put on the MRU end of the chain, and executed. Note that other operations, like CREATE INDEX, can also use the procedure cache even when no procedure is referenced.

The more often that a procedure's plan and definition can be referenced in memory, the better the procedure execution time.

**NOTE:** This statistic is also available on the Memory home page.

Metrics
A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for procedure code reference. Note that when a database is first started, the procedure cache hit rate is not at an optimal level because all code being used is relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

You can drill down into the procedure cache to view the procedures currently in memory along with how much memory they are consuming.

If there is not enough memory to load a requested procedure, or the maximum number of compiled objects is already in use, Sybase returns an error (normally a 701).

Troubleshooting
If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). You should be careful when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed.
For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase server, but in 12.5, if you want to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the procedure cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine’s memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Large I/O Hit Rate

- **Metrics**

- **Troubleshooting**

Large I/O can be enabled by splitting the default or any named cache into pools. By default, Sybase performs I/O operations based on a 2-KB page size. For queries where pages are stored and accessed in a sequential manner, it is possible to read many more data pages in a single I/O operation. Large I/O can greatly reduce disk access time when the right situations exist. Operations that routinely perform large table scans, access image or text data, do bulk copies, scan the leaf level of nonclustered indexes, or initiate DBCC tasks can benefit from large I/O.

If large I/O has been configured and is being used, you should observe a high percentage of hits (the number of times large I/O could be performed vs. the number of times large I/O requests were denied by the server). If large I/O is not configured, no large I/O activity should be present.

**Metrics**

As you might expect, if large I/O is in use, a high hit rate is desirable. You should strive for a hit ratio between 90-100%, with 90% being a good performance benchmark.

**Troubleshooting**

If large I/O is configured, but you see a low hit rate, you should configure more caches for large I/O use.

Clean Buffer Grab Rate

- **Metrics**

- **Troubleshooting**

As information is requested from users, buffers are moved into and out of the Sybase data cache. Pages are also modified in the cache (termed dirty buffers) and need to be written out to disk. If Sybase has to wait for a dirty buffer to be written out to disk before a requested buffer is placed into the cache, performance can suffer.

The clean buffer grab rate represents the percentage of time clean buffers were found and referenced in the cache as opposed to Sybase finding dirty buffers.

**Metrics**

Ideally, the clean buffer grab rate should stay at or near 100%.

**Troubleshooting**

Seeing a poor clean buffer grab rate for either the default or named caches could indicate that the cache size is too small. You can look into adjusting the total memory configuration parameter higher. Keep a careful eye on the actual machine’s memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.
CPU Busy

- **Metrics**

The CPU Busy statistic represents the percent of the sample interval during which the Adaptive Server was busy.

**Troubleshooting**

If you find that your CPU Busy value is consistently above 70%, consider adding additional hardware.

I/O Busy

- **Metrics**

The I/O Busy statistic represents the number of clock ticks in the sample interval during which the user task performed I/O operations.

**Metrics**

High numbers indicate an I/O-intensive process. If idle time is also high, the application could be I/O bound.

**Troubleshooting**

To improve throughput:

- Assign a higher priority to the application.
- Bind the application to a lightly loaded engine or engine group.
- Partition the application's data onto multiple devices.

Server Idle

- **Metrics**

The Server Idle statistic represents the number of clock ticks in the sample interval during which the user task was idle.

**Metrics**

If you see high numbers for idle time and I/O Busy, the application could be I/O-bound.

**Troubleshooting**

To improve throughput:

- Assign a higher priority to the application.
- Bind the application to a lightly loaded engine or engine group.
- Partition the application's data onto multiple devices.
Bottleneck Analysis - Home

When Sybase Adaptive Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem. The Performance Analyst Home page identifies the top system and session waits that might be currently a cause of lessened performance.

When using bottleneck analysis, a DBA cannot rely only on the information contained in the wait events that Sybase Adaptive Server provides. For example, a database might be suspect or offline. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Home page displays space-related bottlenecks as well as other sources of contention/bottlenecks that might be dragging down the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

<table>
<thead>
<tr>
<th>Device I/O Contention</th>
<th>Modify Conflicts</th>
<th>Deadlocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suspect Databases</td>
<td>Databases Low on Space</td>
<td>Logs Low on Space</td>
</tr>
<tr>
<td>Tempdb Percent Full</td>
<td>Task Context Switches</td>
<td>Group Commit Sleeps</td>
</tr>
<tr>
<td>Blocked Connections</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Deadlocks

- **Metrics**
- **Troubleshooting**

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process’ page or object. Each waits for the other to release the necessary lock. Sybase constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates the transaction. The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process’ page or object. Each waits for the other to release the necessary lock. Sybase constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates it (the transaction). The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

The deadlock statistic displays the number of current deadlocks in a Sybase server.

**Metrics**

A nonzero number should alert you to the fact that application conflicts are likely being experienced by your user community.
Troubleshooting
Well-designed applications can minimize deadlocks by always acquiring locks in the same order. You should always do updates to multiple tables in the same order.

Once Sybase discovers a deadlock, it takes action and remedies the situation. Embarcadero Performance Analyst makes it easier to discover how prevalent deadlock activity is on a system; preventing deadlocks from occurring in the first place is more difficult.

Those responsible for writing systems can minimize deadlocks by ensuring that applications acquire locks in the same order. Likewise, you should always do updates and other DML that act on multiple tables in the same order.

You can also shrink the amount of time that Sybase waits to check for deadlocks by modifying the deadlock checking period configuration parameter.

Suspect Databases

• Metrics

Inspect databases

It can be hard to recover quickly from a suspect database. There are times when the cache of the database is suspect and not the actual database itself. If you stop and start the Sybase server, you can verify if the cache is the culprit. If the database itself is damaged, there could be a true recovery situation.

• Metrics

None.

Troubleshooting

The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You then need to re-create the database and perform a recovery operation using the most recent database dump.

Device I/O Contention

• Metrics

• Troubleshooting

Device I/O Contention reflects the number of times a task or process was put to sleep while waiting for a semaphore for a particular database device.

When a task or process involves physical I/O, Sybase first fills out the block I/O structure and links it to a per engine I/O queue. If two or more Sybase engines request an I/O structure from the same device at the exact same time, the server puts one of them to sleep where it waits for the semaphore it needs.

• Metrics

None.

Troubleshooting

If consistent high numbers are viewed for Device I/O Contention, you can try redistributing the tables across devices (that are located on separate hard disks) or by adding devices (again, that are located on separate hard disks) and moving tables and indexes to them.
For some database operations, Sybase uses a specialized protection mechanism to get exclusive access to a page without using actual restrictive page locks. Examples include accessing certain system tables and dirty reads on data pages. These actions need exclusive access to the page in question, even though they do not actually modify it. Modify conflicts record how many times these actions conflicted with other processes that need true database locks.

**Metrics**
None.

**Tempdb Percent Full**
- **Metrics**
- **Troubleshooting**
The Tempdb Percent Full statistic shows the percentage of the Tempdb that is full.

**Metrics**
Tempdb is probably the most important resource in Adaptive Server shared by all applications and processes. It is a server-wide resource used for internal processing tasks like work tables, sorts, reformating, and index builds.

**Troubleshooting**
Problems can be identified by a consistently high percentage registering or receiving a message indicating that your transactions are being suspended until space is made available in the log. In the short term, try using the “dump tran” DBCC command. If you suspect your application requires a very large tempdb, it should be placed on small fast disks. These disks should have no other data on them.

**NOTE:** On some systems, tempdb performs much faster using an operating system file rather than raw devices. This is because an operating system file uses buffering, which is faster than direct writes to a raw device.

**Task Context Switches**
- **Metrics**
Task Context Switches summarizes task-switch activity for all engines on SMP servers. You can use this statistic to observe the effect of reconfigurations.

**Metrics**
You might reconfigure a cache or add memory if tasks appear to block on cache search misses and to be switched out often. Then, check the data to see if tasks tend to be switched out more or less often.

**Group Commit Sleeps**
- **Metrics**
- **Troubleshooting**
For databases needing high throughput, a large log I/O size is very important to prevent disk queuing problems on the transaction log. Group commit sleeps reflect the number of times a task performed a transactional commit operation and was put to sleep by the server until data in the log was written to disk.

**Metrics**
You should examine group commit sleeps in conjunction with the number of committed transactions (found on the Users performance category view).
Troubleshooting

A high count for group commit sleeps is not necessarily a problem if the server's transaction rate is low. If there are a significant number of transactions that result in group commit sleeps, and the log I/O size is greater than 2 KB, a smaller log I/O size can help to reduce commit time by causing more frequent page flushes.

Other factors that can affect group commit sleeps are the size of the server run queue and the speed of the hard disk where the log is located.

Blocked Connections

- Metrics
- Troubleshooting

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although Sybase supports flexible locking mechanisms, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

The blocked statistic displays the number of current processes blocked by other processes.

Metrics

While the lock contention rate is a better measure of the overall lock contention situation, if you consistently see positive numbers for the blocked statistic there may be a bottleneck for some processes. You can easily drill down and discover the exact process(es) holding locks that are blocking out other user activity.

Troubleshooting

Once you discover a blocking lock situation, you can normally remedy it by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing, and usually results in other user processes completing in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in Sybase. You can change this behavior by modifying the lock wait period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

I/O Analysis - Home

I/O statistics gives you a succinct overview of the datasource’s overall performance. The ratios on the Home page are:

<table>
<thead>
<tr>
<th>Server Reads</th>
<th>Server Writes</th>
<th>Transaction Log Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Reads</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Server Reads

- Metrics

Total Server Reads reflect the total number of physical reads performed by the database server since the last refresh.
**Metrics**

Large numbers of physical reads could reflect a too small data or procedure cache. You should examine the data and procedure cache hit rates to determine the overall effectiveness of logical vs. physical I/O.

**Server Writes**

The Total Server Writes value reflects total number of physical writes performed by the database server since the last refresh.

**Metrics**

None.

**Transaction Log Writes**

Transaction Log Writes refers to the number of times Sybase wrote a transaction log page to disk since the last refresh. When the current log page becomes full, Sybase writes it out to disk. Sybase also writes transaction log pages to disk after a transaction commits.

**Metrics**

None.

**I/O Errors**

**Metrics**

The I/O Errors value reflects total number of I/O errors (errors during read and write operations) encountered by the database server since the last refresh.

**Metrics**

You should investigate large numbers of I/O errors by examining the database error log.

**Storage Analysis - Home**

While DBAs focus on memory settings and tuning SQL, they frequently forget how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

- The 'hit-the-wall' variety that can bring things to a complete standstill.
- The 'performance vampire' kind that slowly drains the performance of a database over time.

Storage problems have the capability to bring the curtain down on a database very quickly, as in the case of a transaction log running out of free space. But storage problems can also silently work behind the scenes to slowly, but surely rob a database of its performance. For example, a hub table in a busy database might be accessed very quickly when an application is first given life, but over time, if it develops a heavy forwarded row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page displays the Total Used and Free Space of all tablespaces in the database. It also provides a count of the number of databases, files, and file groups that exist on the server. For more detailed information on database space, you can use the Performance Analyst Space home page or the Space Detail.
The following ratios succinctly communicate the general overall performance levels of the datasource:

- Total Used
- Total Free
- Number of Databases
- Number of Devices

**Total Used**
- **Metrics**

The Total Used statistic shows the total amount of space currently in use by the Sybase Server.

**Metrics**
None.

**Total Free**
- **Metrics**
  - **Troubleshooting**

The Total Free statistic shows the total amount of free space available to the Sybase Server.

**Metrics**
None.

**Troubleshooting**
A low percentage of free space indicates that additional space should be allocated to your database(s)/devices.

**Number of Databases**
- **Metrics**

This value indicates the total number of database currently defined on the Sybase server.

**Metrics**
None.

**Number of Devices**
This value indicates the total number of devices currently defined on the Sybase server.

**Metrics**
None.
Workload Analysis - Home
When your phone starts ringing with complaints of performance slowdowns, some of the first questions you should answer are:

• Who is currently connected to the database?
• What resources are they using?
• What are they currently executing?

The Workload Analysis section of the Performance Analyst Home page provides insight into the leading resource hogs of a server, as well as a count of total active and inactive processes.

Workload Analysis statistics include:

• Top I/O Process
• Top Memory Process
• Top CPU Process
• Active Processes
• Inactive Processes

Top I/O Process

The Top I/O Process statistic identifies the current SPID with the highest percentage of I/O activity.

Metrics
If any one session uses more than 50% of a total resource (CPU, memory, etc.) drill-down into the session to investigate what work is currently executing.

Top Memory Process

The Top Memory Process statistic identifies the current SPID with the highest percentage of memory consumption.

Metrics
If any one session uses more than 50% of a total resource (CPU, memory, etc.) drill-down into the session to investigate what work is currently executing.

Top CPU Process

The Top CPU Process statistic identifies the current SPID with the highest percentage of CPU consumption.

Metrics
If any one session uses more than 50% of a total resource (CPU, memory, etc.) drill-down into the session to investigate what work is currently executing.
Active Processes

- **Metrics**

  The Active Processes statistic is the total number of active and open threads reported on the Server. Active Processes displays the number of processes actively performing work.

 Metrics

  None.

Inactive Processes

- **Metrics**

- **Troubleshooting**

  The Inactive Sessions statistic represents the total number of threads logged onto the server that are currently idle.

 Metrics

  A large number of inactive processes could indicate user sessions that have mistakenly been left logged on. Since each user thread consumes a portion of memory on the server, to reduce resource usage, you should sever any sessions that do not need a connection.

 Troubleshooting

  Drill down into the Session Waits page and check sessions that have many seconds idle and/or that have been logged on for very long periods of time, as indicated by the logon time column. After verifying that a session is no longer necessary, you can KILL it.

Databases Low on Space

- **Metrics**

- **Troubleshooting**

  The Databases Low on Space statistic indicates if any databases in the server are approaching low levels of available free space. Although good to know, you need a more detailed listing by database to determine where any actual space shortages exist in the server.

 Metrics

  If any one database begins to approach 90% used space, and is continuing to dynamically grow, you should take action to prevent any future space allocation errors.

 Troubleshooting

  If a database is approaching its limit on space, a DBA can either extend the database onto a new device, or if space exists on the device where the database currently resides, the DBA can allocate more space for the database on the current device.

Logs Low on Space

- **Metrics**

- **Troubleshooting**
This statistic indicates if any log for a database in the Sybase server is approaching low levels of available free space. Although good to know, a more detailed listing by database is needed to determine where any actual space shortages exist in the server.

**Metrics**

If a database log’s used space begins to approach 90%, you should take action to prevent any future space allocation errors. If the transaction log runs out of space, no transactions can take place in the database until you free up space in the log.

**Troubleshooting**

If a database log consistently approaches its limit on used space, there are a few actions a DBA can take to prevent a database from freezing.

A backup (dump) of the log can be taken:

- If log backups are not required for disaster recovery, the truncate log on checkpoint option can be set for the database. Setting this option deletes any space devoted to transactions in the log that have already completed when a database checkpoint occurs.
- You can extend the log onto a new device. Or, if space exists on the device on which the database log currently resides, you can allocate more space for the log on the current device.

**Memory Statistics - Sybase**

The Memory performance category view displays the following vital Sybase memory statistics:

- Key Ratio Analysis - Memory
- Bottleneck Analysis - Memory
- Metadata Cache Ratio Analysis
- Memory Analysis
- Workload Analysis

**Key Ratio Analysis - Memory**

Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBA’s to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the master.sysperfinfo performance table. This table maintains a count of various statistics for the server and particular databases since the server was brought up. For servers that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA might be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.
A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server’s individual ‘personality’ with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following memory ratios are used on the Performance Analyst Home page to succinctly communicate the general overall memory performance levels of the monitored database:

<table>
<thead>
<tr>
<th>Data Cache Hit Ratio</th>
<th>Procedure Plan Hit Ratio</th>
<th>Large I/O Hit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean Buffer Grab Rate</td>
<td>Dirty Read Restarts</td>
<td>Dirty Read Requests</td>
</tr>
<tr>
<td>Procedure Requests</td>
<td>Procedure Reads from Disk</td>
<td></td>
</tr>
</tbody>
</table>

**Data Cache Hit Ratio**

- **Metrics**
- **Troubleshooting**

Data read from memory produces end-user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is something that the Sybase data cache tries to assist with.

The data cache hit activity rate is an excellent indicator of how often user requests for data are satisfied through memory vs. being physically read from disk.

**Metrics**

To help ensure excellent performance, you want to keep your cache hit rate in the neighborhood of 90% or greater. Lower amounts can be okay for user ad hoc databases where sporadic, large table scan operations occur. However, anything below this general threshold for normal databases can require tuning attention, and the adjustment of the Sybase memory tuning parameters.

If you are using named caches, you can drill down into the cache hit rates for each named cache. This helps you understand which objects/operations are depressing the overall cache hit rate for the server.

**Troubleshooting**

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or reduce the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when reducing the procedure cache, as this could reduce performance in the server as it relates to reading procedures in from disk.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the data cache (and any named caches), but in 12.5, if you want to reduce the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the data cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary as will an examination of unnecessary large table scan operations.

**Dirty Read Requests**

- **Metrics**
Sybase allows dirty reads, which are reads of uncommitted data. To accomplish a dirty read, Sybase uses a special lightweight protection mechanism to gain access to an object without using actual page locks. This statistic displays the number of dirty reads that occurred since the last refresh in Embarcadero Performance Analyst.

**Metrics**

Dirty read page requests can incur significant overhead if they are observed with many dirty read restarts.

**Dirty Read Restarts**

- **Metrics**

Sybase allows dirty reads, which are reads of uncommitted data. To accomplish a dirty read, Sybase uses a special lightweight protection mechanism to gain access to an object without using actual page locks. A dirty read restart occurs when a dirty read is active on an object page, and another process makes changes to the page that cause the page to be deallocated in memory. The scan for the dirty read must be restarted. The amount shown for dirty read restarts are the number of restarts that occurred since the last Embarcadero Performance Analyst refresh.

**Metrics**

Values observed much above zero should serve as a signal that application modifications can be in order. Most applications should do everything possible to avoid restarts because of the large overhead they incur.

**Troubleshooting**

If the numbers observed for dirty read restarts are significant, you can want to look into modifying applications that use dirty reads to accomplish data acquisition.

**Procedure Plan Hit Ratio**

- **Metrics**

The Sybase procedure cache is used to hold the definitions and query plans of stored procedures and triggers and is used for short-term memory needs like statistics and query plans needed for parallel queries. When a user executes a stored procedure, Sybase looks in the procedure cache for a query plan to use. If a query plan is available, Sybase places it on the most recently used (MRU) end of the memory chain and execution of the procedure begins. If no execution plan is in memory, or if all copies of the plan are currently being used, the query tree for the procedure is read in again from the data dictionary, optimized, put on the MRU end of the chain, and executed.

**NOTE:** Other operations, like CREATE INDEX, can also use the procedure cache even when no procedure is referenced.

The percentage of times that a procedure’s plan and definition can be referenced in memory, the better the procedure execution time.
Metrics
A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for procedure code reference. Note that when a database is first started, the procedure cache hit rate is not at an optimal level because all code being used is relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

NOTE: You can drill down into the procedure cache to view the procedures currently in memory along with how much memory they are consuming.

If there is not enough memory to load a requested procedure, or the maximum number of compiled objects is already in use, Sybase returns an error (normally a 701).

Troubleshooting
If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase server, but in 12.5, if you want to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the procedure cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Procedure Requests
The Procedure Requests statistic reports the number of times that stored procedures were executed since Embarcadero Performance Analyst was last refreshed. Such a request could use either an unused copy of the procedure's query plan in memory or if no such copy exists, the procedure must be read in from disk.

Metrics
None.

Procedure Reads from Disk

- Metrics

- Troubleshooting

The Procedure Reads from Disk statistic reports the number of times since Embarcadero Performance Analyst was last refreshed that stored procedures were read from disk rather than copied in the procedure cache.

Metrics
You should examine this number in conjunction with the overall procedure cache hit rate. Observing large numbers in this statistic indicates a lower than ideal procedure cache hit rate. Note that when a database is first started, this statistic is likely larger than desired because all code being used is relatively new and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels and this statistic continues to sport high numbers, you should look into the possibility of increasing the amount of memory allocated to the cache.
Troubleshooting
If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase server, but in 12.5, if you want to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the procedure cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Large I/O Hit Rate

• **Metrics**

• **Troubleshooting**

Large I/O can be enabled by splitting the default or any named cache into pools. By default, Sybase performs I/O operations based on a 2-KB page size. For queries where pages are stored and accessed in a sequential manner, it is possible to read many more data pages in a single I/O operation. Large I/O can greatly reduce disk access time when the right situations exist. Operations that routinely perform large table scans, access image or text data, do bulk copying, scan the leaf level of nonclustered indexes, or initiate DBCC tasks can benefit from large I/O.

If large I/O has been configured and is being used, you should observe a high percentage of hits (the number of times large I/O could be performed vs. the number of times large I/O requests were denied by the server). If large I/O is not configured, no large I/O activity should be present.

**Metrics**
As you might expect, if large I/O is in use, a high hit rate is desirable. You should strive for a hit ratio between 90-100%, with 90% being a good performance benchmark.

**Troubleshooting**
If large I/O is configured, but a low hit rate is being observed, you should configure more caches for large I/O use.

Clean Buffer Grab Rate

• **Metrics**

• **Troubleshooting**

As information is requested from users, buffers are moved into and out of the Sybase data cache. Pages are also modified in the cache (termed dirty buffers) and need to be written out to disk. If Sybase has to wait for a dirty buffer to be written out to disk before a requested buffer is placed into the cache, performance can suffer.

The clean buffer grab rate represents the percentage of time clean buffers were found and referenced in the cache as opposed to Sybase finding dirty buffers.

**Metrics**
Ideally, the clean buffer grab rate should stay at or near 100%.
Troubleshooting

Seeing a poor clean buffer grab rate for either the default or named caches could indicate that the cache size is too small. You can try to adjust the total memory configuration parameter higher. Keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Bottleneck Analysis - Memory

When Sybase Adaptive Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem.

Memory bottlenecks can definitely cause performance degradation in an otherwise well-running database. Typically, these bottlenecks center around the server latches, which are lightweight locks used to protect certain resources in memory. To help you identify such problems, the following statistics are presented on the Performance Analyst Memory home page:

<table>
<thead>
<tr>
<th>Log Semaphore Contention</th>
<th>Latch Contention</th>
<th>Waits on Execution Plans</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large I/Os Denied</td>
<td>Buffers Grabbed Dirty</td>
<td>Procedure Removals</td>
</tr>
</tbody>
</table>

Log Semaphore Contention

- Metrics

The Log Semaphore Contention statistic measures the total number of times tasks requested a log semaphore. This includes those granted immediately and those for which the task had to wait.

Metrics

In high throughput environments with a large number of concurrent users committing transactions, you should expect a certain amount of contention for the log semaphore.

Latch Contention

- Metrics

- Troubleshooting

The Latch Contention statistic reports the number of times a task was switched out because it needed to wait for a latch. If your user tables use only allpages-locking, this latch contention is taking place either on a data-only-locked system table or on allocation pages. If your applications use data-only-locking, the contention reported here includes all waits for latches, including those on index pages and OAM pages as well as allocation pages.
Metrics
In SMP environments where inserts and expanding updates are extremely high, so that page allocations take place very frequently, contention for the allocation page latch can reduce performance. Normally, Adaptive Server allocates new pages for an object on an allocation unit that is already in use by the object and known to have free space. For each object, Adaptive Server tracks this allocation page number as a hint for any tasks that need to allocate a page for that object. When more than one task at a time needs to allocate a page on the same allocation unit, the second and subsequent tasks block on the latch on the allocation page.

Troubleshooting
You can specify a "greedy allocation" scheme, so that Adaptive Server keeps a list of eight allocation hints for page allocations for a table.

The effect of dbcc tune(des_greedyalloc) are not persistent, so you need to reissue the commands after a reboot.

You should use this command only if all of the following are true:

- You have multiple engines. It is rarely useful with fewer than four engines.
- A large number of pages are being allocated for the object. You can use sp_spaceused or optdiag to track the number of pages.
- The latch contention counter shows contention.

Waits on Execution Plans

The Waits on Execution Plans value represents the number of times that a process attempting to use sp_showplan had to wait to acquire read access to the query plan.

Metrics
Query plans may be unavailable if you run sp_showplan before the compiled plan is completed or after the query plan finished executing. In these cases, Adaptive Server tries to access the plan three times and then returns a message to the user.

Large I/Os Denied

The Large I/Os Denied statistic represents the number of times that large I/O could not be performed.

Metrics
Adaptive Server cannot perform large I/O for the following reasons:

- If any page in a buffer already resides in another pool.
- When there are no buffers available in the requested pool.
- On the first extent of an allocation unit, since it contains the allocation page, which is always read into the 2K pool.
**Troubleshooting**

If a high percentage of large I/Os were denied, it indicates that using larger pools might not be as effective as it could be. If a cache contains a large I/O pool, and queries perform both 2K and 16K I/O on the same objects, there will always be some percentage of large I/Os that cannot be performed because pages are in the 2K pool.

If more than half of the large I/Os were denied, and you are using 16K I/O, try moving all of the space from the 16K pool to the 8K pool. Re-run the test to see if total I/O is reduced. Note that when a 16K I/O is denied, Adaptive Server does not check for 8K or 4K pools, but uses the 2K pool.

**Buffers Grabbed Dirty**

- **Metrics**

The Buffers Grabbed Dirty statistic represents the number of times Sybase found dirty buffers since the last refresh in Embarcadero Performance Analyst.

As users request information, buffers are moved into and out of the Sybase data cache. Pages are also modified in the cache (termed dirty buffers) and need to be written out to disk. If Sybase has to wait for a dirty buffer to be written out to disk before a requested buffer is placed into the cache, performance can suffer.

- **Troubleshooting**

Ideally, the dirty buffer grab statistic should stay close to zero.

**Procedure Removals**

- **Metrics**

The Procedure Removals statistic reports the number of times that stored procedures were aged out of the procedure cache since Embarcadero Performance Analyst was last refreshed.

High numbers, along with a lower than desired procedure cache hit rate, could indicate a procedure cache that is too small.

- **Troubleshooting**

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when increasing the procedure cache only, as this could increase query response times due to more physical I/O being performed.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase server, but in 12.5, if you wish to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.
Once the procedure cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can indicate too little memory left for the server machine.

Metadata Cache Ratio Analysis
The following statistics are used on the Performance Analyst for Sybase Memory Home Page to succinctly communicate the general overall performance levels of the memory structures:

<table>
<thead>
<tr>
<th>Open Objects</th>
<th>Open Indexes</th>
<th>Open Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Spinlock Contention</td>
<td>Index Spinlock Contention</td>
<td>Hash Spinlock Contention</td>
</tr>
</tbody>
</table>

Open Objects
- **Metrics**
- **Troubleshooting**

The Open Objects statistic represents the use of the Adaptive Server metadata cache for open objects. The statistic includes number of active objects, free objects, and the maximum number of objects used.

**Metrics**
When Adaptive Server accesses an object, it needs to read information about it in the corresponding system table: sysobjects. The metadata cache for objects lets Adaptive Server access the information that describes it in the sysobjects row directly in its in-memory structure. This improves performance because Adaptive Server bypasses expensive calls that require disk access. It also reduces synchronization and spinlock contention when Adaptive Server has to retrieve object information at runtime.

Managing individual metadata caches for databases, indexes, or objects is beneficial for a database that contains a large number of indexes and objects, and where there is high concurrency among users.

**Troubleshooting**
You can use this information to set the configuration parameter’s number of open objects.

Open Indexes
- **Metrics**
- **Troubleshooting**

The Open Indexes statistic represents the use of the Adaptive Server metadata cache for open indexes. The statistic includes the number of active indexes, free indexes, and the maximum number of indexes.

**Metrics**
When Adaptive Server accesses an index, it has to access information about it in the respective system table: sysindexes. The metadata cache for indexes lets Adaptive Server access the information that describes it in the sysindexes row directly in its in-memory structure. This improves performance because it allows the Adaptive Server to bypass expensive calls that require disk access. It also reduces synchronization and spinlock contention when Adaptive Server has to retrieve index information at runtime.

Managing individual metadata caches for databases, indexes, or objects is beneficial for a database that contains a large number of indexes and objects, and where there is high concurrency among users.
Troubleshooting
You can use this information to set the configuration parameter’s number of open indexes.

Open Databases
- Metrics
- Troubleshooting

The Open Databases statistic shows the use of the Adaptive Server metadata cache for open databases. This statistic charts the number of active databases, free databases, and the maximum number of databases used.

When Adaptive Server opens a database, it has to access information about it in the respective system table: sysdatabases. The metadata cache for databases lets Adaptive Server access the information that describes it in the sysdatabases row directly in its in-memory structure. This improves performance because it allows the Adaptive Server to bypass expensive calls that require disk access. It also reduces synchronization and spinlock contention when Adaptive Server has to retrieve database information at runtime.

Metrics
Managing individual metadata caches for databases, indexes, or objects is beneficial for a database that contains a large number of indexes and objects, and where there is high concurrency among users.

Troubleshooting
You can use this information to set the configuration parameter’s number of open databases.

Object Spinlock Contention
- Metrics
- Troubleshooting

The Object Spinlock Contention statistic represents spinlock contention on the object descriptor caches.

Metrics
None.

Troubleshooting
You can use this information to tune the configuration parameter’s open object spinlock and open index spinlock ratios. If the reported contention is more than 3%, decrease the value of the corresponding parameter to lower the number of objects or indexes that are protected by a single spinlock.

Index Spinlock Contention
- Metrics
- Troubleshooting

The Index Spinlock Contention value represents spinlock contention on the index descriptor caches.

Metrics
None.
Troubleshooting
You can use this information to tune the configuration parameter’s open object spinlock and open index spinlock ratios. If the reported contention is more than 3%, decrease the value of the corresponding parameter to lower the number of objects or indexes that are protected by a single spinlock.

Hash Spinlock Contention
  • Metrics
  • Troubleshooting
The Hash Spinlock Contention statistic represents the contention for the spinlock on the index metadata cache hash table.

Metrics
None.

Troubleshooting
You can use this information to tune the open index hash spinlock ratio configuration parameter. If the reported contention is greater than 3%, decrease the value of the parameter.

Memory Analysis
The following statistics are used on the Performance Analyst for Sybase Memory Home Page to succinctly communicate the general overall performance levels of the memory structures:

<table>
<thead>
<tr>
<th>Buffer Cache</th>
<th>Procedure Cache</th>
<th>Connection Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Database Memory</td>
<td>Open Object Memory</td>
<td>Open Index Memory</td>
</tr>
</tbody>
</table>

Buffer Cache
  • Metrics
The Buffer Cache statistic displays the amount of memory currently allocated to the Buffer Cache.

Metrics
None.

Procedure Cache
  • Metrics
The Procedure Cache value displays the amount of memory currently allocated to the procedure cache.

Metrics
The memory allocated for the procedure cache holds the optimized query plans (and occasionally trees) for all batches, including any triggers.
Connection Memory

**Metrics**
The Connection Memory value indicates the amount of memory currently allocated to user connections.

**Metrics**
Each connection requires approximately 60 KB.

Open Database Memory

**Metrics**
The Open Database Memory value represents the current amount of memory allocated in the number of open databases parameter.

**Metrics**
None.

Open Object Memory

**Metrics**
The Open Object Memory value represents the current amount of memory allocated in the number of open objects parameter.

**Metrics**
None.

Open Index Memory

**Metrics**
The Open Index Memory value represents the current amount of memory allocated in the number of open indexes parameter.

**Metrics**
None.

**Workload Analysis**
The following statistics are used on the Performance Analyst for Sybase Memory Home Page to succinctly communicate the general overall performance levels of the memory structures:

**Top Memory Hogs**
The Top Memory Process statistic identifies the process that currently is using the highest percentage of memory in the database.
**Metrics**

It is not uncommon for one or two users to cause the majority of runtime problems that plague a server. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation.

If your database server does not have an overabundance of memory, then you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 25-50% of the total memory usage, then you should further investigate the sessions to see what activities they are performing.

**Memory Detail** The Memory Detail includes the following tabbed pages:

- Data Cache
- Procedure Cache

**Data Cache Tab**

The Data Cache tab of the Memory Detail View includes the following sections:

<table>
<thead>
<tr>
<th>Data Cache Hit Ratio</th>
<th>Cache Configurations</th>
<th>Cache Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cached Objects</td>
<td>Cache I/O Activity</td>
<td></td>
</tr>
</tbody>
</table>

**Data Cache Hit Ratio**

- **Metrics**
- **Troubleshooting**

Data read from memory produces user response times many times faster than when that same data is read from disk. The Sybase data cache assists with keeping physical I/Os to an absolute minimum.

The data cache hit rate is an excellent indicator of how often user requests for data are satisfied through memory vs. being physically read from disk.

**Metrics**

To help ensure excellent performance, keep your cache hit rate in the neighborhood of 90% or greater. Lower amounts can be acceptable for user ad hoc databases where sporadic, large table scan operations occur. However, anything below this general threshold for normal databases can require tuning attention, and the adjustment of the Sybase memory tuning parameters.

If you are using named caches, you can drill down into the cache hit rates for each named cache. This helps you understand which objects/operations are depressing the overall cache hit rate for the server.

**Troubleshooting**

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter or reduce the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). You should be careful when you reduce the procedure cache, as this could reduce performance in the server as it relates to reading procedures in from disk.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the data cache (and any named caches), but in 12.5, if you wish to reduce the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.
Once the data cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, you may need to increase the configuration parameter again and examine unnecessary large table scan operations. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Cache Configurations

- **Metrics**

  The Cache Configuration view represents the default and named data caches and the memory configuration in MB.

  **Metrics**

  None.

Cache Efficiency

- **Metrics**

  - **Troubleshooting**

  The Cache Efficiency section summarizes behavior for the default data cache and all named data caches combined.

  The table below describes the information available on the Cache Efficiency detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Name</td>
<td>The name of the specified cache</td>
</tr>
<tr>
<td>Hit Rate</td>
<td>The percentage of cache hits compared to the total number of cache searches.</td>
</tr>
<tr>
<td>Percent Used</td>
<td>The percentage of searches using this cache as a percentage of searches across all caches.</td>
</tr>
<tr>
<td>Spinlocks</td>
<td>The number of spinlocks.</td>
</tr>
<tr>
<td>LRU Buffers</td>
<td>The number of buffers that were retrieved and replaced in the Least Recently Used section of the pool.</td>
</tr>
<tr>
<td>MRU Buffers</td>
<td>The number of buffers that were retrieved and replaced in the Most Recently Used section of the pool.</td>
</tr>
<tr>
<td>Large I/Os</td>
<td>The size (bytes) of the I/O buffer for the pool.</td>
</tr>
<tr>
<td>Dirty Reads</td>
<td>The number of dirty reads.</td>
</tr>
</tbody>
</table>

**Metrics**

Interpreting cache hit data requires an understanding of how the application uses each cache. In caches that are created to hold specific objects such as indexes or look up tables, cache hit ratios can reach 100%. In caches used for random point queries on huge tables, cache hit ratios can be quite low but still represent effective cache use.

This data can also help you determine if adding more memory would improve performance. For example, if "Cache Hits" is high, adding memory probably would not help much.

**Troubleshooting**

You can compare the "percent used" value for each cache to determine if there are caches that are over- or under-utilized. If you decide that a cache is not well utilized, you can:

- Change the cache bindings to balance utilization.
- Resize the cache to correspond more appropriately to its utilization.
Cached Objects

The Cached Objects section shows the objects that are currently cached.

The table below describes the information available on the Cached Objects detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Name</td>
<td>The name of the specified cache.</td>
</tr>
<tr>
<td>Database Name</td>
<td>The name of the database where the object resides.</td>
</tr>
<tr>
<td>Object Owner</td>
<td>The owner of the object.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>Object Type</td>
<td>The type of the object.</td>
</tr>
<tr>
<td>Cached (KB)</td>
<td>The amount, in Kilobytes, of the cached object.</td>
</tr>
<tr>
<td>Processes Accessing</td>
<td>The number of processes accessing the specified object.</td>
</tr>
</tbody>
</table>

Metrics

None.

Cache I/O Activity

The Cache I/O Activity view shows how many searches were directed to each cache.

The table below describes the information available on the Cache I/O Activity view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Name</td>
<td>The name of the specified cache.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>The number of physical reads associated with the specified cache.</td>
</tr>
<tr>
<td>Pages Touched</td>
<td>The number of pages touched by the cache.</td>
</tr>
<tr>
<td>Pages Read</td>
<td>The number of pages read by the cache.</td>
</tr>
<tr>
<td>Buffers to MRU</td>
<td>The number of buffers that were retrieved and replaced in the Most Recently Used section of the pool.</td>
</tr>
<tr>
<td>Buffers to LRU</td>
<td>The number of buffers that were retrieved and replaced in the Least Recently Used section of the pool.</td>
</tr>
<tr>
<td>I/O Buffer Size</td>
<td>The size (bytes) of the I/O buffer for the pool.</td>
</tr>
<tr>
<td>Allocated (KB)</td>
<td>The number of KB that have been allocated for the pool.</td>
</tr>
<tr>
<td>Stalls</td>
<td>The number of times a process had to wait for a free procedure cache buffer when installing a stored procedure into cache.</td>
</tr>
</tbody>
</table>

Metrics

You can compare the values in this view for each cache to determine if there are caches that are over- or under-utilized.
Troubleshooting

If you determine that a cache is not used as well as you would like, you can:

- Change which objects are bound to each cache to balance use.
- Resize the cache to correspond more appropriately to its use.

Procedure Cache Tab

The Procedure Cache tab of the Memory Detail View includes the following sections:

<table>
<thead>
<tr>
<th>Procedure Plan Hit Ratio</th>
<th>Procedure Cache Activity</th>
<th>Procedure Cache Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Procedure Usage</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Procedure Plan Hit Ratio

- Metrics
- Troubleshooting

The Sybase procedure cache is used to hold the definitions and query plans of stored procedures and triggers. It is used for short-term memory needs like statistics and query plans for parallel queries. When a user executes a stored procedure, Sybase looks in the procedure cache for a query plan to use. If a query plan is available, Sybase places it on the most recently used (MRU) end of the memory chain and the procedure begins to execute. If no execution plan is in memory, or if all copies of the plan are currently being used, the query tree for the procedure is read in again from the data dictionary, optimized, put on the MRU end of the chain, and executed. Note that other operations, like CREATE INDEX, can also use the procedure cache even when no procedure is referenced.

The more often a procedure's plan and definition can be referenced in memory, the better the procedure execution time.

Metrics

A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for procedure code reference.

NOTE: When a database is first started, the procedure cache hit rate is not at an optimal level because all code being used is relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

You can drill down into the procedure cache to view the procedures currently in memory along with how much memory they are consuming.

If there is not enough memory to load a requested procedure, or the maximum number of compiled objects is already in use, Sybase returns an error (normally a 701).

Troubleshooting

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed.
For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the
Sybase server, but in 12.5, if you wish to increase the size of the procedure cache, note that it is now configured in
terms of literal size instead of a percentage of the overall configured memory.

Once the procedure cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not,
another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap
activity. Increased swap activity can indicate too little memory left for the server machine.

Procedure Cache Activity

*Metrics*

The Procedure Cache Activity view summarizes the key metrics concerning the Procedure Cache.

The table below describes the information available on the Procedure Cache Activity view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requests</td>
<td>The number of stored procedures requested from the cache.</td>
</tr>
<tr>
<td>Loads</td>
<td>The number of stored procedures loaded into cache.</td>
</tr>
<tr>
<td>Writes</td>
<td>The number of times a stored procedure was normalized and the tree was written back to sysprocedures. This happens when a stored procedure (or trigger) is created during the interval between polls. A degradation in performance can occur if an application program generates stored procedures.</td>
</tr>
<tr>
<td>Stalls</td>
<td>The number of times a process had to wait for a free procedure cache buffer when installing a stored procedure into cache.</td>
</tr>
</tbody>
</table>

*Metrics*

None.

Procedure Cache Detail

*Metrics*

The Procedure Cache Detail view summarizes the key metrics concerning the Procedure Cache.

The table below describes the information available on the Procedure Cache Detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>Object Owner</td>
<td>The object’s owner.</td>
</tr>
<tr>
<td>Object Type</td>
<td>The type of object.</td>
</tr>
<tr>
<td>Database</td>
<td>The name of the database where the object resides.</td>
</tr>
<tr>
<td>Compile Date</td>
<td>The last month/day/year object data was compiled.</td>
</tr>
<tr>
<td>Memory Used (KB)</td>
<td>The amount of memory used by the object in kilobytes.</td>
</tr>
</tbody>
</table>

*Metrics*

None.
Current Procedure Usage

- **Metrics**

The Current Procedure Usage view displays the number of procedures currently in use.

The table below describes the information available in the Current Procedure Usage view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The process name.</td>
</tr>
<tr>
<td>User Name</td>
<td>The name of the user attached to the process.</td>
</tr>
<tr>
<td>Database</td>
<td>The database attached to the process.</td>
</tr>
<tr>
<td>Object Owner</td>
<td>The object’s owner.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The object’s name.</td>
</tr>
<tr>
<td>Object Type</td>
<td>The object’s type.</td>
</tr>
<tr>
<td>Memory Used (KB)</td>
<td>The amount of memory the procedure used in kilobytes.</td>
</tr>
<tr>
<td>Program Name</td>
<td>The name of the program where the procedure is used.</td>
</tr>
</tbody>
</table>

**Metrics**

None.

**I/O Statistics - Sybase**

The I/O performance category page displays the following vital Sybase I/O statistics:

- **Bottleneck Analysis - I/O**
  - Key System Analysis - I/O

**Key System Analysis**

The following statistics are used on the Performance Analyst for Sybase I/O Home Page to succinctly communicate the general overall performance levels of I/O:

<table>
<thead>
<tr>
<th>Physical Reads</th>
<th>APF Reads</th>
<th>Physical Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure Requests from Disk</td>
<td>Data Cache Searches</td>
<td>Checkpoints</td>
</tr>
<tr>
<td>Large I/Os Performed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Physical Reads**

- **Metrics**

Physical Reads represents the total number of physical reads performed by the database server since the last refresh.

**Metrics**

Large numbers of physical reads could reflect a too small data or procedure cache. You should examine the data and procedure cache hit rates to determine the overall effectiveness of logical vs. physical I/O.
APF Reads

- **Metrics**

The APF Reads statistic represents the number of pages that were read by asynchronous prefetch since the last refresh.

**Metrics**

Asynchronous prefetch tries to find a page it needs to read in the data cache using a quick scan without holding the cache spinlock. If that does not succeed, it then performs a thorough scan holding the spinlock.

Physical Writes

- **Metrics**

The Physical Writes value reflects total number of physical writes performed by the database server since the last refresh.

**Metrics**

None.

Procedure Requests from Disk

- **Metrics**

  - **Troubleshooting**

The Procedure Requests from Disk value represents the number of times that stored procedures were read from disk rather than found and copied in the procedure cache.

**Metrics**

If you see a relatively high number, it could indicate that the procedure cache is too small.

**Troubleshooting**

You should increase the size of the memory allocated to the procedure cache memory structure.

Data Cache Searches

- **Metrics**

  - **Troubleshooting**

The Data Cache Searches value represents the number of times that a needed page was found in any cache.

**Metrics**

None.

**Troubleshooting**

If you see a high number of cache misses, you should investigate statistics for each cache.

Checkpoints

- **Metrics**
The Checkpoints value represents the combined number of normal and free checkpoints that occurred during the sample interval.

**Metrics**

Checkpoints write dirty pages (pages that have been modified in memory, but not written to disk) to the database device. Adaptive Server's automatic (normal) checkpoint mechanism works to maintain a minimum recovery interval. By tracking the number of log records in the transaction log since the last checkpoint was performed, it estimates whether the time required to recover the transactions exceeds the recovery interval. If so, the checkpoint process scans all data caches and writes out all changed data pages.

When Adaptive Server has no user tasks to process, a housekeeper task begins writing dirty buffers to disk. These writes are done during the server’s idle cycles, so they are known as "free writes." They result in improved CPU use and a decreased need for buffer washing during transaction processing.

If the housekeeper process finishes writing all dirty pages in all caches to disk, it checks the number of rows in the transaction log since the last checkpoint. If there are more than 100 log records, it issues a checkpoint. This is called a "free checkpoint" because it requires very little overhead. In addition, it reduces future overhead for normal checkpoints.

**Large I/Os Performed**

- **Metrics**
- **Troubleshooting**

The Large I/Os Performed statistic measures the number of times that a requested large I/O was performed.

**Metrics**

Adaptive Server cannot perform large I/O for the following situations:

- If any page in a buffer already resides in another pool.
- When there are no buffers available in the requested pool.
- On the first extent of an allocation unit, since it contains the allocation page, which is always read into the 2K pool.

**Troubleshooting**

If a high percentage of large I/Os were denied, it indicates that the use of the larger pools might not be as effective as it could be. If a cache contains a large I/O pool, and queries perform both 2K and 16K I/O on the same objects, there will always be some percentage of large I/Os that cannot be performed because pages are in the 2K pool.

If more than half of the large I/Os were denied, and you are using 16K I/O, try moving all of the space from the 16K pool to the 8K pool. Re-run the test to see if total I/O is reduced. Note that when a 16K I/O is denied, Adaptive Server does not check for 8K or 4K pools, but uses the 2K pool.

**Bottleneck Analysis - I/O**

The following statistics are used on the Performance Analyst for Sybase I/O Home Page to succinctly communicate the general overall performance levels of I/O:

<table>
<thead>
<tr>
<th>Delayed Disk I/Os</th>
<th>Outstanding Disk I/Os</th>
<th>Group Commit Sleeps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device I/O Contention</td>
<td>Disk I/O Structure Contention</td>
<td>Total I/O</td>
</tr>
<tr>
<td>I/O Errors</td>
<td>I/O Busy</td>
<td>Hottest Devices</td>
</tr>
</tbody>
</table>
Delayed Disk IOs

**Metrics**
The Delayed Disk IOs value is the number of I/Os delayed when the limit on disk I/O structures is reached.

**Troubleshooting**
When Adaptive Server exceeds the number of available disk I/O control blocks, I/O is delayed because Adaptive Server requires that tasks get a disk I/O control block before initiating an I/O request.

If you see a nonzero value for delayed disk I/Os, try to add to the number of available disk I/O control blocks by increasing the configuration parameter.

Outstanding Disk I/Os

The Outstanding Disk I/Os statistic reflects the maximum number of I/Os pending for Sybase since the last refresh.

**Metrics**
If you see nonzero numbers for both outstanding and delayed disk I/Os, there could be a problem in many areas.

**Group Commit Sleeps**

**Troubleshooting**
For databases needing high throughput, a large log I/O size is very important to prevent disk queuing problems on the transaction log. Group commit sleeps reflect the number of times a task performed a transactional commit operation and was put to sleep by the server until data in the log was written to disk.

**Metrics**
You should examine group commit sleeps in conjunction with the number of committed transactions (found on the Users performance category view).

**Troubleshooting**
A high count for group commit sleeps is not necessarily a problem if the server's transaction rate is low. If there are a significant number of transactions that result in group commit sleeps, and the log I/O size is greater than 2 KB, a smaller log I/O size can help to reduce commit time by causing more frequent page flushes.

Other factors that can affect group commit sleeps are the size of the server run queue and the speed of the hard disk where the log is located.

Device I/O Contention

**Troubleshooting**
In Sybase, when a task or process needs to perform physical I/O, the server fills out the actual block I/O structure and links it to a per-engine I/O queue. If two Sybase engines request the same I/O structure from the exact same device at the identical time, one of them is put to sleep by the server and it waits for the semaphore it needs.

Device I/O Contention reflects the number of times a task or process was put to sleep while waiting for a semaphore for a particular database device.
Metrics
None.

Troubleshooting
If you consistently see high numbers for Device I/O Contention, you can try redistributing the tables across devices (that are located on separate hard disks) or by adding devices (again, that are located on separate hard disks) and moving tables and indexes to them.

Disk I/O Structure Contention
  • Metrics
  • Troubleshooting

When a Sybase server begins to experience I/O delays, the result can be a very dissatisfied user community. When such problems begin to occur, you should investigate Sybase or operating system limits. It could be that I/O operations are being blocked by one or both.

The Disk I/O Structures statistic represents the number of I/O delays caused by Sybase reaching the limit on disk I/O structures. When Sybase exceeds the number of available disk I/O control blocks, I/O is deferred because Sybase requires that any task get a disk I/O control block before beginning an I/O request.

Metrics
Consistent numbers above zero should be a cause for concern.

Troubleshooting
If you continue to see nonzero numbers for this statistic, you can try increasing the number of available disk I/O control blocks by increasing the configuration parameter disk I/O structures.

Total I/O
  • Metrics

The Total I/O statistic represents the total number of physical reads and writes.

Metrics
None.

I/O Errors
  • Metrics
  • Troubleshooting

I/O error rate reflects total number of I/O errors (errors during read and write operations) encountered by the server since the last refresh inside Performance Analyst. The I/O errors rate is a percentage based on Total I/O (the sum of the physical reads and writes).

Metrics
You should observe few, if any errors.
Troubleshooting
If you notice any errors, you should check the Sybase error log for details.

I/O Busy
- **Metrics**
- **Troubleshooting**

The I/O Busy statistic represents the number of clock ticks in the sample interval during which the user task performed I/O operations.

**Metrics**
High numbers indicate an I/O-intensive process. If idle time is also high, the application could be I/O bound.

**Troubleshooting**
The application might achieve better throughput if you assign it a higher priority, bind it to a lightly loaded engine or engine group, or partition the application's data onto multiple devices.

Hottest Devices
- **Metrics**
- **Troubleshooting**

The Hottest Devices view identifies the configured devices engaged in the most I/O activity. In Sybase, when a task or process needs to perform physical I/O, the server fills out the actual block I/O structure and links it to a per-engine I/O queue. If two Sybase engines request the same I/O structure from the exact same device at the identical time, one of them is put to sleep by the server and it waits for the semaphore it needs.

Device I/O Contention reflects the number of times a task or process was put to sleep while waiting for a semaphore for a particular database device.

The table below describes the information available on the Hottest Devices view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Name</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>APF Reads</td>
<td>The number of asynchronous pre-fetch buffers read.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>The total number of physical reads performed by the database server since the last refresh</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>The total number of physical writes performed by the database server since the last refresh</td>
</tr>
<tr>
<td>Total I/O</td>
<td>The combined total of read and write operations.</td>
</tr>
</tbody>
</table>

**Metrics**
None.
Troubleshooting
If you consistently see high numbers for Device I/O Contention, you can try redistributing the tables across devices (that are located on separate hard disks) or by adding devices (again, that are located on separate hard disks) and moving tables and indexes to them.

Network Analysis - I/O
The following statistics are used on the Performance Analyst for Sybase I/O page to succinctly communicate the general overall performance levels of I/O:

<table>
<thead>
<tr>
<th>Packets Sent</th>
<th>Packets Received</th>
<th>Network Delays</th>
</tr>
</thead>
<tbody>
<tr>
<td>None.</td>
<td>None.</td>
<td></td>
</tr>
</tbody>
</table>

Packets Sent

- **Metrics**

Packets Sent displays the number of times Sybase sent a packet to a client application.

- **Metrics**

None.

Packets Received

- **Metrics**

Total Packets Received reflects the number of times Sybase received a packet from a client application.

- **Metrics**

None.

Network Delays

- **Metrics**

Network Delays statistic reflects the number of times network I/O activity was delayed.

- **Metrics**

If you see a network contention rate greater than 1%, it could indicate a challenged network structure.

- **Troubleshooting**

Beyond ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase-specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase server.
Network Errors

: Metrics

: Troubleshooting

The Network Errors statistic reflects the number of times that network errors were detected by Sybase while reading and writing packets.

Metrics

If you consistently see a value much above zero, it could indicate the network structure is challenged.

Troubleshooting

Beyond ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase-specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase server.

Transaction Log Analysis - I/O

The following statistics are used on the Performance Analyst for Sybase I/O page to succinctly communicate the general overall performance levels of I/O:

<table>
<thead>
<tr>
<th>Log Writes</th>
<th>Log Allocations</th>
<th>ULC Flushes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Number of Writes per Log Page</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Log Writes

: Metrics

Transaction Log Writes refers to the number of times Sybase wrote a transaction log page to disk since the last refresh. When the current log page becomes full, Sybase writes it out to disk. Sybase also writes transaction log pages to disk after a transaction commits.

Metrics

None.

Log Allocations

: Metrics

Transaction Log Allocations refers to the number of times since the last refresh that additional pages were allocated to the transaction log. This statistic gives you a feel for the rate of transaction log growth.

Metrics

None.

ULC Flushes

: Metrics
ULC Flushes is the total number of times that ULCs (user log caches) were flushed to a transaction log during the sample interval.

**Metrics**
None.

**Average Number of Writes per Log Page**

- **Metrics**

  The Average Number of Writes is the average number of times each log page was written to disk.

  **Metrics**

  In high throughput applications, this number should be as low as possible. If the transaction log uses 2K I/O, the lowest possible value is 1; with 4K log I/O, the lowest possible value is .5, since one log I/O can write 2 log pages.

  In low throughput applications, the number will be significantly higher. In very low throughput environments, it may be as high as one write per completed transaction.

**Workload Analysis - I/O**

The following statistics are used on the Performance Analyst for Sybase I/O page to succinctly communicate the general overall performance levels of I/O:

- **Top I/O Hogs**

  **Top I/O Hogs**

  - **Metrics**

  The Top I/O Process statistic identifies the processes that has currently caused the most I/O usage on the database.

  The table below describes the information available from the Top I/O Hogs view of the I/O page:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The process ID.</td>
</tr>
<tr>
<td>Login</td>
<td>The login name the session is using.</td>
</tr>
<tr>
<td>Physical I/O</td>
<td>The physical amount of I/O the process is using.</td>
</tr>
<tr>
<td>% Used</td>
<td>Percent of total I/O on the server this process is consuming.</td>
</tr>
</tbody>
</table>

**Metrics**
None.

**Troubleshooting**

If any one session uses more than 50% of a total resource (CPU, memory, etc.) you should drill down into that particular session and investigate the cause.

**I/O Details**
The following tabbed pages are available on the I/O Detail view:

<table>
<thead>
<tr>
<th>Databases</th>
<th>Devices</th>
<th>Engines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Objects</td>
<td></td>
</tr>
</tbody>
</table>

**Databases**
- **Metrics**
- **Troubleshooting**

The database I/O statistic displays the major I/O statistics as they relate to each associated database.

The table below describes the information available from the I/O Databases detail tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Logical Reads</td>
<td>The number of logical reads associated with the database.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>The number of physical reads associated with the database.</td>
</tr>
<tr>
<td>APF Reads</td>
<td>The number of asynchronous pre-fetch buffer reads associated with the database.</td>
</tr>
<tr>
<td>Pages Read</td>
<td>The number of pages read associated with the database.</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>The number of physical writes associated with the database.</td>
</tr>
<tr>
<td>Pages Written</td>
<td>The number of pages written to the associated database.</td>
</tr>
<tr>
<td>Rows Inserted</td>
<td>The number of rows inserted in the associated database.</td>
</tr>
<tr>
<td>Rows Updated</td>
<td>The number of rows updated in the associated database.</td>
</tr>
<tr>
<td>Rows Deleted</td>
<td>The number of rows deleted from the associated database.</td>
</tr>
<tr>
<td>Lock Requests</td>
<td>The number of requests for a lock on database objects.</td>
</tr>
<tr>
<td>Lock Waits</td>
<td>The number of times a task waited for a lock on a database object.</td>
</tr>
</tbody>
</table>

**Metrics**
None.

**Troubleshooting**
The major guidelines for improving I/O performance in Adaptive Server are as follows:

- Spread data across disks to avoid I/O contention.
- Isolate server-wide I/O from database I/O.
- Separate data storage and log storage for frequently updated databases.
- Keep random disk I/O away from sequential disk I/O.
- Mirror devices on separate physical disks.
- Partition tables to match the number of physical devices in a segment.
Objects
The following views are used on the Performance Analyst for Sybase I/O Drilldown Objects to succinctly communicate the general overall performance levels of I/O:

- Index Scans
- Open Object I/O

Index Scans
The following views are used on the Index Scans view of I/O Drilldown Objects to succinctly communicate the general overall performance levels of I/O:

<table>
<thead>
<tr>
<th>Ascending Scans</th>
<th>DOL Ascending Scans</th>
<th>DOL Descending Scans</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descending Scans</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ascending Scans

- Metrics

The Ascending Scans value represents the number of forward scans on allpages-locked tables.

Metrics
None.

DOL Ascending Scans

- Metrics

The DOL value represents the number of forward scans on data-only-locked tables.

Metrics
None.

DOL Descending Scans

- Metrics

The DOL Descending Scans value represents the number of backward scans on allpages-locked tables.

Metrics
None.

Descending Scans

- Metrics

The Descending Scans value represents the number of backward scans on data-only-locked tables.

Metrics
None.
Open Object I/O

The Open Object I/O view displays a grid that shows a list of each object on the selected database that is either open, or has been open, and related I/O activity.

The table below describes the information available on the Open Object I/O grid:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the object.</td>
</tr>
<tr>
<td>Index ID</td>
<td>The ID number of the associated index.</td>
</tr>
<tr>
<td>Logical Reads</td>
<td>Total number of buffers read.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>Total number of buffers read from disk.</td>
</tr>
<tr>
<td>APF Reads</td>
<td>The number of asynchronous pre-fetch buffer reads associated with the database.</td>
</tr>
<tr>
<td>Pages Read</td>
<td>The number of pages read associated with the database.</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>The number of physical writes associated with the database.</td>
</tr>
<tr>
<td>Pages Written</td>
<td>The number of pages written to the associated database.</td>
</tr>
<tr>
<td>Rows Inserted</td>
<td>The number of rows inserted in the associated database.</td>
</tr>
<tr>
<td>Rows Updated</td>
<td>The number of rows updated in the associated database.</td>
</tr>
<tr>
<td>Rows Deleted</td>
<td>The number of rows deleted from the associated database.</td>
</tr>
<tr>
<td>Used Count</td>
<td>The number of rows used.</td>
</tr>
<tr>
<td>Lock Requests</td>
<td>The number of requests for a lock on database objects.</td>
</tr>
<tr>
<td>Lock Waits</td>
<td>The number of times a task waited for a lock on a database object.</td>
</tr>
</tbody>
</table>

Metrics

Objects are only removed from the list if the configured number of "open objects" is exceeded. Then they are aged out to make room for newly opened objects.

Network

The Network tab of the I/O Detail view presents the following statistics:

<table>
<thead>
<tr>
<th>Network I/O Summary</th>
<th>Network I/O by Engine</th>
<th>Network I/O by Session</th>
</tr>
</thead>
</table>

Network I/O Summary

Here you see a summary of data that includes the following statistics:

- **Packets Sent**: Total Packets Sent reflects the number of times Sybase sent a packet to a client application.
- **Packets Received**: Total Packets Received reflects the number of times Sybase received a packet from a client application.
- **Bytes Sent**: Total Bytes Sent reflects the number of bytes sent to Sybase since the last refresh.
Bytes Received: Total Bytes Received reflects the number of bytes received by Sybase since the last refresh.

Network I/O by Engine
The statistics, presented in tabular format, include:

- **Engine**: The name of the engine under scrutiny.
- **Packets Sent**: The number of packets sent by the engine.
- **Packets Received**: The number of packets received by the engine.
- **Bytes Sent**: The number of bytes sent by the engine.
- **Bytes Received**: The number of bytes received by the engine.

Network I/O by Session

- **Metrics**

This view shows the amount of network I/O committed by registered sessions on the Sybase Server. Presented in tabular format, you see:

- **Login**: The log in name used by the session.
- **SPID**: The process ID.
- **KPID**: The kernel process ID.
- **Packets Sent**: The number of packets sent by the session.
- **Packets Received**: The number of packets received by the session.
- **Bytes Sent**: The number of bytes sent by the session.
- **Bytes Received**: The number of bytes received by the session.
- **Network Packet Size**: Network packet size used by the current session.

**Metrics**

This metric can be used to see if a user or batch job is generating more (or less) network traffic than expected. The process’ network traffic, measured in conjunction with overall server network traffic, could help distinguish between process level or network-wide loads being the source of contention.

**Devices**

The following statistics are used on the Performance Analyst for Sybase I/O Drilldown Devices to succinctly communicate the general overall performance levels of I/O:

<table>
<thead>
<tr>
<th>Device I/O Summary</th>
<th>Device I/O Type</th>
<th>Real Time Device I/O</th>
</tr>
</thead>
</table>

**Device I/O Summary**

- **Metrics**
- **Troubleshooting**
Devices are accessed repeatedly in a dynamic Sybase environment to satisfy user requests for data, to handle write activity that records transactions in a database’s transaction log, and to manage other I/O operations. Viewing the I/O activity for each device is a good way to see what the "hot" devices are in a Sybase server with respect to I/O usage. The same information can be used to spot heavy database usage in systems where the device-to-database mappings are one-to-one.

The table below describes the information available in the Device I/O Summary view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>The device’s name.</td>
</tr>
<tr>
<td>APF Reads</td>
<td>The number of asynchronous pre-fetch buffer reads associated with the database.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>The number of physical reads associated with the database.</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>The number of physical writes associated with the database.</td>
</tr>
<tr>
<td>Total I/O</td>
<td>The combined total of read and write operations.</td>
</tr>
</tbody>
</table>

**Metrics**

If devices have one-to-one relationships with user databases, you can quickly tell which databases are experiencing the most demand on a server. If the device-to-physical drive/file system is a one-to-one relationship, you can also spot which physical drives on a server are getting the heaviest workouts. For servers that have many drives, it is normally desirable to spread devices across different physical drives and controllers so contention does not occur at the disk level. In addition, separating databases and their corresponding logs is normally recommended so that each is located on a distinct physical drive. If possible, write-intensive devices, like log devices, are best suited for non-RAID5 scenarios.

**Troubleshooting**

If device hit rates are low, you can add more devices or redistribute objects among different devices. Typically, segments can be used to redistribute objects among different devices or physical hard disks. Common techniques include placing tables and indexes on different segments and partitioning large tables.

**Device I/O Type**

- **Metrics**
- **Troubleshooting**

The Device I/O Type view shows the type of I/O registering against each configured device.

Devices are accessed repeatedly in a dynamic Sybase environment to satisfy user requests for data, to handle write activity that records transactions in a database’s transaction log, and to manage other I/O operations. Viewing the I/O activity for each device is a good way to see what the "hot" devices are in a Sybase server with respect to I/O usage. The same information can be used to spot heavy database usage in systems where the device-to-database mappings are one-to-one.

The table below describes the information available from the Device I/O Type detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>I/O Type</td>
<td>The type of I/O.</td>
</tr>
<tr>
<td>I/O Time (sec)</td>
<td>The amount of time, in seconds, spent performing I/O operations.</td>
</tr>
</tbody>
</table>
Metrics

If devices have one-to-one relationships with user databases, you can quickly tell which databases are experiencing the most demand on a server. If the device-to-physical drive/file system is a one-to-one relationship, you can also spot which physical drives on a server are getting the heaviest workouts. For servers that have many drives, it is normally desirable to spread devices across different physical drives and controllers so contention does not occur at the disk level. In addition, separating databases and their corresponding logs is normally recommended so that each is located on a distinct physical drive. If possible, write-intensive devices, like log devices, are best suited for non-RAID5 scenarios.

Troubleshooting

If device hit rates are low, you can add more devices or redistribute objects among different devices. Typically, segments can be used to redistribute objects among different devices or physical hard disks. Common techniques include placing tables and indexes on different segments and partitioning large tables.

If device loads appear skewed (one device has much more activity than others), you should focus attention on that device. Again, redistributing objects can lessen the device's workload.

Real Time Device I/O

Devices are accessed repeatedly in a dynamic Sybase environment to satisfy user requests for data, to handle write activity that records transactions in a database's transaction log, and to manage other I/O operations. Viewing the current, real-time I/O activity for each device is a good way to see what the "hot" devices are in a Sybase server with respect to I/O usage. The same information can be used to spot heavy database usage in systems where the device-to-database mappings are one-to-one.

The table below describes the information available on the Real Time Device I/O detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>The logical name of the device.</td>
</tr>
<tr>
<td>Physical Name</td>
<td>The physical name of the device.</td>
</tr>
<tr>
<td>APF Reads</td>
<td>The number of asynchronous pre-fetch buffer reads associated with the database.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>Number of buffers read from disk.</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>Number of buffers written to disk.</td>
</tr>
<tr>
<td>I/O Time (Secs)</td>
<td>The amount of time, in seconds, spent performing I/O operations.</td>
</tr>
<tr>
<td>Semaphore Requests</td>
<td>The number of device semaphore requests immediately granted on the device.</td>
</tr>
<tr>
<td>Semaphore Waits</td>
<td>The number of tasks forced to wait for synchronization of an I/O request.</td>
</tr>
</tbody>
</table>

Metrics

The amount of Reads, APF Reads, and Writes can help you spot overworked devices. You should also keep your eye on the semaphore waits and I/O time for each device to help you determine devices that are experiencing I/O related delays.
Troubleshooting
If device hit rates are low, you can add more devices or redistribute objects among different devices. Typically, segments can be used to redistribute objects among different devices or physical hard disks. Common techniques include placing tables and indexes on different segments and partitioning large tables.

If device loads appear skewed (one device has much more activity than others), you should focus attention on that device. Again, redistributing objects can lessen the device's workload.

Engines Tab
- Metrics
- Troubleshooting

In symmetric multiprocessing (SMP) environments, a DBA can configure the Sybase server to use more than one "engine," which represents a certain amount of CPU power. By default, Sybase configures one engine for use. If you have a server machine with multiple CPUs, you can enable more engines to take advantage of the machine's additional processing ability.

The Engines tab of the I/O Detail view displays information with respect to how each engine is handling I/O. The table below describes the information available on the Engines tab of the I/O Detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine</td>
<td>The name of the configured engine.</td>
</tr>
<tr>
<td>Completed I/Os</td>
<td>The number of I/Os completed during the sample interval.</td>
</tr>
<tr>
<td>Outstanding I/Os</td>
<td>The number of I/Os left outstanding during the sample interval.</td>
</tr>
</tbody>
</table>

TIP: To configure the grid to show/hide row numbers, use the Options Editor.

Metrics
If the number of outstanding I/Os remains high or increases during periods of heavy activity, there may not be enough engines configured for the system.

Troubleshooting
If the server machine has multiple CPUs, you can configure more engines for Sybase to use by following this process:

1. Use the sp_configure procedure to change the current engine configuration. For example, to change the number of engines from one to two, you would run: "sp_configure "max online engines",2"
2. Stop and restart the Sybase server.

Index Scans Tab
- Metrics

Indexes are accessed frequently in dynamic Sybase server environments. The type of index access often determines the response time an end user experiences. Single row index accesses are the quickest, and complete index scans are the most time consuming (for large indexes at least).

The Index Scans tab of the I/O Detail view presents information with respect to index scans. The table below describes the information available on the Index Scans tab of the I/O Detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan Type</td>
<td>The type of index scan.</td>
</tr>
</tbody>
</table>
Metrics
There are two basic scan types - ascending and descending. For ascending scans, Sybase moves up the index page chain from beginning to end. Sybase follows the page chain in reverse order for descending scans. Descending scans are normally the result of requests made for data by descending column order.

Within ascending and descending scans, a data-only lock, or DOL, styled access can also occur.

Space Statistics - Sybase
The Space performance category view displays the following vital Sybase space statistics:

<table>
<thead>
<tr>
<th>Space Analysis</th>
<th>Transaction Log Analysis</th>
<th>Bottleneck Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Space Analysis
• Metrics
A Sybase Server is broken up into many different databases, each of which has its own level of activity. The Space Analysis section displays a summary of activities that have occurred within each database. It also displays the total number of databases defined on the Sybase server.

The table below describes the database information available on the Space Analysis view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the Sybase database.</td>
</tr>
<tr>
<td>Total Space (MB)</td>
<td>The total amount of space allocated to the database expressed in MB.</td>
</tr>
<tr>
<td>Free Space (MB)</td>
<td>The amount of space currently free expressed in MB.</td>
</tr>
<tr>
<td>Used Space (MB)</td>
<td>The amount of space currently in use, expressed in MB.</td>
</tr>
<tr>
<td>Percent Free</td>
<td>The available free space on the database, expressed as a percentage.</td>
</tr>
<tr>
<td>Log Total Space (MB)</td>
<td></td>
</tr>
<tr>
<td>Log Free Space (MB)</td>
<td></td>
</tr>
<tr>
<td>Log Used Space (MB)</td>
<td></td>
</tr>
<tr>
<td>Log Percent Free</td>
<td></td>
</tr>
<tr>
<td>Log Percent Used</td>
<td></td>
</tr>
</tbody>
</table>

Metrics
None.
Transaction Log Analysis
The following statistics are used on the Performance Analyst for Sybase Space page to succinctly communicate the general overall performance levels of space related metrics:

- Transaction Log Analysis

Transaction Log Analysis
- Metrics
- Troubleshooting

The Transaction Log Analysis view indicates if any transaction log for a database in the Sybase server is approaching low levels of available free space.

The table below describes the information available on the Transaction Log Analysis grid:

<table>
<thead>
<tr>
<th>Axis</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Space Used</td>
<td>The amount of space currently used in the log, expressed as a percentage.</td>
</tr>
<tr>
<td>Databases</td>
<td>The name of the Sybase database.</td>
</tr>
</tbody>
</table>

Metrics
If the transaction log runs out of space, no transactions can take place in the database until you free up space in the log.

Troubleshooting
If a database log consistently approaches its limit on used space, there are a few actions a DBA can take to prevent a database from freezing.

A backup (dump) of the log can be taken:
- If log backups are not required for disaster recovery, the truncate log on checkpoint option can be set for the database. Setting this option deletes any space devoted to transactions in the log that have already completed when a database checkpoint occurs.
- You can extend the log onto a new device. Or, if space exists on the device on which the database log currently resides, you can allocate more space for the log on the current device.

Bottleneck Analysis
The following statistics are used on the Performance Analyst for Sybase Space page to succinctly communicate the general overall performance levels of space related metrics:

<table>
<thead>
<tr>
<th>Databases low on space</th>
<th>Logs low on space</th>
<th>Devices low on space</th>
</tr>
</thead>
</table>

Databases low on space
- Metrics
- Troubleshooting
The Databases Low on Space statistic indicates if any databases in the server are approaching low levels of available free space. Although good to know, you need a more detailed listing by database to determine where actual space shortages exist in the server.

**Metrics**

If you see that any one database is approaching 90% of used space, and is continuing to dynamically grow, you should take action to prevent any future space allocation errors.

**Troubleshooting**

If a database is approaching its limit on space, a DBA can extend the database onto a new device. Or, if space exists on the device where the database currently resides, the DBA can allocate more space for the database on the current device.

Logs low on space

- **Metrics**
- **Troubleshooting**

The Logs low on space statistic indicates if any log for a database in the Sybase server is approaching low levels of available free space. Although good to know, you need a more detailed listing by database to determine where any actual space shortages exist in the server.

**Metrics**

If you see a database log's used space begins to approach 90%, you should take action to prevent future space allocation errors. If the transaction log runs out of space, no transactions can take place in the database until you free up space in the log.

**Troubleshooting**

If a database log consistently approaches its limit on used space, there are a few actions a DBA can take to prevent a database from freezing.

You can take a backup (dump) of the log:

- If log backups are not required for disaster recovery, set the truncate log on checkpoint option for the database. Setting this option deletes any space devoted to transactions in the log that have already completed when a database checkpoint occurs.
- You can extend the log onto a new device. Or, if space exists on the device on which the database log currently resides, you can allocate more space for the log on the current device.

Devices low on space

- **Metrics**
- **Troubleshooting**

The Devices low on space statistic indicates if any device in the Sybase server is approaching low levels of available free space.

**Metrics**

The environment of the particular Sybase server, as well as your work style dictate the metrics you should use to evaluating a device that is running into trouble with space. Many DBAs create devices that parallel a single, corresponding database in size, and therefore, such devices show 100% utilization. Other DBAs create large devices that are not completely utilized.
Troubleshooting
If a device has become too full, a DBA can begin the process of manually relocating databases from it onto other devices. The process of moving logs is somewhat easier and can be accomplished via singular commands (sp_logdevice).

Device Analysis

- Metrics
- Troubleshooting

The Device Analysis view presents space related details about a selected device located on the Sybase server.

The table below describes the information available on the Device Analysis view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Devices</td>
<td>Total number of devices configured on the server.</td>
</tr>
<tr>
<td>Device Name</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>Physical Name</td>
<td>The name used by the operating system to identify the device.</td>
</tr>
<tr>
<td>Total Space (MB)</td>
<td>Total space allocated on this device in MB.</td>
</tr>
<tr>
<td>Used Space (MB)</td>
<td>Total space, in MB, currently in use on the device.</td>
</tr>
<tr>
<td>Free Space (MB)</td>
<td>Amount of space that has not been allocated on this device.</td>
</tr>
<tr>
<td>Percent Free (MB)</td>
<td>The available free space on the database, expressed as a percentage.</td>
</tr>
<tr>
<td>Percent Used (MB)</td>
<td>The amount of space used on the database, expressed as a percentage.</td>
</tr>
</tbody>
</table>

Metrics
The environment of the particular Sybase server as well as your work style dictate the metrics you should use to evaluate a device that is running into trouble with space. Many DBAs create devices that parallel a single, corresponding database in size, and therefore, such devices show 100% utilization. Other DBAs create large devices that are not completely utilized.

Troubleshooting
If a device has become too full, a DBA can begin the process of manually relocating databases from it onto other devices. The process of moving logs is somewhat easier and can be accomplished via singular commands (sp_logdevice).

Space Detail View
The following tabbed pages are available on the Space Detail view:

- Databases
- Devices

Databases
The following views are used on the Performance Analyst for Sybase Space drill-down page to succinctly communicate the general overall performance levels of space related metrics:
A Sybase server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The database overview displays details about the space situation for each database in a Sybase server, including the total, used, and free space. The percentage used amount for each database is also shown.

The table below describes the information available on the Database Object Summary detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Total DB Space (KB)</td>
<td>Total amount of space allocated to the database.</td>
</tr>
<tr>
<td>Free DB Space (KB)</td>
<td>Total amount of space allocated but free on the database.</td>
</tr>
<tr>
<td>Used DB Space (KB)</td>
<td>Total amount of space currently used by the database.</td>
</tr>
<tr>
<td>Percent DB Free</td>
<td>The amount of free space expressed as a percentage.</td>
</tr>
<tr>
<td>User Table Count</td>
<td>The number of user tables in the database.</td>
</tr>
<tr>
<td>Table Reserved (KB)</td>
<td>The amount of total space reserved (used and free) for tables.</td>
</tr>
<tr>
<td>Table Data (KB)</td>
<td>The amount of total space used by tables.</td>
</tr>
<tr>
<td>Table Unused (KB)</td>
<td>The amount of total unused space reserved for tables.</td>
</tr>
<tr>
<td>User Index Count</td>
<td>The number of user indexes.</td>
</tr>
<tr>
<td>Index Reserved (KB)</td>
<td>The amount of index space that is reserved.</td>
</tr>
<tr>
<td>Index Data (KB)</td>
<td>The amount of data stored in indexes.</td>
</tr>
<tr>
<td>Index Unused (KB)</td>
<td>The amount of index space that is unused.</td>
</tr>
<tr>
<td>Total Log Space (KB)</td>
<td>The total amount of log space.</td>
</tr>
<tr>
<td>Free Log Space (KB)</td>
<td>The amount of log space that is allocated but unused.</td>
</tr>
<tr>
<td>Used Log Space (KB)</td>
<td>The amount of log space that is currently in use.</td>
</tr>
<tr>
<td>Percent Log Free (KB)</td>
<td>The amount of free log space expressed as a percent.</td>
</tr>
</tbody>
</table>

**Metrics**

If a database's used space percent amount goes above 90%, and the database is dynamic in nature (meaning that users are constantly adding and modifying data), then you should take action to ensure that the database does not run out of available free space.

**Troubleshooting**

If the percent used amount of a database is approaching problematic levels, there are two ways you can rectify the situation:
• If the database device that the database currently resides on contains additional free space, you can ALTER the
  database to consume more available space on that device.

• If the database device that the database currently resides on does not contain additional free space, you can
do one of the following:

• Create a new device and issue an ALTER for the database to use that device for space in addition to those
currently used.

• Choose another existing device that has free space and ALTER the database to use that device in addition to
  those currently in use.

Database Object Detail

- **Metrics**

Tables and indexes comprise every Sybase database. The Database Object Detail view presents space-related
information about tables and indexes. You can choose to view object details by table or index. Or, view objects
matching a pattern you specify in the text box.

The table below describes the information available in the Database Object Detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>The owner of the table or index.</td>
</tr>
<tr>
<td>Table Name</td>
<td>The name of the table.</td>
</tr>
<tr>
<td>Index Name</td>
<td>The name of the index.</td>
</tr>
<tr>
<td>Index Type</td>
<td>The type of index.</td>
</tr>
<tr>
<td>Key Count</td>
<td>The number of keys.</td>
</tr>
<tr>
<td>Fill Factor</td>
<td>The percentage value of the how full each index page can be.</td>
</tr>
<tr>
<td>Reserved (KB)</td>
<td>The amount of space reserved for the object, in KB.</td>
</tr>
<tr>
<td>Index Size (KB)</td>
<td>The size of the index, in KB.</td>
</tr>
<tr>
<td>Unused (KB)</td>
<td>The amount of unused space (free space) that the object contains in KB.</td>
</tr>
<tr>
<td>Segment Name</td>
<td>The name of the segment.</td>
</tr>
<tr>
<td>Extent Estimate</td>
<td>An estimation as to the number of extents for the object.</td>
</tr>
</tbody>
</table>

**Metrics**
None.

**Devices**
The following statistics are used on the Performance Analyst for Sybase Space Drill-down Devices to succinctly
communicate the general overall performance levels of space related metrics:

- **Device Summary**
- **Device Fragment Detail**
Device Summary

- **Metrics**
- **Troubleshooting**

The Device Summary view presents space related details about a selected device located on the Sybase server.

The table below describes the information available in the Device Summary detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Name</td>
<td>The logical name of the device.</td>
</tr>
<tr>
<td>Physical Name</td>
<td>The physical name of the device.</td>
</tr>
<tr>
<td>Total Space (MB)</td>
<td>Amount of space configured when the device was created.</td>
</tr>
<tr>
<td>Used Space (MB)</td>
<td>Amount of space available to any database (via the create or alter database commands).</td>
</tr>
<tr>
<td>Free Space (MB)</td>
<td>The amount of free space left in the device.</td>
</tr>
<tr>
<td>Percent Free</td>
<td>The amount of free space for the device expressed as a percentage of total space.</td>
</tr>
<tr>
<td>Percent Used</td>
<td>The amount of used space for the device expressed as a percentage of total space.</td>
</tr>
</tbody>
</table>

**Metrics**

If a device's used space percent amount goes above 90%, you should take action to ensure that the device does not run out of available free space.

**Troubleshooting**

If the percent used amount of a device is approaching problematic levels, there are two ways you can rectify the situation:

- Create a new device and issue an ALTER for the database(s) to use that device for space in addition to those currently used.
- Choose another existing device that has free space and ALTER the database(s) to use that device in addition to those currently in use.

**Device FragmentDetail**

Space from a device is allocated among one or more databases on a Sybase server. The Device Fragment Detail section presents all databases and their space fragments that reside on a particular device.

The table below describes the information available on the Device Fragment Detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of database fragment (data only, log only, data and log).</td>
</tr>
<tr>
<td>Size (MB)</td>
<td>The size of the fragment in MB.</td>
</tr>
</tbody>
</table>
Databases - Sybase
The Database performance category view displays the following vital Sybase database statistics:

<table>
<thead>
<tr>
<th>Server Analysis - Databases</th>
<th>Bottleneck Analysis - Databases</th>
<th>Backup Analysis - Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine Analysis - Databases</td>
<td>Workload Analysis - Databases</td>
<td></td>
</tr>
</tbody>
</table>

Server Analysis - Databases
The following statistics are used on the Performance Analyst for Sybase Databases Page to succinctly communicate the general overall performance levels of database specific metrics:

- **Sample Query Response Time**
- **CPU Busy**
- **I/O Busy**

Sample Query Response Time

**Metrics**
Performance Analyst allows you to define a query that is periodically run and timed by the product. The sample query helps you gauge the response time effectiveness of the Sybase server.

**Metrics**
Because you control the complexity of the sample query, you must determine what is a “valid” response time for the query. Once you have determined that, you can create an alarm in Performance Analyst that will alert you whenever the sample query falls outside of its acceptable baseline.

CPU Busy

**Metrics**

**Troubleshooting**
The CPU Busy parameter shows the percentage of the sample interval that the Adaptive Server was busy.

**Metrics**
None.

**Troubleshooting**
If you find that your CPU Busy value consistently stays above 70%, consider adding additional hardware.

I/O Busy

**Metrics**

**Troubleshooting**
The I/O Busy statistic represents the number of clock ticks in the sample interval during which the user task performed I/O operations.
Metrics
If the numbers in this category are high, it indicates an I/O-intensive process. If idle time is also high, the application could be I/O bound.

Troubleshooting
The application might achieve better throughput if you assign it a higher priority, bind it to a lightly loaded engine or engine group, or partition the application's data onto multiple devices.

Bottleneck Analysis - Databases
The following statistics are used on the Performance Analyst for Sybase Databases page to succinctly communicate the general overall performance levels of database specific metrics:

<table>
<thead>
<tr>
<th>Databases with Suspect Pages</th>
<th>Engine Configuration Limit</th>
<th>Index Maintenance from Deletes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Maintenance from Insert/Updates</td>
<td>Operating System Limit</td>
<td>Page Splits</td>
</tr>
<tr>
<td>Server Configuration Limit</td>
<td>Suspect Databases</td>
<td>Task Context Switches</td>
</tr>
</tbody>
</table>

Databases with Suspect Pages

Metrics
Suspect objects normally indicate an internal problem in a server. They can also indicate that physical damage has occurred to part of a database.

Metrics
Suspect objects have no place in a production database. If you identify any suspect objects, you should take immediate action.

Troubleshooting
If the suspect object is an index, you could try dropping and recreating it, or use the DBCC REINDEX command. Other damaged objects can complete rebuilding the database. A suspect database can be a difficult thing to recover from quickly. There are times when the cache of the database is suspect and not the database itself. Stopping and starting the Sybase server can verify if this is the case. If the database itself is actually damaged, there could be a true recovery situation. The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You would then need to recreate the database and perform a recovery operation using the most recent database dump.

Engine Configuration Limit

Metrics
An engine can exceed its limit for outstanding asynchronous disk I/O requests based on the maximum number of asynchronous I/Os per engine parameter.

Metrics
You can change this limit with the maximum asynchronous I/Os per engine configuration parameter.

Index Maintenance from Deletes

Metrics
You can change this limit with the maximum asynchronous I/Os per engine configuration parameter.
Indexes can considerably speed data access operations. The trade-off, however, is that indexes require maintenance whenever indexed columns are added to, modified, or removed in the parent table. The index maintenance from insert/updates metric provides a count of how many index maintenance operations have occurred from insert or update operations.

**Metrics**
Some index maintenance activity is to be expected (unless you are operating in a read-only environment), but excessive index maintenance can quickly become a bottleneck in heavy OLTP environments.

**Troubleshooting**
If you observe a lot of index maintenance activity, you should begin a periodic review of your indexing schemes to ensure that all indexes currently maintained on the system are used and necessary. If you are using Sybase 12.5.03 or later (and have installed the monitoring tables), you can use Performance Analysts object I/O drill-down view to diagnose which tables and indexes are the targets of heavy maintenance activity.

---

**Index Maintenance from Insert/Updates**

**Metrics**
Some index maintenance activity is to be expected (unless you are operating in a read-only environment), but excessive index maintenance can quickly become a bottleneck in heavy OLTP environments.

**Troubleshooting**
If you observe a lot of index maintenance activity, you should begin a periodic review of your indexing schemes to ensure that all indexes currently maintained on the system are used and necessary. If you are using Sybase 12.5.03 or later (and have installed the monitoring tables), you can use Performance Analysts object I/O drill-down view to diagnose which tables and indexes are the targets of heavy maintenance activity.

---

**Operating System Limit**

**Metrics**
You should be concerned if you consistently see numbers above zero.
Troubleshooting
In most UNIX operating systems, there is a kernel parameter that limits the number of asynchronous I/Os that can occur at one time. If you continue to see nonzero numbers for this statistic, you should look into raising this limit.

Page Splits
When data is inserted or updated in a table, Sybase may reorganize the data storage in the table's index pages. When an index page becomes full, but a DML operation demands room on that page, Sybase moves half the rows to a new page to accommodate the request. This reorganization is known as a page split.

Metrics
Performance for DML actions can be impaired from page split operations. In addition, more index pages can make for longer index scan times.

Server Configuration Limit
When a Sybase server begins to experience I/O delays, the result can be a very dissatisfied user community. When such problems begin to occur, you should investigate the Sybase or operating system limits. It could be that I/O operations are being blocked by one or both limits.

The Server Configuration Limit statistic shows nonzero numbers if Sybase has exceeded its limit for the number of asynchronous disk I/O requests that can be outstanding for the server at one time.

Metrics
You should be concerned if you consistently see numbers above zero.

Troubleshooting
If you continue to see nonzero numbers for this statistic, you can raise this limit using sp_configure with the max async I/Os per server parameter.

Suspect Databases
The Suspect Databases value represents the number of suspect databases on the server.

Metrics
None.

Troubleshooting
It can be difficult to recover from a suspect database quickly. There are times when the cache of the database is suspect and not the actual database itself. Stopping and starting the Sybase server can verify if this is the case. If the database itself is actually damaged, there could be a true recovery situation. The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You would then need to recreate the database and perform a recovery operation using the most recent database dump.
Task Context Switches

• Metrics

Task Context Switches summarizes task-switch activity for all engines on SMP servers. You can use this statistic to observe the effect of reconfigurations.

Metrics

You might reconfigure a cache or add memory if tasks appear to block on cache search misses and to be switched out often. Then, check the data to see if tasks tend to be switched out more or less often.

Backup Analysis - Databases

The following statistics are used on the Performance Analyst for Sybase Databases page to succinctly communicate the general overall performance levels of database specific metrics:

• Backup Server Status
• Last Transaction Log Dump

Backup Server Status

• Metrics

The Backup Server Status value indicates whether or the Backup Server is up and running.

Metrics

None.

Last Transaction Log Dump

• Metrics

The Last Transaction Log Dump value represents the last time the transaction was dump for the associated database.

Metrics

None.

Engine Analysis - Databases

The following statistics are used on the Performance Analyst for Sybase Databases page to succinctly communicate the general overall performance levels of database specific metrics:

• Engine Busy %
• Maximum Engines

Engine Busy %

• Metrics

• Troubleshooting
The Engine Busy % value represents the percentage of time the Adaptive Server Kernel is busy executing tasks on each Adaptive Server engine (rather than time spent idle).

**Metrics**
When Adaptive Server has no tasks to process, it enters a loop that regularly checks for network and completed disk I/Os, and tasks in the run queue. Operating system commands to check CPU activity may show high usage for an Adaptive Server engine because they are measuring the looping activity, while Engine Busy % does not include time spent looping it is considered idle time.

**Troubleshooting**
This category can help you decide whether there are too many or too few Adaptive Server engines. Adaptive Server's high scalability is due to tunable mechanisms that avoid resource contention. If values are consistently very high (more than 90%), it is likely that response time and throughput could benefit from an additional engine.

**Maximum Engines**

- **Metrics**

The Maximum Engines value represents the maximum number of engines currently connected.

**Metrics**
None.

**Workload Analysis - Databases**
The following statistics are used on the Performance Analyst for Sybase Databases page to succinctly communicate the general overall performance levels of database specific metrics:

- **Workload Analysis**

**Workload Analysis**
The Sybase Server is broken up into many different databases, each of which has its own levels of activity. The Workload Analysis section displays a summary of resource consumption across all databases.

The table below describes the information available in the Workload Analysis section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Name</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Connections</td>
<td>The number of connections to the database.</td>
</tr>
<tr>
<td>Blocked Users</td>
<td>The number of users currently blocked on the associated database.</td>
</tr>
<tr>
<td>Total Memory</td>
<td>The total amount of memory currently being used by the database.</td>
</tr>
<tr>
<td>Total I/O</td>
<td>The total amount of I/O currently being registered by the database.</td>
</tr>
<tr>
<td>Total CPU</td>
<td>The total amount of cumulative CPU for all processes connected to the database.</td>
</tr>
</tbody>
</table>

**Metrics**
None.
Database Detail View - Sybase

The Databases detail view displays statistics on the following tabbed pages:

<table>
<thead>
<tr>
<th>Overview</th>
<th>Objects</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engines</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Overview Tab

The following statistics are used on the Performance Analyst for Sybase Databases Detail Overview Page to succinctly communicate the general overall performance levels of database specific metrics:

<table>
<thead>
<tr>
<th>Database Size Analysis</th>
<th>Database Object Analysis</th>
<th>Database User Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Summary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Database Size Analysis

- **Metrics**

The Database Size Analysis chart displays each database in terms of its total percentage of size to the total size of all databases on the Sybase Server.

**Metrics**

None.

Database Object Analysis

The Database Object Analysis chart displays each database in terms of its total percentage of objects to the total objects for all databases on the Sybase Server.

**Metrics**

None.

Database User Analysis

- **Metrics**

The Database User Analysis chart displays each database in terms of its total percentage of user accounts to the total user accounts of all databases on the Sybase Server.

**Metrics**

None.

Database Summary

- **Metrics**

The Database Summary Analysis section displays summarized metadata for all defined databases on the Sybase Server.

**NOTE:** Right-click any column heading to sort the column(s) in ascending or descending order.
The information available in the Database Summary table is described below:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The name of the database.</td>
</tr>
<tr>
<td>Created</td>
<td>Gives the month/day/year and hour/minutes/second on which the database was first generated.</td>
</tr>
<tr>
<td>Owner</td>
<td>Lets you see who owns the database.</td>
</tr>
<tr>
<td>Database Size (MB)</td>
<td>Displays the size of the database in megabytes.</td>
</tr>
<tr>
<td>Log Size (MB)</td>
<td>Displays the log size in megabytes.</td>
</tr>
<tr>
<td>Tables</td>
<td>Gives the number of tables in the database.</td>
</tr>
<tr>
<td>Indexes</td>
<td>Displays the number of indexes in the table.</td>
</tr>
<tr>
<td>Users</td>
<td>Lets you see the number of database users.</td>
</tr>
<tr>
<td>Last Trans Log Backup</td>
<td>Let you see the last month/day/year and hour/minutes/second on which the transaction logs were most recently backed-up.</td>
</tr>
<tr>
<td>Suspect</td>
<td>Lets you see in YES or NO format whether the database is suspect.</td>
</tr>
<tr>
<td>Suspect Pages</td>
<td>Lets you see in YES or NO format whether there are any suspect pages in the database.</td>
</tr>
</tbody>
</table>

Metrics
The following items merit attention:

- Any critical database that shows a last transaction log backup date that is older than the database's required backup needs.
- Any database that shows a status of offline or suspect.

Objects Tab
The following statistics are used on the Performance Analyst for Sybase Databases Detail Objects Page to succinctly communicate the general overall performance levels of database specific metrics:

| Database Object Detail | Suspect Objects | Cache Bindings |

Database Object Detail
- Metrics
The Database Object Detail view gives you detailed information for each database. Select the database you want to scrutinize and choose between reviewing tables or indexes. You can also enter a preference to see objects matching at pattern you specify in the text box. The table below describes the information available in the Database Object Detail table:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Displays the name of the table or index owner, depending on the view you selected.</td>
</tr>
<tr>
<td>Table/Index Name</td>
<td>Displays the name of the table or index, depending on the view you selected.</td>
</tr>
<tr>
<td>Segment Name</td>
<td>Lets you see associated table/index segment.</td>
</tr>
</tbody>
</table>
Suspect Objects

Suspect objects normally indicate an internal problem in a server or that physical damage has occurred to part of a database. The table below describes the information available in the Suspect Objects grid:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Displays the name of the table owner.</td>
</tr>
<tr>
<td>TableName</td>
<td>The affected table’s name.</td>
</tr>
<tr>
<td>IndexName</td>
<td>The affected index’s name.</td>
</tr>
</tbody>
</table>

Metrics

Suspect objects have no place in a production database. If any suspect objects are identified, you should immediately take action.

Troubleshooting

If the suspect object is an index, you could try dropping and re-creating it, or use the DBCC REINDEX command. Other damaged objects can complete rebuilding the database. A suspect database can be a difficult thing to recover from quickly. There are times when the cache of the database is suspect and not the database itself. Stopping and starting the Sybase server can verify if this is the case. If the database itself is actually damaged, there could be a true recovery situation. The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You would then need to re-create the database and perform a recovery operation using the most recent database dump.

Cache Bindings

The Sybase cache areas are devoted to facilitating the transfer of data and information between clients and the Sybase database. Tables and/or indexes can be assigned to various caches in hopes of speeding up access to table or index information. The Cache Bindings Detail section provides information about the current cache bindings of tables and/or indexes. The table below describes the information available in the Cache Bindings chart:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locking Scheme</td>
<td>TABLE ONLY: Lets you see the type of locking scheme the table uses.</td>
</tr>
<tr>
<td>Last Statistic Date</td>
<td>TABLE ONLY: Lets you see the most recent date and time table statistics were gathered.</td>
</tr>
<tr>
<td>Row Count</td>
<td>TABLE ONLY: Lets you see the number of rows in the target table.</td>
</tr>
<tr>
<td>Reserved (KB)</td>
<td>Lets you see the amount of space reserved for the target object in kilobytes.</td>
</tr>
<tr>
<td>Avg Row Size</td>
<td>TABLE ONLY: Lets you see the average row size for the target table.</td>
</tr>
<tr>
<td>Tree Depth</td>
<td>INDEX ONLY: Lets you see the number of layers, or depth, of the index.</td>
</tr>
<tr>
<td>Leaf Count</td>
<td>INDEX ONLY: Lets you see the number of leaf-level pages in the index.</td>
</tr>
<tr>
<td>Leaf Row Size</td>
<td>INDEX ONLY: Lets you see the row size for each index leaf.</td>
</tr>
</tbody>
</table>
The objects best suited for being bound to a cache include frequently accessed objects that are oftentimes small in nature. Large tables that are scanned are normally not suitable for being bound to a cache and are best left to being indexed properly.

Configuration Tab
The following statistics are used on the Performance Analyst for Sybase Databases Detail Configuration Page to succinctly communicate the general overall performance levels of database specific metrics:

- Database Configuration
- Server Configuration

Database Configuration
- Metrics

The Database Configuration view allows you to see all the settings currently configured parameters and their settings on the associated database. Select a database from the drop-down list to see the parameter/setting information.

Metrics
None.

Server Configuration
- Metrics

The Server Configuration view allows you to see all the settings of currently configured parameters on the Sybase Server. Select a database from the drop-down list to see the following information in the Server Configuration grid:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Lets you see the name of the configured parameter.</td>
</tr>
<tr>
<td>Configured Value</td>
<td>Displays the configured value of the parameter.</td>
</tr>
<tr>
<td>Run Value</td>
<td>Displays the run value.</td>
</tr>
<tr>
<td>Minimum</td>
<td>Displays the minimum configuration value.</td>
</tr>
<tr>
<td>Maximum</td>
<td>Displays the maximum configuration value.</td>
</tr>
<tr>
<td>Dynamic</td>
<td>Indicates whether or not the parameter is dynamic or not.</td>
</tr>
</tbody>
</table>
Metrics
None.

Engines Tab
The following statistic is used on the Performance Analyst for Sybase Databases Detail Engines Page to succinctly communicate the general overall performance levels of database specific metrics:

• Database I/O

Database I/O

• Metrics

The Database I/O view on the Engines tab of the Databases Detail view allows you to see statistics about database objects. The table below describes the information available to you about Database I/O:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the engine.</td>
</tr>
<tr>
<td>Start Date/Time</td>
<td>The date and time when the Sybase engine started.</td>
</tr>
<tr>
<td>Stop Date/Time</td>
<td>The date and time when the Sybase engine stopped</td>
</tr>
<tr>
<td>OSPID</td>
<td>The operating system process ID.</td>
</tr>
<tr>
<td>KPID</td>
<td>The kernel process ID.</td>
</tr>
<tr>
<td>Status</td>
<td>The engine’s current status and whether it is connected or not.</td>
</tr>
<tr>
<td>Context Switches</td>
<td>The current number of context switches registering on the associated engine.</td>
</tr>
<tr>
<td>Current Connections</td>
<td>The current number of connections registering on the associated engine.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The current amount of CPU time consumed by the associated engine.</td>
</tr>
<tr>
<td>CPU Time (System)</td>
<td>The current amount of system CPU time.</td>
</tr>
<tr>
<td>CPU Time (User)</td>
<td>The current amount of user CPU time.</td>
</tr>
</tbody>
</table>

Metrics
None.

Users Statistics - Sybase
The Users performance category view displays the following vital Sybase user statistics:

<table>
<thead>
<tr>
<th>Key User Analysis - Users</th>
<th>Bottleneck Analysis - Users</th>
<th>Index Scan Analysis - Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Analysis - Users</td>
<td>Workload Analysis - Users</td>
<td></td>
</tr>
</tbody>
</table>
Key User Analysis - Users

The following statistics are used on the Performance Analyst for Sybase User page to succinctly communicate the general overall performance levels of user specific metrics:

<table>
<thead>
<tr>
<th>Active user processes</th>
<th>Extended Procedure Requests</th>
<th>Inactive user processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Queries</td>
<td>Procedure Requests</td>
<td>System processes</td>
</tr>
</tbody>
</table>

Active User Processes

• **Metrics**

The Active User Processes statistic is the total number of active and open threads reported on the Server. Active Sessions displays the number of processes actively performing work.

**Metrics**

None.

Extended Procedure Requests

• **Metrics**

The Extended Procedure Requests value represents the number of times extended procedures were executed in the sample interval.

**Metrics**

None.

Inactive User Processes

• **Metrics**

• **Troubleshooting**

The Inactive Sessions statistic represents the total number of threads logged onto the server that are idle at the current time.

**Metrics**

A large number of inactive users could indicate user sessions that have mistakenly been left logged on. Since each user thread consumes a portion of memory on the server, to reduce resource usage, you should sever any sessions not needing a connection.

**Troubleshooting**

Drill down into the Session Waits page and check sessions that have many seconds idle or that have been logged on for very long periods of time, as indicated by the logon time column. After verifying that a session is no longer necessary, you can KILL it.

Parallel Queries

• **Metrics**

The Parallel Queries value represents the total number of queries eligible to be run in parallel.
Metrics
The optimizer determines the best plan, and decides whether a query should be run serially or in parallel and how many worker processes should be used for parallel queries.

Procedure Requests

- **Metrics**

The Procedure Requests value represents the number of times stored procedures were executed in the sample interval.

- **Metrics**

None.

System Processes

- **Metrics**

The System processes value represents the total number of threads logged onto the server that are Sybase internal processes.

- **Metrics**

None.

Bottleneck Analysis - Users

The following statistics are used on the Performance Analyst for Sybase User Home Page to succinctly communicate the general overall performance levels of user specific metrics:

<table>
<thead>
<tr>
<th>Average Hash Chain Length</th>
<th>Average Lock Contention %</th>
<th>Blocked Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Locks</td>
<td>Deadlock %</td>
<td>Deadlocks</td>
</tr>
<tr>
<td>Lock Promotions</td>
<td>Lock Timeouts</td>
<td>Total Priority Changes</td>
</tr>
</tbody>
</table>

Average Hash Chain Length

- **Metrics**

- **Troubleshooting**

The Average Hash Chain Length value represents the average number of locks per hash bucket during the sample interval.

- **Metrics**

None.

**Troubleshooting**

If the average number of locks per hash chain is more than four, consider increasing the size of the hash table. Large inserts with bulk copy are an exception to this guideline. Lock hash chain lengths may be longer during large bulk copies.
The Average Lock Contention % represents the average number of times there was lock contention as a percentage of the total number of lock requests.

**Metrics**
None.

**Troubleshooting**
If the lock contention average is high, investigate the locks to see if the application is causing a lock contention or deadlock-related problem.

**Blocked Connections**
A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although Sybase supports flexible locking mechanisms, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

The blocked statistic displays the number of current processes blocked by other processes.

**Metrics**
While the lock contention rate is a better measure of the overall lock contention situation, if you consistently see positive numbers for the blocked statistic, you should suspect there is a bottleneck for some processes. You can easily drill down and discover the exact process(es) holding locks that are blocking other user activity.

**Troubleshooting**
Once you discover a blocking lock situation, you can normally remedy it by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing, and usually results in other user processes completing instantly. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in Sybase. You can change this behavior by modifying the lock wait period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

**Current Locks**
There are varieties of operations in Sybase that require the use of locks. The Current Locks statistic represents the number of total locks currently active in Sybase.
Metrics
You should watch to see if the if the number approaches the Sybase limit for available locks.

Troubleshooting
If the number of current locks in a Sybase server approaches the Sybase limit for available locks, you can look into increasing the Number of Locks configuration parameter.

Deadlock %

- Metrics
- Troubleshooting

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process' page or object. Each waits for the other to release the necessary lock. Sybase constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates the transaction. The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

The deadlock % rate displays the percentage of times deadlocks occurred vs. the percentage of locks requested and immediately granted.

Metrics
You should immediately investigate a percentage much above zero to prevent the situation from mushrooming. You can easily drill down and discover the exact process(es) holding locks and deadlocks that are blocking out other user activity.

Troubleshooting
Well-designed applications can minimize deadlocks by always acquiring locks in the same order. You should always do updates to multiple tables in the same order.

Once Sybase discovers a deadlock, it takes action and remedies the situation. Embarcadero Performance Analyst makes it easier to discover how prevalent deadlock activity is on a system; preventing deadlocks from occurring in the first place is more difficult.

Those responsible for writing systems can minimize deadlocks by ensuring that applications acquire locks in the same order. Likewise, you should always do updates and other DML that act on multiple tables in the same order.

You can also shrink the amount of time that Sybase waits to check for deadlocks by modifying the deadlock checking period configuration parameter.

Deadlocks

- Metrics
- Troubleshooting

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process' page or object. Each waits for the other to release the necessary lock. Sybase constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates it (the transaction). The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

The deadlock statistic displays the number of current deadlocks in a Sybase server.
Metrics
If you see a nonzero number, you should suspect that your user community is experiencing application conflicts.

Troubleshooting
Well-designed applications can minimize deadlocks by always acquiring locks in the same order. Updates to multiple tables should always be performed in the same order.

Once Sybase discovers a deadlock, it takes action and remedies the situation. Discovering how prevalent deadlock activity is on a system is made easier by using tools like Performance Analyst, but preventing deadlocks from occurring in the first place is more difficult.

Those responsible for writing systems can minimize deadlocks by ensuring that applications acquire locks in the same order. Likewise, updates and other DML that act on multiple tables should always be performed in the same order.

You can also shrink the amount of time that Sybase waits before performing a deadlock check by reducing the deadlock checking period parameter.

Lock Promotions

- Metrics
- Troubleshooting

The Lock Promotions value represents the average number of lock promotion types combined per second and per transaction.

Metrics
Lock promotions occur when the following escalations take place:

- "Ex-Page to Ex-Table" - Exclusive page to exclusive table.
- "Sh-Page to Sh-Table" - Shared page to shared table.
- "Ex-Row to Ex-Table" - Exclusive row to exclusive table.
- "Sh-R to Sh-Table" - Shared row to shared table.
- "Sh-Next-Key to Sh-Table" - Shared next-key to shared table.

Troubleshooting
If lock contention is high and lock promotion is frequent, you should consider changing the lock promotion thresholds for the tables involved. You can configure the lock promotion threshold either server-wide or for individual tables.

Lock Timeouts

- Metrics
- Troubleshooting

The Lock Timeouts value represents the total number of locks timing out during the sample period.

Metrics
You can specify the time that a task waits for a lock:

- At the server level, with the lock wait period configuration parameter
- For a session or in a stored procedure, with the set lock wait command
For a lock table command

**Troubleshooting**

Except for lock table, a task that attempts to acquire a lock and fails to acquire it within the time period returns an error message and the transaction is rolled back. Using lock timeouts can be useful for removing tasks that acquire some locks, and then wait for long periods of time blocking other users. However, since transactions are rolled back, and users may simply resubmit their queries, timing out a transaction means that the work needs to be repeated.

**Total Priority Changes**

- **Metrics**
- **Troubleshooting**

Total Priority Changes represent the total number of priority changes during the sample period.

**Metrics**

It is normal to see some priority switching due to system-related activity. Such priority switching occurs, for example, when:

- A task sleeps while waiting on a lock - Adaptive Server temporarily raises the task's priority.
- The housekeeper task sleeps - Adaptive Server raises the priority to medium while the housekeeper sleeps, and changes it back to low when it wakes up.
- A task executes a stored procedure - the task assumes the priority of the stored procedure and resumes its previous priority level after executing the procedure.

**Troubleshooting**

If you are using logical process management and there are a high number of priority changes compared to steady state values, it may indicate that an application, or a user task related to that application, is changing priorities frequently. Check priority change data for individual applications. You should verify that applications and logins are behaving expected.

If you determine that a high-priority change rate is not due to an application or to related tasks, then it is likely due to system activity.

**Index Scan Analysis - Users**

The following statistics are used on the Performance Analyst for Sybase User page to succinctly communicate the general overall performance levels of user specific metrics:

<table>
<thead>
<tr>
<th>Ascending</th>
<th>Descending</th>
<th>DOL Ascending</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOL Descending</td>
<td>Total Index Scans</td>
<td></td>
</tr>
</tbody>
</table>

**Ascending**

- **Metrics**

The Ascending value represents the number of forward scans on allpages-locked tables.

**Metrics**

None.
Descending

Metrics

The Descending statistic represents the number of backward scans on allpages-locked tables.

DOL Ascending

Metrics

The DOL Ascending value represents the number of forward scans on data-only-locked tables.

DOL Descending

Metrics

The DOL Descending value represents the number of backward scans on data-only-locked tables.

Total Index Scans

Metrics

The Total Index Scans statistic represents the total number of index scans (forward and backward).

Transaction Analysis - Users

The following statistics are used on the Performance Analyst for Sybase User page to succinctly communicate the general overall performance levels of user-specific metrics:

Active Transactions

Committed Transactions

Active Transactions

Metrics

The Active Transaction statistic represents a count of the number on in-process transactions for SQL Server.
Metrics
None.

Committed Transactions

Committed Transactions gives the number of transactions committed since the last refresh inside Performance Analyst. This includes transactions that meet explicit, implicit, and ANSI definitions for committed transactions. Note that multi-database transactions are counted.

Metrics

Multi-database transactions generally incur more overhead than single database transactions (for example, a transaction that modifies two databases is counted as two transactions). They usually involve more log activity and two-phase commits between the different databases, as well as cause more ULC flushes. You can improve performance by reducing the number of multi-database transactions.

Workload Analysis - Users

The following statistics are used on the Performance Analyst for Sybase User page to succinctly communicate the general overall performance levels of user specific metrics:

Workload Analysis

When your phone starts ringing with complaints of performance slowdowns, some of the first questions you should answer are:

- What users are currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst page provides insight into the leading resource hogs of a server.

The table below describes the information available in the Workload Analysis section of the Users page:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>Statistics are presented for the Top CPU Process, Top I/O Process, or Top Memory Process.</td>
</tr>
<tr>
<td>SPID</td>
<td>The process ID.</td>
</tr>
<tr>
<td>Login</td>
<td>The logon name the session is using.</td>
</tr>
<tr>
<td>% Used</td>
<td>The percentage of CPU, I/O, or Memory consumed by the process.</td>
</tr>
</tbody>
</table>

Metrics

If any one session appears to be using more than 50% of a total resource (CPU, memory, etc.), then you should drill down into the session to find out what they are currently executing.
Users Detail View

The following tabbed pages are available on the Users Detail view:

<table>
<thead>
<tr>
<th>Top Sessions</th>
<th>Locks</th>
<th>Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Waits</td>
<td>Session Waits</td>
<td></td>
</tr>
</tbody>
</table>

Top Sessions

The following statistics are used on the Performance Analyst for Sybase Users Drill-down Top Sessions to succinctly communicate the general overall performance levels of user specific metrics:

<table>
<thead>
<tr>
<th>Top I/O Processes</th>
<th>Top Memory Processes</th>
<th>Top CPU Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Sessions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related Topic

Session Details

Top I/O Processes

- **Metrics**
- **Troubleshooting**

The Top I/O Processes statistic identifies the current Sybase process with the highest % of I/O activity.

**Metrics**

None.

**Troubleshooting**

If any one session uses more than 50% of a total resource (CPU, memory, etc.), you should drill down into that particular session and investigate the cause.

Top Memory Processes

- **Metrics**

The Top Memory Process statistic identifies the Sybase process that currently is using the highest percentage of memory in the database.

**Metrics**

None.

Top CPU Processes

- **Metrics**

The Top CPU Process statistic identifies the Sybase process that currently has the highest percentage of CPU usage in the database.
Metrics
None.

Top Sessions

When a system experiences heavy activity, sometimes you will find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are causing most of the activity. It might be that a large batch load or other typical process is running that is perfectly okay for your system. Or it might be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

The table below describes the information available on the Top Sessions detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The process ID.</td>
</tr>
<tr>
<td>User</td>
<td>The name of the session user.</td>
</tr>
<tr>
<td>FID</td>
<td>The process ID of the worker process’ parent.</td>
</tr>
<tr>
<td>Status</td>
<td>The current status of the session.</td>
</tr>
<tr>
<td>Database</td>
<td>The database the session is attached to.</td>
</tr>
<tr>
<td>Host</td>
<td>The client machine name the session is using.</td>
</tr>
<tr>
<td>Program</td>
<td>This identifies the program being run by the session.</td>
</tr>
<tr>
<td>Mem Usage</td>
<td>The amount of memory being used by the session.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The cumulative CPU time for the process.</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>The number of physical reads produced by the session.</td>
</tr>
<tr>
<td>Logical Reads</td>
<td>The number of logical reads produced by the session.</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>The number of physical writes generated by the session.</td>
</tr>
</tbody>
</table>
To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, Sybase itself issues lock requests to carry out its internal duties. The Locks section gives information about the locks currently on the system and also indicates if any blocking situations are occurring. You can elect to show only blocking locks in the Lock detail view.

The table below describes the information available on the Locks detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The process ID.</td>
</tr>
<tr>
<td>User</td>
<td>The name of the current user</td>
</tr>
<tr>
<td>Database</td>
<td>The name of the database where the process is running.</td>
</tr>
<tr>
<td>Lock Type</td>
<td>The type of lock (database, table, row id, etc.).</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the object involved in the lock.</td>
</tr>
<tr>
<td>Status</td>
<td>The lock's status (waiting or granted).</td>
</tr>
<tr>
<td>Lock Page</td>
<td>The number of pages, if any, that are experiencing locks.</td>
</tr>
<tr>
<td>Lock Class</td>
<td>Lets you see the lock class.</td>
</tr>
<tr>
<td>Host</td>
<td>The client machine name the session is using.</td>
</tr>
<tr>
<td>Program</td>
<td>The executable the process is using against the server.</td>
</tr>
<tr>
<td>Command</td>
<td>The command currently being issued by the process.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The amount of time the process took to parse, execute, and fetch the data needed to satisfy the process’ execution.</td>
</tr>
<tr>
<td>I/O</td>
<td>The amount of I/O activity for the target process.</td>
</tr>
<tr>
<td>Mem Usage</td>
<td>The amount of memory being used by the target process.</td>
</tr>
<tr>
<td>FID</td>
<td>The process ID of the worker process’ parent.</td>
</tr>
<tr>
<td>Transaction</td>
<td>The current transactions generated by processes on the Sybase Server</td>
</tr>
</tbody>
</table>

**Metrics**
A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period, a process waiting for a lock will wait indefinitely.

**Troubleshooting**
Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then nearly always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.
Links by Database

Metrics
None.

Locks by Lock Type

Metrics
None.

Transactions

Metrics

Troubleshooting

The Transactions view displays current transactions generated by processes on the Sybase Server.

The table below describes the information available on the Transactions detail view:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The process ID.</td>
</tr>
<tr>
<td>User</td>
<td>The name of the user generating the session.</td>
</tr>
<tr>
<td>Status</td>
<td>The current status of the session.</td>
</tr>
<tr>
<td>Start Time</td>
<td>The time the session started.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of session.</td>
</tr>
<tr>
<td>State</td>
<td>The current state of the session, for example, whether or not it is active.</td>
</tr>
<tr>
<td>Command</td>
<td>The command currently being issued by the session.</td>
</tr>
<tr>
<td>Connection</td>
<td>The amount of memory allocated to the session’s connections</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The amount of time the process took to parse, execute, and fetch the data</td>
</tr>
<tr>
<td>Mem Usage</td>
<td>The amount of memory being used by the session.</td>
</tr>
<tr>
<td>Physical I/O</td>
<td>The amount of physical I/O for the session.</td>
</tr>
</tbody>
</table>

Metrics

Transaction activity is a good measurement of general performance by a user. If the total or committed numbers do not increase over a period of time, either the process is inactive or it may be hung or blocked.
Troubleshooting
Look at CPU and I/O statistics to see if any activity for the process is being noted there.

System Waits
The following statistics are used on the Performance Analyst for Sybase User Drill-down System Waits tab to succinctly communicate the general overall performance levels of user specific metrics:

<table>
<thead>
<tr>
<th>System Waits</th>
<th>Wait Percent by Total Waits (By Wait Class)</th>
<th>Wait Percent by Time Waited (By Wait Class)</th>
</tr>
</thead>
</table>

System Waits

- **Metrics**

When the Sybase Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The System Waits section identifies the waits that have occurred on the Sybase Server.

The table below describes the information available on the System Waits detail view. You can elect to display the internal system and scheduler events.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait Event</td>
<td>The name of the wait event.</td>
</tr>
<tr>
<td>Wait Class</td>
<td>The type of wait, often called a wait category or class.</td>
</tr>
<tr>
<td>Wait Count</td>
<td>The number of waits that have occurred.</td>
</tr>
<tr>
<td>Percent Waits</td>
<td>The percent of waits as compared to the total number of waits.</td>
</tr>
<tr>
<td>Wait Time</td>
<td>The amount of time waited in seconds.</td>
</tr>
<tr>
<td>Percent Wait Time</td>
<td>The percent of wait time as compared to the total wait time.</td>
</tr>
</tbody>
</table>

**Metrics**
None.

Wait Percent by Total Waits (By Wait Class)

- **Metrics**

When the Sybase Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Wait Percent by Total Waits section graphically depicts the top waits by their percentage to all waits.

**Metrics**
None.

Wait Percent by Time Waited (By Wait Class)

- **Metrics**
When the Sybase Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Wait Percent by Time Waited section graphically depicts the top waits by their percentage of time waited to total time waited.

**Metrics**
None.

**Session Waits**
The following statistics are used on the Performance Analyst for Sybase User Drill-down Locks tab to succinctly communicate the general overall performance levels of user specific metrics:

| Session Waits | Wait Percent by Total Waits (By Wait Class) | Wait Percent by Time Waited (By Wait Class) |

**Session Waits**
- **Metrics**
- **Troubleshooting**

Session waits is merely a subset of contention that is viewable at the global database level. Often it takes analysis at the session level to pinpoint the exact source of the wait that is occurring globally. Therefore, you need to become accustomed to viewing contention statistics and waits at the user process level.

When monitoring waits with respect to user sessions, there are two areas of interest:

- What HAS the user session been waiting on?
- What IS the user session waiting on?

The table below describes the information available on the Session Waits detail view. You can elect to include background process in the detail display.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The unique identifier for the process.</td>
</tr>
<tr>
<td>User</td>
<td>The name of the process’ user.</td>
</tr>
<tr>
<td>FID</td>
<td>The process ID of the worker process’ parent.</td>
</tr>
<tr>
<td>Status</td>
<td>The current status of the process.</td>
</tr>
<tr>
<td>Database</td>
<td>The name of the database where the process is running.</td>
</tr>
<tr>
<td>Hostname</td>
<td>The client machine name where the session is running.</td>
</tr>
<tr>
<td>Program Name</td>
<td>The name of the program being run by the process.</td>
</tr>
<tr>
<td>Wait Event</td>
<td>The name of the wait event.</td>
</tr>
<tr>
<td>Wait Class</td>
<td>The type of wait, often called a wait category or class.</td>
</tr>
<tr>
<td>Waits</td>
<td>The number of waits that have occurred.</td>
</tr>
<tr>
<td>Process Waits</td>
<td>The percent of waits as compared to the total number of waits.</td>
</tr>
<tr>
<td>Wait Time (s)</td>
<td>The number of seconds the process spent waiting for this wait.</td>
</tr>
</tbody>
</table>
Metrics
To determine the actual wait causes currently experienced by user connections, you should drill down from the global count of users waiting, into the actual system and user wait details.

Troubleshooting
If you find a problem, drill down into wait details to determine whether the waits are resource-related.

Wait Percent by Total Waits (By Wait Class)
The Wait Percent by Total Waits section graphically depicts the waits that have experienced the highest percentage of occurrences on the system.

Wait Percent by Time Waited (By Wait Class)

• Metrics

When the Sybase Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Session Wait Percent by Time Waited section graphically depicts the top session waits by their percentage of time waited to total time waited.

Metrics
None.

Session Details
The Sessions Detail view includes the following tabbed pages:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Waits</td>
<td>The percent of wait time as compared to the total wait time.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQL</th>
<th>I/O</th>
<th>Object Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locks/Waits</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SQL
The SQL tab of the Sessions Detail view offers the following information:

• Current SQL
  • Current Procedures

Current SQL
This view shows a snippet of the SQL associated with the user in question.

Current Procedures
Here, in tabular format, you find the following information about the user’s current procedures:
Owner
Object Name
Object Type

**Database:** The name of the database on which the procedure resides.

**Compile Date:** The last date on which the object was compiled.

**Memory (KB):** The number of pages in the procedure cache that is currently allocated to the process. A negative number indicates that pages are being released (freed) from the process.

**I/O**
The I/O tab of the Session Detail view displays the following statistical information about the current user’s activities in a graphical display:

- **Physical Reads**
- **Logical Reads**
- **Pages Read**
- **Pages Written**
- **Physical Writes**
- **Table Accesses (scans):** The number of table accesses (scans) by the user.
- **Index Accesses:** The number of index accesses by the user.
- **Tempdb Objects:** The number of times the Tempdb was accessed by the user.

**Object Access**
The Object Access tab of the Session Detail view displays information about the current objects being accessed. Here you’ll see, in tabular format:

- **Database:** The name of the database
- **Owner ID:** The ID number assigned to the process accessing the object.
- **Object Name**
- **Object Type:** The type of object being accessed.
- **Index ID**
- **Physical Reads:** The number of physical disk reads issued against the object.
- **Physical APF Reads:** Number of physical Asynchronous Pre-Fetch buffers read.
- **Logical Reads:** The number logical reads issued against the object.

**Locks/Waits**
The Lock/Waits tab of the Session Details page offers statistics on:

- **Locks**
- **Waits**
Locks
To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, Sybase itself issues lock requests to carry out its internal duties. The Locks section gives information about the locks currently on the system and also indicates if any blocking situations are occurring. You can elect to show only blocking locks in the Lock detail view.

The table below describes the information available on the Locks section of Session Details:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID</td>
<td>The process ID.</td>
</tr>
<tr>
<td>User</td>
<td>The name of the current user</td>
</tr>
<tr>
<td>Database</td>
<td>The name of the database where the process is running.</td>
</tr>
<tr>
<td>Lock Type</td>
<td>The type of lock (database, table, row id, etc.).</td>
</tr>
<tr>
<td>Object Name</td>
<td>The name of the object involved in the lock.</td>
</tr>
<tr>
<td>Status</td>
<td>The lock's status (waiting or granted).</td>
</tr>
<tr>
<td>Lock Page</td>
<td>The number of pages, if any, that are experiencing locks.</td>
</tr>
<tr>
<td>Lock Class</td>
<td>Lets you see the lock class.</td>
</tr>
<tr>
<td>Host</td>
<td>The client machine name the session is using.</td>
</tr>
<tr>
<td>Program</td>
<td>The executable the process is using against the server.</td>
</tr>
<tr>
<td>Command</td>
<td>The command currently being issued by the process.</td>
</tr>
<tr>
<td>CPU Time</td>
<td>The amount of time the process took to parse, execute, and fetch the data needed to satisfy the process’ execution.</td>
</tr>
<tr>
<td>I/O</td>
<td>The amount of I/O activity for the target process.</td>
</tr>
<tr>
<td>Mem Usage</td>
<td>The amount of memory being used by the target process.</td>
</tr>
<tr>
<td>FID</td>
<td>The process ID of the worker process’ parent.</td>
</tr>
<tr>
<td>Transaction</td>
<td>The current transactions generated by processes on the Sybase Server</td>
</tr>
</tbody>
</table>

Waits
When the Sybase Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Waits section shows processes that are currently waiting on the Sybase Server.

Here you find, in tabular format, details for:

**Login:** The logon name used by the session.

**Database:** The name of the database the process is attached to.

**Wait Event:** The name of the wait event.

**Wait Class:** The type of wait, often called a wait category or class.

**Waits:** The number of waits that have occurred.

**Wait Time (sec):** The amount of time waited in seconds.
**OS Page Statistics**

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Analyst lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX
- Linux
- Solaris
- Unix
- Windows XP and 2000

**NOTE:** The statistics available on the OS page depend on the operating system platform.

**TIP:** If you magnify a graph to change back to the original size or minimize the size, close the OS Page and then reopen it again. Performance Analyst displays the graph in its original form and size.

The OS home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Disk Analysis</th>
<th>Key Resource Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Analysis</td>
<td>Workload Analysis - Top Resource Hogs</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topic**

**OS Detail**

---

**Key Resource Usage - OS**

The following ratios are used on the Performance Analyst OS home page to communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Disk Time</th>
<th>Load Average</th>
<th>Processor Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paged Memory Used (Windows)</td>
<td>Swap Memory Used (AIX, HP-UX, Linux, Solaris, Unix)</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the CPU tab, right-click Processor Time. To open the I/O tab, right-click Disk Time. To open the Memory tab, right-click Paged Memory Used.
Bottleneck Analysis - OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Metric</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Disk Queue Length</td>
<td>Interrupts/Sec</td>
<td>Network Output Queue Length (Windows)</td>
</tr>
<tr>
<td>Network Queue Length (Solaris)</td>
<td>Page Faults/Sec</td>
<td>Processor Queue Length</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>Processor Speed</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the I/O tab, right-click any Details menu item. To open the CPU tab, right-click the Item Processor Queues Length Details menu item. To open the Network tab, right-click Network Output Queues Length.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Memory Analysis - OS
The following metrics are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Metric</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Paged Memory (Windows)</td>
<td>Available Physical Memory</td>
<td>Available Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)</td>
</tr>
<tr>
<td>Total Paged Memory (Windows)</td>
<td>Total Physical Memory</td>
<td>Total Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)</td>
</tr>
</tbody>
</table>

**TIP:** To open the Memory tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Disk Analysis - OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Metric</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free Disk Space</td>
<td>Total Disk Space</td>
<td>Used Disk Space</td>
</tr>
</tbody>
</table>

**TIP:** To open the Space tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.
Workload Analysis - Top Resource Hogs OS

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Number of Logins</th>
<th>Number of Processes</th>
<th>Top CPU Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top I/O Process</td>
<td>Top Memory Process</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the Processes tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.

**Processor Time**

The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

**Metrics**

If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.

**Troubleshooting**

Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

**Processor Speed**

The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

**Processor**

The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

**Disk Time**

The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

**Metrics**

You should avoid consistently seeing values for this statistic greater then 90%.

**Troubleshooting**

Add more disk drives and partition the files among all of the drives.

**Load Average**

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.
Metrics
High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Paged Memory Used
The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

**NOTE:** This statistic is available for the Windows platform.

Metrics
This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.

Troubleshooting
Increase the size of page file.

Swap Memory Used
The Swap Memory Used statistic is the percentage of swap space currently in use.

Metrics
If the percentage of swap memory used is above 90%, you may be running out of memory.

Troubleshooting
Increase the size of your swap files.

Average Disk Queue Length
The Average Disk Queue Length statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Metrics
This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.

Troubleshooting
Some things you can do if you have problems with this statistic include:

- Redistribute the data on the disk with the large average disk queue to other disks.
- Upgrade to faster disk(s).
Interrupts/Sec
Interrupts/Sec is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every 10 milliseconds, creating a background of interrupt activity. This statistic shows the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Metrics
A high value indicates possible excessive hardware interrupts; justification is dependent on device activity.

Page Faults/Sec
The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics
This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue Length
The Processor Queue Length statistic is the number of threads in the processor queue.

Metrics
Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Network Output Queue Length/Network Queue Length
The Network Output Queue Length statistic is the number of threads in the processor queue.

NOTE: The name of this statistic depends on the platform of the operating system.
Metrics
Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Available Physical Memory
The Available Physical Memory statistic represents the amount of RAM available to all processes.

Metrics
This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Troubleshooting
Some things you can do if you have problems with this statistic include:

- Check the running processes to see if there are any memory leaks.
- Stop any services that are not required.
- Install additional RAM.

Available Paged Memory
The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

NOTE: This statistic is available for the Windows platform.

Metrics
If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.

Troubleshooting
Increase the size of page file.

Available Swap Memory
The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Metrics
If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.
Troubleshooting
Increase the size of swap files.

Total Physical Memory
The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Metrics
This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Total Paged Memory/Total Swap Memory
The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics
It is recommended that this be 1½ - 3 times the amount of RAM on the system.

Used Disk Space
The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

• If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.

• You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Total Disk Space
Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two:

1 If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.

2 You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.
Free Disk Space
The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Top Memory Process
Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics
If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Processes Overview
The Processes Overview of the OS Summary includes the following sections:

| Top CPU Process | Top I/O Process | Top Memory Process |

Top CPU Process
Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics
If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.

Troubleshooting
Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process
The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.
Number of Logins
This statistic displays the total number of logins on the server.

Number of Processes
This statistic displays the total number of processes on the server.

OS Detail
The following tabbed pages are available on the OS Detail page:

<table>
<thead>
<tr>
<th>CPU</th>
<th>I/O</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Processes</td>
<td>Space</td>
</tr>
</tbody>
</table>

CPU Tab
The CPU tab of the OS Detail includes the following sections:

<table>
<thead>
<tr>
<th>Context Switches/Sec</th>
<th>CPU Events</th>
<th>CPU Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrupts/Sec</td>
<td>Load Averages</td>
<td>Processor Queue Length</td>
</tr>
</tbody>
</table>

NOTE: The sections available on this tab depend on the platform of operating system.

CPU Utilization
The CPU Utilization section includes the following information:

- % Privileged Time
- % User Time

% Privileged Time
The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

NOTE: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

Metrics
The ideal range should be 0-40% (less than 40% indicates excessive system activity).

Troubleshooting
If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).
% User Time
The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Metrics
If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

CPU Events
The CPU Events section includes the following information:

<table>
<thead>
<tr>
<th>Inturrupts/Sec</th>
<th>Context Switches/Sec</th>
<th>System Calls/Sec</th>
</tr>
</thead>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Inturrupts/Sec
The Inturrupts/Sec statistic is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. It does not include deferred procedure calls (DPCs), which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every ten milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Metrics
The ideal rage should be 0-5000. A number greater then 5000 indicates possible excessive hardware interrupts; justification is dependent on device activity.

Context Switches/Sec
The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Metrics
The ideal range should be between 0-10,000. GA number greater then 10,000 may indicate too many threads contending for resources.

System Calls/Sec
System calls are functions which require the services of the operating system. Usually they involve access to some data that users must not be permitted to corrupt or even change.
Troubleshooting
Try to find a good ratio between the amount of time the operating system runs (fulfilling system calls and doing process scheduling), and the amount of time the application runs. Your goal should be running 60% to 75% of the time in application mode, and 25% to 40% of the time in operating system mode. If you find that the system is spending 50% of its time in each mode, then you should investigate to determine what is wrong.

Processor Queue Length
The Processor Queue Length statistic is the number of threads in the processor queue. There is a single queue for processor time even on computers with multiple processors.

**NOTE:** For Windows systems, unlike the disk counters, this counter shows ready threads only, not threads that are running.

Metrics
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Load Averages
The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Metrics
High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Processes Tab
The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>The name of the process.</td>
</tr>
<tr>
<td>User</td>
<td>The user of the process.</td>
</tr>
<tr>
<td>ID</td>
<td>The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.</td>
</tr>
<tr>
<td>CPU</td>
<td>The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.</td>
</tr>
<tr>
<td>User Mode</td>
<td>The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.</td>
</tr>
<tr>
<td>Memory <strong>WINDOWS ONLY</strong></td>
<td>Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.</td>
</tr>
<tr>
<td>Memory (MB)</td>
<td>Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.</td>
</tr>
<tr>
<td>Memory</td>
<td>Memory is the percentage of the memory used of the total memory.</td>
</tr>
</tbody>
</table>
I/O Tab

The I/O tab includes the following sections:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Memory</td>
<td>Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.</td>
</tr>
<tr>
<td>I/O Data</td>
<td>The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>The total elapsed time, in seconds, that this process has been running.</td>
</tr>
<tr>
<td>Thread Count</td>
<td>The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.</td>
</tr>
<tr>
<td>Handle Count</td>
<td>The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.</td>
</tr>
<tr>
<td>Priority</td>
<td>The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.</td>
</tr>
<tr>
<td>Creating Proc ID</td>
<td>The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.</td>
</tr>
<tr>
<td>Page Faults/Sec</td>
<td>Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.</td>
</tr>
<tr>
<td>Page File</td>
<td>Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.</td>
</tr>
<tr>
<td>Private</td>
<td>Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.</td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of operating system.
I/O Details
The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk</td>
<td>The disk number assignment.</td>
</tr>
<tr>
<td>Reading (KB/s)</td>
<td>The amount of bytes read from the device.</td>
</tr>
<tr>
<td>Writing (KB/s)</td>
<td>The amount of bytes written to the device.</td>
</tr>
<tr>
<td>Disk Read Time</td>
<td>Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.</td>
</tr>
<tr>
<td>Disk Write Time</td>
<td>Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.</td>
</tr>
<tr>
<td>Disk Time</td>
<td>Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.</td>
</tr>
<tr>
<td>Avg. Read Queue</td>
<td>Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.</td>
</tr>
<tr>
<td>Avg. Write Queue</td>
<td>Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.</td>
</tr>
<tr>
<td>Disk Reads/Sec</td>
<td>Disk Reads/Sec is the rate of read operations on the disk.</td>
</tr>
<tr>
<td>Disk Writes/Sec</td>
<td>Disk Writes/Sec is the rate of write operations on the disk.</td>
</tr>
</tbody>
</table>

**NOTE:** The columns available in this table depend on the platform of operating system.

Transfer Rate
The Transfer Rate section succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading</td>
<td>The average number of bytes transferred from the disk during read operations.</td>
<td>It is measured in number of KBs per second.</td>
</tr>
<tr>
<td>Writing</td>
<td>The average number of bytes transferred to the disk during write operations.</td>
<td>It is measured in number of KBs per second.</td>
</tr>
<tr>
<td>Transferred</td>
<td>The amount of time required to transfer the data between the device's cache and the actual device media.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Bytes per I/O Operation
The Bytes per I/O Operation section of the OS Detail succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading</td>
<td>The average number of bytes transferred from the disk during read operations.</td>
<td>It is measured in number of bytes per second.</td>
</tr>
<tr>
<td>Writing</td>
<td>The average number of bytes transferred to the disk during write operations.</td>
<td>It is measured in number of bytes per second.</td>
</tr>
</tbody>
</table>
Average Disk Queue/Total Disk Queue
The Average Disk Queue section of the OS Detail page succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Read Queue</td>
<td>Average Read Queue Length is the average number of read requests that</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system.</td>
</tr>
<tr>
<td></td>
<td>were queued for a particular disk.</td>
<td></td>
</tr>
<tr>
<td>Average Write Queue</td>
<td>Average Write Queue is the average number of write requests that were queued</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system.</td>
</tr>
<tr>
<td></td>
<td>for a particular disk.</td>
<td>Contact your Windows System Administrator for assistance in troubleshooting these problems.</td>
</tr>
<tr>
<td>Average Queue</td>
<td>Average time in ms. to service I/O requests which for disks includes seek,</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system.</td>
</tr>
<tr>
<td></td>
<td>rotational latency and data transfer time).</td>
<td>Contact your Unix System Administrator for assistance in troubleshooting these problems.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Troubleshooting
Contact your Windows System Administrator for assistance in troubleshooting these problems.

Disk I/O Time
The Disk I/O Time section of the OS Detail page succinctly communicates the general overall performance levels of the disk's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Disk Read Time</td>
<td>% Disk Read Time is the percentage of time during the sample interval that</td>
<td>Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
<tr>
<td></td>
<td>the disk is busy servicing read requests.</td>
<td></td>
</tr>
<tr>
<td>% Disk Write Time</td>
<td>% Disk Write Time is the percentage of elapsed time that the selected disk</td>
<td>Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
<tr>
<td></td>
<td>drive was busy servicing write requests.</td>
<td></td>
</tr>
<tr>
<td>% Disk Time</td>
<td>% Disk Time is the percentage of time during the sample interval that the</td>
<td>Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
<tr>
<td></td>
<td>disk is busy servicing requests.</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.
Disk Transfers/Sec
The Disk Transfers/Sec section of the OS Detail page succinctly communicates the general overall performance levels of the disk transfers. The table below describes the statistics in this section.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads/Sec</td>
<td>The rate of read operations on the disk.</td>
</tr>
<tr>
<td>Writes/Sec</td>
<td>The rate of write operations on the disk.</td>
</tr>
<tr>
<td>Transfers/Sec</td>
<td>The rate of transfer operations per second.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Memory Tab
The Memory tab of the OS Detail page includes the following sections:

<table>
<thead>
<tr>
<th>Buffer Size</th>
<th>Cache Efficiency</th>
<th>Cache Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Available</td>
<td>Paging Activity</td>
<td>Page Faults</td>
</tr>
<tr>
<td>Page Replacements</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of operating system.

Paging Activity
The Paging Activity section includes the following statistics:

<table>
<thead>
<tr>
<th>Blocks Input/Sec</th>
<th>Blocks Output/Sec</th>
<th>Paged In</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paged Out</td>
<td>Pages Input/Sec</td>
<td>Pages Output/Sec</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Blocks Input/Sec
The Blocks Input/sec statistic is the number of blocks sent to a block device per second.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck:

1. Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.

2. Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.

Blocks Output/Sec
The Blocks Output/sec statistic is the number of blocks received from a block device per second.)
Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck.

1. Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.

2. Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.

Paged In
The number of disk block paged into core from disk which occurs when the server is experiencing a shortage of RAM. Any nonzero value indicates excessive swap activity. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Troubleshooting
High values indicate the possible need to increase the amount of RAM on the system.

Paged Out
The number of disk block paged out of core from disk. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Pages Input/Sec
The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Metrics
This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.

- Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.
Pages Output/Sec
The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Metrics
Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Page Faults/Sec
The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics
This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.
Memory Available
The Memory Available section of the OS Detail page succinctly communicates the general overall performance levels of the server's memory. The statistics available in this section depend on the platform of operating system. The table below describes these statistics:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Memory</td>
<td>The Physical Memory statistic is the amount of committed virtual memory, in megabytes. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk.</td>
<td>This counter displays the last observed value only and not an average.</td>
<td>N/A</td>
</tr>
<tr>
<td>Physical</td>
<td>The Physical statistic is the total physical memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Paged Memory</td>
<td>The Paged Memory statistic is the amount of committed virtual memory, in bytes. Used Memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.</td>
<td>This counter displays the last observed value only; it is not an average.</td>
<td>N/A</td>
</tr>
<tr>
<td>Swap Memory</td>
<td>The Swap Memory statistic is the free, unreserved swap space.</td>
<td>N/A</td>
<td>Increase the size of your swap file or add additional files if you consistently see a shortage of free, unreserved swap space.</td>
</tr>
<tr>
<td>Swap</td>
<td>The Swap statistic is the total swap memory being used.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total Physical</td>
<td>The Total Physical statistic is the total physical memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total Swap</td>
<td>The Total Swap statistic is the total swap memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Swap-ins</td>
<td>The number of memory pages paged in from swap space.</td>
<td>N/A</td>
<td>If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.</td>
</tr>
<tr>
<td>Swap-outs</td>
<td>The number of memory pages paged in from swap space.</td>
<td>N/A</td>
<td>If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.</td>
</tr>
<tr>
<td>Buffers</td>
<td>The Buffers statistic represents the amount of buffers used by the kernel in kilobytes.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
NOTE: The statistics available in this section depend on the platform of operating system.

Buffer Size
The Buffer Size statistic represents the amount of cache used by the kernel in kilobytes.

Cache Size
The Cache Size statistic represents the amount of cache used by the kernel in kilobytes.

Page Replacements
The following statistics are available in this section:

- Memory Freed (Pages/sec)
- Clock Algorithm Scans (Pages/sec)

Memory Freed (Pages/sec)
Freed or destroyed (Kb/s).

Clock Algorithm Scans (Pages/sec)
The VMM uses a technique known as the clock algorithm to select pages to be replaced. This technique takes advantage of a referenced bit for each page as an indication of what pages have been recently used (referenced). When the page-stealer routine is called, it cycles through the PFT, examining each page’s referenced bit.

Cache Efficiency
The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server’s memory. The following statistics are available in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy Read Hits%</td>
<td>The percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.</td>
</tr>
<tr>
<td>Data Map Hits%</td>
<td></td>
</tr>
<tr>
<td>MDL Read Hits%</td>
<td></td>
</tr>
<tr>
<td>Pin Read Hits%</td>
<td></td>
</tr>
</tbody>
</table>
Metrics
When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Data Map Hits %
The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Metrics
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %
The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Metrics
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %
The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Metrics
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab
The Space tab of the OS Detail page includes the following sections:
Device Summary
The Device Summary section of the Space tab displays a bar chart for all devices. The Device Summary section succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
</tbody>
</table>

Disk Space Free
The Disk Space Free metric displays the amount of free disk space in megabytes.

Metric
None.

Disk Space Detail
The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server’s disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>The drive letter of the disk.</td>
</tr>
<tr>
<td>Local Filesystem</td>
<td>The name of the file system.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of file system.</td>
</tr>
<tr>
<td>Total Space</td>
<td>Total size of the disk/device's capacity expressed in MBs.</td>
</tr>
<tr>
<td>Used Space</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free Space</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
<tr>
<td>Capacity</td>
<td>The percentage of space used on the device.</td>
</tr>
<tr>
<td>Mounted On</td>
<td>The mount point of the device.</td>
</tr>
</tbody>
</table>
Device Detail
The Device Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>The drive letter of the disk.</td>
</tr>
<tr>
<td>Local Filesystem</td>
<td>The name of the file system.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of file system.</td>
</tr>
<tr>
<td>Total Space</td>
<td>Total size of the disk/device's capacity expressed in MBs.</td>
</tr>
<tr>
<td>Used Space</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free Space</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
<tr>
<td>Capacity</td>
<td>The percentage of space used on the device.</td>
</tr>
<tr>
<td>Mounted On</td>
<td>The mount point of the device.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Network Tab
The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The Network tab of the OS Detail page includes the following sections:

<table>
<thead>
<tr>
<th>Network Details</th>
<th>Packet Collisions</th>
<th>Packet Discards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Errors</td>
<td>Transmission Rate</td>
<td>Transmission Rate (Bytes)</td>
</tr>
<tr>
<td>Transmission Queue Length</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of operating system.

Network Details
The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Interface</td>
<td>The name of network interface.</td>
</tr>
<tr>
<td>INET Address/Address</td>
<td>The IP address assigned to the network interface.</td>
</tr>
<tr>
<td>Pkts Sent/Sec</td>
<td>The number of packets sent per second.</td>
</tr>
<tr>
<td>Pkts Received/Sec</td>
<td>The number of packets received per second.</td>
</tr>
<tr>
<td>Sent (KB/Sec)</td>
<td>The number of bytes sent per second.</td>
</tr>
<tr>
<td>Received (KB/Sec)</td>
<td>The number of bytes received per second.</td>
</tr>
<tr>
<td>Out Pkts Discarded</td>
<td>The number of outbound packets discarded.</td>
</tr>
<tr>
<td>In Pkts Discarded</td>
<td>The number of inbound packets discarded.</td>
</tr>
<tr>
<td>Out Pkt Errors</td>
<td>The number of outbound packet errors.</td>
</tr>
<tr>
<td>In Pkt Errors</td>
<td>The number of inbound packet errors.</td>
</tr>
</tbody>
</table>
Transmission Rate
The Transmission Rate section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

- **Packets Received/Sec**
- **Packets Sent/Sec**

Packets Sent/Sec
The Packets Sent/Sec statistic is the number of packets sent over each network adapter per second.

Packets Received/Sec
The Packets Received/Sec statistic is the number of packets received over each network adapter per second.

Transmission Rate (Bytes)
The Transmission Rate (Bytes) section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

- **Received (KB/Sec)**
- **Sent (KB/Sec)**

Sent (KB/Sec)
The Sent (KB/Sec) statistic is the rate at which bytes are sent over each network adapter including framing characters.

Received (KB/Sec)
The Received (KB/Sec) statistic is the rate at which bytes are received over each network adapter, including framing characters.

Transmission Queue Length
The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue

Packet Collisions
A collision occurs when two devices on the network sense that the network is idle and end up trying to send data at the same time. Some collisions are normal on a repeated network, but excess collisions can cause serious performance problems.
**Troubleshooting**
Contact your network administrator if there is high rate of persistent Packet Collisions.

**Packet Discards**
The Packet Discards section of the Network tab succinctly communicates the packet discard rate. The following statistics are available in this section:

| Inbound Packets Discarded | Outbound Packets Discarded | Packet Discards |

**Outbound Packets Discarded**
The Outbound Packets Discarded statistic is the number of outbound packets that were discarded even though no errors had been detected to prevent from being transmitted.

**Troubleshooting**
A possible reason for discarding such a packet could be to free up buffer space.

**Inbound Packets Discarded**
Received Packets Discarded is the number of received packets that were discarded even though no errors had been detected to prevent from being transmitted.

**Troubleshooting**
A possible reason for discarding such a packet could be to free up buffer space.

**Packet Discards**
The Packet Discards statistic represents the number of network packets discarded.

**Troubleshooting**
Contact your network administrator if there is high rate of persistent packet discards.

**Packet Errors**
The Packet Errors section of the Network tab succinctly communicates the packet error rate. The following statistics are available in this section:

- Inbound Packet Errors
- Outbound Packet Errors

**Outbound Packet Errors**
The Outbound Packet Errors statistic is the outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
Inbound Packet Errors
The Packets Received Errors statistic is the received packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Top SQL Statistics

- **Metrics**

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements or stored procedures can drag an otherwise well-configured database down in terms of user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The Top SQL results grid displays the top SQL statements across various performance categories.

The SQL Details tab of the Top SQL view presents information as follows:

- **Active SQL Results**
- **Selected SQL Text**

**Metrics**
When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer:

- What has been the worst SQL that’s historically been run in my database?
- What is the worst SQL that’s running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

**Active SQL Results**
The Active SQL results section displays the following information in grid format:

- **SQL Text**: A snippet of the full SQL statement. Clicking on the statement will cause the full statement to be presented in the Selected SQL Text grid.
- **Database**: The name of the database.
- **SPID**: The process ID.
- **CpuTime**: The accumulated CPU time for the statement.
- **Wait Time**: The time the process has been waiting to be serviced, in milliseconds. A value of zero indicates there was no waiting.
- **Memory Usage (KB)**: The number of pages in the procedure cache that is currently allocated to the process. A negative number indicates that pages are being released (freed) from the process.
- **Physical Reads**: The current cumulative number of physical disk reads issued by the process.
- **Logical Reads**: The current cumulative number logical reads issued by the process.
Pages Modified: The number of pages that were modified by the process.

Packets Received

Packets Sent

Start Time: The date/time when the statement was first executed.

Batch ID

**Selected SQL Text**

In this section of SQL details, you see the fully expanded SQL statement that you highlighted in the Active SQL Results grid. By clicking Explain SQL you open the Explain Plan with its ISQL window.
Performance Analyst for IBM DB2 for Linux, Unix and Windows Tutorial

Embarcadero Performance Analyst provides key insight into the complex world of database performance. Running inside the Embarcadero DBArtisan console, or standalone in a Web browser, Performance Analyst helps you pinpoint performance problems that threaten a database’s availability or response times. The powerful analysis facilities will satisfy advanced experts, yet novices will find Performance Analyst’s interface easy to use and the expert help a welcome aid to troubleshooting difficult performance issues.

About This Performance Analyst for IBM DB2 Tutorial

This tutorial is intended to help you get started with Embarcadero Performance Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Performance Analyst. You will have learned how to accurately monitor your database using the most popular performance methodologies in use today. You will know how to drill down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. Finally, you’ll understand how to fix performance problems using the integration that exists between Performance Analyst and Embarcadero DBArtisan.

This tutorial is divided into 8 sections.

Session 1: Getting Started with Performance Analyst
Session 2: Using Ratio-based Performance Analysis
Session 3: Using Bottleneck/Wait-based Performance Analysis
Session 4: Using Workload Performance Analysis
Session 5: Using SQL Performance Analysis
Session 6: Using Operating System Performance Analysis
Session 7: Identifying and Getting Help with Performance Problems
Session 8: Fixing Identified Problems

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see Additional Evaluation Resources.

Session 1: Getting Started with Performance Analyst for DB2

Download and Install

You can obtain the latest version of Performance Analyst by downloading the DBArtisan software from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Performance Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Performance Analyst (although Performance Analyst can also run in standalone fashion within a Microsoft Internet Explorer Web browser).

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.
Overview for Performance Analyst for DB2
When DBArtisan is connected to a DB2 database, you can open Performance Analyst two ways:

1. On the toolbar, click the Performance Analyst icon.
2. On Utilities menu, select Performance Analyst.

DBArtisan opens Performance Analyst.

DB2 requires that certain monitor switches be turned on for collecting various performance metrics. When Performance Analyst is first invoked, you will be prompted to turn on the monitor switches. Also, you can increase or decrease the size of the monitor heap depending on the activity of your database.

If you do not want to see this dialog each time you start Performance Analyst, you can check the "Save these settings as default" check box.

Product Navigation
Navigating through Performance Analyst is easily accomplished through two main mechanisms:

1. The main Performance Analyst toolbar contains buttons and drop-down menus that invoke the performance analysis main views and drill-downs within the product as well as frequently used actions like Refresh and Performance Threshold Customizations.
2. Positioning the mouse pointer on any area within Performance Analyst and right-clicking brings up a navigation menu that allows you to easily move between any of the main Performance Analyst home pages and drill-down views. The right mouse menu also contains context-sensitive options like drilling down further into a particular statistic, copying the display to a text file, and more.

Session 2: Using Ratio-Based Performance
Ratio-based analysis has been around for many years, and used to be the only technique database administrators used when they were called on to diagnose the cause of a database slowdown. Performance ratios are very good roll-up mechanisms for busy DBAs, making possible the analysis-at-a-glance approach, however ratios must be computed correctly for them to be meaningful. Key ratios in Performance Analyst are calculated using delta statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. Performance Analyst makes it easy to see the ratios that are computed with delta statistics vs. cumulative. A triangle symbol denotes a delta-based statistic, while square symbols represent a cumulative-based measure.

Ratio-based analysis is available for all performance categories (memory, etc.) where ratios can assist in determining the performance level of a database. For example, if you would like to see memory-specific ratios, on the Performance Analyst toolbar click Memory. Performance Analyst displays the Memory home page that showcases all important memory-related ratios.

Obtaining Detail Information for a Ratio Using Performance Analyst for DB2
A ratio gives you a global picture for a particular performance measure. However, you often need more information regarding a statistical measure so you can make an informed decision. Performance Analyst has easy-to-use drill down features that make obtaining detailed information for various ratios a breeze.
For example, to view more information for the Buffer Cache Hit Ratio statistic, do the following:

1. On either the main Home page or the I/O home page, locate the Buffer Pool Hit Ratio statistic.
2. Position the pointer over the statistic. The pointer changes to a hand symbol.
3. Double-click the left mouse button. Performance Analyst opens the Buffer Cache Tab of the memory drill down view.
4. Or right-click the statistic, select Details or Navigate to... > I/O > Buffer Pool.

Once either of these actions has been taken, you will be taken to a drill-down view that provides more information on the DB2 buffer pools.

Session 3: Using Bottleneck/Wait-Based Performance Analysis in Performance Analyst for DB2

When a DB2 database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things, or it can be an indicator that a database bottleneck exists. Bottleneck analysis is a valid method of measuring performance because it helps you track where a database and user sessions have been spending their time. If lock contentions or heavy table-scan activity has been dragging a database's performance down, you can use bottleneck analysis to confirm the actual root cause.

Performance Analyst uses several key areas of DB2 metrics to form its bottleneck analysis. These areas are usually the main sources of performance drag on the databases.

Performance Analyst prominently displays bottleneck analysis on every performance home page. On the main home page, you will see a summary of all key bottlenecks threatening the database. You can also see specific bottlenecks for memory, I/O, space, objects, and users by navigating to each home page and viewing the Bottleneck Analysis section for that home page.

For example, to see bottlenecks that specifically relate to user applications, follow these steps:

1. Click the Users Performance Analyst toolbar button
2. Locate the Bottleneck Analysis section at the lower-left portion of the screen

The Users Bottleneck Analysis section shows the top bottlenecked applications, concurrency issues, and bottleneck activity count.

Obtaining Detail on System Bottlenecks

You can easily obtain more information regarding global bottlenecks that are plaguing a database by drilling down into a particular bottleneck metric. For example, if you would like to see more data regarding the current top concurrency bottlenecks, do the following:

1. Open the Performance Analyst main Home page.
2. Double-click the Concurrency area in the Bottlenecks section to be taken to the Lock Waits drill-down section. Performance Analyst opens the System Waits view.
3. Or right-click anywhere in Performance Analyst, point to Users, and then click Lock Waits. Performance Analyst opens the System Waits view.
The Lock Waits view displays critical details regarding all the lock wait events that DB2 has recorded. You can easily see what applications are waiting on locks and the average lock wait times for all applications.

**NOTE:** If you would like to see an expanded view of the Lock Waits grid shown on the top of the Lock Waits view, click the Expand button located at the top right corner of the section. All section views in Performance Analyst can be maximized/minimized using this control.

### Viewing Detail for Application Bottlenecks

Not only can you obtain detail information for system bottlenecks, but you can also use Performance Analyst to get drill-down information regarding the locks held by and waited on by applications.

To see more information regarding application locks, do the following:

1. Open the Performance Analyst main Home page.
2. Double-click the Concurrency Bottlenecks area to be taken to the Lock Waits drill-down view.
   Performance Analyst opens the Lock Waits tab.
3. Or right-click anywhere in Performance Analyst, point to Users, and then click Locks.

Once on the Locks drill-down view, you can view the details for locks held and waited on by an application details by following these steps:

1. Examine the Application Details grid
2. Select an application that appears to be experiencing lock waits problem
   You will see details on the locks held by, and the locks waited on, by the selected application. You can get further details on the application that is holding the locks by navigating to the Attributes drill-down and selecting the appropriate Application ID.

### Session 4: Using Workload Performance Analysis

Key ratios give you a global perspective on database activity. Bottleneck analysis gives you insight into things that are holding up user activity and throughput. But another technique is necessary if a database professional is to really get a handle on what's occurring inside a badly performing database.

Workload analysis involves the investigation of two critical areas of a database's performance:

- Application resource consumption and activity
- SQL execution

Without looking at these two key performance categories, you can miss a lot of what could be responsible for perceived performance problems. When performance on a database takes a sudden nosedive, it is not uncommon to find one or two sessions that are causing the bulk of the workload. Finding these processes can be easily accomplished by viewing session metadata coupled with resource consumption and statistical execution statistics.

### Pinpointing Top Resource Consumers

Performance Analyst identifies top resource consumers in a couple of different ways. On the main home page, Performance Analyst highlights the top resource consumers across lock usage, CPU usage, and I/O usage. The percentage used across all statistical categories are displayed so a DBA can immediately pinpoint a session that is using all or most of a particular resource. The top resource sessions are also displayed on the memory, I/O, and other home pages as well with a listing for the top sessions for that category (for example, the top IO consumers appear on the I/O home page, etc.).
Getting Workload Details Using Performance Analyst for DB2

Performance Analyst offers several details on top resource applications in addition to the summaries provided on each home page. A detailed and flexible top applications view lets you view a summary of all the major resource applications on the database as well as every critical detail regarding each application connected to the database.

Further drill-downs are available that highlights specific details regarding each application’s activities and submitted SQL. To see detailed information regarding an application identified in one of Performance Analyst’s Workload Analysis sections, do the following:

1. In a Workload Analysis display, position the pointer over one of the processes.
2. Double-click the process or right-click, and then select Details.

If you want like to see all key workload analysis metadata for sessions, do the following:

1. On the toolbar, click the Top Sessions button.
2. Or right-click anywhere in Performance Analyst, point to Users, and then click Top Sessions.

Session 5: Using SQL Performance Analysis

SQL-based performance analysis is really a subset of Workload Analysis. Understanding current and historical SQL execution patterns lets you to have the second set of data points necessary to properly perform workload analysis. You may find that optimizing SQL code produces some of the best performance-enhancing boosts available for a database.

Understanding SQL Activity

Performance Analyst offers plenty of insight into SQL performance and execution patterns. Each one of the home pages depicts the current levels of SQL activity as determined by a variety of different metrics for that performance category. For example, Performance Analyst might show that rows reads account for 75% of all SQL statement I/O on the database.

Getting Details on Top SQL

Getting details on identified SQL statements is easy in Performance Analyst. For example, if you would like to see further information on a SQL statement identified on the Performance Analyst Home page, do the following:

1. In the SQL Analysis section, position the pointer over Hash Values.
2. Double-click or right-click, and then select Details.

   Performance Analyst opens the Top SQL summary page that shows a breakdown of all top SQL statements along with their execution statistics.

If you would like to see the full SQL text of a SQL statement, do the following

1. In the Top SQL Details section, double-click a statement.

   Performance Analyst opens the SQL Details Tab of the Top SQL view to display the entire SQL statement and the execution metrics. While on this pane, you can also perform an EXPLAIN of the SQL statement and examine its access path information.

   **NOTE:** To automatically tune a problem SQL statement if you have Embarcadero SQL Tuner installed on your workstation, do the following:
2. On the **SQL Details Tab**, select a problem statement, and then click the **Tune SQL** button.

   SQL Tuner opens and you can begin a tuning session.

---

### Session 6: Using Operating System Performance Analysis

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O or Memory), a slow CPU, insufficient I/O devices, and insufficient memory. Performance Analyst allows you to examine the relevant operating system metrics for DB2.

To view the operating system metrics of the machine hosting the DB2 server you are monitoring, do the following:

1. Click the **OS** Performance Analyst toolbar button.

   OR

2. Navigate to the OS Home Page by going to **OS**, and then **Home**.

   **NOTE:** The first time you invoke the OS feature you will be prompted for an operating system login. Please specify the correct operating system, host name, username and password, and protocol.

The Operating System home page has been designed in a similar fashion as the datasource home page. On this page you can see the pertinent operating system metrics displayed in five quadrants broken down into the following categories:

- Resource Usage
- Memory Analysis
- Disk Analysis
- Workload Analysis
- Bottleneck Analysis

### Obtaining Details on the Operating System

You can easily obtain more information regarding operating system problems that are degrading database performance by drilling down into a particular analysis quadrant. For example if you’d like to see more information regarding the processes running on the host machine, do the following:

1. Navigate to the Performance Analyst **Operating System** home page.

2. Double-click the **Top Memory Process** in the **Workload Analysis** quadrant.

   Performance Analyst opens the Processes detail view.

   OR

3. Right-click anywhere in Performance Analyst and select the **Navigate to**, **OS**, and then **Processes**.

The Processes detail tab shows all the processes on the host machine. From this view you can easily determine which processes are showing high marks in I/O or CPU, for example, by simply double clicking on the column headers and sorting the processes in ascending or descending order.
Session 7: Identifying and Getting Help with Performance Problems

You do not need to be an advanced expert in performance tuning to spot trouble in a database if you use Performance Analyst. There are a number of mechanisms in the tool that help you quickly locate areas of concern within a database.

Using the Alarm Log

Performance Analyst has a central and flexible alarm log that lists every performance violation found in all monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst lists all alarms for each database you are monitoring in one central log so you can see every problem in one place. To view the Performance Analyst Alarm log, do the following:

1. On the toolbar, click the Alarms button.

Once in the Alarm log, you can perform a number of actions including:

- Selecting an alarm and viewing the application page that shows details regarding the metric in violation.
- Filtering the types of alarms shown through the Options dialog.
- Clearing all current alarms.
- Copying or saving the Alarm log into a different format.

To select any of these options, do the following:

1. Position the pointer over any area of the Alarm Log.
2. In the shortcut menu, select an option.

Recognizing Performance Violations

Performance Analyst makes it easy to visually pinpoint performance violations in a database. Any statistic that does not conform to your predefined performance standards is colorized in any view that it appears so you can easily see potential trouble spots.

The colors used to highlight a particular performance violation depend on the severity of the problem. Performance Analyst uses several layers of performance standards/threshold violations with each layer being represented by a different color. You can control everything regarding threshold violations, including how thresholds are determined, their color representation in Performance Analyst’s display, and more by using the Threshold Template feature of Performance Analyst.

How can Performance Analyst inform you of performance violations in your database if you minimize the product to your desktop and are working on other things? Even if Performance Analyst isn’t right in front of you, it can still let you know that a problem has been found in your database through the systray alarm icon. When Performance Analyst detects a problem in one of your databases, one of the things the product does is enable an alarm icon in your workstation’s systray area that communicates the presence of alarms in your database. So even if the Performance Analyst interface isn’t in sight, the product can let you know that something in your database requires attention.
Creating Threshold Templates

How does Performance Analyst know what to signal as a problem in a database? It uses predefined performance standards that you define and enforce. All of this is accomplished by using Performance Analyst’s Threshold templates. A template is a set of performance metrics and their thresholds that can be reused to enforce performance standards across any number of databases. Embarcadero provides a default performance template that contains nearly all of the most-watched performance statistics so you can begin monitoring with confidence right out of the box. However, you can create your own Threshold templates so you only are alerted to performance violations that are important to you and your particular environment.

To view or create Threshold templates, do the following:

1. On the toolbar, click the Thresholds button.
   Performance Analyst opens the Threshold maintenance dialog box.

   The Threshold Template Properties dialog box lets you select the options to add or edit templates, as well as other maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your own needs. You can control what statistics are sampled for performance violations as well as the levels of severity you would like reported. Once you are satisfied with your template, you can save it and then apply it to one or more datasources in one action to immediately begin enforcing your performance standards.

Getting Help with Performance Problems

As databases become more complex, so do the problems that plague them. Knowing what to do for a particular performance problem can be hard in some cases, but Performance Analyst helps by providing a complete set of expert help that you can use as a guide when performance problems occur. To obtain expert help for any statistic shown in Performance Analyst, do the following:

1. Position the pointer over a statistic that you need help with.

   Right-click and select Help.

   The Expert Help gives you a general description of the statistic, the range of values that are considered optimal, and some suggestions on how to fix a problem situation. Expert help can also be invoked for any of the Ratio, Bottleneck, SQL, or Workload Analysis sections shown on every home page by clicking on the help icon for that section, which is located at the top right corner of the section.

Session 8: Fixing Performance Problems Using Performance Analyst for DB2

Finding performance problems is one thing, but fixing them is another. Many performance monitors point out a particular problem in a database, but then offer no way to actually correct the situation. Such is not the case with Performance Analyst. Because of its tight integration with DBArtisan, certain problems found in Performance Analyst can be visually corrected by DBArtisan.

If a DB2 tablespace is about to run out of free space, Performance Analyst can issue an alarm that a space shortage exists. Once the space problem has been identified, do the following:

1. On the Space home page toolbar, click the Space button.

2. In the Space Analyst grid, right-click a tablespace, and then select Manage with DBArtisan.

   You will then be taken to DBArtisan’s Tablespace Editor where you can perform corrective actions, such as adding new containers to the tablespace, or resizing the tablespace’s existing container(s)
Performance Analyst for IBM DB2 for Linux, Unix, and Windows Expert Guide

This section includes expert help for all DB2 categories and statistics in the Embarcadero Performance Analyst views and pages. This help is divided into the following sections:

<table>
<thead>
<tr>
<th>Home View</th>
<th>Memory Statistics</th>
<th>I/O Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Statistics</td>
<td>Database Configuration</td>
<td>Instance Statistics</td>
</tr>
<tr>
<td>Users Statistics</td>
<td>OS Detail</td>
<td>Top SQL</td>
</tr>
</tbody>
</table>

Home Page Statistics - DB2

The Embarcadero Performance Analyst Home view lets you review availability and overall performance of all monitored databases from a single window. Statistics on the Home view are organized into the following categories:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Key Ratio Analysis</th>
<th>SQL Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Analysis</td>
<td>Workload Analysis</td>
<td></td>
</tr>
</tbody>
</table>

Key Ratio Analysis - Home

Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the DB2 Snapshot Monitor. The snapshot monitor returns metrics for the Instance and particular databases since the start of monitoring. Some metrics are cumulative (e.g., counters) and others are instantaneous (e.g., gauges).

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Home page to succinctly communicate the general overall performance levels of the monitored database:
Counters/Efficiency

<table>
<thead>
<tr>
<th>Buffer Pool Hit Ratio</th>
<th>Catalog Cache Hit Ratio</th>
<th>Package Cache Hit Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Workspace Hit Ratio</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Resource Utilization

<table>
<thead>
<tr>
<th>Database Heap Utilization</th>
<th>Log Space Utilization</th>
<th>Lock List Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Sort Memory Utilization</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Buffer Pool Hit Ratio

- **Metrics**
- **Troubleshooting**

The DB2 database server reads and updates all data from a bufferpool because memory access is much faster than disk access. Data is copied from disk to a bufferpool as needed by the applications using the database. When the server needs to read/write data and the data is already in the bufferpool, no disk access is required. However, if the data is not in the bufferpool, it needs to be read from the disk, which is significantly slower process.

The buffer pool hit ratio indicates the percentage of time that the database server did not need to load a page to service the request for a page. The calculation takes into account all the index and data pages that were requested.

**NOTE:** This statistic is also available on the *Memory* home page.

**Metrics**

Avoiding disk I/O is the main issue when you try to improve the performance tuning. A high buffer pool hit ratio is desirable because it indicates a lower frequency of synchronous disk I/O. A database where data is accessed uniformly from very large tables will have a poor hit ratio. There is little you can do to improve the performance in such cases.

**Troubleshooting**

The buffer pool hit ratio on the database Home page is the overall hit ratio of the database. First, drill down to the bufferpool level and check the individual buffer pool hit ratios to identify the elements that may require tuning.

Increasing the bufferpool size generally improves the buffer pool hit ratio. You can use one of the following methods to increase the bufferpool size depending on the layout of the data:

1. If the tablespaces using the bufferpool have tables and indexes on them, increase the bufferpool size incrementally until the index page hit ratio stops increasing for that bufferpool. You will not usually get any benefit increasing the bufferpool size after you reach that point.

2. If the tablespaces associated with the bufferpool have only indexes on them, increase the bufferpool size until the index page hit ratio stops increasing.

3. If the tablespaces associated with the bufferpool have only data on them, increase the bufferpool size until the data page hit ratio stops increasing.

One general design guideline is to try and use different tablespaces for indexes and tables and associate different bufferpools with them. This generally improves performance and makes bufferpool tuning much easier.

Catalog Cache Hit Ratio

- **Metrics**
Catalog cache is used to cache the following types of catalog information:

- Metadata for tables, views, and aliases.
- Database authorization information that is used to check authorization when performing operations like CONNECT, LOAD, CREATE, BIND, and so on.
- Execute privileges for user-defined functions and stored procedures.

When a database operation accesses catalog information, it inserts this information into the catalog cache so operations accessing the same information can read it from the cache and avoid disk reads.

The catalog cache hit ratio indicates how well the catalog cache avoids accesses to the catalog.

**NOTE:** The catalog cache hit ratio is also available on the Memory home page.

**Metrics**

A high catalog cache hit ratio (> 80%) is desirable and it indicates that the catalog cache is working well. A smaller ratio can indicate that this parameter needs tuning. You may see a smaller ratio immediately following the first connection to the database and execution of DDL/DCL statements since these require heavy catalog accesses.

**Troubleshooting**

If the catalog cache hit ratio is consistently small, the database configuration parameter `catalogcache_sz` should be increased. When you increase the value of this parameter, pause to consider whether it would be more effective to allocate the extra memory you are reserving to another purpose such as package cache or bufferpools.

In a partitioned database environment, make the `catalogcache_sz` larger on the catalog partition than on other partitions because the catalog information required for all partitions will go through this partition.

When tuning this parameter, it is advisable to monitor changes to the database catalog via DDL statements. During the execution of DDL statements, there may be a large drop in the catalog cache hit ratio due to invalidation of the cached data.

**Package Cache Hit Ratio**

**Metrics**

Embarcadero DBArtisan’s Database Manager caches the sections for frequently used dynamic and static SQL statements in the package cache.

The package cache hit ratio indicates how well the package cache is avoiding catalog accesses to packages and recompileations.

**NOTE:** The package cache hit ratio is also available on the Memory page.

**Metrics**

A high package cache hit ratio (> 80%) is a good thing. It indicates that the package cache is working well. In the case of static SQL statements, package caching allows the Database Manager to reduce the internal overhead by eliminating the need to access system catalogs when reloading a package. For dynamic SQL, the benefit of package caching is even greater since it a cache hit eliminates the need for recompilation.

The package cache hit ratio metric is particularly important for transaction-processing applications since a typical workload involves repeatedly executing the same SQL statements.
Troubleshooting
Executing DDL statements can invalidate sections of SQL statements in the cache, causing the hit ratio to decrease dramatically. Before attempting to tune this parameter, you should check the DDL activity to see if that is causing a skew. No amount of tuning will improve the package cache performance if the DDL activity is causing package invalidations in the cache.

If the DDL activity is minimal and package cache hit ratio is consistently small, consider increasing the package cache size (pckcachesz) in the database configuration. When increasing the value of this parameter, consider whether it might be more effective to allocate the extra memory being reserved to another purpose such as catalog cache or bufferpools.

Shared Workspace Hit Ratio
- Metrics
- Troubleshooting

When sections are required by an application for executing dynamic or static SQL statements, they are placed in the shared workspace. The shared workspace exists at the application level and is shared among applications using the database.

The hit ratio is a percentage indicating how well the shared SQL workspace is helping to avoid initialization of sections for SQL statements that are about to be executed. A high ratio indicates the shared workspace is successful in avoiding this action.

**NOTE:** The shared workspace hit ratio is also available on the Memory page.

Metrics
A shared workspace is shared by many applications. If applications have similar database access patterns, they can benefit greatly if they find required sections in the shared workspace. If an application finds a section in the shared workspace (e.g., a hit), that application avoids the setup and initialization cost for that section. A high hit ratio is desirable for this metric. Typically, you should expect to see a high ratio (>80%) in transaction processing environments.

Troubleshooting
Shared workspace memory is allocated from the application control heap (app_ctl_heap_sz database configuration parameter) and increasing this may improve the hit ratio.

Database Heap Utilization
- Metrics
- Troubleshooting

There is one database heap per database, and the database manager uses it on behalf of all applications connected to the database. The database heap utilization is the percentage of database heap that is currently being used.

Metrics
The database heap contains control block information for tables, indexes, table spaces, and bufferpools. It also contains space for event monitor buffers, the log buffer, and temporary memory used by utilities. Thus, the heap utilization can increase or decrease whenever any of the constituent elements change. If the utilization goes above 85% several times or stays above 85% for a sustained period, it may mean that you need to increase the maximum database heap size.
Troubleshooting
The dbheap database configuration parameter determines the maximum amount of heap memory that can be allocated. Examine the individual memory pools and how they are using the memory before increasing this parameter. An excessively large use of one of the memory pools may be the cause of this problem.

NOTE: On the main toolbar, click Memory and then Memory Pools to see the Memory Pools Details.

Log Space Utilization

Log space utilization is the percentage of total available log space on the database that is currently being used.

Metrics
Very high log space utilization (>75%) can indicate that the database might run out of log space soon. Usually, consistently high log space utilization has one of the following causes:

- Large transaction size. Applications are performing large numbers of updates on the database but are not committing the transactions frequently.
- The log space configured for the database is insufficient for the level of database activity

Very low log space utilization (< 5) % may indicate an excessively large log file and that you may be able to free some of the space occupied by log files on disk.

Troubleshooting
If the log utilization is mostly low but increases to near 100% for short periods, then you can increase the number of secondary log files (logsecond database configuration parameter). This allocates and deallocates additional log files as needed. If however, the log size is consistently high and changes frequently, you can increase the primary log size and/or the log file size (logfilsiz and logprimary database configuration parameters).

Also, check the SQL activity on the SQL Activity tab of the Users Detail section to see if any applications are performing infrequent commits. If this is the case, evaluate whether such applications can be changed to have shorter transactions and hence less need for log space.

Lock List Utilization

Lock list utilization is the percentage of total database memory allocated for locks that is currently being used.

Metrics
There is only one lock list for each database and it contains the locks held by all applications connected to the database. Once the lock list is full, the database manager starts escalating row locks to table locks to free up space. This escalation may result in serious performance degradation because of reduced concurrency. Additionally, the number of deadlocks and transaction rollbacks may go up.

If this metric reaches the 75% mark, you should consider bringing this percentage down with tuning.

Troubleshooting
Depending on the database’s activity level, you may be able to reduce the lock utilization by following these recommendations:
Increase size of lock list: If there is not enough lock list space available, lock escalations will occur, thereby increasing contention and reducing concurrency. Update the `locklist` database configuration parameter to increase this value.

Tune applications to reduce locking: On the Locks tab of the Users detail section, identify the applications that are holding many locks and then consider the following steps for controlling the size of the lock list:

- Make the transactions shorter by increasing the COMMIT frequency. This ensures that locks are released frequently, thus freeing up lock space.
- Before you update many rows in a single table, lock the entire table (using the SQL LOCK TABLE statement). This prevents many row-level locks from being obtained (although this decreases concurrency)
- To control how locking is done for a specific table, use the LOCKSIZE parameter of the ALTER TABLE.
- To decrease the number of share locks held, use the Cursor Stability isolation level when possible. If the applications’ integrity requirements are not compromised, use Uncommitted Read instead of Cursor Stability to further decrease the amount of locking.

Decrease percentage of lock list: If a small number of applications are consuming most of the lock space, decrease the percentage of lock list for each application. You can throttle back those applications by decreasing the `maxlocks` database configuration parameter. This reduces the amount of lock list memory available to each application thereby allowing for better distribution of lock list memory.

**NOTE:** Decreasing the percentage of lock list should be the last resort, and used only if you cannot decrease utilization with the other recommendations. It can cause a large number of lock escalations.

Shared Sort Memory Utilization

- **Metrics**
- **Troubleshooting**

The amount of memory allocated for each sort may be some or all of the available sort heap size. Sort heap size is the amount of memory available for each sort as defined in the database configuration parameter `sortheap`. Shared sort memory utilization gives the percentage of the sort heap being used.

It is possible for a single application to have concurrent sorts active. For example, in some cases a SELECT statement with a subquery can cause concurrent sorts.

**Metrics**

Memory estimates do not usually include sort heap space. If excessive sorting is occurring, the extra memory used for the sort heap should be added to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Typically the shared sort utilization should be less than or equal to 70%. You should consider tuning the database if you see a utilization value greater than this.

**Troubleshooting**

To bring the sort heap utilization to an acceptable level, use the following guidelines:

- Examine the queries you are running on the database to see if you can add indexes to columns frequently accessed in the WHERE clause. This minimizes the use of the sort heap.
- If you are using dynamic bitmaps or hash join buffers in your queries, or your queries frequently require large sorts, increase the `sortheap` size.
- If you adjust the `sortheap` size, also look at the `sheapthres` database manager configuration parameter to see if it too needs to be adjusted.
If you are adjusting the sorheap size, you may also benefit from rebinding your packages since the optimizer takes this parameter into account when determining the access paths.

Bottleneck Analysis - Home
When a DB2 database is active, the applications accessing the database can either successfully access the database resources, or they can wait for resources to become available. An application that is waiting can indicate normal database activity, or it can be an indicator that a database bottleneck exists. This is where wait-based or bottleneck analysis comes into play. A database administrator can use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Performance Analyst Bottleneck analysis section helps a DBA isolate potential problem areas where there are waits, resource overloads or potential failures. If concurrency issues or heavy table scan activity has been dragging a database's performance down, a DBA can use bottleneck analysis to isolate the root cause. Once one or more bottlenecks have been isolated as possible areas of interest, the DBA can drill down and examine details on why the applications or resources are causing the problem. The Performance Analyst home page identifies the top system and application bottlenecks that may be currently a cause of lessened performance.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

Concurency
- Apps Waiting on Locks
- Lock Escalations
- Avg. Lock Wait Time
- Lock Timeouts
- Deadlocks Detected
- Lock Timeouts

I/O Activity
- Log Pages I/O
- Log Pages I/O
- Synchronous I/O
- Lock Timeouts

Object Bottlenecks
- Accesses to Overflowed Records
- Page Reorganizations
- Accesses to Overflowed Records
- Page Reorganizations

Sorts and Joins
- Hash Join Overflows
- Sort Overflows
- Hash Join Overflows
- Sort Overflows

Storage Bottlenecks
- DMS Tablespaces Low on Space
- Inaccessible Containers
- Tablespace in Abnormal State
- Node with Smallest Log
- Tablespace in Abnormal State

Apps Waiting on Locks
- Metrics
- Troubleshooting

The Apps Waiting on Locks metric gives the percentage of all currently connected applications that are waiting for locks.
Metrics
If this number is high, you should investigate whether the applications are having concurrency problems.

Troubleshooting
Compare this metric with the lock escalations metric to identify if the lock list memory is too small.

Go to the Locks Waiting Details section of the Locks tab of the Users detail view and examine the lock activity at application level to identify the applications that are holding a large number of row locks and table-level locks. You may be able to tune the applications with high number of locks.

Avg. Lock Wait Time

The average lock wait time metric gives the average amount of time applications waited for a lock in the database.

Metrics
This metric gives an overview of the concurrency performance of the database. If this value is too high, it may mean that either too many applications are vying for the same resources or that applications are executing long transactions that make other applications wait longer. You should examine this metric over a period of time to determine the normal operational state of the database.

Troubleshooting
As a first step, examine the lock activity at the application level to identify if any particular application is causing excessive lock contentions.

NOTE: You can find the lock activity in the Applications Waiting on Locks section of the Lock Waits tab of the User detail page.

If you see a high value, consider tuning the application to provide better concurrency. If lock timeouts are excessive, and average lock wait times are very short, you can increase the locktimeout database configuration parameter to make the applications wait longer before timing out.

Deadlocks Detected

Deadlocks detected is the total number of deadlocks that have occurred since this instance of Performance Analyst started monitoring the database.

Metrics
If a large number of deadlocks are detected, it can indicate that applications are experiencing lock contention problems. Deadlocks are usually caused by one of the following situations:

- Lock escalations on the database
- Catalog tables locked for Repeatable Read
- Applications are using inappropriate isolation levels at bind time
- Applications are obtaining the same locks in a different order
• Applications are locking tables explicitly where row level locks are sufficient.

Troubleshooting
You may be able to modify the applications causing lock contentions for better concurrency.

To identify the applications that may be causing contentions, go to the Lock Waits tab of the Users detail page and review the Application Details section.

Lock Escalations
• Metrics
• Troubleshooting

The lock escalations metric indicates the number of times that locks have been escalated from row locks to table locks, since this instance of Performance Analyst started monitoring the database.

NOTE: This metric also appears on the Memory page.

Metrics
A lock is escalated when the total number of locks held by an application reaches the maximum amount it is allowed in the lock list memory. There are several possible causes of lock escalations:

• The database lock list size is too small for the concurrent workload
• The maximum percentage of lock list memory allowed for each application is too small
• One or more applications are obtaining an excessive number of locks

Monitor the lock escalations over a period of time to determine what levels are acceptable in your environment. If the escalations are excessive, or are accompanied by deadlocks or long lock waits, consider tuning the database.

Troubleshooting
Examine the locklist database configuration parameter together with the lock list utilization metric to see if the locklist size needs to be increased. If the lock list utilization is within acceptable range, try raising the maxlocks database configuration parameter to increase the percentage of lock list space available to each application.

You may also want to examine the detailed lock usage for each application on the Locks tab of the Users detail page to see if any application is holding an inordinately high number of locks. If this is the case, tune your application for better concurrency.

Lock Timeouts
• Metrics
• Troubleshooting

The lock timeouts metric identifies the number of times that a request to lock an object timed out without being granted.

Metrics
If the number of lock timeouts becomes excessive when compared to the acceptable range for your database, it can indicate that an application is holding locks for long durations. It can also indicate that the amount of time an application waits for a lock before timing out is too short.

If you have too few lock timeouts and the average lock wait time is too high, it can indicate that the lock timeout configuration parameter is set to an excessively high value.
Troubleshooting
First you should examine the lock activity at the application level to identify any particular application that is causing excessive lock contentions. If so, you can tune the application to provide better concurrency. If lock timeouts are excessive, and average lock wait times are very short, you can increase the `locktimeout` database configuration parameter to make the applications wait longer before timing out.

Log Pages I/O

- **Metrics**

The log pages I/O statistic is the total number of pages that are read and written to the disk by the database logger.

**Metrics**
You can use this element with an operating system monitor to get a bird's eye view of the amount of I/O on a device that is attributable to database activity.

Synchronous I/O

- **Metrics**
- **Troubleshooting**

The synchronous I/O metric presents the number of I/O operations that had to be performed synchronously by the database agents.

**Metrics**
DB2 read and write operations are always performed through bufferpools. If the page being read or written is not present in the bufferpool, it is first read into the bufferpool and then the I/O operation is performed on it. DB2 pre-fetchers fetch the data page into the bufferpool asynchronously in anticipation of a read/write operation. If the page is not available at the time of I/O, the database manager agent has to perform synchronous I/O. Synchronous I/O can decrease the responsiveness of the database, and if the database tablespaces are spread across multiple physical devices you may be able to reduce the number of synchronous operations by tuning the database configuration.

**Troubleshooting**
Generally as a rule of thumb, you can set the `num_iopensource` database configuration parameter to 1-2 more than the number of physical storage devices used by the database to ensure that parallel I/O is occurring.

To minimize synchronous I/O, you may also need to layout your tablespaces such that the containers are defined on different physical devices.

Accesses to Overflowed Records

- **Metrics**

The accesses to overflow records is the number of accesses (reads and writes) to overflowed rows in the database tables.

**Metrics**
A row overflows if it is updated and no longer fits in the data page where it was originally written. This usually happens as a result of an update of a VARCHAR or an ALTER TABLE statement.

Overflowed rows indicate that data fragmentation has occurred. If this number is high, you can try to improve performance by reorganizing the tables using the Space Analyst.
Page Reorganizations

- **Metrics**

The page reorganization statistic is the number of page reorganizations executed in the database.

**Metrics**

Too many page reorganizations can result in less than optimal insert performance. You can use the Space Analyst to reorganize tables and eliminate fragmentation. You can also use the APPEND parameter for the ALTER TABLE statement to indicate that all inserts are appended at the end of a table and so avoid page reorganizations. In situations where updates to rows cause the row length to increase, the page may have enough space to accommodate the new row, but a page reorganization can be required to defragment that space. Or if the page does not have enough space for the new larger row, an overflow record is created that causes overflow_accesses during reads. You can avoid both situations by using fixed length columns instead of varying length columns.

Hash Join Overflows

- **Metrics**
- **Troubleshooting**

Hash join is an option for the DB2 optimizer. A hash join compares the hash codes before comparing the predicates for tables involved in a join. This reduces the number of comparisons. The hash join overflows metric gives the percentage of all hash join operations that ran out of sort heap and may have required disk space for temporary storage since the current instance of Performance Analyst started monitoring the database.

**Metrics**

An overflowing hash join incurs a large overhead because of the amount of disk I/O required to complete the operation. If this value crosses the 30% mark, the DBA should take action.

**Troubleshooting**

Increase the sortheap database configuration parameter to reduce the number of overflows.

**NOTE:** This statistic is also available on the Memory page.

Sort Overflows

- **Metrics**
- **Troubleshooting**

The sort overflows statistic is the total number of sorts that ran out of sort heap and that may have required disk space for temporary storage.

**NOTE:** This metric also appears on the Memory page.

**Metrics**

When a sort overflows, additional overhead is incurred. The sort requires a merge phase and can potentially require more I/O if data needs to be written to disk.

**Troubleshooting**

Sort overflows can be reduced by increasing the sortheap database configuration parameter.
DMS Tablespaces Low on Space

**Metrics**

The DMS Tablespaces Low on Space metric gives the number of DMS tablespaces in the database where more than 90% of the total available space is in use.

**Troubleshooting**

This metric is an indicator that some of the tablespaces in your database may be running out of space. You should drill down to the Space metrics to identify the tablespaces that may be at the critical utilization level. In tablespaces where there are little on no inserts after the initial data is loaded, little or no free space may have been left on purpose. In such cases, it is normal to see a high utilization.

If this metric goes beyond the normal operational value for your database, consider creating additional space on the DMS tablespaces that are running out of space. You can do this by performing one or more one of the following tasks:

- Clean up unused space by deleting the tables/indexes that are no longer required.
- Resize or extend the existing containers in the tablespaces.
- Add new containers to the tablespaces.

Inaccessible Containers

**Metrics**

The inaccessible containers metric identifies the number of containers on all tablespaces that are currently inaccessible.

**Troubleshooting**

Inaccessible containers represent a serious problem on the database. The database is unable to access/write any data on an inaccessible container. Usually this is caused by either media errors or modifications to the container files/directories from outside DB2.

Drill down to the Space metrics to examine the inaccessible containers and the tablespaces on which they are contained. You may need to perform a redirected restore on the tablespace from a backup to correct this.

Node with Smallest Log

**Metrics**

The node with smallest log metric is only returned for global snapshots and indicates the node with the least amount of available log space.

**Troubleshooting**

Use this metric in conjunction with the log space utilization metrics to check if the log space or applications on the specified node need to be adjusted.
Troubleshooting
If the log utilization is mostly low but increases to near 100% for short periods, then you can increase the number of secondary log files (logsecond database configuration parameter). This allocates and deallocates additional log files as needed. If, however, the log size is consistently high and changes frequently, you can increase the primary log size and/or the log file size (logfilsiz and logprimary database configuration parameters).

Also, check the SQL activity, under Users Drilldown > SQL activity to see if any applications are performing infrequent commits. If possible, such applications can be changed to have shorter transactions and hence less need for log space.

Tablespaces in Abnormal State

- Metrics
- Troubleshooting

The tablespaces in abnormal state metric is a count of the number of tablespaces in your database that are not in a 'Normal State'.

Metrics
Tablespaces in not normal state are not necessarily a cause for alarm. They may have been transitioned to that state intentionally by the DBA for maintenance related tasks. If this metric shows a value that is higher than you anticipated, you may need to drilldown to the Space metrics to see which tablespaces are not normal.

A tablespace can be in one or more of the following states when it is not 'Normal':

- Quiesced: SHARE, UPDATE, EXCLUSIVE
- Pending: Load, delete, backup, roll forward, restore, disable, drop
- In progress: Roll forward, reorg, backup, TBS deletion, TBS creation
- Storage must be defined
- Restore in progress
- Offline and not accessible
- Storage may be defined
- Storage definition is in 'final' state
- Storage definition was changed prior to rollforward
- DMS rebalancer is active

Troubleshooting
If the number of 'Not Normal' tablespaces is higher than you expect, drill down to the space metrics to examine the abnormal tablespaces and take appropriate action.

SQL Analysis - Home
A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured server down in terms of user response times.

Before you can identify problem SQL in your database, you need to understand the level of SQL activity that is occurring in the database. A high level overview of the type of SQL activity helps determine not only they types of statements that are being executed but also the type of internal activity they may be causing.
The following indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

- **Static SQL**
- **Dynamic SQL**
- **Units of Work**
- **Internal Rebinds**
- **Internal Rollbacks Due to Deadlocks**

### Static SQL

- **Metrics**

  The static SQL metric is the number of static SQL statements that were executed on the database since the start of this instance of Performance Analyst.

### Dynamic SQL

- **Metrics**

  The dynamic SQL metric is the number of static SQL statements that were executed on the database since the start of this instance of Performance Analyst.

### Failed SQL

- **Metrics**

  The failed SQL metric is the number of failed SQL statements that were executed on the database since the start of this instance of Performance Analyst.

### Units of Work

- **Metrics**

  Units of work is the number of transactions executed on the database since the start of this instance of Performance Analyst.

This metric may also help you determine reasons for poor performance, since failed statements mean time wasted by the database manager. This results in lower throughput for the database.
Metrics
A low number of units of work compared to the overall SQL activity (static + dynamic SQL statements) indicate long transactions. This may in turn be an indicator of poor concurrency and heavy log usage.

Internal Rebinds

The internal rebinds metric is the number of automatic rebinds (or recompiles) that have been attempted on the database since the beginning of this instance of the Performance Analyst.

Metrics
Automatic rebinds are the internal binds the system performs when a package has been invalidated. The rebind is performed the first time the database manager needs to execute an SQL statement from the package. For example, packages are invalidated when you:

- Drop an object, such as a table, view, or index, on which the plan depends
- Add or drop a foreign key
- Revoke object privileges on which the plan depends.

You can use this element to determine the level of database activity at the application or database levels. Since internal rebinds can significantly degrade performance, they should be minimized whenever possible.

Internal Rollbacks Due to Deadlocks

The internal rollbacks due to deadlocks metric is the total number of forced rollbacks initiated by the database manager because of a deadlock for the current instance of Performance Analyst. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.

Metrics
This metric shows the number of deadlocks that have been broken and can be used as an indicator of concurrency problems. It is important because this metric lowers the throughput of the database.

Storage Analysis - Home

While DBAs focus on memory settings and tuning SQL, they frequently forget how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

- The 'hit-the-wall' variety that can bring things to a complete standstill.
- The 'performance vampire' kind that slowly drains the performance of a database over time.

Storage problems have the capability to bring the curtain down on a database very quickly, as in the case of a transaction log running out of free space. But storage problems can also silently work behind the scenes to slowly, but surely rob a database of its performance. For example, a hub table in a busy database might be accessed very quickly when an application is first given life, but over time, if it develops a heavy forwarded row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page displays the total used and free space of all DMS tablespaces in the database. It also provides the total and available log space for the database.
The following ratios succinctly communicate the general overall performance levels of the datasource:

<table>
<thead>
<tr>
<th>DMS Space</th>
<th>Total SMS Space Used</th>
<th>Log Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Backup</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DMS Space**
- **Metrics**
- **Troubleshooting**

The total used and total free metrics show the space details for DMS tablespaces. Specifically, they show the amount of used and free space on the DMS tablespaces in the database.

**Metrics**

Unlike SMS tablespaces, the total available space to the DMS tablespaces is defined at the time the database is defined. The DBA needs to explicitly increase the amount of space on DMS tables by adding/extending tablespace containers (using the `ALTER TABLESPACE` statement). If left unattended, a DMS tablespace can either remain underutilized or fill up completely. Keeping an eye on the DMS space is important because the once it fills up, the applications trying to write to the database will come to a halt suddenly.

**Troubleshooting**

Go to the Space home page or drill-down to see the usage metrics for individual tablespaces and allocate/deallocate space to containers accordingly using `ALTER TABLESPACE` command.

**Total SMS Space Used**
- **Metrics**
- **Troubleshooting**

This metric shows the total amount of SMS space used by the database.

**Metrics**

The maximum size of SMS tablespaces is not set at the time the tablespaces are created. The maximum size that can be reached is the space available on the drives/volumes that the SMS tablespace containers are defined on. You need to periodically examine the available space of the drives/volumes (using OS metrics Space page) to make sure there is enough space available for your database requirements. Since the space is determined by the space available on the drives/volumes, remember that if other applications are reading and writing to the same devices (especially logs, traces, etc), DB2 may be periodically competing for space with these applications.

**Troubleshooting**

Ideally, try and isolate the DB2 SMS drives/volumes from other applications. If the OS level metrics show that you are running out of space on a volume that is used by a tablespace's containers, you can add more containers defined on volumes with more space, to the tablespace using the `ALTER TABLESPACE` command.

**Log Space**
- **Metrics**
- **Troubleshooting**

The log space metrics show the amount of free and used log space on the database.
Metrics

Very high log space utilization (>75%) indicates that the database could run out of log space soon. Usually, consistently high log space utilization has one of the following causes:

- Large transaction size. Applications are performing a large number of updates on the database but are not committing the transactions frequently.
- The log space configured for the database is insufficient for the level of database activity

Very low log space utilization (< 5) % can indicate an excessively large log file. It's possible you can free some of the space occupied by log files on disk.

Troubleshooting

If the log utilization is mostly low but increases to near 100% for short periods, you can increase the number of secondary log files (logsecond database configuration parameter). This allocates and deallocates log files as needed. If however, the log size is consistently high, and changes frequently, you can increase the primary log size and/or the log file size (logfilsiz and logprimary database configuration parameters).

Also, check the SQL activity, under Users Drilldown>SQL activity to see if any applications are performing infrequent commits, and if possible such applications can be changed to have shorter transactions and hence less need for log space.

Last Backup

Metrics

The last backup metric indicates when the database was last backed-up.

Metrics

None.

Workload Analysis - Home

When your phone starts ringing with complaints of performance slowdowns, some of the first questions you should answer are:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Home page provides insight into the leading resource hogs of a server, as well as a count of total active and inactive processes. Drill-down's are available so you can easily get detailed information into what each leading application is currently involved with.

Workload Analysis statistics include:

- Application Activity Analysis that includes top lock escalations, top CPU processes, and top I/O processes
- Applications Currently Executing
- Idle Applications

Application Activity Analysis

Metrics
Ideally, database resource usage should be spread evenly across connected applications. In many cases, however, a couple of renegade application can hog most of the resources and starve the other applications that are competing for the same resources.

If you see a slowdown of the database, but cannot identify the root cause from ratio / bottleneck analysis, you need to examine any connected applications that are using more than their fair share of system resources (CPU, I/O, locklist, etc.)

Performance Analyst provides an easy way to identify the top users of the database resources and what percentage of total usage can be attributed to these applications.

**Metrics**

If any one application is using more than 50% of the total resource you should drill down into that application (Users drill-down) to isolate the problem.

**Applications Currently Executing**

**Metrics**

The applications currently executing statistic gives the number of applications for which the database manager is currently executing requests.

**Metrics**

You can use this number to understand how many of the database manager agent tokens are being used by applications connected to this database

**Idle Applications**

**Metrics**

The idle applications statistic the number of applications that are currently connected to the database for which the database manager is not executing any requests

**Metrics**

You can use this statistic to help you understand the level of activity within a database and the amount of system resources currently in use.

**Memory Statistics - DB2**

The Memory performance category view displays the following vital DB2 memory statistics:

<table>
<thead>
<tr>
<th>Key Ratio Analysis - Memory</th>
<th>Bottleneck Analysis - Memory</th>
<th>SQL Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Analysis</td>
<td>Workload Analysis</td>
<td></td>
</tr>
</tbody>
</table>

**Related Topic**

Memory Detail View
Key Ratio Analysis - Memory
Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards you should always adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Most of the global ratios that a DBA will examine come from the DB2 Snapshot Monitor. The Snapshot Monitor returns various metrics for the instance and database since the start of monitoring. Some of the metrics are cumulative (i.e., counters) and others are instantaneous (i.e., gauges).

In the case of cumulative metrics, the values can grow quite large over the monitoring period and the ratios calculated on these metrics can hide the skews that may occur during the monitoring. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the database activity over the sampling period can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each DB2 databases' individual personality with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following memory ratios are used on the Performance Analyst Key Ratio Analysis section to succinctly communicate the general overall memory performance levels of the monitored database:

<table>
<thead>
<tr>
<th>Buffer Pool Hit Ratio</th>
<th>Catalog Cache Hit Ratio</th>
<th>Data Page Hit Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Page Hit Ratio</td>
<td>Package Cache Hit Ratio</td>
<td>Private Workspace Hit Ratio</td>
</tr>
<tr>
<td>Shared Workspace Hit Ratio</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Buffer Pool Hit Ratio
- **Metrics**
- **Troubleshooting**

The DB2 database server reads and updates all data from a bufferpool since memory access is much faster than disk access. Data is copied from disk to a bufferpool as it is required by the applications using the database. When the server needs to read/write data and the data is already in the bufferpool, no disk access is required. However, if the data is not in the bufferpool, it needs to be read from the disk which is a significantly slower process.

The buffer pool hit ratio indicates the percentage of time that the database server did not need to load a page from in order to service the request for a page. The calculation takes into account all of the index and data pages that were requested.

**NOTE:** This statistic is also available on the [Home](#) page.

**Metrics**

Avoiding disk I/O is the main issue when trying to improve the performance tuning. A high buffer pool hit ratio is desirable because it indicates a lower frequency of synchronous disk I/O. A database where data is accessed uniformly from very large tables has a poor hit ratio. There is little you can do to improve the performance in such cases.
Troubleshooting
The buffer pool hit ratio on the database home page is the overall hit ratio of the database. First drill-down to the bufferpool level and check the hit ratios for individual bufferpools to identify the elements that require tuning. Increasing the bufferpool size generally improves the buffer pool hit ratio. You can use one of the following methods to increase the bufferpool size depending on the layout of the data:

1. If the tablespaces using the bufferpool have both tables and indexes on them, increase the bufferpool size incrementally until the index page hit ratio stops increasing for that bufferpool. Usually you will not get any benefit by increasing the bufferpool size further.

2. If the tablespaces associated with the bufferpool have only indexes on them, increase the bufferpool size until the index page hit ratio stops increasing.

3. If the tablespaces associated with the bufferpool have only data on them, increase the bufferpool size until the data page hit ratio stops increasing.

One general design guideline is to try and use different tablespaces for indexes and tables and associate different bufferpools with them. This generally improves performance and makes bufferpool tuning much easier.

Catalog Cache Hit Ratio

- Metrics
- Troubleshooting

Catalog cache is used to cache the following types of catalog information:

- Metadata for tables, views, and aliases.
- Database authorization information that is used to check authorization when performing operations like CONNECT, LOAD, CREATE, BIND, and so on.
- Execute privileges for user-defined functions and stored procedures.

When a database operation accesses catalog information, it inserts this information into the catalog cache so operations accessing the same information can read it from the cache and avoid disk reads.

The catalog cache hit ratio indicates how well the catalog cache avoids accesses to the catalog.

**NOTE:** The catalog cache hit ratio is also available on the Home page.

Metrics
A high catalog cache hit ratio (> 80%) is desirable and it indicates that the catalog cache is working well. A smaller ratio can indicate that this parameter needs tuning. You may see a smaller ratio immediately following the first connection to the database and execution of DDL/DCL statements since these require heavy catalog accesses.

Troubleshooting
If the catalog cache hit ratio is consistently small, the database configuration parameter `catalogcache_sz` should be increased. When you increase the value of this parameter, pause to consider whether it would be more effective to allocate the extra memory you are reserving to another purpose such as package cache or bufferpools.

In a partitioned database environment, make the `catalogcache_sz` larger on the catalog partition than on other partitions because the catalog information required for all partitions will go through this partition.

When tuning this parameter, it is advisable to monitor changes to the database catalog via DDL statements. During the execution of DDL statements, there may be a large drop in the catalog cache hit ratio due to invalidation of the cached data.
Data Page Hit Ratio

- **Metrics**

The data page hit ratio is the percentage of all data reads that were satisfied because the page was already available in a bufferpool.

- **Metrics**

The index page hits are used in conjunction with data page hits to calculate the overall buffer pool hit ratio. Use this element, in conjunction with the index page hit ratio when tuning the overall buffer pool hit ratio.

Private Workspace Hit Ratio

- **Metrics**

Each application executing on the database has access to the private workspace of the agents working for it.

The private workspace hit ratio is a percentage indicating how well the private SQL workspace is helping to avoid initialization of sections for SQL statements that are about to be executed. A high ratio indicates it is successful in avoiding this action.

- **Metrics**

None.

- **Troubleshooting**

Shared workspace memory is allocated from the application heap (\texttt{app\_heap\_sz} database configuration parameter) and increasing this may improve the private workspace hit ratio.

Package Cache Hit Ratio

- **Metrics**

Embarcadero DBArtisan’s Database Manager caches the sections for frequently used dynamic and static SQL statements in the package cache.

The package cache hit ratio indicates how well the package cache is avoiding catalog accesses to packages and recompilations.

- **NOTE:** The package cache hit ratio metric is also available on the [Home](#) page.

- **Metrics**

A high package cache hit ratio (\textgreater{} 80\%) is a good thing. It indicates that the package cache is working well. In the case of static SQL statements, package caching allows the Database Manager to reduce the internal overhead by eliminating the need to access system catalogs when reloading a package. For dynamic SQL, the benefit of package caching is even greater since it a cache hit eliminates the need for recompilation.

The package cache hit ratio metric is particularly important for transaction-processing applications since a typical workload involves repeatedly executing the same SQL statements.
Troubleshooting
Executing DDL statements can invalidate sections of SQL statements in the cache, causing the hit ratio to decrease dramatically. Before attempting to tune this parameter, you should check the DDL activity (Space home page - SQL analysis) to see if that is causing a skew. No amount of tuning will improve the package cache performance if the DDL activity is causing package invalidations in the cache.

If the DDL activity is minimal and package cache hit ratio is consistently small, consider increasing the package cache size (pckcachesz) in the database configuration. When increasing the value of this parameter, consider whether it might be more effective to allocate the extra memory being reserved to another purpose such as catalog cache or bufferpools.

Index Page Hit Ratio

- **Metrics**

  The index page hit ratio is the percentage of all index reads that were satisfied because the page was already available in a bufferpool.

  **Metrics**
  The index page hits are used in conjunction with data page hits to calculate the overall buffer pool hit ratio. Use this element, in conjunction with the data page hit ratio when tuning the overall buffer pool hit ratio.

Shared Workspace Hit Ratio

- **Metrics**
- **Troubleshooting**

  When sections are required by an application for executing dynamic or static SQL statements, they are placed in the shared workspace. The shared workspace exists at the application level and is shared among applications using the database.

  The hit ratio is a percentage indicating how well the shared SQL workspace is helping to avoid initialization of sections for SQL statements that are about to be executed. A high ratio indicates the shared workspace is successful in avoiding this action.

  **NOTE:** The shared workspace hit ratio is also available on the Home page.

  **Metrics**
  A shared workspace is shared by many applications. If applications have similar database access patterns, they can benefit greatly if they find required sections in the shared workspace. If an application finds a section in the shared workspace (e.g., a hit), that application avoids the setup and initialization cost for that section. A high hit ratio is desirable for this metric. Typically, you should expect to see a high ratio (>80%) in transaction processing environments.

  **Troubleshooting**
  Shared workspace memory is allocated from the application control heap (app_ctl_heap_sz database configuration parameter) and increasing this may improve the hit ratio.

Bottleneck Analysis - Memory

The Bottleneck Analysis section identifies the number of times when overflows or over-utilization may have occurred in areas of memory. An overflow occurs when the usage of a memory area goes beyond the allocated amount.
Overflows are serious bottlenecks and may cause unnecessary lock escalations, out of memory errors or performance degradations.

The following statistics are used on the Performance Analyst for DB2 Memory home page to succinctly communicate the general overall performance levels of the memory structures:

<table>
<thead>
<tr>
<th>Catalog Cache Overflows</th>
<th>Hash Join Overflows</th>
<th>Hash Join Small Overflows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock Escalations</td>
<td>Package Cache Overflows</td>
<td>Private Workspace Overflows</td>
</tr>
<tr>
<td>Shared Workspace Overflows</td>
<td>Sort Overflows</td>
<td></td>
</tr>
</tbody>
</table>

**Catalog Cache Overflows**
- **Metrics**
- **Troubleshooting**

The catalog cache overflow statistic is the number of times the catalog cache overflowed the bounds of its allocated memory.

**Metrics**
Catalog cache overflows can cause unnecessary lock escalations. This can result in loss of concurrency, or ‘out of memory’ errors from other heaps allocated to the database’s shared memory. Overflows of the catalog cache can also cause performance degradation.

DB2 reclaims the catalog cache space by evicting table descriptor information for tables, views, or aliases and/or authorization information that is not currently being used by any transaction.

**Troubleshooting**
Use this element with the catalog cache high watermark (Memory> Caches Drilldown) to determine whether the size of the catalog cache needs to be increased to avoid overflowing.

If the number of overflows is large, the catalog cache may be too small for the workload. Enlarging the catalog cache can improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures in a single unit of work, then compiling fewer SQL statements in a single transaction can improve the performance of the catalog cache. Or, if the workload includes binding packages that contain many SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures, you can try splitting packages so that they include fewer SQL statements to improve performance.

**Hash Join Overflows**
- **Metrics**
- **Troubleshooting**

The hash join overflows statistic is the number of times that hash join data exceeded the available sort heap space.

**NOTE:** This statistic is also available on the Home page.

**Metrics**
If the value of hash join small overflows is greater than 10% of this metric, then you should consider increasing the sort heap size. Values at the application level can be used to evaluate hash join performance for individual applications.
Troubleshooting
Increase the *sortheap* database configuration parameter to reduce the number of overflows.

Hash Join Small Overflows

- **Metrics**
- **Troubleshooting**

The hash join small overflows statistic is the number of times that hash join data exceeded the available sort heap space by less than 10%.

**Metrics**
If the value of hash join small overflows is greater than 10% of hash join overflows, then you should consider increasing the sort heap size. Values at the application level can be used to evaluate hash join performance for individual applications.

**Troubleshooting**
Increase the *sortheap* database configuration parameter to reduce the number of overflows.

Lock Escalations

- **Metrics**
- **Troubleshooting**

The lock escalations metric indicates the number of times that locks have been escalated from row locks to table locks, since this instance of Performance Analyst started monitoring the database.

**NOTE:** This metric also appears on the Home page.

**Metrics**
A lock is escalated when the total number of locks held by an application reaches the maximum amount it is allowed in the lock list memory. There are several possible causes of lock escalations:

- The database lock list size is too small for the concurrent workload
- The maximum percentage of lock list memory allowed for each application is too small
- One or more applications are obtaining an excessive number of locks

Monitor the lock escalations over a period of time to determine what levels are acceptable in your environment. If the escalations are excessive, or are accompanied by deadlocks or long lock waits, consider tuning the database.

**Troubleshooting**
Examine the *locklist* database configuration parameter together with the *lock list utilization* metric to see if the *locklist* size needs to be increased. If the lock list utilization is within acceptable range, try raising the *maxlocks* database configuration parameter to increase the percentage of lock list space available to each application.

You may also want to examine the detailed lock usage for each application on the Locks tab of the Users detail page to see if any application is holding an inordinately high number of locks. If this is the case, tune your application for better concurrency.
Package Cache Overflows

- **Metrics**
- **Troubleshooting**

The package cache overflows metric is the number of times that the package cache overflowed the bounds of its allocated memory.

**Metrics**

Package cache overflows can cause unnecessary lock escalations. This can result in loss of concurrency, or ‘out of memory’ errors from other heaps allocated to the database’s shared memory. Overflows of the package cache can also cause performance degradation.

**Troubleshooting**

Use this element with the package cache high watermark (Memory > Caches Drilldown) to determine whether the size of the package cache needs to be increased to avoid overflowing.

Private Workspace Overflows

- **Metrics**
- **Troubleshooting**

The private workspace overflows statistic is the number of times that private workspaces overflowed the bounds of their allocated memory.

**Metrics**

Private workspace overflows can cause performance degradation as well as ‘out of memory’ errors from other heaps allocated to the agent’s private memory.

**Troubleshooting**

Use this element with the private workspace high watermark (Memory -> Caches Drilldown) to determine whether the size of the private workspace needs to be increased to avoid overflowing.

Shared Workspace Overflows

- **Metrics**
- **Troubleshooting**

The shared workspace overflows metric is the number of times that shared workspaces overflowed the bounds of their allocated memory.

**Metrics**

Overflows of shared workspaces can cause performance degradation. Overflows can also ‘out of memory’ errors from the other heaps allocated out of application’s shared memory.

**Troubleshooting**

Use this element with the shared workspace high watermark (Memory > Caches Drilldown) to determine whether the size of the shared workspaces need to be increased to avoid overflowing.
Sort Overflows

- **Metrics**
- **Troubleshooting**

The sort overflows metric is the total number of sorts that ran out of sort heap and that may have required disk space for temporary storage.

**NOTE:** This metric is also available on the Home page.

**Metrics**

When a sort overflows, additional overhead is incurred because the sort requires a merge phase. If data needs to be written to disk, it potentially requires more I/O.

**Troubleshooting**

Sort overflows can be reduced by increasing the *sortheap* database configuration parameter.

Memory Analysis

The Memory Analysis section helps you to track the memory usage for database processes through *memory analysis heaps*.

**NOTE:** The memory metrics for DB2 are only available if you are using a DB2 version 8 server and a DB2 version 8 client.

**NOTE:** The memory management for DB2 (and hence the metrics) varies between platforms. You only see the memory metrics at the database level for UNIX system. If you are tracking the memory metrics for a Windows system, please refer to Instance -> Memory Pool Drilldown.

Memory Analysis Heaps

- **Metrics**
- **Troubleshooting**

Under normal operating conditions you can track the memory usage for the following elements:

<table>
<thead>
<tr>
<th>Application Heap</th>
<th>Database Heap</th>
<th>Application Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock Manager Heap</td>
<td>Backup/Restore/Utility Heap</td>
<td>Statistics Heap</td>
</tr>
<tr>
<td>Package Cache Heap</td>
<td>Catalog Cache Heap</td>
<td>DFM Heap</td>
</tr>
<tr>
<td>Query Heap</td>
<td>Database Monitor Heap</td>
<td>Statement Heap</td>
</tr>
<tr>
<td>FCMBP Heap</td>
<td>Import Pool</td>
<td>Other Memory</td>
</tr>
<tr>
<td>Buffer Pool Heap</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SQL Analysis
A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL
statements can drag an otherwise well-configured server down in terms of user response times.

Before you can identify problem SQL in your database, you need to understand the level of SQL activity that is
occurring in the database. A high level overview of the type of SQL activity helps determine not only they types of
statements that are being executed but also the type of internal activity they may be causing.

The following indicators are used on the Performance Analyst Memory page to succinctly communicate the general
overall performance level of the monitored database:

<table>
<thead>
<tr>
<th>Static SQL</th>
<th>Dynamic SQL</th>
<th>Units of Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Rebinds</td>
<td>Internal Rollbacks Due to Deadlocks</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** These statistics are also available on the Home page.

Workload Analysis
When your phone starts ringing with complaints of performance slowdowns, one of the first things you should get a
handle on is:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Memory page provides insight into the leading applications
that caused memory overflows to disk or lock escalations. Drill down's are available so you can easily get detailed
information into what each leading application is currently involved with.

Application Activity Analysis
- **Metrics**

Top memory issues are displayed in the following categories:

- Top Locks
- Top Lock Escalations
- Top Cache Overflows
- Top Sort Overflows
- Top Workspace Overflows

**Metrics**
The list of top memory hogs can help you quickly identify the applications that are causing memory usage problems.
You can drill down into the Memory detail views to see how these applications are using memory compared to their
peers and the overall memory being used by the database. This is a good place to start, if you think that the memory
problems you are noticing on the database may be as a result of poorly written applications rather than database
configuration or overall system memory availability.

Memory Detail
The Memory Detail includes the following tabbed pages:

<table>
<thead>
<tr>
<th>Caches Tab</th>
<th>Memory Pools Tab</th>
<th>Lock Memory Tab</th>
</tr>
</thead>
</table>

Caches Tab
The Caches tab of the Memory Detail page offers detailed views of cache usage metrics and includes the following statistics:

<table>
<thead>
<tr>
<th>Application Details</th>
<th>Catalog Cache Usage Details</th>
<th>Catalog Cache Hit Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package Cache Usage Details</td>
<td>Package Cache Hit Ratio</td>
<td>Private Workspace Usage Details</td>
</tr>
<tr>
<td>Private Workspace Hit Ratio</td>
<td>Shared Workspace Usage Details</td>
<td>Shared Workspace Hit Ratio</td>
</tr>
</tbody>
</table>

Application Details

- **Metrics**
- **Troubleshooting**

The Application Details section shows the list of applications connected to the database and the various cache usage metrics for each application. It also shows the rolled up metrics at the database level in the same list. You can select the type of cache monitoring you intend to perform for the applications from the following:

- **Catalog Cache**: Shows the details of all catalog cache activity for connected applications
- **Package Cache**: Displays the details of all package cache activity for connected applications
- **Shared Workspace**: Gives the details of all shared workspace activity for connected applications
- **Private Workspace**: This shows the details all of private workspace activity for connected applications

**Metrics**
None.

**Troubleshooting**
Depending on the category of cache details you select, you can view cache activity associated with the applications. This section is very useful in isolating the applications and the cache areas you need to tune by comparing the cache performance for all the applications and also the overall cache performance at the database level.

Catalog Cache Usage Details

- **Metrics**
- **Troubleshooting**

This section shows the detailed catalog cache usage metrics for the selected item. The metrics presented here include:

- **Catalog Cache Lookups**: The number of times that the catalog cache was referenced to obtain table descriptor information or authorization information
- **Catalog Cache Inserts**: The number of times that the system tried to insert table descriptor or authorization information into the catalog cache.
- **Catalog Cache Overflows**: The number of times that the catalog cache overflowed the bounds of its allocated memory
- **Catalog Cache High Watermark**: The largest size reached by the catalog cache

**Metrics**
These metrics are used for calculating the catalog cache hit ratio. They also give you an estimate of catalog access activity occurring in the database.

You can use the catalog cache overflow metric with the catalog cache high watermark to determine if the catalog cache size is too small. If the catalog cache overflows frequently, you consider increasing the catalog cache size.

**Troubleshooting**
If catalog cache overflows are large, the catalog cache may be too small for the workload. Enlarging the catalog cache may improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures in a single unit of work, then compiling fewer SQL statements in a single transaction may improve the performance of the catalog cache. Or if the workload includes binding packages that contain many SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures, you can try splitting packages. If they include fewer SQL statements, performance can improve.

Monitor the catalog cache metrics for your normal workload. If you see frequent overflows, you can increase the value of `catalogcache_sz` database configuration parameter to the `catalog cache high watermark/4096` (rounded up to whole integer).

**Catalog Cache Hit Ratio**

- **Metrics**
- **Troubleshooting**

Catalog cache is used to cache the following types of catalog information:

- Metadata for tables, views, and aliases.
- Database authorization information that is used to check authorization when performing operations like CONNECT, LOAD, CREATE, BIND, and so on.
- Execute privileges for user-defined functions and stored procedures.

When a database operation accesses catalog information, it inserts this information into the catalog cache so operations accessing the same information can read it from the cache and avoid disk reads.

The catalog cache hit ratio indicates how well the catalog cache avoids accesses to the catalog.

**NOTE:** The catalog cache hit ratio is also available on the Home page.

**Metrics**
A high catalog cache hit ratio (> 80%) is desirable and it indicates that the catalog cache is working well. A smaller ratio can indicate that this parameter needs tuning. You may see a smaller ratio immediately following the first connection to the database and execution of DDL/DCL statements since these require heavy catalog accesses.

**Troubleshooting**
If the catalog cache hit ratio is consistently small, the database configuration parameter `catalogcache_sz` should be increased. When you increase the value of this parameter, pause to consider whether it would be more effective to allocate the extra memory you are reserving to another purpose such as package cache or bufferpools.
In a partitioned database environment, make the `catalogcache_sz` larger on the catalog partition than on other partitions because the catalog information required for all partitions will go through this partition.

When tuning this parameter, it is advisable to monitor changes to the database catalog via DDL statements. During the execution of DDL statements, there may be a large drop in the catalog cache hit ratio due to invalidation of the cached data.

**Package Cache Usage Details**

- **Metrics**
- **Troubleshooting**

The Package Cache Usage Details section shows the detailed package cache usage metrics for the selected item. The metrics presented here include:

- **Package Cache Lookups**: The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset.

- **Package Cache Inserts**: The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system.

- **Package Cache Overflows**: The number of times that the package cache overflowed the bounds of its allocated memory.

- **Package Cache High Watermark**: The largest size reached by the package cache.

**Metrics**

These metrics are used to calculate the package cache hit ratio. They also give you an estimate of package cache access activity occurring on the database.

You can use the package cache overflow metric with the package cache high watermark to determine if the package cache size is too small. If the package cache overflows frequently, you may need to increase the package cache size.

**Troubleshooting**

Monitor the package cache metrics for your normal workload. If you see frequent overflows, you can increase the value of the `pckcachesz` database configuration parameter to the `package cache high watermark/4096` (rounded up to whole integer).

**Package Cache Hit Ratio**

- **Metrics**
- **Troubleshooting**

Embarcadero DBArtisan’s Database Manager caches the sections for frequently used dynamic and static SQL statements in the package cache.

The package cache hit ratio indicates how well the package cache is avoiding catalog accesses to packages and recompile.

**NOTE:** The package cache hit ratio metric is also available on the Home page.
Metrics

A high package cache hit ratio (> 80%) is a good thing. It indicates that the package cache is working well. In the case of static SQL statements, package caching allows the Database Manager to reduce the internal overhead by eliminating the need to access system catalogs when reloading a package. For dynamic SQL, the benefit of package caching is even greater since it a cache hit eliminates the need for recompilation.

The package cache hit ratio metric is particularly important for transaction-processing applications since a typical workload involves repeatedly executing the same SQL statements.

Troubleshooting

Executing DDL statements can invalidate sections of SQL statements in the cache, causing the hit ratio to decrease dramatically. Before attempting to tune this parameter, you should check the DDL activity (Space home page - SQL analysis) to see if that is causing a skew. No amount of tuning will improve the package cache performance if the DDL activity is causing package invalidations in the cache.

If the DDL activity is minimal and Tr is consistently small, consider increasing the package cache size (pckcachesz) in the database configuration. When increasing the value of this parameter, consider whether it might be more effective to allocate the extra memory being reserved to another purpose such as catalog cache or bufferpools.

Private Workspace Usage Details

This section shows the detailed private workspace usage metrics for the selected item. The metrics presented here include:

- **Private Workspace Lookups**: Lookups of SQL sections by an application in its agents' private workspace.
- **Private Workspace Inserts**: Inserts of SQL sections by an application into the private workspace.
- **Private Workspace Overflows**: The number of times that the private workspaces overflowed the bounds of its allocated memory.
- **Private Workspace High Watermark**: The largest size reached by the Private Workspace.

Metrics

These metrics are used to calculate the private workspace hit ratio. They also give you an estimate of private workspace access activity occurring on the database.

When the private workspace overflows, memory is temporarily borrowed from other entities in agent private memory. This can result in memory shortage errors from these entities or possibly performance degradation.

Troubleshooting

You can reduce the chance of overflow by increasing APPLHEAPSZ.

Private Workspace Hit Ratio
The private workspace hit ratio is a percentage indicating how well the private SQL workspace is helping to avoid initialization of sections for SQL statements that are about to be executed. A high ratio indicates it is successful in avoiding this action.

**Metrics**
None.

**Troubleshooting**
Shared workspace memory is allocated from the application heap (app_heap_sz database configuration parameter) and increasing this may improve the private workspace hit ratio.

**Share Workspace Usage Details**

- **Metrics**

- **Troubleshooting**

This section shows the detailed shared workspace usage metrics for the selected item. The metrics presented here include:

- **Shared Workspace Lookups**: Lookups of SQL sections by applications in shared workspaces.
- **Shared Workspace Inserts**: Number of inserts of SQL sections by applications into shared workspaces.
- **Shared Workspace Overflows**: The number of times that shared workspaces overflowed the bounds of their allocated memory.
- **Shared Workspace High Watermark**: The largest size reached by shared workspaces.

**Metrics**
These metrics are used to calculate the shared workspace hit ratio. They also give you an estimate of shared workspace access activity occurring on the database.

When the shared workspace overflows, memory is temporarily borrowed from other entities in application shared memory. This can result in memory shortage errors from these entities or possibly performance degradation.

**Troubleshooting**
You can reduce the chance of overflow by increasing APPL_CTL_HEAP_SZ.

**Shared Workspace Hit Ratio**

- **Metrics**

- **Troubleshooting**

When sections are required by an application for executing dynamic or static SQL statements, they are placed in the shared workspace. The shared workspace exists at the application level and is shared among applications using the database.

The hit ratio is a percentage indicating how well the shared SQL workspace is helping to avoid initialization of sections for SQL statements that are about to be executed. A high ratio indicates the shared workspace is successful in avoiding this action.

**NOTE:** The shared workspace hit ratio is also available on the Home page.
Metrics
A shared workspace is shared by many applications. If applications have similar database access patterns, they can benefit greatly if they find required sections in the shared workspace. If an application finds a section in the shared workspace (e.g., a hit), that application avoids the setup and initialization cost for that section. A high hit ratio is desirable for this metric. Typically, you should expect to see a high ratio (>80%) in transaction processing environments.

Troubleshooting
Shared workspace memory is allocated from the application control heap (app_ctl_heap_sz database configuration parameter) and increasing this may improve the hit ratio.

Memory Pools Tab
The Memory Pools tab offers the following detailed views:

<table>
<thead>
<tr>
<th>Application Details</th>
<th>Memory Pool Details</th>
<th>Memory Pool Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Pool Utilization</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Application Details
- Metrics
This section shows the list of applications connected to the database. You can select an application to show the breakdown of the memory pool usage by the application.

Metrics
None.

Memory Pool Details
- Metrics
This section shows the memory pool usage details for the selected application (or database level) in the application details section. The metrics shown here include:
  - Pool ID: This is the kind of memory pool. Each pool type is only shown once
  - Pool Current Size: The current size of a memory pool.
  - Pool Max Size: The internally configured size of a memory pool in DB2.
  - Pool High Watermark: The largest size of a memory pool since its creation.

Metrics
The nature of memory_pool data elements varies between platforms. On Windows systems, no memory usage is reported at the database level, while on UNIX systems, memory is reported at the database level. Instead of reporting this memory at the database level, the system monitor for Windows systems reports it in instance -level snapshots. This divergence in reporting is due to differences in the underlying memory architecture between Windows systems and UNIX systems.
Memory Pool Usage
This section shows the memory pool usage details of the selected memory pool over time including the current size of the memory pool, its high watermark, and the maximum size of the memory pool.

Memory Pool Utilization
- Metrics
- Troubleshooting

This section shows the percentage utilization of the selected memory pool

Metrics
You can use this metric to see if a memory pool is nearly full. You can diagnose the problems with specific memory pools by monitoring their utilization over time.

Troubleshooting
If the value of the pool utilization is consistently close to or exceeds 100%, you may need to increase the configuration parameter associated with that pool.

Lock Memory Tab
The Lock Memory tab of the Memory Detail view shows the following statistics:
- Locklist Utilization
- Lock Escalations

Locklist Utilization
- Metrics
- Troubleshooting

Lock list utilization is the percentage of total database memory allocated for locks that is currently being used.

**NOTE:** This metric is also available on the Home page.

Metrics
There is only one lock list for each database and it contains the locks held by all applications connected to the database. Once the lock list is full, the database manager starts escalating row locks to table locks to free up space. This escalation may result in serious performance degradation because of reduced concurrency. Additionally, the number of deadlocks and transaction rollbacks may go up.

If this metric reaches the 75% mark, you should consider bringing this percentage down with tuning.

Troubleshooting
Depending on the database’s activity level, you may be able to reduce the lock utilization by following these recommendations:

**Increase size of lock list:** If there is not enough lock list space available, lock escalations will occur, thereby increasing contention and reducing concurrency. Update the locklist database configuration parameter to increase this value.
Tune applications to reduce locking: On the Locks tab of the Users detail section, identify the applications that are holding many locks and then consider the following steps for controlling the size of the lock list:

- Make the transactions shorter by increasing the COMMIT frequency. This ensures that locks are released frequently, thus freeing up lock space.

- Before you update many rows in a single table, lock the entire table (using the SQL LOCK TABLE statement). This prevents many row-level locks from being obtained (although this decreases concurrency)

- To control how locking is done for a specific table, use the LOCKSIZE parameter of the ALTER TABLE.

- To decrease the number of share locks held, use the Cursor Stability isolation level when possible. If the applications’ integrity requirements are not compromised, use Uncommitted Read instead of Cursor Stability to further decrease the amount of locking.

Decrease percentage of lock list: If a small number of applications are consuming most of the lock space, decrease the percentage of lock list for each application. You can throttle back those applications by decreasing the maxlocks database configuration parameter. This reduces the amount of lock list memory available to each application thereby allowing for better distribution of lock list memory.

Decreasing the percentage of lock list should be the last resort, and used only if you cannot decrease utilization with the other recommendations. It can cause a large number of lock escalations.

Lock Escalations
- Metrics
- Troubleshooting

This section shows the metrics for lock escalations in the database. The metrics shown here are:

- Lock Escalations: The number of times that locks have been escalated from several row locks to a table lock.

- Exclusive Lock Escalations: The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.

Metrics
A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. The amount of lock list space available is determined by the maxlocks and locklist configuration parameters.

When an application reaches the maximum number of locks allowed and there are no more locks to escalate, it will then use space in the lock list allocated for other applications. When the entire lock list is full, an error occurs.

This data item includes a count of all lock escalations, including exclusive lock escalations.

There are several possible causes for excessive lock escalations:

- The lock list size (locklist) may be too small for the number of concurrent applications

- The percent of the lock list usable by each application (maxlocks) may be too small

- One or more applications may be using an excessive number of locks.

Troubleshooting
To resolve these problems, you may be able to:

- Increase the locklist configuration parameter value. See the Administration Guide for a description of this configuration parameter.
• Increase the maxlocks configuration parameter value. See the Administration Guide for a description of this configuration parameter.

• Identify the applications with large numbers of locks (see locks_held_top), or those that are holding too much of the lock list, using the following formula:

\[
(((\text{locks held} \times 36) / (\text{locklist} \times 4096)) \times 100)
\]

and comparing the value to maxlocks. These applications can also cause lock escalations in other applications by using too large a portion of the lock list. These applications may need to resort to using table locks instead of row locks, although table locks may cause an increase in lock_waits and lock_wait_time.

### I/O Statistics - DB2

The I/O performance category page displays the following vital DB2 I/O statistics:

<table>
<thead>
<tr>
<th>Bottleneck Analysis - I/O</th>
<th>Key System Analysis - I/O</th>
<th>SQL Analysis - I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Buffered I/O Activity</td>
<td>Workload Analysis - I/O</td>
<td></td>
</tr>
</tbody>
</table>

### Key System Analysis

The root cause of many database problems can be attributed to I/O-related operations. I/O monitoring is essentially a review of your physical database design.

All physical storage characteristics and placements, table and index designs, and the speed with which all processes works are on display when I/O is monitored. Because a database’s main index of performance is measured by how fast I/O needs are satisfied, it is your responsibility to quickly determine if a reported database slowdown is I/O related.

The following statistics are used on the Performance Analyst for DB2 I/O Home Page to succinctly communicate the general overall performance levels of I/O:

#### Efficiency

<table>
<thead>
<tr>
<th>Async Read Ratio</th>
<th>Async Write Ratio</th>
<th>Victim Page Cleaners %</th>
</tr>
</thead>
</table>

#### Resource Utilization

<table>
<thead>
<tr>
<th>Log Space Cleaners Triggered</th>
<th>Threshold Cleaners Triggered</th>
<th>Victim Page Cleaners Triggered</th>
</tr>
</thead>
</table>

Async Read Ratio

• **Metrics**

• **Troubleshooting**

Async read ratio is the percentage of all index and data reads that performed asynchronously by the database manager prefachers.
Metrics
Prefetching refers to the technique whereby the Database Manager can read several pages on the physical reads device simultaneously into a bufferpool in anticipation of access to these pages. Since each physical read operation is costly, a 50% async read ratio is desirable. It shows that the prefellers are working well and read waits are being minimized.

Troubleshooting
The num_ioserver database configuration parameter defines the number of prefellers that are available for the database. To get an optimal async read ratio, you should set the num_ioserver database configuration parameter to 1-2 more than the number of physical storage devices used by the database to ensure that asynchronous I/O is occurring. This parameter should be changed judiciously. Having too many prefellers invariably results in high system I/O because prefellers can read many more pages than required into the bufferpools.

Async Write Ratio

Metrics
Page cleaners write changed pages from bufferpool to disk before the space in the bufferpool is needed by a database agent. If the page cleaners are not working well, the problem may manifest itself in two ways:

• The database agents need to synchronously free space in bufferpools resulting in poorer response time.
• If the system crashes, the recovery time of the system is greater because there will be a large number of pages that would not have been committed to disk at the time of the crash and they require processing for a large number of log records.

An overall async write ratio of 50% is desirable for most transactional database systems. If your database is 'query only' (i.e., updates are rarely or never performed), it is fine to have an async write ratio of 0 to less than 50%. Even in 'query only' databases if the queries create temp tables on the database for intermediate results, a 30-40% async write ratio is desirable.

Troubleshooting
The async write ratio can be tuned using the num_iocleaners database configuration parameter, which specifies the number of asynchronous page cleaners for a database. Increasing the number of page cleaners generally results in a higher async write ratio. The following rules of thumb should be followed when tuning this parameter:

• For most transactional systems set this parameter to between one and the number of physical storage devices used by the database.
• For workloads that consists mainly of simple reads from the database, set this parameter to zero.
• When you define very large bufferpools on your database you may need to increase the num_iocleaners
• Monitor the victim page cleaners% metric when tuning the num_iocleaners.

Victim Page Cleaners %

• Metrics
• Troubleshooting
The victim page cleaners % metric is the percentage of times the page cleaners were invoked because synchronous write was needed during the victim buffer replacement in the database.

**Metrics**

Usually this metric should be maintained around the 50-60% level. If this ratio much lower than 50%, it can indicate that you have defined too many page cleaners. If the `chnpgs_thresh` database configuration parameter is set too low, the page cleaners may be writing out pages that will be dirtied later. Aggressive cleaning defeats one purpose of the bufferpool, that is to defer writing to the last possible moment.

If this ratio is high, it may indicate that you have too few page cleaners defined. Too few page cleaners will increase recovery time after failures.

**Troubleshooting**

You may increase or decrease the `chnpgs_thresh` database configuration parameter to bring the victim page cleaner % to optimal range. Usually, for databases with heavy transaction workloads, `chnpgs_thresh` should be set to around 60%. For a database with only a few very large tables you can try and increase this parameter to a higher value to see if you get performance improvements.

**Note:** DB2 version 8.1.4 introduces the DB2_USE_ALTERNATE_PAGE_CLEANING registry variable. If this is set to ON, this metric may not be relevant.

---

**Log Space Cleaners Triggered**

**Metrics**

The log space cleaners triggered metric is the number of times a page cleaner was triggered because log space usage reached a predefined threshold for the database.

**Troubleshooting**

The log space cleaners are triggered every time the space used by the DB2 log reaches the limit set in the `softmax` database configuration parameter. This parameter specifies the percentage of primary log size at which the cleaners are triggered. By default this parameter is set to 100.

A low rate of log space cleaners getting triggered may indicate that logs are not being written to disk frequently enough and that you may need to process a large number of log records and redundant log records in the event of a crash recovery.

High rates of log space cleaners getting triggered can indicate that your primary log is too small or that you have set the `softmax` too high. In either case, a very high rate of log cleaners being triggered may adversely impact database performance.

It is important to keep in mind, however, that more page cleaner triggers and more frequent soft checkpoints increase the overhead associated with database logging. This can have an impact on the performance of the Database Manager. Also, more frequent soft checkpoints may not reduce the time required to restart a database, if you have:

- Very long transactions with few commit points.
- A very large bufferpool and the pages containing the committed transactions are not written back to disk very frequently. The use of asynchronous page cleaners can help avoid this situation.

In both of these cases, the log control information kept in memory does not change frequently and there is no advantage in writing the log control information to disk, unless it has changed.
Troubleshooting
Examine this metric together with other page cleaner metrics and the async write ratio to determine if excessive numbers of log page cleaners are being triggered. If this is true, you need to either increase your primary log size or adjust the *softmax* parameter to a higher value.

**NOTE:** Decreasing the log page cleaner triggering rate can impact the overall crash recovery time so it needs to be done judiciously.

Victim Page Cleaners Triggered

- **Metrics**
- **Troubleshooting**

The victim page cleaners triggered statistic is the number of times the page cleaners were invoked because synchronous write was necessary during the victim buffer replacement in the database.

**Metrics**

If the *chngpgs_thresh* database configuration parameter is set too low, the page cleaners may be writing out pages that will be dirtied later. Aggressive cleaning defeats one purpose of the bufferpool, which is to defer writing to the last possible moment.

If this ratio is high, it may indicate that you have too few page cleaners defined. Too few page cleaners increases recovery time after failures.

**Troubleshooting**

You may increase or decrease the *chngpgs_thresh* database configuration parameter to bring the victim page cleaner triggering rate to optimal range. For databases with heavy transaction workloads, *chngpgs_thresh* should be set to around 60%. For a database with only a few very large tables you can try and increase this parameter to a higher value to see performance improvements.

**NOTE:** DB2 v8.1.4 introduces the DB2_USE_ALTERNATE_PAGE_CLEANING registry variable. If this is set to ON, this metric may not be relevant.

Threshold Cleaners Triggered

- **Metrics**
- **Troubleshooting**

The threshold cleaners triggered metric is the number of times a page cleaner was invoked because a bufferpool had reached the dirty page threshold criterion for the database.

**Metrics**

The dirty page threshold is set by the *chngpgs_thresh* database configuration parameter. It is a percentage applied to the bufferpool size. When the number of dirty pages in the pool exceeds this value, the cleaners are triggered.

If this value is set too low, pages might be written out too early, requiring them to be read back in. If set too high, then too many pages may accumulate, requiring users to write out pages synchronously.

**Troubleshooting**

For databases with frequent update transactions, you can generally ensure that there are enough clean pages in the bufferpool by setting the *chngpgs_thresh* parameter value to be equal-to or less-than the default value. A percentage larger than the default can help performance if your database has a small number of very large tables.
**Bottleneck Analysis - I/O**

High prefetch waits and file closures are good indicators of I/O waits at the database.

Often, heavy I/O activity may not be due to non-optimal settings of the database configuration parameters. Rather, the underlying cause can be poor physical layout of a few tables/tables spaces or a few poorly written/suboptimal database applications.

It is usually best to try and tune individual objects/applications before attempting to tune database configuration parameters. Database configuration changes can have a profound impact on every application and object that resides in the database and if there are only a couple of apps/objects dragging down the performance of the entire database, it may be best to first isolate and tune these.

I/O Bottleneck analysis provides information on applications and tablespaces that are performing the highest number of read and write operations on the database at a glance. It also shows the bufferpool I/O metrics that point to direct I/O waits.

The following statistics are used on the Performance Analyst for DB2 I/O home page to succinctly communicate the general overall performance levels of I/O:

<table>
<thead>
<tr>
<th>Hottest Tablespaces</th>
<th>Hottest Apps</th>
<th>Total Database Files Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Time Spent Waiting for Prefetchers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Hottest Tablespaces**

- **Metrics**
- **Troubleshooting**

The hottest tablespaces section displays the list of top three tablespaces with the highest number of physical reads and writes.

**Metrics**

Understanding the storage-level hot spots of a database is beneficial for a two reasons. First, you can get a feel for overworked physical disks by viewing I/O statistics at the table space level. If a particular table space is under too much strain, you can attempt to relocate table space containers to other less-used devices. You can also try to create new table spaces on different disks and move hot objects to them (assuming, of course, you have extra disks available). Second, if you have used standard DBA practices and placed indexes in their own table space, you can review the I/O statistics for the hottest table spaces and see if the indexes are actually being used.

**Troubleshooting**

Some areas to consider when reviewing the hottest table spaces display are:

- A temp table space showing high volumes of physical I/O could indicate a problem with excessive sort overflows.
- Quickly review all physical I/O for each drive/file system at the OS level and get a feel for the overworked disks on your server. If you have underutilized disk drives (with their own controllers), then you should consider relocating some table spaces/containers that exhibit high I/O characteristics to those drives.
- To minimize contention at the physical disk level, and to improve performance overall, it is generally recommended that a database have its table spaces and log files physically separated onto different drives/file systems.

**Hottest Apps**

- **Metrics**
**Metrics**

A few applications with suboptimal queries/database accesses can drag down the performance of the entire database. The hottest apps section allows you to get a good feel for the applications that are causing the highest number of reads and writes on the database. It is also a good indicator of where you will reap the greatest benefit when tuning the application's performance.

**Troubleshooting**

- Drill down to the application I/O pages and see how the high I/O applications are behaving compared to other applications.
- Check the SQL Snapshot Metrics for the high read/write apps to see what queries these applications are running on the database and try to optimize these queries.

**Total Database Files Closed**

**Metrics**

Both SMS tablespaces and DMS tablespace file containers are treated as files in the Database Manager's interaction with the operating system, and file handles are required. There is a limit to the number of files that can be open concurrently by each database agent and a file is closed when the limit of open files for an agent is reached and the agent needs to open another file. This metric gives the number of times a database file was closed because the limit for concurrently open files was reached.

**Troubleshooting**

If you notice that files are being closed during the monitoring, first check the rate at which the files are being closed at the database level, from the I/O drill-down. If you see files being closed frequently, increase the `maxfilop` database configuration parameter. Note that the operating system on which the database is running also has a limit on the number of files that can be opened concurrently. Make sure that you do not increase the `maxfilop` parameter beyond the operating system limit.

**Total Time Spent Waiting for Prefetchers**

This is the total time database applications spent waiting for prefetchers to finish loading.
Metrics
None.

Troubleshooting
If you see a high number for this metric, it indicates there is a bottleneck in the prefetcher operations. Experiment with changing the number of I/O servers (num_ioserver) and the I/O server sizes.

SQL Analysis - I/O
Most of database’s I/O performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag down an otherwise well-configured server in terms of user response times.

This section shows a snapshot of row level SQL operations that have been performed on the database since you started monitoring. This gives you an insight into the row level of read/write activity that is currently occurring within the database and comparing this to the prefetcher, and cleaner activity allows you to relate the two.

The following statistics are used on the Performance Analyst for DB2 I/O home page to succinctly communicate the general overall performance levels of I/O:

<table>
<thead>
<tr>
<th>Rows Deleted</th>
<th>Rows Inserted</th>
<th>Rows Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows Updated</td>
<td>Rows Read</td>
<td></td>
</tr>
</tbody>
</table>

Rows Deleted
- **Metrics**

The rows deleted metric is the total number of row deletions attempted on the database. It includes the internal row deletions that were caused by constraints or triggers.

**Metrics**
None.

Rows Inserted
- **Metrics**

The rows inserted metric is the total number of row insertions attempted on the database. This metric includes the internal row insertions that were caused by triggers.

**Metrics**
None.

Rows Selected
- **Metrics**

The rows selected statistic is the total number of rows selected at the database. This metric does not include a count of rows accesses for operations such as COUNT(*) or joins.

**Metrics**
None.
Rows Updated
   • Metrics

The rows updated metric is the total number of row updates attempted on the database. This metric includes the internal row updates that were caused by triggers or constraints.

Metrics
None.

Rows Read
   • Metrics

The rows read statistic is the number of rows read from the table. This count is not the number of rows that were returned to the calling application. Rather, it is the number of rows that had to be read in order to return the result set.

Metrics
None.

Non Buffered I/O Activity

Some database operations such as backups, loads, restores, LOB (large object I/O) bypass the bufferpools and read and write directly to storage. Sometimes, even if your database I/O configurations and all of the other activities are within the normal thresholds, the I/O performance may be poor because LOB reads/writes or load/backup/restore operations are causing or waiting on I/O contentions on your storage devices.

The non buffered I/O activity section allows you to quickly identify the level of direct I/O activity on the database. The level of direct I/O activity is succinctly presented using the following metrics:

<table>
<thead>
<tr>
<th>Direct Reads from the Database</th>
<th>Direct Writes to the Database</th>
<th>Direct I/O Averages</th>
</tr>
</thead>
</table>

Direct Reads from the Database
   • Metrics

The direct reads from the database metric identifies the number of read operations that did not use a bufferpool.

Direct reads are performed for the following database operations:

- Reading LONG VARCHAR columns
- Reading LOB (BLOB/CLOB/DBCLOB) columns
- Performing a backup

Metrics
None.

Direct Writes to the Database
   • Metrics

The direct writes to the database metric identifies the number of write operations that did not use a bufferpool.
Direct writes are performed for the following database operations:

- Writing LONG VARCHAR columns
- Writing LOB (BLOB/CLOB/DBCLOB) columns
- Performing a load
- Performing a restore

**Metrics**
None.

**Direct I/O Averages**

- **Metrics**
- **Troubleshooting**

The following values show how well the direct I/O operations are performing:

- Avg. Sectors / Read: Average number of sectors read for each direct read operation
- Avg Sectors / Write: Average number of sectors written for each direct write operation
- Avg Read Time: Average read time per sector for direct reads
- Avg Write Time: Average write time per sector for direct writes

**Metrics**
Understanding the level of direct I/O averages is important. A high level of activity with a slowdown in database performance can indicate that you need to have dedicated and/or high performance storage devices for your LOB objects or backup and load images.

These metrics, when viewed with the device metrics at the OS level can also indicate I/O resource conflicts with other processes.

**Troubleshooting**
You can monitor these metrics for the LOB storage devices at the OS level to see if they are inordinately high for the database compared to the OS level. If so, there may be an I/O conflict with another process.

**Workload Analysis - I/O**

When your phone starts ringing with complaints of performance slowdowns, some of the first questions you should answer are:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst I/O page provides insight into the leading resource hogs of a server, as well as a count of total active and inactive processes. Drill-downs are available so you can easily get detailed information into what each leading application is currently involved with.

Workload Analysis statistics includes:
Top I/O Apps

The Top I/O Apps section gives a list of the apps with performing that are performing the highest number of most aggressive I/O activities. The list includes:

- Application with most bufferpool misses
- Application with highest direct reads and writes
- Application with highest sort overflows
- Application with highest direct read/write time
- Application with highest bufferpool read/write time

Metrics

None.

Troubleshooting

Drill down to the application I/O level to see the details and compare the metrics with the application’s peers, and the database. In a lot of cases simple SQL rewrites, logic changes, and defining indexes based on access patterns can make a world of difference.

I/O Detail View

The following tabbed pages are available on the I/O Detail view:

<table>
<thead>
<tr>
<th>Applications Tab</th>
<th>Bufferpools Tab</th>
<th>Logging Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables Spaces Tab</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Applications Tab

- Metrics
- Troubleshooting

This section shows the list of applications connected to the database and the various I/O metrics for each application. It also shows the rolled up metrics at the database level in the same list. You can select the type of I/O monitoring you intend to perform for the applications from the following:

- Direct I/O: This shows the details of all direct I/O activity for the applications.
- Buffered I/O: This shows the details of all bufferpool I/O activity for the applications connected to the database.
- Extended Storage: This shows the level of extended storage usage by the applications.
- Sorts & Joins: This shows the details for the number of files closed for the various applications.
- Block / Vectored I/O: This shows the detailed of block and vectored I/O performed by the applications.
Metrics
None.

Troubleshooting
Depending on the type of details you select, you can view the different types of I/O activity associated with the applications. This section is very useful in isolating the applications, and the I/O areas you need to tune by comparing the I/O performance for all the applications and also the overall I/O performance at the database level.

Direct I/O
The Direct I/O Application or Tablespace Details view lets you the following statistics:

<table>
<thead>
<tr>
<th>Direct Read and Writes</th>
<th>Direct Read and Write Requests</th>
<th>Direct Read and Writes Time</th>
</tr>
</thead>
</table>

Direct Reads and Writes
- **Metrics**
  This section shows the rate at which direct read/write operations are occurring for the selected element. Direct I/O operations are those that do not use the bufferpools. Direct read and write operations are performed when:
  - Reading/Writing LONG VARCHAR columns
  - Reading/Writing LOB data
  - Performing a backup operation
  - Performing a restore operation
  - Performing a load operation

Metrics
When using the performance analyst to track I/O, these elements help you distinguish the database I/O from the non-database I/O on storage devices.

Direct Read and Write Requests
- **Metrics**
- **Troubleshooting**
  This section shows the rate at which direct read/write operations requests are occurring for the selected element. Direct I/O operations are those that do not use the bufferpools. Direct read and write operations are performed when:
  - Reading/Writing LONG VARCHAR columns
  - Reading/Writing LOB data
  - Performing a backup operation
  - Performing a restore operation
  - Performing a load operation
Metrics
Use these metrics together with the Direct R/W metrics and Direct R/W request metrics to determine if the rate at which the read/write occurring is similar to the rate at which they are being satisfied. If there is a large discrepancy between the Direct I/O request rates and Direct I/O rates, it may mean that there are I/O contentions on the device(s) where the direct I/O operations are being performed.

Troubleshooting
If you suspect there are I/O contentions, you can investigate whether the contentions are being caused by the DB2 processes or other processes by examining the I/O metrics at the operating system level. Consider using dedicated devices for the direct I/O operations to reduce the contention levels.

Direct Read and Writes Time
- **Metrics**
- **Troubleshooting**

This section shows the elapsed time to perform the direct I/O operations. Direct I/O operations are those that do not use the bufferpools. Direct read and write operations are performed when:

- Reading/Writing LONG VARCHAR columns
- Reading/Writing LOB data
- Performing a backup operation
- Performing a restore operation
- Performing a load operation

**Metrics**
Use these metrics together with the Direct R/W metrics to determine average time it is taking to perform the read write operations. If the level of Direct I/O rates is low and the Direct RW times, it may mean that there are I/O contentions on the device(s) where the direct I/O operations are being performed.

**Troubleshooting**
If you suspect there are I/O contentions, you can investigate whether the contentions are being caused by the DB2 processes or other processes by examining the I/O metrics at the operating system level. Consider using dedicated devices for the direct I/O operations to reduce the contention levels. If the contention levels are low, you may also benefit from using faster storage devices for direct I/O.

Buffered I/O
The Buffered I/O Application Details view lets you see the following statistics:

- **Hit Ratios**
- **Write/Read Ratios**

**Hit Ratios**
- **Metrics**
- **Troubleshooting**
The DB2 database server reads and updates all data from a bufferpool because memory access is much faster than disk access. Data is copied from disk to a bufferpool as needed by the applications using the database. When the server needs to read/write data and the data is already in the bufferpool, no disk access is required. However, if the data is not in the bufferpool, it needs to be read from the disk, which is significantly slower process.

The buffer pool hit ratio indicates the percentage of time that the database server did not need to load a page to service the request for a page. The calculation takes into account all the index and data pages that were requested.

NOTE: This statistic can also be found on the Home page.

Metrics
Avoiding disk I/O is the main issue when you try to improve the performance tuning. A high buffer pool hit ratio is desirable because it indicates a lower frequency of synchronous disk I/O. A database where data is accessed uniformly from very large tables will have a poor hit ratio. There is little you can do to improve the performance in such cases.

Troubleshooting
The buffer pool hit ratio on the database Home page is the overall hit ratio of the database. First, drill down to the bufferpool level and check the individual buffer pool hit ratios to identify the elements that may require tuning. Increasing the bufferpool size generally improves the buffer pool hit ratio. You can use one of the following methods to increase the bufferpool size depending on the layout of the data:

1. If the tablespaces using the bufferpool have tables and indexes on them, increase the bufferpool size incrementally until the index page hit ratio stops increasing for that bufferpool. You will not usually get any benefit increasing the bufferpool size after you reach that point.

2. If the tablespaces associated with the bufferpool have only indexes on them, increase the bufferpool size until the index page hit ratio stops increasing.

3. If the tablespaces associated with the bufferpool have only data on them, increase the bufferpool size until the data page hit ratio stops increasing.

One general design guideline is to try and use different tablespaces for indexes and tables and associate different bufferpools with them. This generally improves performance and makes bufferpool tuning much easier.

Write/Read Ratios
This section shows the overall level of read / write activity being performed on the selected element.

Extended Storage
The Extended Storage Application or Transaction Details view lets you see the following statistics:

- Index I/O on Extended Storage
- Data I/O on Extended Storage

Index I/O on Extended Storage
- Metrics
- Troubleshooting
This section shows the Index Pages copied to/from extended storage for the selected element.
On 32-bit platforms, virtual addressable memory is usually limited to between 2 and 4 GB. If your 32-bit machine has more real addressable memory than the maximum amount, you can configure any additional real addressable memory beyond virtual addressable memory as an extended storage cache. Any of the defined bufferpools can use an extended storage cache to improve performance. You define the extended storage cache as a number of memory segments.

The bufferpools perform first-level caching, and any extended storage cache is used by the bufferpools as secondary-level caching. Ideally, the bufferpools hold the data that is most frequently accessed, while the extended storage cache hold data that is accessed less frequently.

Pages are copied from the bufferpool to extended storage, when they are selected as victim pages. This copying is required to make space for new pages in the bufferpool.

Required index pages are copied from extended storage to the bufferpool, if they are not in the bufferpool, but are in extended storage. This copying may incur the cost of connecting to the shared memory segment, but saves the cost of a disk read.

**Metrics**

Because an extended storage cache is an extension to a bufferpool, it must always be associated with one or more specific bufferpools. Therefore, only the bufferpools that are declared to take advantage of the extended storage use it. The CREATE and ALTER BUFFERPOOL statements have the attributes NOT EXTENDED STORAGE and EXTENDED STORAGE that control cache usage. By default neither IBMDEFAULTBP nor any newly created bufferpool will use extended storage.

Use these metrics together with the Data I/O on Extended Storage metrics to determine the level of I/O being performed on extended storage. If there is significant activity on this section, you may improve performance by increasing the size of extended storage.

**Troubleshooting**

The amount of extended storage for a database is determined by the following two database configuration parameters:

- `num_estore_segs` defines the number of extended storage memory segments. The default for this configuration parameter is zero, which specifies that no extended storage cache exists.
- `estore_seg_sz` defines the size of each extended memory segment. This size is determined by the platform on which the extended storage cache is used.

**Data I/O on Extended Storage**

This section shows the DataPages copied to/from extended storage for the selected element.

On 32-bit platforms, virtual addressable memory is usually limited to between 2 and 4 GB. If your 32-bit machine has more real addressable memory than the maximum amount, you can configure any additional real addressable memory beyond virtual addressable memory as an extended storage cache. Any of the defined bufferpools can use an extended storage cache to improve performance. You define the extended storage cache as a number of memory segments.

The bufferpools perform the first-level caching, and any extended storage cache is used by the bufferpools as secondary-level caching. Ideally, the bufferpools hold the data that is most frequently accessed, while the extended storage cache hold data that is accessed less frequently.

Pages are copied from the bufferpool to extended storage, when they are selected as victim pages. This copying is required to make space for new pages in the bufferpool.
Required pages are copied from extended storage to the buffer pool, if they are not in the bufferpool, but are in extended storage. This copying may incur the cost of connecting to the shared memory segment, but saves the cost of a disk read.

**Metrics**

Because an extended storage cache is an extension to a bufferpool, it must always be associated with one or more specific bufferpools. Therefore, only the bufferpools that are declared to take advantage of advantage of the extended storage use it. The CREATE and ALTER BUFFERPOOL statements have the attributes NOT EXTENDED STORAGE and EXTENDED STORAGE that control cache usage. By default neither IBMDEFAULTBP nor any newly created bufferpool will use extended storage.

Use these metrics together with the Index I/O on Extended Storage metrics to determine the level of I/O being performed on extended storage. If there is significant activity on this section, you can improve performance by increasing the size of extended storage.

**Bufferpools Tab**

- **Metrics**
- **Troubleshooting**

This section shows the list of bufferpools on the database and the various I/O metrics for each bufferpool. It also shows the rolled up metrics at the database level in the same list. You can select the type of monitoring you intend to perform for the bufferpools from the following:

- **Direct I/O**: This shows the details of all direct I/O activity for the various bufferpools
- **Buffered I/O**: This shows the details of all bufferpool I/O activity for the various bufferpools in the database
- **Extended Storage**: This shows the level of extended storage usage by the various bufferpools that use extended storage
- **Files Closed**: This shows the details for the number of files closed for the various bufferpools in the database
- **Block / Vectored I/O**: This shows the detailed of block and vectored I/O performed by the various bufferpools.

**Metrics**

None.

**Troubleshooting**

Depending on the type of details you select, you can view the different types of I/O activity associated with the bufferpools. This section is very useful in isolating the bufferpools, and the I/O areas you need to tune by comparing the I/O performance for all the bufferpools and also the overall I/O performance at the database level.

**Block/Vectored I/O**

The Block/Vectored I/O section of the Bufferpool Details view displays the following statistics:

- **Vectored I/O**
- **Block I/O**

**Vectored I/O**

The Vectored I/O section gives the metrics for vectored I/O requests and reads.
**Metrics**
Use this element to determine how often vectored I/O is being done. The number of vectored I/O requests is monitored only during sequential prefetching.

**Block I/O**
- **Metrics**
- **Troubleshooting**

The Block I/O section gives the metrics for block I/O requests and reads. If block-based bufferpool is enabled, this section reports how often block I/O is being done. The number of block I/O requests is monitored only during sequential prefetching when using block-based bufferpools.

**Metrics**
If block-based bufferpool is enabled and the block I/O activity is very low, or close to the values in the vectored I/O section, consider changing the block size. This state can be an indication of the following:

- The extent size of one or more tablespaces bound to the bufferpool is smaller than the block size specified for the bufferpool.
- Some pages requested in the prefetch request are already present in the page area of the bufferpool.

The prefetcher allows some wasted pages in each bufferpool block, but if too many pages are wasted, then the prefetcher will decide to perform vectored I/O into the page area of the bufferpool.

To take full advantage of the sequential prefetch performance improvements that block-based bufferpools provide, it is essential to choose an appropriate value for the block size. This can, however, be difficult because multiple tablespaces with different extent sizes can be bound to the same block-based bufferpool. For optimal performance, it is recommended that you bind tablespaces with the same extent size to a block-based bufferpool with a block size equal to the extent size. Good performance can be achieved when the extent size of the tablespaces are greater than the block size, but not when the extent size is smaller than the block size.

For example, if extent size is 2 and block size is 8, vectored I/O would be used instead of block I/O (block I/O would have wasted 6 pages). A reduction of the block size to 2 would solve this problem.

**Troubleshooting**
You can alter the block size for a buffer pool using the ALTER BUFFERPOOLddl statement.

**Logging Tab**
- **Metrics**

The Logging tab of the I/O Detail view displays Logging I/O statistics. This section shows the level of read/write activity being performed by the logger. The information presented here is:

**Log Reads:** The number of log pages read from the disk by the logger per second.

**Log Writes:** The number of log pages written to the disk by the logger.

**Metrics**
You can use this page with the operating system level I/O metrics, to quantify the amount of I/O on a device that is attributable to database activity.
Tablescape Tab
- Metrics
- Troubleshooting

The Tablespace Tab of the I/O detail view shows the list of tablespaces in the database and the I/O metrics for each tablespace. It also shows the rolled-up metrics at the database level in the same list. You can select the type of monitoring you intend to perform for the tablespaces from the following:

- **Direct I/O**: This shows the details of all direct I/O activity for the various tablespaces
- **Buffered I/O**: This shows the details of all bufferpool I/O activity for the various tablespaces in the database
- **Extended Storage**: This shows the level of extended storage usage by the various tablespaces that use extended storage
- **Files Closed**: This shows the details for the number of files closed for the various tablespaces in the database

**Metrics**
None.

**Troubleshooting**
Depending on the type of details you select, you can view the different types of I/O activity associated with the tablespaces. This section is very useful in isolating the tablespaces, and the I/O areas you need to tune by comparing the I/O performance for all the tablespaces and also the overall I/O performance at the database level.

**Buffered I/O**
The following statistics are available on the Buffered I/O section of the Tablespace Details of the I/O view:

<table>
<thead>
<tr>
<th>Hit Ratios</th>
<th>Async I/O</th>
<th>Write/Read Ratios</th>
</tr>
</thead>
</table>

**Async I/O**
- Metrics
- Troubleshooting

Async read ratio is the percentage of all index and data reads that were performed asynchronously by the database manager prefetchers.

Async write ratio is the percentage of all index and data writes that were performed asynchronously by the database manager bufferpool page cleaners.

**Metrics**
Prefetching refers to the technique whereby the Database Manager can read several pages on the physical reads device simultaneously into a bufferpool in anticipation of an access to these pages. Since each physical read operation is costly, a 50% async read ratio is desirable since it shows that the prefetchers are working well and read waits are being minimized.

Page cleaners write changed pages from bufferpool to disk before the space in the bufferpool is needed by a database agent. If the page cleaners are not working well, the problem may manifest itself in two ways:

- The database agents will need to synchronously free up space in bufferpools resulting in poorer response time.
If the system crashes, the recovery time of the system will be greater because there will be a large number of pages that would not have been committed to disk at time of crash and will require processing of a large number of log records.

An overall async write ratio of 50% is desirable for most transactional database systems. If your database is ‘query only’ (i.e., updates are rarely or never performed), it is okay to have an async write ratio of 0 to less than 50%. Even in ‘query only’ databases if the queries create temp tables on the database for intermediate results, a 30-40% async write ratio is desirable.

Troubleshooting

num_ioserver database configuration parameter defines the number of prefetchers that are available for the database. To get an optimal async read ratio, you should set the num_ioserver database configuration parameter to 1-2 more than the number of physical storage devices used by the database to ensure that asynchronous I/O is occurring. This parameter should be changed judiciously since having too many prefetchers invariably results in high system I/O because prefetchers can read many more pages than required into the bufferpools.

Async write ratio can be tuned using the num_iocleaners database configuration parameter. This parameter specifies the number of asynchronous page cleaners for a database. Increasing the number of page cleaners generally results in a higher async write ratio. The following rules of thumb can be followed when tuning this parameter:

- For most transactional systems set this parameter to between 1 and the number of physical storage devices used by the database.
- For workloads that consists of mainly simple reads from the database, set this parameter to 0.
- When you define very large bufferpools on your database you may need to increase the num_iocleaners.
- Monitor the victim page cleaners% metric when tuning the num_iocleaners.

Space Statistics - DB2

While DBAs focus on memory settings and tuning SQL, they often forget just how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

- The 'hit-the-wall' variety that can bring things to a complete standstill
- The ‘performance vampire’ kind that slowly drains the performance of a database over time

Storage problems can bring the curtain down on a database very quickly, as in the case of a transaction log running out of free space. But storage problems can also silently work behind the scenes to slowly but surely rob a database of its performance.

The Space performance category view displays the following vital DB2 space statistics:

<table>
<thead>
<tr>
<th>Tablespace Analysis</th>
<th>Bottleneck Analysis</th>
<th>SQL Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Analysis</td>
<td>Log Analysis</td>
<td></td>
</tr>
</tbody>
</table>

Related Topic

Space Detail View
Tablespace Analysis
The Tablespace Analysis section includes a list of all of the tablespaces in the database. The following information is presented for each tablespace:

- **Tablespace Name**: The name of the tablespace
- **Type**: system managed (SMS) or database managed (DMS) tablespace
- **State**: This element describes the current state of the tablespace
- **Used Space**: The space currently in use on tablespace
- **Free Space**: The space currently free on the tablespace. (This is not applicable to SMS tablespaces)
- **Content Type**: The type of data that can be stored on the tablespace. It can be one of the following:
  - Any (any type of data)
  - Long (LOB/Structured type / Index data)
  - System Temporary (work area used by database manager to perform operations like sorts and joins)
  - User Temporary (Stores declared global temporary tables)
- **Page Size**: Size of pages used by the tablespace
- **Extent Size**: Number of pages that will be written to a container before writing to the next container on the tablespace
- **Prefetch Size**: Number of pages that will be read from the tablespace by the prefetchers in a single prefetch
- **Total Space**: Total space (Used + Free) used by the tablespace on the storage device(s). This will be the same as used space for SMS tables
- **High Watermark**: Highest amount of used space for the tablespace (This is not applicable to SMS tablespaces)

Other statistics appearing in this section include:

<table>
<thead>
<tr>
<th>Total Tablespaces</th>
<th>SMS Tablespaces</th>
<th>DMS Tablespaces</th>
</tr>
</thead>
</table>

**Total Tablespaces**
The total tablespaces statistic is the total number of tablespaces in the database.

**SMS Tablespaces**
The SMS tablespaces statistic is the number of system managed (SMS) tablespaces in the database.

**DMS Tablespaces**
The DMS tablespaces statistic is the number of database managed (DMS) tablespaces in the database.

**Bottleneck Analysis**
Space shortfalls or inaccessible tablespaces can quickly bring a database to a halt. The space bottleneck analysis section provides an at a glance look at the items that may require your attention. You can drilldown to the details pages to see further details of the various metrics.
The metrics presented here are:

<table>
<thead>
<tr>
<th>DMS Tablespaces at Utilization &gt;80%</th>
<th>Inaccessible Tablespaces</th>
<th>Inaccessible Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abnormal Tablespaces</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DMS Tablespaces at Utilization >80%**

- **Metrics**
- **Troubleshooting**

The DMS tablespaces at utilization greater than 80% metric gives the number of DMS tablespaces in the database where the used space on the tablespace is more than 80% of the total available space.

**Metrics**

This metric is an indicator that some of the tablespaces in your database may be running out of space. You should drill down to the Space metrics to identify the tablespaces that may be at the critical utilization level. In tablespaces where there are little on no inserts after the initial data is loaded, little or no free space may have been left on purpose. In such cases, it is normal to see a high utilization.

**NOTE:** This statistic is also available on the Home page.

**Troubleshooting**

If this metric goes beyond the normal operational value for your database, consider creating additional space on the DMS tablespaces that are running out of space. You can do this by performing one or more one of the following tasks:

- Clean up unused space by deleting the tables/indexes that are no longer required.
- Resize or extend the existing containers in the tablespaces.
- Add new containers to the tablespaces.

**Inaccessible Tablespaces**

- **Metrics**
- **Troubleshooting**

The inaccessible tablespaces statistic is the number of tablespaces that are offline.

**Metrics**

Sometimes a problem (such as inaccessible containers) will cause a tablespace to go offline. Even when the problem is corrected, such tablespaces are not accessible by any application and need to be brought online explicitly.

**Troubleshooting**

1. From Tablespace Analysis section, identify the tablespaces whose state is set to Offline
2. Drill down to space details to make sure all of the containers for these tablespaces are accessible. If they are not, first fix the problem with the containers
3. Disconnect all applications from the database (you can use DBArtisan DB Monitor).
4. Reconnect to the database.
5. Use the ALTER TABLESPACE .. SWITCH ONLINE command to bring the tablespace back online.

- If the tablespace can be brought up successfully after issuing the command, or if the tablespace was not in the OFFLINE state to begin with, DB2 returns an SQLCODE of 0.
- If the tablespace cannot be brought up successfully because there are still problems with one or more of the containers, DB2 returns an SQLCODE of --293. You can force the database to restart by using the RESTART ...

Inaccessible Containers

- **Metrics**
- **Troubleshooting**

The inaccessible containers metric identifies the number of containers on all tablespaces that are currently inaccessible.

**NOTE:** This statistic is also available on the Home page.

**Metrics**

Inaccessible containers represent a serious problem on the database. The database is unable to access/write any data on an inaccessible container. Usually this is caused by either media errors or modifications to the container files/directories from outside DB2.

**Troubleshooting**

Drill down to the Space metrics to examine the inaccessible containers and the tablespaces on which they are contained. You may need to perform a redirected restore on the tablespace from a backup to correct this.

Abnormal Tablespaces

- **Metrics**
- **Troubleshooting**

The tablespaces in abnormal state metric is a count of the number of tablespaces in your database that are not in a 'Normal State'.

**NOTE:** This statistic is also available on the Home page.

**Metrics**

Tablespaces in not normal state are not necessarily a cause for alarm. They may have been transitioned to that state intentionally by the DBA for maintenance related tasks. If this metric shows a value that is higher than you anticipated, you may need to drilldown to the Space metrics to see which tablespaces are not normal.

A tablespace can be in one or more of the following states when it is not 'Normal':

- Quiesced: SHARE, UPDATE, EXCLUSIVE
- Pending: Load, delete, backup, roll forward, restore, disable, drop
- In progress: Roll forward, reorg, backup, TBS deletion, TBS creation
- Storage must be defined
- Restore in progress
- Offline and not accessible
• Storage may be defined
• Storage definition is in 'final' state
• Storage definition was changed prior to rollforward

DMS rebalancer is active

SQL Analysis
The metrics presented here are:

<table>
<thead>
<tr>
<th>DDL Statements Executed</th>
<th>Insert/Updates/Deletes Statements Executed</th>
<th>Select Statements Executed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total SQL Executed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DDL Statements Executed**

*Metrics*

The DDL statements executed element indicates the number of SQL Data Definition Language (DDL) statements that have been executed.

**Metrics**

You can use this element to determine the level of database activity at the application or database level. DDL statements are expensive to run because of their impact on the system catalog tables. As a result, if the value of this element is high, you should determine the cause, and possibly restrict this activity from being performed.

DDL statements can also impact:

• The catalog cache, by invalidating table descriptor information and authorization information that are stored there and causing additional system overhead to retrieve the information from the system catalogs
• The package cache, by invalidating sections that are stored there and causing additional system overhead due to section recompilation.

**Insert/Updates/Deletes Statements Executed**

*Metrics*

This is the number of SQL UPDATE, INSERT, and DELETE statements that were executed.

**Metrics**

You can use this element to determine the level of database activity at the application or database level. This information can be useful for analyzing application activity and throughput.

**Select Statements Executed**

*Metrics*

This statistic is the number of SQL SELECT statements that were executed.
Metrics
You can use this element to determine the level of database activity at the application or database level. This information can be useful for analyzing application activity and throughput.

Total SQL Executed
- **Metrics**

You can use this element to calculate the total number of SQL statements at the database level.

Metrics
This element can also help you in determine reasons for poor performance. Failed statements mean time wasted by the database manager, and as a result, lower throughput for the database. You can use this element in conjunction with the other SQL metrics to understand the portion of SQL activity that can be attributed to various statement types.

Storage Analysis
While DBAs focus on memory settings and tuning SQL, they frequently forget how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:
- The 'hit-the-wall' variety that can bring things to a complete standstill.
- The 'performance vampire' kind that slowly drains the performance of a database over time.

Storage problems have the capability to bring the curtain down on a database very quickly, as in the case of a transaction log running out of free space. But storage problems can also silently work behind the scenes to slowly, but surely rob a database of its performance. For example, a hub table in a busy database might be accessed very quickly when an application is first given life, but over time, if it develops a heavy forwarded row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page displays the total used and free space of all DMS tablespaces in the database. It also provides the total and available log space for the database.

The following ratios succinctly communicate the general overall performance levels of the datasource:

<table>
<thead>
<tr>
<th>DMS Space</th>
<th>Total SMS Space Used</th>
<th>Log Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Backup</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DMS Space
- **Metrics**
- **Troubleshooting**

The total used and total free metrics show the space details for DMS tablespaces. Specifically, they show the amount of used and free space on the DMS tablespaces in the database.
Metrics
Unlike SMS tablespaces, the total available space to the DMS tablespaces is defined at the time the database is defined. The DBA needs to explicitly increase the amount of space on DMS tables by adding/ extending tablespace containers (using the ALTER TABLESPACE statement). If left unattended, a DMS tablespace can either remain underutilized or fill up completely. Keeping an eye on the DMS space is important because the once it fills up, the applications trying to write to the database will come to a halt suddenly.

Troubleshooting
Go to the Space home page or drilldown to see the usage metrics for individual tablespaces and allocate/deallocate space to containers accordingly using ALTER TABLESPACE command.

Total SMS Space Used

This metric shows the total amount of SMS space used by the database.

Metrics
The maximum size of SMS tablespaces is not set at the time the tablespaces are created. The maximum size that can be reached is the space available on the drives/volumes that the SMS tablespace containers are defined on. You need to periodically examine the available space of the drives/volumes (using OS metrics Space page) to make sure there is enough space available for your database requirements. Since the space is determined by the space available on the drives/volumes, remember that if other applications are reading and writing to the same devices (especially logs, traces, etc), DB2 may be periodically competing for space with these applications.

Troubleshooting
Ideally, try and isolate the DB2 SMS drives/volumes from other applications. If the OS level metrics show that you are running out of space on a volume that is used by a tablespace’s containers, you can add more containers defined on volumes with more space, to the tablespace using the ALTER TABLESPACE command.

Log Analysis
The Log Analysis section gives you at-a-glance information about all your log configuration parameters and the high watermarks for the log space usage. You can use this information to quickly determine the logging mechanism in place for your database and if the high watermarks are getting too close to the maximum allocations for the log space.

The metrics presented here are:

Log Configuration

The log configuration metric gives a list of space related log configuration parameters for your database. The values of these parameters determine how the logger uses storage space on the database:
• Block on Log Disk Full (blk_log_dsk_ful): this configuration parameter can be set to prevent disk full errors from being generated when DB2 cannot create a new log file in the active log path. Instead, DB2 will attempt to create the log file every five minutes until it succeeds. After each attempt, DB2 will write a message to the administration notification log. The only way to confirm that your application is hanging because of a log disk full condition is to monitor the administration notification log. Until the log file is successfully created, any user application that attempts to update table data is not able to commit transactions. Read-only queries may not be directly affected; however, if a query needs to access data that is locked by an update request or a data page that is fixed in the bufferpool by the updating application, read-only queries also appear to hang.

Setting blk_log_dsk_ful to YES causes applications to hang when DB2 encounters a log disk full error. You are then able to resolve the error and the transaction can continue. A disk full situation can be resolved by moving old log files to another file system, or by increasing the size of the file system so that hanging applications can complete.

If blk_log_dsk_ful is set to NO, a transaction that receives a log disk full error fails and is rolled back. In some cases, the database comes down if a transaction causes a log disk full error.

• Log File Size (logfilsiz): specifies the size of each configured log, in number of 4-KB pages.

There is a 256-GB logical limit on the total active log space that you can configure. This limit is the result of the upper limit on logfilsiz, which is 262144, and the upper limit on \((logprimary + logsecond)\), which is 256.

The size of the log file has a direct bearing on performance. There is a performance cost for switching from one log to another. So, from a pure performance perspective, the larger the log file, size the better. This parameter also indicates the log file size for archiving. In this case, a larger log file is size it not necessarily better, since a larger log file size may increase the chance of failure or cause a delay in log shipping scenarios. When considering active log space, it may be better to have a larger number of smaller log files. For example, if there are two very large log files and a transaction starts close to the end of one log file, only half of the log space remains available.

Every time a database is deactivated (all connections to the database are terminated), the log file that is currently being written is truncated. So, if a database is frequently being deactivated, it is better not to choose a large log file size because DB2 will create a large file only to have it truncated. You can use the ACTIVATE DATABASE command to avoid this cost, and having the bufferpool primed will also help with performance.

Assuming that you have an application that keeps the database open to minimize processing time when opening the database, the log file size should be determined by the amount of time it takes to make offline archived log copies.

Minimizing log file loss is also an important consideration when setting the log size. Archiving takes an entire log. If you use a single large log, you increase the time between archiving. If the medium containing the log fails, some transaction information will probably be lost. Decreasing the log size increases the frequency of archiving but can reduce the amount of information loss in case of a media failure since the smaller logs before the one lost can be used.

• Log Retain (logretain): if logretain is set to RECOVERY, archived logs are kept in the database log path directory and the database is considered to be recoverable, meaning that rollforward recovery is enabled.

**NOTE:** The default value for the logretain database configuration parameter does not support rollforward recovery and must be changed if you are going to use it.

• User Exit (userexit): causes the database manager to call a user exit program for archiving and retrieving logs. The log files are archived in a location that is different from the active log path. If userexit is set to ON, rollforward recovery is enabled.

**NOTE:** The userexit database configuration parameter must be enabled to set logsecond parameter to 1.
• Primary Logs(logprimary): specifies the number of primary logs of size logfilsz that will be created.

A primary log, whether empty or full, requires the same amount of disk space. Thus, if you configure more logs than you need, you use disk space unnecessarily. If you configure too few logs, you can encounter a log-full condition. As you select the number of logs to configure, you must consider the size to make each log and whether your application can handle a log-full condition. The total log file size limit on active log space is 256 GB.

If you are enabling an existing database for rollforward recovery, change the number of primary logs to the sum of the number of primary and secondary logs, plus 1. Additional information is logged for LONG VARCHAR and LOB fields in a database enabled for rollforward recovery.

• Secondary Logs(logsecond): specifies the number of secondary log files that are created and used for recovery, if needed.

If the primary log files become full, secondary log files (of size logfilsiz) are allocated, one at a time as needed, up to the maximum number specified by this parameter. If this parameter is set to -1, the database is configured with infinite active log space. There is no limit on the size or number of in-flight transactions running on the database.

**NOTE:** If this parameter is set to -1, crash recovery time may be increased since DB2 may need to retrieve archived log files.

**Metrics**
None.

**High Watermarks**

- **Metrics**
- **Troubleshooting**

The high watermarks statistic shows the maximum total and secondary log space used by the database.

**Metrics**
The *total log space used high watermark* in conjunction with the total log space allocated, helps you determine if applications have come close to using the total log space available to your database.

The *total secondary log space used high watermark* in conjunction with total log space used high watermark helps you determine your current dependency on secondary logs.

**Troubleshooting**
High values of either high watermarks can indicate that you either need to increase the log space on the database or that the applications using the database need to perform more frequent COMMITS.

You can increase the overall log space available by adjusting one or more of the following database configuration parameters:

- logfilsiz
- logprimary
- logsecond
- logretain
**Space Detail View**

The following tabbed pages are available on the Space Detail view:

<table>
<thead>
<tr>
<th>Usage</th>
<th>Logging</th>
<th>Bufferpools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>Quiesce</td>
<td>Rebalance</td>
</tr>
<tr>
<td>Reorganization</td>
<td>Rollforward</td>
<td></td>
</tr>
</tbody>
</table>

**Usage**

The Space Detail Usage tab section shows you the list of tablespaces for the database. The following information is presented for each tablespace:

- **Tablespace Details**
- **Container Details**

**Tablespace Details**

- **Metrics**

**ID:** Internal DB2 Tablespace identifier.

**Name:** Tablespace Name

**Type:** Tablespace Type – SMS (System Managed Space) or DMS (Database Managed Space)

**Content Type:** Shows the type of data that can be stored on the tablespace. The valid types are:

- *Any* meaning can store any type of data
- *Sys Temp* meaning, used internally by DB2 to store intermediate result data
- *User Temp* meaning, can be used to create temporary tables
- *Long* meaning, can be used to store LOB/LONG/Index data

**State:** Can be one or more of the following:

- Normal (see the definition SQLB_NORMAL in squtil.h)
- Quiesced: SHARE
- Quiesced: UPDATE
- Quiesced: EXCLUSIVE
- Load pending
- Delete pending
- Backup pending
- Roll forward in progress
- Roll forward pending
- Restore pending
- Recovery pending (not used)
- Disable pending
• Reorg in progress
• Backup in progress
• Storage must be defined
• Restore in progress
• Offline and not accessible
• Drop pending
• Storage may be defined
• Storage Definition is in 'final' state
• Storage Definition was changed prior to rollforward
• DMS rebalancer is active
• TBS deletion in progress
• TBS creation in progress

**Page Size:** The size of tablespace pages

**Extent Size:** Number of pages that will be written to a container before writing to the next container on the tablespace

**Prefetch Size:** The maximum number of pages the prefetcher gets from the disk at a time

**Usable Space:** The total space in a table space minus overhead space. (Note: For SMS tablespaces, this element will have the same value as Used Space).

**Used Space:** The total space in the tablespace that is currently used in the tablespace (including overhead)

**Free Space:** The total space that is free to be used for storage of data. This element is not applicable for SMS tablespace. free space for an SMS tablespace is the OS file system free space for the SMS containers.

**Total Space:** The total space used by the tablespace

**Pending Free Pages:** The number of pages in the tablespace that would become free if all pending transactions are committed or rolled back and new space is requested for an object. (Applies only to DMS tablespaces)

**# Containers:** The number of containers in the tablespace

**Usable Pages:** The total pages in a table space minus overhead pages. (Note: For SMS tablespaces, this element will have the same value as Used Pages).

**Used Pages:** The total pages in the tablespace that is currently used in the tablespace (including overhead)

**Free Pages:** The total pages that are free to be used for storage of data. This element is not applicable for SMS tablespace. Free pages for an SMS tablespace are dependent on the OS file system free space for the SMS containers.

**Total Pages:** The total pages used by the tablespace

**Page HWM:** For DMS, this element represents the page number of the first free extent following the last allocated extent of a table space. Note that this is not really a "high water mark", but rather a "current water mark", since the value can decrease. For SMS, this is not applicable.

**Metrics**

None.
Container Details

This section shows the list of containers for the selected tablespace. The information presented here includes:

**ID:** A value that uniquely identifies the container within the tablespace

**Name:** The name of the container. Typically this is the full path of the file/directory/device where the container exists.

**Type:** The type of container. An SMS Container will be a directory. A DMS Containers will be a raw device/file/stripped disk/ or stripped file. Together with Container name, and partition, this metric identifies the physical location of the container.

**DB Partition:** The database partition where the container exists

**Usage Pages:** Usable pages in the container (Applicable to DMS Tablespaces only).

**Total Pages:** Total pages in the container

**Stripe Set:** The stripe set that a container belongs to.

**Accessible:** This element describes if a container is accessible or not

**Metrics**

This section can be used to get a better understanding of the physical layout of a tablespace. Also, it lets you quickly identify if any containers are inaccessible.

Logging

The Logging tab of the Space Detail view displays Logging Space statistics:

<table>
<thead>
<tr>
<th>Log Read/Writes</th>
<th>Indoubt Transactions</th>
<th>Log Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Secondary Logs</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Log Read/Writes**

This section shows the level of read/write activity being performed by the logger. The information presented here is:

**Log Reads:** The number of log pages read from the disk by the logger per second.

**Log Writes:** The number of log pages written to the disk by the logger.

**Metrics**

You can use this page with the operating system level I/O metrics, to quantify the amount of I/O on a device that is attributable to database activity.

**Indoubt Transactions**

This section shows the number of outstanding indoubt transactions in the database.
Metrics
Indoubt transactions hold log space for uncommitted transactions, which can cause the logs to become full. When the logs are full, further transactions cannot be completed. The resolution of this problem involves a manual process of heuristically resolving the indoubt transactions. This monitor element provides a count of the number of currently outstanding indoubt transactions that must be heuristically resolved.

Log Usage
- Metrics
- Troubleshooting
This section shows the following log usage details:

Total Used Space: Total space used by log files
Total Free Space: Total space still available to the logger
Total Used Space Highwatermark: The maximum amount of total log space used.
Secondary Space Used High Watermark: The maximum amount of secondary log space used.

Metrics
Use these metrics to check your current log space usage and the highest amount of space used by the logger. If the usage is high and the high watermarks are also near the total available space, you may need to increase the amount of log space or end/commit/rollback the oldest transaction.

Troubleshooting
You can use the following database configuration parameters to change the amount of log space available:

- logfilsiz
- logprimary
- logsecond

You can end (force) a connection with a long running transaction using the DBArtisan database monitor.

Secondary Logs
- Metrics
- Troubleshooting
This section shows the secondary logs allocations by the database over the monitoring period.

Metrics
When the primary log files become full, the secondary log files are allocated one at a time as needed, up to a maximum number as controlled by the logsecond database configuration parameter. Secondary log files are useful in preventing occasional log fill ups but they may not be as efficient as primary log and they also increase the recovery time. If you see a constant reliance on secondary log files, it may indicate infrequent commits by database applications or insufficient primary log space.
Troubleshooting
First examine the database applications to see if secondary log files are due to long running transactions and whether these transactions can be shortened. If the transactions cannot be shortened or you still see a very frequent use of secondary logs after transaction tuning, increase the primary log size (logprimary, logfilsiz Database configuration parameters) to reduce the dependence on secondary logs.

If there is an occasional long running transaction, and you see your see transaction log full (SQL0964C) error messages, you can either increase the number of secondary log files (logsecond database configuration parameter) or set the number to -1 (no limit on number of secondary log files).

Bufferpools
The Bufferpools tab of the Space Detail View shows the list of tablespaces, with the current bufferpool associated with each tablespace and also the bufferpool used by the database at next startup.

Objects
The Objects tab of the Space Detail View shows the following statistics:

- **Tablespace Object Summary**
- **Tablespace Object Details**

Tablespace Object Summary
This section shows information about how tablespaces are being used on the database. The information presented here includes:

- **Tablespace**: Name of the table space
- **Table Count**: Number of tables on the table space
- **Page Size**: Tablespace page size
- **Index Count**: Number of indexes on the table space
- **Table Space**: Space used by tables on the table space
- **Index Space**: Space used by indexes on the table space

Tablespace Object Details
This section shows the details of the objects on the table spaces shown in the Tablespace Object summary section. The user can toggle between the list of tables and indexes for the table space.

The information presented here for tables includes:

- **Table Schema**: Schema name of the table
- **Table Name**: Name of the tables
- **Avg Row Length**: Approximate length of the table row.
- **# Rows**: Table Cardinality
- **#Pages Allocated**: Number of pages allocated to the table
- **Table Size**: Total space used by the rows on the table
- **Available**: Total available space on the table
**Total:** Total space used by the table pages

**NOTE:** The information presented here is based on the table/index statistics and if the statistics haven’t been collected/updated, this information may be out of date.

**Quiesce**
The Quiesce tab of the Space Detail view shows the following information:

- **Quiesced Tablespaces**
- **Quiescer Details**

**Quiesced Tablespaces**

- **Metrics**
- **Troubleshooting**

This section shows a list of quiesced tablespaces. A quiesce is a persistent lock. Its benefit is that it persists across transaction failures, connection failures, and even across system failures (such as power failure, or reboot).

**Metrics**

A quiesce is owned by a connection. If the connection is lost, the quiesce remains, but it has no owner, and is called a phantom quiesce. For example, if a power outage caused a load operation to be interrupted during the delete phase, the tablespaces for the loaded table would be left in delete pending, quiesce exclusive state. With a database restart, this quiesce would be an unowned (or phantom) quiesce. The removal of a phantom quiesce requires a connection with the same user ID used when the quiesce mode was set. This section is useful in identifying the tablespaces that are in a quiesced state.

**Troubleshooting**

To remove a phantom quiesce:

- Connect to the database from DBArtisan with the same user ID used when the quiesce mode was set.
- Re-quiesce the table space using the current quiesce state from DBArtisan.
- Reset the quiesce from DBartisan.

**Quiescer Details**

- **Metrics**
- **Troubleshooting**

This section shows the list of quiescers for the selected quiesced table.

**Metrics**

Any quiescers whose agent ids are not listed under Users Detail >Attributes are phantom.

**Troubleshooting**

To remove a phantom quiesce:

- Connect to the database from DBArtisan with the same user ID used when the quiesce mode was set.
- Re-quiesce the table space using the current quiesce state from DBArtisan.
• Reset the quiesce from DBartisan.

Rebalance
The Rebalance tab of the Space Detail page displays the following information:

- Tablespace Details
- Rebalance Status

Tablespace Details
- Metrics

This section shows a list of tablespaces being rebalanced. The elements shown in this section are:

ID: Internal DB2 Tablespace identifier.

Name: Tablespace Name

Type: Tablespace Type (SMS or DMS)

Content Type: Shows the type of data that can be stored on the tablespace. The valid types are
  • Any meaning can store any type of data
  • Sys Temp meaning, used internally by DB2 to store intermediate result data
  • User Temp meaning, can be used to create temporary tables
  • Long meaning, can be used to store LOB/LONG/Index data

Extent Size: Number of pages that will be written to a container before writing to the next container on the tablespace

State: Tablespace State

Containers: Number of containers on the tablespace

Ranges: The number of ranges (entries) in the table space map

Rebalancer Mode: Indicates the whether the rebalancing is adding or removing space from the tablespace. Its values can be Forward(adding space) or Reverse (removing space)

Rebalancer Start Time: Timestamp of when the rebalancer started

Rebalancer Restart Time: A timestamp of when (if) the rebalancer was restarted after being paused or stopped.

Rebalancer Extents Remaining: The number of extents that haven’t been processed yet.

Rebalancer Extents Processed: The number of extents that the rebalancer has already moved since the rebalancer has been started or restarted (whichever is most recent).

Rebalancer Last Extent Moved: The last extent moved by the rebalancer

Rebalancer Priority: The priority at which the rebalancer is running in the database.

Metrics
Rebalancing involves moving table space extents from one location to another and it is done in an attempt to keep data striped within the table space. Rebalancing starts automatically when containers are added or extended in a tablespace.
The metrics in this section help you identify the status of current rebalance operations and when you can expect the rebalancing to complete.

Rebalance Status
This section shows a graph of the percentage of extents that have been processed and the percentage that remains incomplete for the selected tablespace.

Reorganization
The Reorganization tab of the Space Details View displays the following statistics:

<table>
<thead>
<tr>
<th>Table Details</th>
<th>Reorganization Status</th>
<th>Reorganization Attributes</th>
</tr>
</thead>
</table>

Table Details
This section shows the details of all the tables on which reorganization has been attempted since the start of monitoring. It also shows the status of tables that are presently undergoing reorganization. The information presented in this includes:

Table Schema: Table Schema of the table on which reorganization is being/was performed

Table Name: Table Name of the table on which reorganization is being/was performed

Tablespace: Tablespace of the table on which reorganization is being/was performed

Start: Start time of the reorganization

End: End Time of the reorganization

Phase Start: Start time of a phase of reorganization

Current Counter: Percentage of reorganization operation that has been completed (0 if the reorganization status is complete)

Max Phase: The maximum number of reorganization phases that will occur during reorganization processing. This applies to classic (offline) reorganizations only.

Status: The status of an in-place (online) table reorganization. This is not applicable to classic (offline) table reorganizations. An in-place table reorganization can be in one of the following states:

- Started/Resumed
- Paused
- Stopped
- Completed
- Truncate

Completion: Table reorganize success indicator.

Phase: Table reorganize phase. This applies to classic (offline) table reorganizations only.

For classic table reorganization, the following phases are possible:

- Sort
- Build
• Replace
• Index Recreate

% Complete: Shows the completion status of the reorganization jobs

Reorganization Status
Shows a graphical status view of the currently selected reorganization.

Reorganization Attributes
This section shows the reorganization attributes of the currently selected reorganization. The attributes include:

• Allow Write Access
• Allow Read Access
• Allow No Access
• Recluster Via Index Scan
• Reorg Long Field LOB Data
• No Table Truncation
• Reclustering
• Reclaiming
• Inplace Table Reorg
• Table Reorg
• Recluster Via Table Scan
• Reorg Data Only.

Rollforward
Recovering database changes can be a time consuming process. You can use the database system monitor to monitor the progression of a recovery. This section provides the following elements to help you monitor the status of rollforwards:

Timestamp: Timestamp of the log being processed. This is an indicator of the data changes that will be recovered

Tablespace: Tablespace being rolled forward. This element identifies the tablespaces involved in the rollforward

Type: An indicator of whether the recovery is happening at a database or tablespace level

Log#: The log being currently processed. This identifies the log involved in the rollforward.

Status: The status of rollforward recovery. This metric indicates if the recovery is in an undo (rollback) or redo (rollforward) phase.

#Tablespaces: Number of tablespaces involved in the rollforward

Node: Indicates the database partition number for the rollforward operation
Users Statistics - DB2
The database activity can be examined using key performance metrics and wait/bottleneck-based analysis.

The Users performance category view displays the vital DB2 user statistics in the following categories:

- Key User Analysis - Users
- SQL Analysis - Users
- Connection Analysis - Users

Key User Analysis - Users
Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the DB2 Snapshot Monitor. The snapshot monitor returns metrics for the Instance and particular databases since the start of monitoring. Some metrics are cumulative (e.g., counters) and others are instantaneous (e.g., gauges).

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst User page to succinctly communicate the general overall performance levels of the monitored database:

**Efficiency**

<table>
<thead>
<tr>
<th>Transactions/sec</th>
<th>Static SQL/sec</th>
<th>Dynamic SQL/sec</th>
</tr>
</thead>
</table>

**Activity**

<table>
<thead>
<tr>
<th>Applications Idle</th>
<th>Applications Executing</th>
<th>Connections High Watermark</th>
</tr>
</thead>
</table>

**Transactions/sec**

- **Metrics**
- **Troubleshooting**

Transactions/sec is the number of transactions (units of work) completed per second on the database.

**Metrics**

A small rate of transactional activity on the database can indicate that applications are not doing frequent commits, which may lead to logging and concurrency problems.
Troubleshooting
Drill down to the Users Detail>SQL Activity page to check which applications are running their transactions for long periods of time.

Static SQL/sec
The static SQL/sec metric is the number of static SQL statement executions attempt on the database each second.

Dynamic SQL/sec
The dynamic SQL/sec metric is the number of dynamic SQL statement executions being attempted on the database each second.

Applications Idle
- Metrics
The idle applications metric indicates the number of applications that are currently connected to the database for which the database manager is not executing any requests.

Metrics
You may use this element to help you understand the level of activity within a database and the amount of system resource being used.

Applications Executing
- Metrics
The executing applications statistic indicates the number of applications for which the database manager is currently executing requests.

Metrics
You can use this element to understand how many of the database manager agent tokens are being used by applications connected to this database.

Connections High Watermark
- Metrics
- Troubleshooting
The connections high watermark is the highest number of simultaneous connections to the database since the database was activated.

Metrics
You can use this element to evaluate the setting of the maxappls configuration parameter. If the value of this element is the same as the maxappls parameter, it is likely that some database connection requests were rejected, since maxappls limits the number of database connections allowed.

Troubleshooting
Increase the value of the maxappls parameter to allow more connections.
Bottleneck Analysis - Users
When a DB2 database is active, the applications accessing the database can either successfully access the database resources, or they can wait for resources to become available. An application that is waiting can indicate normal database activity, or it can be an indicator that a database bottleneck exists. This is where wait-based or bottleneck analysis comes into play. A database administrator can use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Performance Analyst Bottleneck analysis section helps a DBA isolate potential problem areas where there are waits, resource overloads or potential failures. If concurrency issues or heavy table scan activity has been dragging a database's performance down, a DBA can use bottleneck analysis to isolate the root cause. Once one or more bottlenecks have been isolated as possible areas of interest, the DBA can drill down and examine details on why the applications or resources are causing the problem. The Performance Analyst Users page identifies the top system and application bottlenecks that may be currently a cause of lessened performance.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

- Top Bottlenecked User Processes

Activity Bottlenecks

<table>
<thead>
<tr>
<th>Automatic Rebounds</th>
<th>Failed SQL Operations</th>
<th>Internal Rollbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort Overflows</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lock Bottlenecks

<table>
<thead>
<tr>
<th>Waits</th>
<th>Deadlocks</th>
<th>Escalations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeouts</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Top Bottlenecked User Processes
The top bottlenecked user processes grid shows the top bottlenecked applications in the system. These applications are either performing the top I/O operations or have idled the longest to obtain locks.

Automatic Rebounds
- Metrics

The automatic rebinds statistic is the number of automatic rebinds (or recompiles) that have been attempted.

Metrics
Automatic rebinds are the internal binds the system performs when an package has been invalidated. The rebind is performed the first time the database manager needs to execute an SQL statement from the package. For example, packages are invalidated when you:

- Drop an object, such as a table, view, or index, on which the plan depends
- Add or drop a foreign key
- Revoke object privileges on which the plan depends.

Since automatic rebinds can have a significant impact on performance, they should be minimized where possible.
Failed SQL Operations

- **Metrics**

The failed SQL operations statistic gives the number of attempted SQL statements that failed.

**Metrics**

This element may also help you in determine reasons for poor performance, since failed statements can mean time wasted by the database manager and, as a result, lower throughput for the database.

Internal Rollbacks

- **Metrics**

The internal rollbacks statistic is the total number of rollbacks initiated internally by the database manager.

**Metrics**

An internal rollback occurs when any of the following cannot complete successfully:

- A reorganization
- An import
- A bind or pre-compile
- An application ends as a result of a deadlock situation or lock timeout situation
- An application ends without executing an explicit commit or rollback statement (on Windows).

Sort Overflows

- **Metrics**

The sort overflows metric is the total number of sorts that ran out of sort heap and that may have required disk space for temporary storage.

  **NOTE:** This metric is also available on the Home page.

**Metrics**

When a sort overflows, additional overhead is incurred because the sort requires a merge phase. If data needs to be written to disk, it potentially requires more I/O.

**Troubleshooting**

Sort overflows can be reduced by increasing the sortheap database configuration parameter.

Waits

- **Metrics**

The lock waits metric is the total number of times that applications or connections waited for locks within the database.

**Metrics**

None.
Deadlocks

- **Metrics**
- **Troubleshooting**

The deadlocks statistic is the total number of deadlocks that have occurred since this instance of Performance Analyst started monitoring the database.

**Metrics**

If a large number of deadlocks are detected, it can indicate that applications are experiencing lock contention problems. Deadlocks are usually caused by one of the following situations:

- Lock escalations on the database
- Catalog tables locked for Repeatable Read
- Applications are using inappropriate isolation levels at bind time
- Applications are obtaining the same locks in a different order
- Applications are locking tables explicitly where row level locks are sufficient.

**Troubleshooting**

You may be able to modify the applications causing lock contentions for better concurrency.

To identify the applications that may be causing contentions, go to the Lock Waits tab of the Users detail page and review the Application Details section.

Escalations

- **Metrics**
- **Troubleshooting**

The lock escalations metric indicates the number of times that locks have been escalated from row locks to table locks, since this instance of Performance Analyst started monitoring the database.

**NOTE:** This metric also appears on the Home page.

**Metrics**

A lock is escalated when the total number of locks held by an application reaches the maximum amount it is allowed in the lock list memory. There are several possible causes of lock escalations:

- The database lock list size is too small for the concurrent workload
- The maximum percentage of lock list memory allowed for each application is too small
- One or more applications are obtaining an excessive number of locks

Monitor the lock escalations over a period of time to determine what levels are acceptable in your environment. If the escalations are excessive, or are accompanied by deadlocks or long lock waits, consider tuning the database.

**Troubleshooting**

Examine the locklist database configuration parameter together with the lock list utilization metric to see if the locklist size needs to be increased. If the lock list utilization is within acceptable range, try raising the maxlocks database configuration parameter to increase the percentage of lock list space available to each application.
You may also want to examine the detailed lock usage for each application on the Locks tab of the Users detail page to see if any application is holding an inordinately high number of locks. If this is the case, tune your application for better concurrency.

Timeouts

- **Metrics**
- **Troubleshooting**

The lock timeouts metric identifies the number of times that a request to lock an object timed out without being granted.

**Metrics**

If the number of lock timeouts becomes excessive when compared to the acceptable range for your database, it can indicate that an application is holding locks for long durations. It can also indicate that the amount of time an application waits for a lock before timing out is too short.

If you have too few lock timeouts and the average lock wait time is too high, it can indicate that the lock timeout configuration parameter is set to an excessively high value.

**Troubleshooting**

First you should examine the lock activity at the application level (Users>Lock, Lock Waits drilldown) to identify any particular application that is causing excessive lock contentions. If so, you can tune the application to provide better concurrency. If lock timeouts are excessive, and average lock wait times are very short, you can increase the `locktimeout` database configuration parameter to make the applications wait longer before timing out.

**SQL Analysis - Users**

Most of database's I/O performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag down an otherwise well-configured server in terms of user response times.

This section shows a snapshot of row level SQL operations that have been performed on the database since you started monitoring. This gives you an insight into the row level of read/write activity that is currently occurring within the database and comparing this to the prefetcher, and cleaner activity allows you to relate the two.

The following statistics are used on the Performance Analyst for DB2 User home page to succinctly communicate the general overall performance levels of I/O:

<table>
<thead>
<tr>
<th>Rows Deleted</th>
<th>Rows Inserted</th>
<th>Rows Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows Updated</td>
<td>Rows Read</td>
<td></td>
</tr>
</tbody>
</table>

**Rows Deleted**

- **Metrics**

The rows deleted metric is the total number of row deletions attempted on the database. It includes the internal row deletions that were caused by constraints or triggers.

**Metrics**

None.
Rows Inserted

- **Metrics**

The rows inserted metric is the total number of row insertions attempted on the database. This metric includes the internal row insertions that were caused by triggers.

**Metrics**
None.

Rows Selected

- **Metrics**

The rows selected statistic is the total number of rows selected at the database. This metric does not include a count of rows accesses for operations such as COUNT(*) or joins.

**Metrics**
None.

Rows Updated

- **Metrics**

The rows updated metric is the total number of row updates attempted on the database. This metric includes the internal row updates that were caused by triggers or constraints.

**Metrics**
None.

Rows Read

- **Metrics**

The rows read statistic is the number of rows read from the table. This count is not the number of rows that were returned to the calling application. Rather, it is the number of rows that had to be read in order to return the result set.

**Metrics**
None.

**Connection Analysis - Users**

The Connection Analysis section displays the level of application connect/disconnect activity occurring on the database.

The following statistics are used on the Performance Analyst for DB2 User home page to succinctly communicate general overall connectivity:

<table>
<thead>
<tr>
<th>Current Connections</th>
<th>Disconnects</th>
<th>Connections Since Database Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Watermark for Concurrent Connections</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Current Connections

- **Metrics**
- **Troubleshooting**

The current connections metric is the number of applications currently connected to the database.

**Metrics**

You can use this metric to help you get an overview of the level of database activity and the amount of system resources in use.

**Troubleshooting**

This metric can help you adjust the setting of the `maxappls` and `max_coordagents` configuration parameters. For example, if this value is always the same as `maxappls`, you consider increasing the value of `maxappls`.

Disconnects

- **Metrics**

The disconnects statistic indicates the number of disconnects from the database since the first connect, activate, or last reset.

**Metrics**

None.

Connections Since Database Activation

- **Metrics**
- **Troubleshooting**

The connections since database activation statistic indicates the number of connections to the database since the first connect, activate, or last reset.

**Metrics**

Use this metric in conjunction with the database connection time and the database start time to see if the application connection frequency is low. When there are no connections to db2, by default the database is automatically deactivated until the next connection. The connection that reactivates the database encounters slow connection time because it needs to wait for the database to reinitialize. If very few applications are connecting to the database, some connections can be slow because of this.

**Troubleshooting**

If there are short periods when your database has no applications connected to it, activate the database explicitly using the `ACTIVATE DATABASE` command. This significantly improves the connection performance for the database.

High Watermark for Concurrent Connections

- **Metrics**
- **Troubleshooting**

The connections high watermark is the highest number of simultaneous connections to the database since the database was activated.
Metrics
You can use this element to evaluate the setting of the `maxappls` configuration parameter. If the value of this element is the same as the `maxappls` parameter, it is likely that some database connection requests were rejected, since `maxappls` limits the number of database connections allowed.

Troubleshooting
Increase the value of the `maxappls` parameter to allow more connections.

Workload Analysis - Users
When your phone starts ringing with complaints of performance slowdowns, one of the first things you should get a handle on is:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Users page provides insight into the leading resource hogs of a server. Drill-downs are available so you can easily get detailed information into what each leading application is currently involved with.

The Workload Analysis section presents the following metrics:

<table>
<thead>
<tr>
<th>Top Lock Escalations</th>
<th>Top CPU Process</th>
<th>Oldest Transaction</th>
</tr>
</thead>
</table>

Top Lock Escalations

- **Metric**
  
  The top lock escalation displays the application with largest number of lock escalations.

  **Metric**
  None.

Top CPU Process

- **Metric**
  
  The top CPU process displays the application with the highest CPU usage.

  **Metric**
  None.

Top I/O Process

- **Metric**
  
  The top I/O process displays the application with the highest I/O read/write time.

  **Metric**
  None.
Oldest Transaction

- Metrics

The oldest transaction displays the application with the longest running current transaction.

Metric
None.

Users Detail View

The following tabbed pages are available on the Users Detail view:

<table>
<thead>
<tr>
<th>Top Sessions</th>
<th>Cursors</th>
<th>SQL Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Lock Waits</td>
<td>Locks</td>
</tr>
</tbody>
</table>

Top Sessions

Very often, one renegade client application brings a healthy database down by hogging most of the available resources. This page shows the top resource hogs in each of the resource usage areas to help you quickly identify the worst offenders in each area. You see:

<table>
<thead>
<tr>
<th>Top I/O Process</th>
<th>Top Memory Process</th>
<th>Top CPU Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Sessions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Top I/O Process

- Metrics

This section shows the application with the top IO time and compares it with the IO time of all other applications.

Metrics

The metrics are calculated by measuring the total time spent by each application doing reads and writes (buffered and non-buffered).

Top Memory Process

- Metrics

This section shows the application with the largest memory pool usage aggregate. It also shows the total memory pool usage for all other applications.

Metrics

This metric is calculated by aggregating all of the memory pools for each application and showing the top application. This section is useful in determining if the top application is using a lot more resources.

Top CPU Process

- Metrics
This section shows the application with the top CPU time usage and compares it with the CPU usage of all other applications.

**Metrics**
The metrics are calculated by measuring the total user and system CPU time for each application connected to the database.

**Top Sessions**
Shows a list of all applications connected to the database and the resource usage for these applications.

The information presented for each application includes:

- **Auth ID**: The authorization ID of the user who invoked the application that is being monitored
- **Name**: The name of the application running at the client as known to the database manager
- **Handle**: A system-wide unique ID for the application. On a single-partitioned database, this identifier consists of a 16 bit counter. On a multi-partitioned database, this identifier consists of the coordinating partition number concatenated with a 16 bit counter. In addition, this identifier will be the same on every partition where the application may make a secondary connection.
- **ID**: This identifier is generated when the application connects to the database at the database manager
- **Client PID**: The process ID of the client application that made the connection to the database
- **I/O Time**: The total time spent by application in performing buffered and non-buffered reads and writes
- **Memory Usage**: Total memory pool usage for the application
- **CPU Time**: The total user + system cpu time used by the application agents
- **# Agents**: Number of agents for the application
- **Status**: Current Status of the application. Values for this field are:
  - **Database Connect Pending**: The application has initiated a database connection but the request has not yet completed.
  - **Database Connect Completed**: The application has initiated a database connection and the request has completed.
  - **Unit of Work Executing**: The database manager is executing requests on behalf of the unit of work.
  - **Unit of Work Waiting**: The database manager is waiting on behalf of the unit of work in the application. This status typically means that the system is executing in the application's code.
  - **Lock Wait**: The unit of work is waiting for a lock. After the lock is granted, the status is restored to its previous value.
  - **Commit Active**: The unit of work is committing its database changes.
  - **Rollback Active**: The unit of work is rolling back its database changes.
  - **Recompiling**: The database manager is recompiling (that is, rebinding) a plan on behalf of the application.
  - **Compiling**: The database manager is compiling an SQL statement or precompiling a plan on behalf of the application.
  - **Request Interrupted**: An interrupt of a request is in progress.
• **Database Disconnect Pending**: The application has initiated a database disconnect but the command has not yet completed executing. The application may not have explicitly executed the database disconnect command. The database manager will disconnect from a database if the application ends without disconnecting.

• **Decoupled from Agent**: The application has been decoupled from an agent.

• **Transaction Prepared**: The unit of work is part of a global transaction that has entered the prepared phase of the two-phase commit protocol.

• **Transaction Heuristically Committed**: The unit of work is part of a global transaction that has been heuristically committed.

• **Transaction Heuristically Rolled Back**: The unit of work is part of a global transaction that has been heuristically rolled-back.

• **Transaction Ended**: The unit of work is part of a global transaction that has ended but has not yet entered the prepared phase of the two-phase commit protocol.

• **Creating Database**: The agent has initiated a request to create a database and that request has not yet completed.

• **Restarting Database**: The application is restarting a database in order to perform crash recovery.

• **Restoring Database**: The application is restoring a backup image to the database.

• **Backing Up Database**: The application is performing a backup of the database.

• **Data Fast Load**: The application is performing a "fast load" of data into the database.

• **Data Fast Unload**: The application is performing a "fast unload" of data from the database.

• **Wait to Disable Tablespace**: The application has detected an I/O error and is attempting to disable a particular table space. The application has to wait for all other active transactions on the table space to complete before it can disable the table space.

• **Quiescing a Tablespace**: The application is performing a quiesce table space request.

• **Wait for Remote Partition**: The application is waiting for a response from a remote partition in a partitioned database instance.

• **Remote Request Pending**: The application is waiting for results from a federated data source.

**Last UOW Start Time**: The date and time that the unit of work first required database resource

**Last UOW End Time**: The date and time that the most recent unit of work completed, which occurs when database changes are committed or rolled back

**Current UOW Elapsed Time**: The elapsed execution time of the most recently completed unit of work.

**Login Timestamp**: The date and time that a connection request was granted.

**Inbound Comm Address**: This is the communication address of the client.

**Client Protocol**: The communication protocol that the client application is using to communicate with the server.

**Cursors**

- **Metrics**
- **Troubleshooting**

The Cursors tab of the Users Detail view displays the Application SQL Cursor Activity. It details the SQL cursors open for each application. The information presented here includes:
Open Cursors: The number of cursors (local and remote) currently open for an application.

Blocking Cursors: The number of blocking cursors (local and remote) currently open for the application.

Block Requests Accepted: The percentage of all blocking I/O requests that were accepted by the database.

Block Requests Rejected: The percentage of all blocking I/O requests that were rejected by the database.

Metrics
The number of blocking cursors for an application should be close to the number of open cursors. If this is not the case, the application performance may be affected.

If there are many cursors blocking data, the communication heap may become full. When this happens, instead of returning an error, the database stops allocating I/O blocks for blocking cursors.

Troubleshooting
If the number of blocking cursors in an application is not close to the number of open cursors, you may be able to improve application performance by tweaking the precompile options for record blocking for the treatment of ambiguous cursors, and by redefining the cursors with FOR FETCH ONLY clause where possible.

If a large number of cursors were unable to perform blocking, you may be able to improve performance by increasing the query_heap database manager configuration parameter.

SQL Activity
The SQL Activity tab of the Users Detail view displays the following statistics:

<table>
<thead>
<tr>
<th>Application List</th>
<th>SQL Executed</th>
<th>SQL Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOW Details</td>
<td>Binds and Precompiles Executed</td>
<td></td>
</tr>
</tbody>
</table>

Application List
This section shows a list of applications connected to the database. You can select an application from this list and see details for the SQL activity for this application.

SQL Executed
- Metrics

This section shows the details for the various types of SQL statements executed by the particular application. It also gives you the total application throughput. The information presented here includes:

Selects: Percentage of statements that are select statements.

Updates/Inserts/Deletes: Percentage of statements that modify/write data on the database.

DDLs: Percentage of CREATE/DROP/ALTER statements.

Others: All other types of statements (Fetch, Open cursor etc).

Metrics
Various types of statements have varying impact on the performance and concurrency of the database. For example - DDL statements can cause catalog and package cache invalidations, Insert/update/delete statements usually obtain locks and perform log IO.

This section can help you determine the contributing statement types to the activity for an application.
SQL Activity

• Metrics

This section gives details on the sql activity for the selected application. It allows you to quick compare the number of static dynamic and failed sql statements with the overall application throughput. The metrics presented here include:

Static SQL: The number of static SQL statements that the application tried to execute
Dynamic SQL: The number of dynamic SQL statements that the application tried to execute
Failed SQL: The number of SQL statements (static and dynamic) that the application tried to execute but failed
SQL Throughput: Number of SQL statements (dynamic and static) that the application executed successfully

Metrics
These elements may help you determine the nature of the application. Failed SQL statements metric may also help identify the cause of poor performance on the database since failed statements mean time wasted by the database manager.

UOW Details

• Metrics
• Troubleshooting

Shows the units of work being completed by the application over the refresh time interval.

Metrics
A low rate of unit of work completion may be an indicator of one of the following:

• The application is not committing transactions frequently and may be causing concurrency issues for other applications
• The application is idling and not performing any work on the server

Troubleshooting
Examine the locks obtained by the application (Users Detail > Locks page) to see if a low number here should be a cause for concern.

Binds and Precompiles Executed

• Metrics

This section shows the total number of implicit and explicit binds and precompiles that were executed by the application.

Implicit rebinds are performed by the system when the application tries to execute a sql statement from a package and the package has been invalidated. A package is usually invalidated when you perform DDL/DCL operations on objects on which the package plan is dependent.

Explicit rebinds occur when the user executes the REBIND PACKAGE command.

Metrics
Bind and precompile operations have significant overhead for the database operations and they should be minimized during periods of heavy usage when possible.
Attributes
The Attributes tab of the Users Detail page displays the following statistics:

<table>
<thead>
<tr>
<th>Application List</th>
<th>Application Attributes</th>
<th>Direct Authorities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indirect Authorities</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Application List
This section shows the list of applications connected to the database. The user can obtain further details for an application from other sections on this page by selecting the application from this list.

Application Attributes
Shows the main information for the selected application. This information is useful in determining the applications connection attributes and its state. The information shown here includes:

**Application Status:** The status of the application. Values for this field are:

- **Database Connect Pending:** The application has initiated a database connection but the request has not yet completed.
- **Database Connect Completed:** The application has initiated a database connection and the request has completed.
- **Unit of Work Executing:** The database manager is executing requests on behalf of the unit of work.
- **Unit of Work Waiting:** The database manager is waiting on behalf of the unit of work in the application. This status typically means that the system is executing in the application’s code.
- **Lock Wait:** The unit of work is waiting for a lock. After the lock is granted, the status is restored to its previous value.
- **Commit Active:** The unit of work is committing its database changes.
- **Rollback Active:** The unit of work is rolling back its database changes.
- **Recompiling:** The database manager is recompiling (that is, rebinding) a plan on behalf of the application.
- **Compiling:** The database manager is compiling an SQL statement or precompiling a plan on behalf of the application.
- **Request Interrupted:** An interrupt of a request is in progress.
- **Database Disconnect Pending:** The application has initiated a database disconnect but the command has not yet completed executing. The application may not have explicitly executed the database disconnect command. The database manager will disconnect from a database if the application ends without disconnecting.
- **Decoupled from Agent:** The application has been decoupled from an agent.
- **Transaction Prepared:** The unit of work is part of a global transaction that has entered the prepared phase of the two-phase commit protocol.
- **Transaction Heuristically Committed:** The unit of work is part of a global transaction that has been heuristically committed.
- **Transaction Heuristically Rolled Back:** The unit of work is part of a global transaction that has been heuristically rolled-back.
- **Transaction Ended:** The unit of work is part of a global transaction that has ended but has not yet entered the prepared phase of the two-phase commit protocol.
• **Creating Database**: The agent has initiated a request to create a database and that request has not yet completed.

• **Restarting Database**: The application is restarting a database in order to perform crash recovery.

• **Restoring Database**: The application is restoring a backup image to the database.

• **Backing Up Database**: The application is performing a backup of the database.

• **Data Fast Load**: The application is performing a “fast load” of data into the database.

• **Data Fast Unload**: The application is performing a “fast unload” of data from the database.

• **Wait to Disable Tablespace**: The application has detected an I/O error and is attempting to disable a particular table space. The application has to wait for all other active transactions on the table space to complete before it can disable the table space.

• **Quiescing a Tablespace**: The application is performing a quiesce table space request.

• **Wait for Remote Partition**: The application is waiting for a response from a remote partition in a partitioned database instance.

• **Remote Request Pending**: The application is waiting for results from a federated data source.

**ID of code page used by the application**: The code page identifier.

**Application status change time**: The date and time the application entered its current status.

**Sequence Number**: This identifier is incremented whenever a unit of work ends (that is, when a COMMIT or ROLLBACK terminates a unit of work). Together, the application id and sequence number uniquely identify a transaction.

**Client Product/Version ID**: The product and version of DB2 that is running on the client.

**Database Alias used by the Application**: The alias of the database provided by the application to connect to the database.

**User login id**: The ID that the user specified when logging in to the operating system. This ID is distinct from auth_id, which the user specifies when connecting to the database.

**DRDA correlation token**: The DRDA correlation token is used for correlating the processing between the application server and the application requester. It is an identifier dumped into logs when errors arise, that you can use to identify the conversation that is in error. In some cases, it will be the LUWID of the conversation.

**Client operating platform**: The operating system on which the client application is running.

**Database territory code**: The territory code of the database for which the monitor data is collected.

**Application agent priority**: The priority of the agents working for this application.

**Application priority type**: Operating system priority type for the agent working on behalf of the application.

**Coordinating node**: In a multinode system, the node number of the node where the application connected or attached to the instance.

**Connection request start timestamp**: The date and time that an application started a connection request.

**Connection request completion timestamp**: The date and time that a connection request was granted.

**Previous unit of work completion timestamp**: This is the time the unit of work completed.

**Unit of work start timestamp**: The date and time that the unit of work first required database resources.

**Unit of work stop timestamp**: The date and time that the most recent unit of work completed, which occurs when database changes are committed or rolled back.
Most recent unit of work elapsed time: The elapsed execution time of the most recently completed unit of work.

Unit of work completion status: The status of the unit of work and how it stopped.

Application idle time: Number of seconds since an application has issued any requests to the server. This includes applications that have not terminated a transaction, for example not issued a commit or rollback.

Direct Authorities
Shows the highest level of direct authority granted to the application

Indirect Authorities
Shows the highest level of indirect authority granted to the application

Lock Waits
The Lock Waits tab of the Users Detail view shows the details for all applications in the database that are waiting on locks. This information is useful in tracking concurrency issues in the database. The statistics available here are:

<table>
<thead>
<tr>
<th>Application Details</th>
<th>Average Lock Wait Time</th>
<th>Applications Waiting on Locks</th>
</tr>
</thead>
</table>

Application Details
- **Metrics**
- **Troubleshooting**

This section shows the list of applications waiting for locks to be released on objects they need to access. The information presented here includes:

**Auth ID**: The authorization ID of the user who invoked the application that is being monitored.

**Name**: Name of the application executable

**Handle**: A systemwide unique ID for the application

**ID**: This identifier is generated when the application connects to the database at the database manager

**Client PID**: The process ID of the client application that made the connection to the database

**Total Lock Waits**: The total number of times that applications or connections waited for locks

**Total Lock Waiting Time**: This is the total amount of elapsed time that this connection or transaction has waited for a lock to be granted to it.

**Locks Waiting**: Indicates the number of agents waiting on a lock

**UOW Lock Waiting Time**: The total amount of elapsed time current unit of work has spent waiting for locks

**Wait Start Time**: The date and time that this application started waiting to obtain a lock on the object that is currently locked by another application

**Lock App Handle**: System wide Unique ID of the application holding the lock

**Lock App ID**: ID of the application holding the lock

**Lock Sequence ID**: The sequence number of the application that is holding a lock on the object that this application is waiting to obtain.
Metrics
AuthID, Name, Handle, ID, and Client PID help you identify the application waiting on the lock. Lock App Handle, Lock App ID, and Lock Sequence number help you identify the application that is holding the waited on lock. You can use this information on Users Detail Locks tab to get further details on the application holding the locks and the locks it’s holding.

Troubleshooting
If an application is not performing well, and it has a high number of total lock waits, and a high total lock waiting time you may need to improve concurrency of the applications that are holding the locks.

Average Lock Wait Time

• Metrics
• Troubleshooting

This is the average time, applications waited for locks in the database.

Metrics
This metric helps you determine if the applications are spending a large amount of time waiting to obtain locks.

Troubleshooting
If the average lock wait time is high, you should look for applications that hold many locks or have lock escalations. If appropriate, you may need to tune such applications for better concurrency. If escalations are the reason for high average lock wait time, the values of one or both of the locklist and maxlocks database configuration parameters may be too low.

Applications Waiting on Locks

• Metrics
• Troubleshooting

This metric indicates the percentage of all connected applications waiting on locks.

Metrics
If this number is high, the application may have concurrency problems.

Troubleshooting
You can identify the applications that are holding locks or exclusive locks for a long time from the Users Detail> Locks page and tune such applications for better concurrency.

Locks
The Locks tab of the Users Detail view gives you information for the following statistics:
Application Details
This section shows a list of applications connected to the database. You can look at the lock details for an application in the list by selecting the application. The attributes presented for each application include:

**Handle**: A system-wide unique ID for the application

**ID**: This identifier is generated when the application connects to the database at the database manager

**Seq No**

**Name**: Name of the application executable

**Auth ID**: The authorization ID of the user who invoked the application that is being monitored.

**Status**

**Code Page**

**Locks Held**

**Total Wait Time**: UOW Lock Waiting Time

Locks Held Details
This section gives the details of all the locks that the selected application is currently holding. The details presented here include:

**Lock Name**: Internal binary lock name. This element serves as a unique identifier for locks

**Lock Attributes**: Gives the lock attribute settings

**Release Flags**: The lock release flags can either be *untracked* or *SQL Compiler*

**Lock Count**: The number of locks on the lock being held. This value ranges from 1 to 255. It is incremented as new locks are acquired, and decremented as locks are released. When lock_count has a value of 255, this indicates that a transaction duration lock is being held. At this point, lock_count is no longer incremented or decremented when locks are acquired or released. The lock_count element is set to a value of 255 in one of two possible ways:

1. lock_count is incremented 255 times due to new locks being acquired.
2. A transaction duration lock is explicitly acquired. For example, with a LOCK TABLE statement, or an INSERT.

**Hold Count**: The number of holds placed on the lock. Holds are placed on locks by cursors registered with the WITH HOLD clause and some DB2 utilities. Locks with holds are not released when transactions are committed.

**Object Name**: It is the name of the object for table-level locks is the file ID (FID) for SMS and DMS table spaces. For row-level locks, the object name is the row ID (RID). For table space locks, the object name is blank. For buffer pool locks, the object name is the name of the buffer pool.

**Object Type**: The type of object against which the application holds a lock (for object-lock-level information), or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information).

**Tablespace**: This is the name of the table space against which the lock is held.

**Table Schema**: Schema of the table that the lock is on

**Table Name**: Name of the table that the lock is on. This element is only set if **Object Type** indicates Table

**Lock Mode**: The type of lock being held.
Locks Waiting Details
This section gives the details of all the locks that the selected application is currently waiting on. The details include:

**Subsection**: Identifies the subsection associated with the waited on lock

**Agent ID**: The application handle of the agent holding a lock for which this application is waiting

**App ID**: The application ID of the application that is holding a lock on the object that this application is waiting to obtain.

**Lock Name**: Internal binary lock name. This element serves as a unique identifier for locks.

**Lock Attributes**: Gives the lock attribute settings

**Release Flags**: Lock release flags

**Object Type**: The type of object against which the application holds a lock

**Lock Mode**: The type of lock being held

**Lock Mode Requested**: The lock mode being requested by the application

**Tablespace**: This is the name of the table space against which the lock is held.

**Table Schema**: Schema of the table that the lock is on

**Table Name**: Name of the table that the lock is on. This element is only set if **Object Type** indicates Table

**Wait Start Timestamp**: The date and time that this application started waiting to obtain a lock on the object that is currently locked by another application

**Escalation**: Indicates whether a lock request was made as part of a lock escalation.

Database Configuration Statistics (Cfg)
The Database Configuration view displays the following tabbed pages:

<table>
<thead>
<tr>
<th>Configuration</th>
<th>History</th>
<th>Indoubt Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Configuration**
This tab lets you view the current values for the DB2 database configuration parameters.

**History**
The History tab of the Database Configuration Details viewlets you browse the history file entries for the database. The history file contains a record of recovery and administrative events. Recovery events include full database and table space level backup, incremental backup, restore, and rollforward operations. Additional logged events include create, alter, drop, or rename table space, reorganize table, drop table, and load.

On this tab, you’ll find the following information:

<table>
<thead>
<tr>
<th>History Details</th>
<th>SQLCA Structure</th>
<th>Tablespaces List</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDL View</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
History Details
This section allows you to select the operation for which you want to see the history details. You can also narrow the history list to operations since a certain date or operations involving a certain object name. You can show the history for the following types of operations:

- Backups
- Dropped Tables
- Load
- Quiesce
- Reorganizations
- Rollforward
- Tablespace Alters
- Tablespace Renames
- Tablespace Creates
- Tablespace Drops

You can see further detail on some operations in the lower sections, by selecting that operation from the history detail operation list.

SQLCA Structure
Shows the SQLCA structure for the selected operation from the history detail. This information is not available for Quiesce, Rollforward, and Reorganization operations.

Tablespaces List
Shows the list of tablespaces involved in the selected operation in history detail.

DDL View
Shows the DDL for the selected operation in the history detail.

Indoubt Transactions
The Indoubt Transactions tab provides a list of transactions that are in doubt. The Indoubt Transaction Detail metrics displayed for each indoubt transaction include:

**Application ID:** Specifies the application identifier assigned by the database manager for this transaction

**Sequence Number:** Specifies the sequence number assigned by the database manager as an extension to the application ID

**Status:** Indicates the status of this indoubt transaction. Valid values are:

- SQLXA_TS_PREP
  - The transaction is prepared. The connected parameter can be used to determine whether the transaction is waiting for the second phase of normal commit processing or whether an error occurred and resynchronization with the transaction manager is required.

- SQLXA_TS_HCOM
• The transaction has been heuristically committed.
• SQLXA_TS_HROL
• The transaction has been heuristically rolled back.
• SQLXA_TS_MACK
• The transaction is missing commit acknowledgement from a node in a partitioned database.
• SQLXA_TS_END
• The transaction has ended at this database. This transaction may be re-activated, committed, or rolled back at a later time. It is also possible that the transaction manager encountered an error and the transaction will not be completed. If this is the case, this transaction requires heuristic actions, because it may be holding locks and preventing other applications from accessing data.

Timestamp: Specifies the time when the transaction entered the indoubt state.

Auth ID: Specifies the authorization ID of the user who ran the transaction.

Log Full: Indicates whether or not this transaction caused a log full condition

Originator

XID: Specifies the XA identifier assigned by the transaction manager to uniquely identify a global transaction

Objects
Tables and indexes consume the storage in all databases. The Database Object Detail grid displays object space details for the database selected in the Database Object Summary grid.

The following statistics are presented for tables:

Table Schema: The owner of the object

Table Name: The name of the object

File Group: File group where the object resides

Reserved (KB): The amount of space (in KB) reserved by the object

Used (KB): The amount of space (in KB) used by the object

Free (KB): The amount of free space (in KB) used by the object

Percent Used: The percentage of space used by the object

Instance Statistics - DB2
The Embarcadero Performance Analyst Instance view lets you review availability and overall performance of all monitored databases from a single window. Statistics on the Instance home view are organized into the following categories:

<table>
<thead>
<tr>
<th>Agent Analysis</th>
<th>Bottleneck Analysis</th>
<th>FCM Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Information</td>
<td>Key Ratio Analysis</td>
<td></td>
</tr>
</tbody>
</table>

Related Topic
• Instance Detail View
Agent Analysis
The Agent Analysis section presents the following metrics:

<table>
<thead>
<tr>
<th>Connections</th>
<th>Connections Executing</th>
<th>Agents Registered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agents Waiting</td>
<td>Idle Agents</td>
<td></td>
</tr>
</tbody>
</table>

Connections

- **Metrics**
- **Troubleshooting**

The connections statistic gives the total Number of connections to databases in the current instance.

**Metrics**
Shows the number of connections from remote and local clients to databases in this instance. This value will change frequently, so you may need to sample it at specific intervals over an extended period of time to get a realistic view of system usage.

**Troubleshooting**
This element can help you adjust the setting of the *max_coordagents* configuration parameter.

Connections Executing

- **Metrics**
- **Troubleshooting**

The connections executed statistic is the number if applications that are currently connected to a database and are currently processing a unit of work within the Database Manager instance being monitored.

**Metrics**
The connections executing metric can help you determine the level of concurrent processing occurring on the Database Manager. This value can change frequently, so you may need to sample it at specific intervals over an extended period of time to get a realistic view of system usage

**Troubleshooting**
This element can help you adjust the setting of the *maxcagents* configuration parameter.

Agents Registered

- **Metrics**
- **Troubleshooting**

The agents registered metric is the number of agents registered in the Database Manager instance that is being monitored (coordinator agents and subagents).

**Metrics**
None.
Troubleshooting
You can use this element to help evaluate your setting for the `maxagents` configuration parameter.

Agents Waiting
- **Metrics**
- **Troubleshooting**

The number of agents waiting for a token so they can execute a transaction in the database manager.

**Metrics**
Each application has a dedicated coordinator agent to process database requests within the Database Manager. Furthermore, each agent has to get a token before it can execute a transaction. The maximum number of agents that can execute Database Manager transactions is limited by the configuration parameter `maxcagents`.

Troubleshooting
You can use this element to help evaluate your setting for the `maxcagents` configuration parameter.

Idle Agents
- **Metrics**
- **Troubleshooting**

The idle agents metric is the number of agents in the agent pool that are currently unassigned to an application and are, therefore, “idle.”

**Metrics**
Having idle agents available to service requests for agents can improve performance.

Troubleshooting
You can use this element to help set the `num_poolagents` configuration parameter.

Bottleneck Analysis
The Bottleneck Analysis section presents the following metrics:

**Sorts and Joins**

<table>
<thead>
<tr>
<th>Post Threshold Sorts</th>
<th>Piped Sorts Rejects</th>
<th>Hash Join Thresholds</th>
</tr>
</thead>
</table>

**Agent Bottlenecks**

<table>
<thead>
<tr>
<th>Agents Waiting on a Token</th>
<th>Agents Created Due to Empty Pool</th>
</tr>
</thead>
</table>

**Post Threshold Sorts**
- **Metrics**
The post threshold sorts is the number of sorts that have requested heaps after the sort heap threshold has been exceeded.

Metrics
Under normal conditions, the Database Manager allocates sort heap using the value specified by the `sortheap` configuration parameter. If the amount of memory allocated to sort heaps exceeds the sort heap threshold (`sheapthres` configuration parameter), the database manager allocates sort heap using a value less than that specified by the `sortheap` configuration parameter.

Each active sort on the system allocates memory, which may result in sorting taking up too much of the system memory available. Sorts that start after the sort heap threshold has been reached may not receive an optimum amount of memory to execute. As a result, however, the entire system may benefit.

Troubleshooting
By modifying the sort heap threshold and sort heap size configuration parameters, the performance of sort operations and/or the overall system can be improved. If this element's value is high, you can:

- Increase the sort heap threshold (`sheapthres`) or,
- Adjust applications to use fewer or smaller sorts via SQL query changes.

Piped Sorts Rejects

The piped sorts rejects statistic is the number of piped sorts that were rejected by the database manager.

Metrics
Each active sort on the system allocates memory, which may result in sorting taking up too much of the available system memory. A piped sort is not accepted if the sort heap threshold is exceeded when the sort heap is allocated for the sort.

The sort list heap (`sortheap`) and sort heap threshold (`sheapthres`) configuration parameters help control the amount of memory used for sort operations. These parameters are also used to determine whether a sort will be piped.

Since piped sorts may reduce disk I/O, allowing more piped sorts can improve the performance of sort operations and possibly the performance of the overall system.

Troubleshooting
If piped sorts are being rejected, you might consider decreasing your sort heap or increasing your sort heap threshold. You should be aware of the possible implications of either of these options. If you increase the sort heap threshold there is the possibility that more memory will remain allocated for sorting. This could cause paging memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort.

Hash Join Thresholds

The hash join thresholds metric is the total number of times that a hash join heap request was limited because of concurrent use of shared or private sort heap.
Metrics

Each active sort on the system allocates memory, which may result in sorting taking up too much of the available system memory. A piped sort is not accepted if the sort heap threshold is exceeded when the sort heap is allocated for the sort.

The sort list heap (sortheap) and sort heap threshold (sheapthres) configuration parameters help control the amount of memory used for sort operations. These parameters are also used to determine whether a sort will be piped.

Since piped sorts may reduce disk I/O, allowing more piped sorts can improve the performance of sort operations and possibly the performance of the overall system.

Troubleshooting

If this value is large (greater than 5% of hash join overflows), the sort heap threshold should be increased.

Agents Waiting on a Token

- Metrics
- Troubleshooting

The agents waiting statistic is the percentage of agents registered that are waiting for a token so they can execute a transaction in the Database Manager.

Metrics

Each application has a dedicated coordinator agent to process database requests within the Database Manager. Each agent has to get a token before it can execute a transaction.

The maximum number of agents that can execute database manager transactions is limited by the configuration parameter maxcagents. This parameter is used to control the load on the system during periods of high simultaneous application activity. For example, you may have a system requiring a large number of connections but with a limited amount of memory to serve those connections. Adjusting this parameter can be useful in such an environment, where a period of high simultaneous activity could cause excessive operating system paging.

This parameter does not limit the number of applications that can have connections to a database. It only limits the number of Database Manager agents that can be processed concurrently by the Database Manager at any one time, thereby limiting the usage of system resources during times of peak processing.

Troubleshooting

In cases where the high concurrency of applications is causing problems, you can use benchmark testing to tune this parameter to optimize the performance of the database.

Agents Created Due to Empty Pool

- Metrics

The agents created due to empty pool metric shows the number of agents created because the agent pool was empty. It includes the number of agents started when the current instance started (num_initagents).

Metrics

None.
FCM Analysis

The fast communication manager (FCM) provides communication support for Enterprise Server Editions. Each database partition server has one FCM thread to provide communications between database partition servers to handle agent requests and to deliver message buffers. The FCM thread starts when you start the instance.

The following ratios are used on the FCM Analysis section to succinctly communicate the general overall FCM performance levels of the monitored instance:

<table>
<thead>
<tr>
<th>Buffer Throughput</th>
<th>Buffer Utilization</th>
<th>Connection Entry Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Anchor Utilization</td>
<td>Request Block Utilization</td>
<td></td>
</tr>
</tbody>
</table>

Buffer Throughput

- **Metrics**
- **Troubleshooting**

The buffer throughput metric is the total number of buffers that have been sent and received by this database partition since the start of monitoring.

**Metrics**

This gives the level of traffic to and from the selected partition.

**Troubleshooting**

If this traffic level is high, you should consider redistributing one or more databases on the instance or move tables to reduce internode traffic.

Buffer Utilization

- **Metrics**
- **Troubleshooting**

This element indicates the percentage of all FCM buffers that are currently being used by the fast communication manager.

**Metrics**

None.

**Troubleshooting**

You can use this information to tune `fcm_num_anchors`. If the utilization percentage is high, you should increase the `fcm_num_anchors` to ensure that operations do not run out of FCM message anchors. If the utilization is low, you can decrease `fcm_num_anchors` to conserve system resources.

Connection Entry Utilization

- **Troubleshooting**

The connection entry utilization element indicates the percentage of all connection entries that are currently being used by the fast communication manager.

**NOTE:** This metric is only applicable to DB2 version 7. In DB2 version 8, the maximum connection entries are adjusted dynamically and automatically.
Troubleshooting
You can use this information to tune `fcm_num_connect`. If the utilization percentage is high, you should increase the `fcm_num_connect` to ensure that operations do not run out of FCM connection entries. If the utilization is low, you can decrease `fcm_num_connect` to conserve system resources.

Message Anchor Utilization

- **Metrics**
- **Troubleshooting**

This element indicates the percentage of all message anchors that are currently being used by the fast communication manager.

**NOTE:** This metric is only applicable to DB2 version 7. In DB2 version 8, the maximum message anchors are adjusted dynamically and automatically.

**Metrics**
None.

**Troubleshooting**
You can use this information to tune `fcm_num_buffers`. If the utilization percentage is high, you should increase the `fcm_num_buffers` to ensure that operations do not run out of FCM buffers. If the utilization is low, you can decrease `fcm_num_buffers` to conserve system resources.

Request Block Utilization

- **Troubleshooting**

This element indicates the percentage of all request blocks that are currently being used by the fast communication manager.

**NOTE:** This metric is only applicable to DB2 version 7. In DB2 version 8, the maximum request blocks are adjusted dynamically and automatically.

**Troubleshooting**
You can use this information to tune `fcm_num_rqb`. If the utilization percentage is high, you should increase the `fcm_num_rqb` to ensure that operations do not run out of FCM request blocks. If the utilization is low, you can decrease `fcm_num_rqb` to conserve system resources.

Instance Information
Here you see the following information:

- Product Name
- FixPack Level
- Instance Type
- Operating System
- Instance Name
- Number of Partitions
Active Local Databases

Key Ratio Analysis
The Key Ratio Analysis section presents the following metrics:

Efficiency

<table>
<thead>
<tr>
<th>Agents Registered</th>
<th>Agents Waiting</th>
<th>Monitor Heap</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCMBP</td>
<td>Private Sort Memory</td>
<td></td>
</tr>
</tbody>
</table>

Agents Registered

- **Metrics**
- **Troubleshooting**

The agents registered metric shows the percentage of total allowable agents (coordinator agents and subagents) that are currently working in the instance.

Metrics
When this value is close to 100% it means that the number of agents currently working is close to the hard limit set in the database manager configuration. If this value stays high for long periods of time it means that there is high level of activity occurring in the databases on the instance.

Troubleshooting
Use this element to help you evaluate your setting for the `maxagents` database manager configuration parameter. When you increase the value of this parameter, make sure that your system has enough memory available to accommodate the additional agents that may be created.

Agents Waiting

- **Metrics**
- **Troubleshooting**

The agents waiting statistic is the percentage of agents registered that are waiting for a token so they can execute a transaction in the Database Manager.

Metrics
Each application has a dedicated coordinator agent to process database requests within the Database Manager. Each agent has to get a token before it can execute a transaction.

The maximum number of agents that can execute database manager transactions is limited by the configuration parameter `maxcagents`. This parameter is used to control the load on the system during periods of high simultaneous application activity. For example, you may have a system requiring a large number of connections but with a limited amount of memory to serve those connections. Adjusting this parameter can be useful in such an environment, where a period of high simultaneous activity could cause excessive operating system paging.

This parameter does not limit the number of applications that can have connections to a database. It only limits the number of Database Manager agents that can be processed concurrently by the Database Manager at any one time, thereby limiting the usage of system resources during times of peak processing.
Troubleshooting
In cases where the high concurrency of applications is causing problems, you can use benchmark testing to tune this parameter to optimize the performance of the database.

Monitor Heap
• Metrics
• Troubleshooting

The monitor heap statistic is the percentage of total monitor heap that is currently being used by the Database Manager.

Metrics
The amount of memory required for monitoring activity depends on the number of monitoring applications (applications taking snapshots or event monitors), which switches are set, and the level of database activity.

Memory is allocated from the monitor heap when you perform database monitoring activities such as taking a snapshot, turning on a monitor switch, resetting a monitor, or activating an event monitor.

If this metric is running at over 70% utilization and you have many applications performing snapshot/event monitoring, you may need to increase the size of the monitor heap.

Troubleshooting
Increase the value of `mon_heap_sz` database manager configuration parameter to increase the available monitor heap size.

FCMBP
• Metrics

The fast communication manager bufferpool (FCMBP) is the percentage of total amount of the FCMBP that is currently being used. This metric is only applicable to multipartitioned database environments.

NOTE: FCM buffers are used for internal communication both among and within database servers. The FCMBU metric gives the percentage utilization of the available fcm buffers in the Database Manager.

Metrics
You can use this information to tune the `fcm_num_buffers` database manager configuration parameter.

Private Sort Memory
• Metrics

The private sort memory metric is the percentage of the total private sort memory heap that is currently in use.

Metrics
You can use this information to tune the `sheapthres` database manager configuration parameter. If the value of this element is close to 100%, it means that the sorts for some databases are not getting the full sort heap as defined by the `sortheap` database configuration parameter.
Instance Detail View
The following tabbed pages are available from the Instance Detail View:

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Agents &amp; Connections</th>
<th>Utilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCM Resources</td>
<td>FCM Throughput</td>
<td>Memory Pools</td>
</tr>
<tr>
<td>Sorts &amp; Joins</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Configuration
The Configuration tab of the Instance Detail view lets you view and change the instance configuration parameter. It also allows you to look at some information about the instance such as Product Name, Type, Fixpak Level, Operating System, instance name, partitions, and number of active databases.

Agents & Connections
The Agents & Connections tab of the Instance Detail view gives you the following analyses:

<table>
<thead>
<tr>
<th>Connection Analysis</th>
<th>Agent Pool Analysis</th>
<th>Agent Analysis</th>
</tr>
</thead>
</table>

Connection Analysis

- **Metrics**

This section presents the metrics for the current local and remote connections to databases in the instance. The information presented here includes:

**Local Connections**: The number of local applications that are currently connected to a database within the database manager instance being monitored.

**Remote Connections**: The current number of connections initiated from remote clients to the instance of the database manager that is being monitored.

**Local Connections Executing**: The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.

**Remote Connections Executing**: The number of remote applications that are currently connected to a database and are currently processing a unit of work within the database manager instance being monitored.

**Metrics**
These metrics can help you determine the level of concurrent processing occurring in the database manager. These metrics will change frequently so it is important to obtain a sampling over an extended period.

These metrics are useful when tuning the `max_coordagents`, and `maxcagents` database manager configuration parameters.

Agent Pool Analysis

- **Metrics**

- **Troubleshooting**

The metrics in this section help determine how often the database agents are created due to empty pool and how often they are assigned from the agent pool.
Metrics
If the number of agents created due to empty pool is high compared to the agents assigned from pool, it may mean that the number of pool agents should be increased. It may also mean that the overall workload is too high.

Troubleshooting
You can change the maxcagents and num_poolagents database manager configuration parameters to tune these metrics.

Agent Analysis
The metrics presented here give a detailed view of how the database manager agents are working. The metrics presented here include:

Registered Agents: The number of agents registered in the database manager instance that is being monitored (coordinator agents and subagents).

Agents Waiting for Token: The number of agents waiting for a token so they can execute a transaction in the database manager.

Idle Agents: The number of agents in the agent pool that are currently unassigned to an application and are, therefore, "idle".

Stolen Agents: The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.

Maximum Agents Registered: The maximum number of agents that the database manager has ever registered, at the same time, since it was started (coordinator agents and subagents).

Maximum Agents Waiting: The maximum number of agents that have ever been waiting for a token, at the same time, since the database manager was started.

Maximum Coordinating Agents: The maximum number of coordinating agents working at one time.

Utilities
The Utilities tab of the Instance Detail View gives information on the utilities currently executing in the database manager (available in DB2 version 8 only). The Utility Execution Details are as follows:

Utility ID: The unique identifier corresponding to the utility invocation

Type: Class of the utility (Rebalance, Backup, Restore, Reorg etc.)

Utility Description: A brief description of the work the utility is performing

Utility Start Time: The time the utility was invoked

DBName: Database on which the utility is operating.

FCM Resources
The FCM Resources tab of the Instances Detail View presents statistics on the following:

<table>
<thead>
<tr>
<th>Node Details</th>
<th>FCM Buffer Utilization</th>
<th>FCM Message Anchor Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCM Connection Entries Utilization</td>
<td>FCM Request Block Utilization</td>
<td></td>
</tr>
</tbody>
</table>
Node Details
This section gives a detailed view of inter-node communication in a multipartition environment. The information shown here includes the connection status and buffers sent and received between various combinations of partitions in a multi-partitioned environment. The columns presented here include:

Source Node: Partition that sends the information
Target Node: Partition that received the information
Buffers Sent: Number of Buffers sent from source node to the target node
Buffers Received: Number of Buffers sent from source node to the target node

Connection Status

FCM Buffer Utilization

Metrics
This section provides details on the current utilization and low watermark of FCM buffers.

Troubleshooting
If you notice a very small value for the low watermark and similar values for free FCM buffers over a long period, you may need to increase the `fcm_num_buffers` database manager configuration parameter to increase the number of available FCM buffers.

If on the other hand you notice a very high value for the low watermark – you may have allocated too many buffers and you can reduce the value of `fcm_num_buffers` database manager configuration parameter.

FCM Message Anchor Utilization

Metrics
This section provides details on the current utilization and low watermark of FCM message anchors.

Troubleshooting
If you notice a very small value for the low watermark and similar values for free message anchors over a long period, you may need to increase the `fcm_num_anchors` database manager configuration parameter to increase the number of available message anchors.

If on the other hand you notice a very high value for the low watermark – you may have allocated too many message anchors and you can reduce the value of `fcm_num_anchors` database manager configuration parameter.

NOTE: DB2 version 8 automatically adjusts `fcm_num_anchors` parameter, so you do not need to adjust this manually in DB2 version 8.
FCM Connection Entries Utilization

- Metrics
- Troubleshooting

This section provides details on the current utilization and low watermark of FCM connection entries.

**Metrics**

You can monitor the connection entry utilization for the selected partition together with the low watermark to determine the usage trend of connection entries.

**Troubleshooting**

If you notice a very small value for the low watermark and similar values for free connection entries over a long period, you may need to increase the `fcm_num_connect` database manager configuration parameter to increase the number of available connection entries.

If on the other hand you notice a very high value for the low watermark – you may have allocated too many connection entries and you can reduce the value of `fcm_num_connect` database manager configuration parameter.

**NOTE:** DB2 version 8 automatically adjusts `fcm_num_rqb` parameter, so you do not need to adjust this manually in DB2 version 8.

FCM Request Block Utilization

- Metrics
- Troubleshooting

This section provides details on the current utilization and low watermark of FCM request blocks.

**Metrics**

You can monitor the request block utilization for the selected partition together with the low watermark to determine the usage trend of the request blocks.

**Troubleshooting**

If you notice a very small value for the low watermark and similar values for free request blocks over a long period, you may need to increase the `fcm_num_rqb` database manager configuration parameter to increase the number of available request blocks.

If on the other hand you notice a very high value for the low watermark – you may have allocated too many request blocks and you can reduce the value of `fcm_num_rqb` database manager configuration parameter.

**NOTE:** DB2 version 8 automatically adjusts `fcm_num_rqb` parameter, so you do not need to adjust this manually in DB2 version 8.

FCM Throughput

The FCM Throughput tab of the Instance Detail View includes the following statistics:

- Node Details
- FCM Throughput
Node Details
This section gives a detailed view of inter node communication in a multipartition environment. The information shown here includes the connection status and buffers sent and received between various combinations of partitions in a multi-partitioned environment. The columns presented here include:

**Source Node**: Partition that sends the information
**Target Node**: Partition that received the information
**Buffers Sent**: Number of Buffers sent from source node to the target node
**Buffers Received**: Number of Buffers sent from source node to the target node

Connection Status

FCM Throughput

- **Metrics**

This section shows the delta statistics for the FCM buffers sent and received between the selected nodes in the Node Details section

**Metrics**
If the total number of FCM buffers sent or received between selected nodes is high, you may want to redistribute the database, or move tables to reduce the inter-node traffic.

Memory Pools
The Memory Pools tab of the Instance Detail View gives information on the following statistics:

<table>
<thead>
<tr>
<th>Memory Pool List</th>
<th>Memory Pool Details</th>
<th>Memory Pool Utilization</th>
</tr>
</thead>
</table>

Memory Pool List
This section shows the memory pool usage details of the selected memory pool over time.

Sorts & Joins
The Sorts & Joins tab of the Instance Detail View presents the following information:

<table>
<thead>
<tr>
<th>Sort Heap Details</th>
<th>Sort Threshold Details</th>
<th>Hash Join Threshold Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piped Sort Details</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sort Heap Details

- **Metrics**
- **Troubleshooting**

This section shows the metrics for the total number of allocated pages of sort heap space for all sorts. It represents the sum of sort heap space allocated for all sorts in all active databases in the database manager.
Metrics
Normal memory estimates do not include sort heap space. If excessive sorting is occurring, the extra memory used for the sort heap should be added to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Appropriate use of indexes can reduce the amount of sorting required.

Troubleshooting
You may use the information in this section to help you tune the `sheapthres` configuration parameter. If the element value is greater than or equal to `sheapthres`, it means that the sorts are not getting the full sort heap as defined by the `sorheap` parameter.

Sort Threshold Details
- **Metrics**
- **Troubleshooting**

This section details the number of sorts that have requested heaps after the sort heap threshold has been exceeded.

**Metrics**
Under normal conditions, the database manager will allocate sort heap using the value specified by the `sorheap` configuration parameter. If the amount of memory allocated to sort heaps exceeds the sort heap threshold (`sheapthres` configuration parameter), the database manager will allocate sort heap using a value less than that specified by the `sorheap` configuration parameter.

Each active sort on the system allocates memory, which may result in sorting taking up too much of the system memory available.Sorts that start after the sort heap threshold has been reached may not receive an optimum amount of memory to execute, but, as a result, the entire system may benefit. By modifying the sort heap threshold and sort heap size configuration parameters, the performance of sort operations and/or the overall system can be improved.

**Troubleshooting**
If this element's value is high, you can:

- Increase the sort heap threshold (`sheapthres`) or,
- Adjust applications to use fewer or smaller sorts via SQL query changes.

Hash Join Threshold Details
- **Troubleshooting**

This section shows the total number of times that a hash join heap request was limited due to concurrent use of shared or private sort heap space.

**Troubleshooting**
If this value is large (greater than 5% of hash join overflows), the sort heap threshold should be increased.

Piped Sort Details
- **Metrics**
- **Troubleshooting**

This section gives the details on the piped sorts that have been requested and accepted.
Metrics
Each active sort on the system allocates memory, which may result in sorting taking up too much of the available system memory.

When the number of accepted piped sorts is low compared to the number requested, you can improve sort performance by adjusting one or both of the following configuration parameters:

- `sortheap`
- `sheapthres`

Troubleshooting
If piped sorts are being rejected, you might consider decreasing your sort heap or increasing your sort heap threshold. You should be aware of the possible implications of either of these options. If you increase the sort heap threshold, then there is the possibility that more memory will remain allocated for sorting. This could cause the paging of memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort.

OS Page Statistics
In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Analyst lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX
- Linux
- Solaris
- Unix
- Windows XP and 2000

**NOTE:** The statistics available on the OS page depend on the operating system platform.

TIP: If you magnify a graph to change back to the original size or minimize the size, close the OS Page and then reopen it again. Performance Analyst displays the graph in its original form and size.

The OS home page includes the following sections:

<table>
<thead>
<tr>
<th>Bottleneck Analysis</th>
<th>Disk Analysis</th>
<th>Key Resource Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Analysis</td>
<td>Workload Analysis - Top Resource Hogs</td>
<td></td>
</tr>
</tbody>
</table>

Related Topic
OS Detail
Key Resource Usage - OS
The following ratios are used on the Performance Analyst OS home page to communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Disk Time</th>
<th>Load Average</th>
<th>Processor Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paged Memory Used (Windows)</td>
<td>Swap Memory Used (AIX, HP-UX, Linux, Solaris, Unix)</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the CPU tab, right-click Processor Time. To open the I/O tab, right-click Disk Time. To open the Memory tab, right-click Paged Memory Used.

Bottleneck Analysis - OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Average Disk Queue Length</th>
<th>Interrupts/Sec</th>
<th>Network Output Queue Length (Windows)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Queue Length (Solaris)</td>
<td>Page Faults/Sec</td>
<td>Processor Queue Length</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>Processor</td>
<td></td>
</tr>
</tbody>
</table>

**TIP:** To open the I/O tab, right-click any Details menu item. To open the CPU tab, right-click the Item Processor Queues Length Details menu item. To open the Network tab, right-click Network Output Queues Length.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Memory Analysis - OS
The following metrics are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Available Paged Memory (Windows)</th>
<th>Available Physical Memory</th>
<th>Available Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Paged Memory (Windows)</td>
<td>Total Physical Memory</td>
<td>Total Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)</td>
</tr>
</tbody>
</table>

**TIP:** To open the Memory tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.
Disk Analysis - OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Free Disk Space</th>
<th>Total Disk Space</th>
<th>Used Disk Space</th>
</tr>
</thead>
</table>

**TIP:** To open the Space tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Workload Analysis - Top Resource Hogs OS
The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

<table>
<thead>
<tr>
<th>Number of Logins</th>
<th>Number of Processes</th>
<th>Number of Processors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top CPU Process</td>
<td>Top I/O Process</td>
<td>Top Memory Process</td>
</tr>
</tbody>
</table>

**TIP:** To open the Processes tab, right-click any Details menu item.

**NOTE:** The statistics available in this section depend on the platform of operating system.

Processor Time
The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

**Metrics**
If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.

**Troubleshooting**
Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

Processor Speed
The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Processor
The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Disk Time
The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.
Metrics
You should avoid consistently seeing values for this statistic greater than 90%.

Troubleshooting
Add more disk drives and partition the files among all of the drives.

Load Average
The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Metrics
High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Paged Memory Used
The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

**NOTE:** This statistic is available for the Windows platform.

Metrics
This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.

Troubleshooting
Increase the size of page file.

Swap Memory Used
The Swap Memory Used statistic is the percentage of swap space currently in use.

Metrics
If the percentage of swap memory used is above 90%, you may be running out of memory.

Troubleshooting
Increase the size of your swap files.

Average Disk Queue Length
The Average Disk Queue Length statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.
**Metrics**

This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.

**Troubleshooting**

Some things you can do if you have problems with this statistic include:

- Redistribute the data on the disk with the large average disk queue to other disks.
- Upgrade to faster disk(s).

**Interrupts/Sec**

Interrupts/Sec is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every 10 milliseconds, creating a background of interrupt activity. This statistic shows the difference between the values observed in the last two samples, divided by the duration of the sample interval.

**Metrics**

A high value indicates possible excessive hardware interrupts; justification is dependent on device activity.

**Page Faults/Sec**

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

**Metrics**

This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

**Troubleshooting**

If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

**Processor Queue Length**

The Processor Queue Length statistic is the number of threads in the processor queue.

**Metrics**

Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Troubleshooting
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Network Output Queue Length/Network Queue Length
The Network Output Queue Length statistic is the number of threads in the processor queue.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics
Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Available Physical Memory
The Available Physical Memory statistic represents the amount of RAM available to all processes.

Metrics
This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Troubleshooting
Some things you can do if you have problems with this statistic include:

- Check the running processes to see if there are any memory leaks.
- Stop any services that are not required.
- Install additional RAM.

Available Paged Memory
The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

NOTE: This statistic is available for the Windows platform.

Metrics
If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.
Troubleshooting
Increase the size of page file.

Available Swap Memory
The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Metrics
If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.

Troubleshooting
Increase the size of swap files.

Total Physical Memory
The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Metrics
This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Total Paged Memory/Total Swap Memory
The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics
It is recommended that this be 1½ - 3 times the amount of RAM on the system.

Used Disk Space
The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

• If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.

• You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.
Total Disk Space
Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two:

1. If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
2. You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Free Disk Space
The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Troubleshooting
There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

• If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
• You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Top Memory Process
Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics
If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Processes Overview
The Processes Overview of the OS Summary includes the following sections:

<table>
<thead>
<tr>
<th>Top CPU Process</th>
<th>Top I/O Process</th>
<th>Top Memory Process</th>
</tr>
</thead>
</table>

Top CPU Process
Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.
Metrics
If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be
the bottleneck on the server.

Troubleshooting
Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU
speed for the process. You may need to upgrade your CPU.

Top I/O Process
The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information
displayed is dependent on the platform of the operating system. Information displayed includes the name of the
process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the
amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Number of Logins
This statistic displays the total number of logins on the server.

Number of Processes
This statistic displays the total number of processes on the server.

OS Detail
The following tabbed pages are available on the OS Detail page:

<table>
<thead>
<tr>
<th>CPU</th>
<th>I/O</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Processes</td>
<td>Space</td>
</tr>
</tbody>
</table>

CPU Tab
The CPU tab of the OS Detail includes the following sections:

<table>
<thead>
<tr>
<th>Context Switches/Sec</th>
<th>CPU Events</th>
<th>CPU Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrupts/Sec</td>
<td>Load Averages</td>
<td>Processor Queue Length</td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of operating system.

CPU Utilization
The CPU Utilization section includes the following information:

- % Privileged Time
- % User Time
% Privileged Time

The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

**NOTE:** For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

**Metrics**

The ideal range should be 0-40% (less than 40% indicates excessive system activity).

**Troubleshooting**

If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

% User Time

The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

**Metrics**

If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

CPU Events

The CPU Events section includes the following information:

<table>
<thead>
<tr>
<th>Interrupts/Sec</th>
<th>Context Switches/Sec</th>
<th>System Calls/Sec</th>
</tr>
</thead>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

**Interrupts/Sec**

The Interrupts/Sec statistic is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. It does not include deferred procedure calls (DPCs), which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every ten milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

**Metrics**

The ideal range should be 0-5000. A number greater then 5000 indicates possible excessive hardware interrupts; justification is dependent on device activity.
Context Switches/Sec
The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Metrics
The ideal range should be between 0-10,000. GA number greater than 10,000 may indicate too many threads contending for resources.

System Calls/Sec
System calls are functions which require the services of the operating system. Usually they involve access to some data that users must not be permitted to corrupt or even change.

Troubleshooting
Try to find a good ratio between the amount of time the operating system runs (fulfilling system calls and doing process scheduling), and the amount of time the application runs. Your goal should be running 60% to 75% of the time in application mode, and 25% to 40% of the time in operating system mode. If you find that the system is spending 50% of its time in each mode, then you should investigate to determine what is wrong.

Processor Queue Length
The Processor Queue Length statistic is the number of threads in the processor queue. There is a single queue for processor time even on computers with multiple processors.

NOTE: For Windows systems, unlike the disk counters, this counter shows ready threads only, not threads that are running.

Metrics
A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Load Averages
The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Metrics
High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Processes Tab
The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>The name of the process.</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>User</td>
<td>The user of the process.</td>
</tr>
<tr>
<td>ID</td>
<td>The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.</td>
</tr>
<tr>
<td>CPU</td>
<td>The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.</td>
</tr>
<tr>
<td>User Mode</td>
<td>The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.</td>
</tr>
<tr>
<td>Memory (MB)</td>
<td>Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.</td>
</tr>
<tr>
<td>Memory (WINDOWS ONLY)</td>
<td>Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.</td>
</tr>
<tr>
<td>Memory</td>
<td>Memory is the percentage of the memory used of the total memory.</td>
</tr>
<tr>
<td>Active Memory</td>
<td>Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.</td>
</tr>
<tr>
<td>I/O Data</td>
<td>The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>The total elapsed time, in seconds, that this process has been running.</td>
</tr>
<tr>
<td>Thread Count</td>
<td>The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.</td>
</tr>
<tr>
<td>Handle Count</td>
<td>The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.</td>
</tr>
<tr>
<td>Priority</td>
<td>The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process’ base priority.</td>
</tr>
<tr>
<td>Creating Proc ID</td>
<td>The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.</td>
</tr>
<tr>
<td>Page Faults/Sec</td>
<td>Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.</td>
</tr>
<tr>
<td>Page File</td>
<td>Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.</td>
</tr>
<tr>
<td>Private</td>
<td>Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.</td>
</tr>
</tbody>
</table>
I/O Tab

The I/O tab includes the following sections:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk</td>
<td>The disk number assignment.</td>
</tr>
<tr>
<td>Reading (KB/s)</td>
<td>The amount of bytes read from the device.</td>
</tr>
<tr>
<td>Writing (KB/s)</td>
<td>The amount of bytes written to the device.</td>
</tr>
<tr>
<td>Disk Read Time</td>
<td>Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.</td>
</tr>
<tr>
<td>Disk Write Time</td>
<td>Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.</td>
</tr>
<tr>
<td>Disk Time</td>
<td>Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.</td>
</tr>
<tr>
<td>Avg. Read Queue</td>
<td>Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.</td>
</tr>
<tr>
<td>Avg. Write Queue</td>
<td>Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.</td>
</tr>
<tr>
<td>Disk Reads/Sec</td>
<td>Disk Reads/Sec is the rate of read operations on the disk.</td>
</tr>
<tr>
<td>Disk Writes/Sec</td>
<td>Disk Writes/Sec is the rate of write operations on the disk.</td>
</tr>
</tbody>
</table>

**NOTE:** The columns available in this table depend on the platform of operating system.

I/O Details

The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading</td>
<td>The average number of bytes transferred from the disk during read operations.</td>
<td>It is measured in number of KBs per second.</td>
</tr>
<tr>
<td>Writing</td>
<td>The average number of bytes transferred to the disk during write operations.</td>
<td>It is measured in number of KBs per second.</td>
</tr>
<tr>
<td>Transferred</td>
<td>The amount of time required to transfer the data between the device's cache and the actual device media.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.
Bytes per I/O Operation
The Bytes per I/O Operation section of the OS Detail succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading</td>
<td>The average number of bytes transferred from the disk during read operations.</td>
<td>It is measured in number of bytes per second.</td>
</tr>
<tr>
<td>Writing</td>
<td>The average number of bytes transferred to the disk during write operations.</td>
<td>It is measured in number of bytes per second.</td>
</tr>
</tbody>
</table>

Average Disk Queue/Total Disk Queue
The Average Disk Queue section of the OS Detail page succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Read Queue</td>
<td>Average Read Queue Length is the average number of read requests that were queued for a particular disk.</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system.</td>
</tr>
<tr>
<td>Average Write Queue</td>
<td>Average Write Queue is the average number of write requests that were queued for a particular disk.</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Windows System Administrator for assistance in troubleshooting these problems.</td>
</tr>
<tr>
<td>Average Queue</td>
<td>Average time in ms. to service I/O requests which for disks includes seek, rotational latency and data transfer time).</td>
<td>Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Unix System Administrator for assistance in troubleshooting these problems.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Troubleshooting
Contact your Windows System Administrator for assistance in troubleshooting these problems.
Disk I/O Time
The Disk I/O Time section of the OS Detail page succinctly communicates the general overall performance levels of the disk's I/O. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Disk Read Time</td>
<td>% Disk Read Time is the percentage of time during the sample interval that the disk is busy servicing read requests.</td>
<td>Use this in conjunction with the % Disk Write Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
<tr>
<td>% Disk Write Time</td>
<td>% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.</td>
<td>Use this in conjunction with the % Disk Read Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
<tr>
<td>% Disk Time</td>
<td>% Disk Time is the percentage of time during the sample interval that the disk is busy servicing requests.</td>
<td>Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Disk Transfers/Sec
The Disk Transfers/Sec section of the OS Detail page succinctly communicates the general overall performance levels of the disk transfers. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads/Sec</td>
<td>The rate of read operations on the disk.</td>
</tr>
<tr>
<td>Writes/Sec</td>
<td>The rate of write operations on the disk.</td>
</tr>
<tr>
<td>Transfers/Sec</td>
<td>The rate of transfer operations per second.</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

Memory Tab
The Memory tab of the OS Detail page includes the following sections:

<table>
<thead>
<tr>
<th>Buffer Size</th>
<th>Cache Efficiency</th>
<th>Cache Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Available</td>
<td>Paging Activity</td>
<td>Page Faults</td>
</tr>
<tr>
<td>Page Replacements</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** The sections available on this tab depend on the platform of operating system.
Paging Activity
The Paging Activity section includes the following statistics:

<table>
<thead>
<tr>
<th>Blocks Input/Sec</th>
<th>Blocks Output/Sec</th>
<th>Paged In</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paged Out</td>
<td>Pages Input/Sec</td>
<td>Pages Output/Sec</td>
</tr>
</tbody>
</table>

**NOTE:** The statistics available in this section depend on the platform of operating system.

**Blocks Input/Sec**
The Blocks Input/sec statistic is the number of blocks sent to a block device per second.

**Troubleshooting**
Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck:

1. Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.

2. Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.

**Blocks Output/Sec**
The Blocks Output/sec statistic is the number of blocks received from a block device per second.

**Troubleshooting**
Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck:

1. Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.

2. Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.

**Paged In**
The number of disk block paged into core from disk which occurs when the server is experiencing a shortage of RAM. Any nonzero value indicates excessive swap activity. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

**Troubleshooting**
High values indicate the possible need to increase the amount of RAM on the system.
Paged Out
The number of disk block paged out of core from disk. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Pages Input/Sec
The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Metrics
This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Pages Output/Sec
The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Metrics
Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.
Page Faults/Sec
The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics
This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting
If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Memory Available
The Memory Available section of the OS Detail page succinctly communicates the general overall performance levels of the server's memory. The statistics available in this section depend on the platform of operating system. The table below describes these statistics:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
<th>Metrics</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>The Physical Memory statistic is the amount of committed virtual memory, in</td>
<td>This counter displays the last observed value only and not an average.</td>
<td>N/A</td>
</tr>
<tr>
<td>Memory</td>
<td>megabytes. Committed memory is where memory space has been reserved in the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>paging file if it needs to be written to disk.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td>The Physical statistic is the total physical memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Memory</td>
<td>The Paged Memory statistic is the amount of committed virtual memory, in</td>
<td>This counter displays the last observed value only; it is not an average.</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>bytes. Used Memory is the physical memory which has space reserved on the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>disk paging file(s). There can be one or more paging files on each physical</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>drive.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Swap Memory</td>
<td>The Swap Memory statistic is the free, unreserved swap space.</td>
<td>N/A</td>
<td>Increase the size of your swap file or add additional files if you consistently see a shortage of free, unreserved swap space.</td>
</tr>
<tr>
<td>Swap</td>
<td>The Swap statistic is the total swap memory being used.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total Physical</td>
<td>The Total Physical statistic is the total physical memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total Swap</td>
<td>The Total Swap statistic is the total swap memory available.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Swap-ins</td>
<td>The number of memory pages paged in from swapspace.</td>
<td>N/A</td>
<td>If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.</td>
</tr>
</tbody>
</table>
**NOTE:** The statistics available in this section depend on the platform of operating system.

### Buffer Size
The Buffer Size statistic represents the amount of cache used by the kernel in kilobytes.

### Cache Size
The Cache Size statistic represents the amount of cache used by the kernel in kilobytes.

### Page Replacements
The following statistics are available in this section:

- **Memory Freed (Pages/sec)**
- **Clock Algorithm Scans (Pages/sec)**

### Memory Freed (Pages/sec)
Freed or destroyed (Kb/s).

### Clock Algorithm Scans (Pages/sec)
The VMM uses a technique known as the clock algorithm to select pages to be replaced. This technique takes advantage of a referenced bit for each page as an indication of what pages have been recently used (referenced). When the page-stealer routine is called, it cycles through the PFT, examining each page's referenced bit.
Cache Efficiency
The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

<table>
<thead>
<tr>
<th>Copy Read Hits %</th>
<th>Data Map Hits %</th>
<th>MDL Read Hits %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin Read Hits %</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copy Read Hits %
The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

**Metrics**
When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

**Troubleshooting**
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Data Map Hits %
The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

**Metrics**
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

**Troubleshooting**
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %
The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

**Metrics**
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

**Troubleshooting**
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %
The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.
Metrics
Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting
Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab
The Space tab of the OS Detail page includes the following sections:

- Device Summary
- Device Detail

Disk Space Free
The Disk Space Free metric displays the amount of free disk space in megabytes.

Metric
None.

Disk Space Detail
The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server’s disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>The drive letter of the disk.</td>
</tr>
<tr>
<td>Local Filesystem</td>
<td>The name of the file system.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of file system.</td>
</tr>
<tr>
<td>Total Space</td>
<td>Total size of the disk/device's capacity expressed in MBs.</td>
</tr>
<tr>
<td>Used Space</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free Space</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
<tr>
<td>Capacity</td>
<td>The percentage of space used on the device.</td>
</tr>
<tr>
<td>Mounted On</td>
<td>The mount point of the device.</td>
</tr>
</tbody>
</table>

Device Summary
The Device Summary section of the Space tab displays a bar chart for all devices. The Device Summary section succinctly communicates the general overall performance levels of the server’s disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
</tbody>
</table>
Device Detail
The Device Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>The drive letter of the disk.</td>
</tr>
<tr>
<td>Local Filesystem</td>
<td>The name of the file system.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of file system.</td>
</tr>
<tr>
<td>Total Space</td>
<td>Total size of the disk/device's capacity expressed in MBs.</td>
</tr>
<tr>
<td>Used Space</td>
<td>Amount of MBs currently allocated on the particular disk/device.</td>
</tr>
<tr>
<td>Free Space</td>
<td>Amount of MBs currently unallocated and free on the particular disk/device.</td>
</tr>
<tr>
<td>Capacity</td>
<td>The percentage of space used on the device.</td>
</tr>
<tr>
<td>Mounted On</td>
<td>The mount point of the device.</td>
</tr>
</tbody>
</table>

NOTE: The statistics available in this section depend on the platform of operating system.

Network Tab
The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The Network tab of the OS Detail page includes the following sections:

<table>
<thead>
<tr>
<th>Network Details</th>
<th>Packet Collisions</th>
<th>Packet Discards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Errors</td>
<td>Transmission Rate</td>
<td>Transmission Rate (Bytes)</td>
</tr>
<tr>
<td>Transmission Queue Length</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE: The sections available on this tab depend on the platform of operating system.

Network Details
The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Interface</td>
<td>The name of network interface.</td>
</tr>
<tr>
<td>INET Address/Address</td>
<td>The IP address assigned to the network interface.</td>
</tr>
<tr>
<td>Pkts Sent/Sec</td>
<td>The number of packets sent per second.</td>
</tr>
<tr>
<td>Pkts Received/Sec</td>
<td>The number of packets received per second.</td>
</tr>
<tr>
<td>Sent (KB/Sec)</td>
<td>The number of bytes sent per second.</td>
</tr>
<tr>
<td>Received (KB/Sec)</td>
<td>The number of bytes received per second.</td>
</tr>
<tr>
<td>Out Pkts Discarded</td>
<td>The number of outbound packets discarded.</td>
</tr>
<tr>
<td>In Pkts Discarded</td>
<td>The number of inbound packets discarded.</td>
</tr>
<tr>
<td>Out Pkt Errors</td>
<td>The number of outbound packet errors.</td>
</tr>
<tr>
<td>In Pkt Errors</td>
<td>The number of inbound packet errors.</td>
</tr>
</tbody>
</table>
The Transmission Rate section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

- **Packets Received/Sec**
- **Packets Sent/Sec**

**Packets Sent/Sec**
The Packets Sent/Sec statistic is the number of packets sent over each network adapter per second.

**Packets Received/Sec**
The Packets Received/Sec statistic is the number of packets received over each network adapter per second.

**Transmission Rate (Bytes)**
The Transmission Rate (Bytes) section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

- **Received (KB/Sec)**
- **Sent (KB/Sec)**

**Sent (KB/Sec)**
The Sent (KB/Sec) statistic is the rate at which bytes are sent over each network adapter including framing characters.

**Received (KB/Sec)**
The Received (KB/Sec) statistic is the rate at which bytes are received over each network adapter, including framing characters.

**Transmission Queue Length**
The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue.

**Packet Collisions**
A collision occurs when two devices on the network sense that the network is idle and end up trying to send data at the same time. Some collisions are normal on a repeated network, but excess collisions can cause serious performance problems.
Troubleshooting
Contact your network administrator if there is a high rate of persistent Packet Collisions.

Packet Discards
The Packet Discards section of the Network tab succinctly communicates the packet discard rate. The following statistics are available in this section:

| Inbound Packets Discarded | Outbound Packets Discarded | Packet Discards |

Outbound Packets Discarded
The Outbound Packets Discarded statistic is the number of outbound packets that were discarded even though no errors had been detected to prevent from being transmitted.

Troubleshooting
A possible reason for discarding such a packet could be to free up buffer space.

Inbound Packets Discarded
Received Packets Discarded is the number of received packets that were discarded even though no errors had been detected to prevent from being transmitted.

Troubleshooting
A possible reason for discarding such a packet could be to free up buffer space.

Packet Discards
The Packet Discards statistic represents the number of network packets discarded.

Troubleshooting
Contact your network administrator if there is a high rate of persistent packet discards.

Packet Errors
The Packet Errors section of the Network tab succinctly communicates the packet error rate. The following statistics are available in this section:

- Inbound Packet Errors
- Outbound Packet Errors

Outbound Packet Errors
The Outbound Packet Errors statistic is the outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
Inbound Packet Errors
The Packets Received Errors statistic is the received packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Top SQL
- **Metrics**

One or two bad queries can cause a lot of trouble for the remaining sessions in a database, and so can one or two bad queries. It is important to find them before they get into a production system, but sometimes a few sneak through.

By applying custom filters and performance-related thresholds, the Top SQL view locates inefficient SQL. By applying filters to certain I/O and statistical counters, you hope to isolate queries that far exceed their nearest competitors in the same area (like disk reads). When you find them, you should reduce the number of sorts a query performs. Or, for a query that returns only a few records, you should try to minimize the number of rows a query processes.

**Metrics**
When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer:

- What has been the worst SQL that's historically been run in my database?
- What is the worst SQL that's running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

The following tabs are available on the Top SQL page:

<table>
<thead>
<tr>
<th>Summary</th>
<th>Criteria</th>
<th>SQL Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic SQL Cache</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Summary**
The Summary tab is divided into two sections:

- **Top SQL Statements**
- **Top SQL Details**

**Top SQL Statements**
A lot of a database’s overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements or stored procedures can drag an otherwise well-configured database down in terms of user response times.

Before you can identify problem SQL in your database, you have to ask the question of what ‘bad SQL’ is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL, so there is no way to sufficiently answer this question to every SQL Server professional’s satisfaction. The Top SQL results grid displays the top SQL statements across various performance categories including Reads, Writes, CPU statistics, and Elapsed Time.
Top SQL Details
This section shows the details of the top SQL statements. The columns presented here include:

**Usage:** % of all operations of this type performed by the selected SQL statement.

**SQL text:** Text of the selected SQL statement.

**Total Reads:** Total number of read operations performed by the selected SQL statement.

**Total Writes:** Total number of write operations performed by the selected SQL statement.

**Executions:** Number of times the selected SQL statement was executed.

**First Start Time:** First time the selected SQL statement was executed.

**Avg CPU Time:** Avg CPU time used by the selected SQL statement.

**Avg Elapsed Time:** Avg Elapsed time for selected SQL statement.

Criteria
The Top SQL view displays all SQL running or executed in the current datasource. You can use the lists and boxes to reorder the display according to your specific needs. The Criteria tab of the Top SQL page includes the following sections that you can mix and match to give you the information you want to find.

- Demographic Criteria
- Statistical Criteria
- Sort Criteria

SQL Details
The SQL Details tab presents the following sections:

<table>
<thead>
<tr>
<th>Top SQL Results</th>
<th>Selected SQL Text</th>
<th>Selected SQL Last Snapshot Details</th>
</tr>
</thead>
</table>

Top SQL Results
This section shows the details of the SQL statements and (their metrics) captured at each refresh of the performance analyst. The information presented here includes

**SQL text:** Partial text of the captured SQL statement. The full text is shown in the selected SQL text section.

**Agents Top:** The maximum number of agents that were used when executing the selected statement at the time of last refresh

**Max Agents Top:** the maximum number of agents that were used when executing the selected statement over all captured executions

**Fetch Count:** Number of successful fetches performed for the specific statement at the time of last refresh

**Avg Fetch Count:** Avg number of successful fetches for the selected statement over all its captured executions

**Int. Rows Deleted:** Number of internal rows deleted for the selected statement as a result of internal activity

**Avg Int Rows Deleted:** over all its captured executions

Int Rows Inserted
**Avg Int Rows Inserted** over all its captured executions

Int. Rows Updated

**Avg Internal Rows Updated** over all its captured executions

**#Agents**

**#Rows Estimate** over all its captured executions

**Avg Cost Estimate** over all its captured executions

Rows Read

**Avg Rows Read** over all its captured executions

Rows Written

**Avg Rows Written** over all its captured executions

**Avg Sorts** over all its captured executions

Sys CPU Time

**Avg Sys CPU Time** over all its captured executions

Usr CPU Time

**Avg Usr CPU Time** over all its captured executions

Total Sort Time

**Avg Total Sort Time** over all its captured executions

Buffer pool Hit Ratio

**Overall Buffer pool Hit Ratio** over all its captured executions

Temp Data Hit Ratio

**Overall Temp Data Hit Ratio** over all its captured executions

**Selected SQL Text**

The Selected SQL Text window allows you to select any of the Top SQL result rows and get the complete SQL text. From here you can open an Explain Plan or Tune the SQL.

**Selected SQL Last Snapshot Details**

This displays attributes for the last snapshot of the selected SQL statement

**Dynamic SQL Cache**

The Dynamic SQL Cache tab of the Top SQL view displays the following sections:

<table>
<thead>
<tr>
<th>Dynamic SQL Cache Details</th>
<th>Dynamic SQL Statement</th>
<th>Execution and Preparation Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bufferpool Analysis</td>
<td>Execution Time Details</td>
<td></td>
</tr>
</tbody>
</table>
Dynamic SQL Cache Details

**Metrics**

The DB2 statement cache stores packages and statistics for frequently used SQL statements. This section shows the statements in this cache and the metrics associated with each statement.

**Metrics**

By examining the contents of this cache, you can identify the dynamic SQL statements that are most frequently executed and the queries that consume the most resources. Using this information, you can examine the most commonly executed and most expensive SQL operations, to determine if SQL tuning could result in better database performance.

Dynamic SQL Statement

The Dynamic SQL Text window allows you to select any of the Dynamic SQL Cache Detail and get the complete SQL text. From here you can open an Explain Plan or Tune the SQL.

Execution and Preparation Details

**Metrics**

Here you see the number of times the selected SQL statement was compiled and executed.

**Metrics**

You can use the execution information to identify the most commonly executed SQL statements in your system. You will probably see the greatest performance benefit from tuning them.

You can use the compilation information to determine the number of different compilations for the selected SQL statement. Use this information together with the number of executions to determine if a bad compilation environment is skewing the metrics.

Bufferpool Analysis

**Metrics**

This section shows the various bufferpool usage metrics for the selected SQL statement.

**Metrics**

This information helps you determine the scope when you tune bufferpools.

Execution Time Details

**Metrics**

The information presented here may be used to identify the longest running SQL.

**Metrics**

The longest running SQL statements together with the most frequently used SQL statements help you identify the most likely candidates for SQL tuning.
Space Analyst

Space management plays a key role in a database's uptime and overall performance. Because of this, it's important that a database administrator understand the various types of space bottlenecks that can occur in a database.

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to help you effectively locate and troubleshoot bottlenecks and performance inefficiencies that result from the challenges of space management. Running inside the Embarcadero DBArtisan console, Space Analyst automates all diagnostic and correction procedures so any space-related headache can be quickly detected and corrected before it reaches critical mass.

The table below describes the major sections of the help available for the Space Analyst.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Provides information about opening, and licensing Space Analyst.</td>
</tr>
<tr>
<td>Application Basics</td>
<td>Provides conceptual information about Space Analyst's product design and navigation.</td>
</tr>
<tr>
<td>Using Space Analyst</td>
<td>Shows how to use each component of the application. Using Space Analyst breaks each process and procedure into simple step-by-step instructions.</td>
</tr>
<tr>
<td>Space Analyst for Oracle Tutorial</td>
<td>Helps you get started with Embarcadero Space Analyst.</td>
</tr>
<tr>
<td>Space Analyst for Microsoft SQL Server Tutorial</td>
<td>Helps you get started with Embarcadero Space Analyst.</td>
</tr>
<tr>
<td>Space Analyst for Sybase ASE Tutorial</td>
<td>Helps you get started with Embarcadero Space Analyst.</td>
</tr>
<tr>
<td>Space Analyst for IBM DB2 for Linux, Unix, and Windows Tutorial</td>
<td>Helps you get started with Embarcadero Space Analyst.</td>
</tr>
<tr>
<td>Space Analyst Expert Guide</td>
<td>Provides detailed information on each tab of the Space Analyst and discusses fragmentation concepts.</td>
</tr>
</tbody>
</table>
Welcome to Space Analyst

Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance problems that result from poor space management. Space Analyst helps ensure that database storage is being used as efficiently as possible.

Space Analyst also includes a Reorganization Wizard as part of the Space Management Wizard that handles full storage structure and object reorganization needs.

For more information, see:

- What You Should Know Before Installation
- Opening the Space Analyst
- Licensing Space Analyst
- Using Space Analyst
- Using Help
- Embarcadero Technologies Technical Support

What You Should Know Before Installation

When you install the Analyst, several Embarcadero-generated stored objects are pushed onto the database server. These objects are as follows:

**SQL Server**

**Procedures**

<table>
<thead>
<tr>
<th>sp_embt_sa_database_map</th>
<th>sp_embt_sa_database_overview</th>
<th>sp_embt_sa_dbbottlenecks</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_embt_sa_filegroups</td>
<td>sp_embt_sa_files</td>
<td>sp_embt_sa_index_space_general</td>
</tr>
<tr>
<td>sp_embt_sa_index_space_reorg</td>
<td>sp_embt_sa_index_space_reorg_summary</td>
<td>sp_embt_sa_server_overview</td>
</tr>
<tr>
<td>sp_embt_sa_table_space_general</td>
<td>sp_embt_sa_table_space_reorg</td>
<td>sp_embt_sa_table_space_reorg_summary</td>
</tr>
<tr>
<td>sp_embt_sa_index_reorg_summary</td>
<td>sp_embt_sa_table_reorg_summary</td>
<td>sp_embt_sa_version</td>
</tr>
</tbody>
</table>

**SYBASE**

**Procedures**

<table>
<thead>
<tr>
<th>sp_embt_sa_dbspace</th>
<th>sp_embt_sa_device_summary</th>
<th>sp_embt_sa_device_detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_embt_sa_database_overview</td>
<td>sp_embt_sa_db_obj_space</td>
<td>sp_embt_sa_database_detail</td>
</tr>
<tr>
<td>sp_embt_sa_db_frag_summary</td>
<td>sp_embt_sa_index_diag_summary</td>
<td>sp_embt_sa_index_diag</td>
</tr>
<tr>
<td>sp_embt_sa_server_overview</td>
<td>sp_embt_sa_table_diag_summary</td>
<td>sp_embt_sa_table_diag</td>
</tr>
<tr>
<td>sp_embt_sa_version</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** No procedures are installed in Space Analyst for Oracle or DB2.
Opening an Analyst
To open the Space Analyst, do the following:

1. On the Analyst menu, click Space Analyst.

   **NOTE:** The first time that you open the Space Analyst, DBArtisan alerts you that you need to install the Analyst’s procedures. When you install the Analyst, several procedures and tables are pushed onto the database server. For a list of these procedures and tables, see What You Should Know Before Installation.

Uninstalling a Repository
To uninstall an Analyst repository, do the following:

1. On the Command Menu, select Uninstall Repository.

   The Analyst Repository Uninstallation Wizard opens.

Analyst Repository Uninstallation Wizard - Panel 1
The table below describes the options and functionality on this panel of the Analyst Repository Uninstallation Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator Login</td>
<td>Displays the login name, password, and login as information.</td>
</tr>
<tr>
<td>Database to hold the procedures</td>
<td>Displays the database that stores the procedures.</td>
</tr>
<tr>
<td>Uninstall</td>
<td>Uninstalls the repository and opens Panel 2.</td>
</tr>
</tbody>
</table>

For more information see Uninstalling Repository.

Analyst Repository Uninstallation Wizard - Panel 2
The table below describes the options and functionality on this panel of the Analyst Repository Uninstallation Wizard:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository Login</td>
<td>Displays repository login.</td>
</tr>
<tr>
<td>Execution Log</td>
<td>Displays the repository execution log and links to ETRepoLog.txt.</td>
</tr>
<tr>
<td>Execution Error(s)</td>
<td>Displays any errors.</td>
</tr>
<tr>
<td>Finish</td>
<td>Closes wizard.</td>
</tr>
</tbody>
</table>

For more information see Uninstalling Repository.

Licensing Space Analyst
The Space Analyst is licensed as a component add-on to DBArtisan. You are issued one license key for the number of databases/servers you want to license. For example, if you purchase ten licenses for Space Analyst, you only need to obtain one license key from Embarcadero.
For more information, see Analyst License Manager.

Space Analyst License Manager
The Analyst License Manager lets you manage licenses.

The table below describes the options and functionality on the Analyst License Manager:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Summary</td>
<td>Displays basic licensing metadata for the product.</td>
</tr>
<tr>
<td>Available Datasources</td>
<td>Lets you select a datasource from the list of datasources available to license.</td>
</tr>
<tr>
<td>License</td>
<td>Click to add a license to a datasource.</td>
</tr>
<tr>
<td>Licensed Datasource(s)</td>
<td>Displays datasources already licensed.</td>
</tr>
<tr>
<td>Remove</td>
<td>Click to remove a license from a datasource.</td>
</tr>
<tr>
<td>Apply</td>
<td>Click to make your license choices permanent.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Click to close the Licensing Dialog box.</td>
</tr>
</tbody>
</table>

**NOTE:** If unsaved license changes have been made, the Analyst asks if you want to discard their changes before exiting.

Completing the Space Analyst License Manager
To complete the Analyst License Manager, do the following:

1. Open DBArtisan.
2. On the Help menu, click Licensing.
   DBArtisan opens the License Configuration Dialog box.
3. Click License.
   DBArtisan opens the Analyst License Manager.
4. Select options.
5. To make your license choices permanent, click Apply.
6. To close the Analyst License Manager, click Cancel.

For more information, see Analyst License Manager.

Space Analyst Add an Analyst License Dialog Box
The Add an Analyst License Details dialog box lets you add a license. The table below describes the options and functionality on the Add a Analyst License Details dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a DBMS platform</td>
<td>Lets you select the DBMS platform.</td>
</tr>
</tbody>
</table>

For more information, see Completing the Analyst License Manager.
Using Help
DBArtisan provides comprehensive, context-sensitive Help. Help includes detailed information on using the application. DBArtisan Help includes the following standard conventions:

Fonts
The table below describes the conventions in DBArtisan Help to distinguish elements of text

<table>
<thead>
<tr>
<th>Font</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Sequential list of steps to complete a task.</td>
</tr>
<tr>
<td>Result Font</td>
<td>Result of an action in the previous step.</td>
</tr>
<tr>
<td>Bold Font</td>
<td>Elements of the application, including dialog box and window titles, menus, commands on menus, dialog box options, buttons, icons.</td>
</tr>
<tr>
<td>Italic Font</td>
<td>Variables.</td>
</tr>
<tr>
<td>Monospace Font</td>
<td>Code samples.</td>
</tr>
<tr>
<td>UPPERCASE Font</td>
<td>Keyboard combinations. SQL keywords and elements.</td>
</tr>
<tr>
<td>Underlined Blue Font</td>
<td>Hyperlinked words and phrases. Click to jump to a new location with related information.</td>
</tr>
</tbody>
</table>

For more information, see Using Help.

Opening DBArtisan Help
Although the most common way to access Help is through context-sensitive Help, DBArtisan also offers three additional ways to access Help:

1. On the Help menu, click Help Topics.
2. On the Windows Start button, point to Programs, Embarcadero DBArtisan, and then click DBArtisan Help.
3. On the Windows desktop, open the Embarcadero DBArtisan folder, and then double-click Help.

TIP: To toggle between Help and the application, on the Windows Start button, open DBArtisan Help, and then press ALT+TAB.

For more information, see Using DBArtisan Help.

Using DBArtisan Help
Help is a customizable window that lets you access DBArtisan Help. The table below describes the tabs in the DBArtisan Help window:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>The Table of Contents for the Help system.</td>
</tr>
<tr>
<td>Index</td>
<td>A comprehensive index of DBArtisan Help.</td>
</tr>
<tr>
<td>Search</td>
<td>A search engine for every word in the DBArtisan Help. DBArtisan Help features an advanced full-text search, letting you search using Boolean, wildcard, and nested expressions.</td>
</tr>
</tbody>
</table>
Embarcadero Technologies Technical Support

Space Analyst is fully supported by the Embarcadero Technologies Technical Support team. The Technical Support team is available to assist you with any problems you have with this application. When contacting Embarcadero Technologies Technical Support please provide the following:

• Product name.
• Product version.
• Operating system (server and client).
• Database platform in which the error occurs (including version).
• Error message (exact text or a screen shot if possible).
• Detailed description of the problem including steps to reproduce.

Every Embarcadero Technologies application includes sixty days free technical support. We strongly recommend that you purchase a maintenance contract to continue receiving technical support after this period.

Contacting Embarcadero Technologies Technical Support

Embarcadero Technologies offers three ways to contact Embarcadero Technical Support:

1. Web
2. Phone
3. E-mail

Web
Embarcadero Technologies Technical Support on the Web is available through:

• Embarcadero Technologies Web Site
• DBArtisan Help

Embarcadero Technologies Web Site
• Complete a technical support form at the Embarcadero Technologies Technical Support Web page.

DBArtisan Help
To open Embarcadero Technologies Technical Support on the Web using DBArtisan:

1. On the Help menu, point to Embarcadero on the Web, and then click Online Technical Support.
   DBArtisan connects to the Web and opens the Embarcadero Technologies Technical Support Request Web page.
2. Complete the online form.
3 Click Submit.

Phone
The Embarcadero Technologies Technical Support phone number is (415) 834-3131 option 2, and then follow the prompts. The hours are Monday through Friday, 6:00 A.M. to 6:00 P.M. Pacific time.

For licensing issues, including Product Unlock Codes, call (415) 834-3131 option 2, and then follow the prompts. The hours are Monday through Friday, 6:00 A.M. to 6:00 P.M., Saturday and Sunday 8 A.M. to 5 P.M. Pacific time.

The Embarcadero Technologies Technical Support fax number is (415) 495-4418.

E-mail
Embarcadero Technologies offers two ways to contact Technical Support via e-mail:

1 Complete a technical support form at the Embarcadero Technologies Technical Support Web page.

OR

2 Send an e-mail message to support@Embarcadero.com.

Please include the following in your e-mail message:

• Product name.
• Product version.
• Operating system (client and server).
• Database platform in which the error occurs (including version).
• Error message (exact text or a screen shot if possible).
• Detailed description of the problem including steps to reproduce.
Space Analyst Application Basics

The Space Analyst utility assists DBAs with space diagnostics and reorganizations by displaying pertinent reorganization diagnostics and providing a Reorganization Wizard as part of the Space Management Wizard that builds scripts that you can execute or schedule to perform reorganizations.

For more information, see Product Design.

Space Analyst Product Design

The Space Analyst product design includes tabs for Oracle, Microsoft SQL Server and IBM DB2.

Oracle Space Analyst Tabs

The Space Analyst includes the following tabs:

<table>
<thead>
<tr>
<th>Database Overview Tab</th>
<th>Datafiles Tab</th>
<th>Tables Tab</th>
<th>Indexes Tab</th>
<th>Tablespace Map Tab</th>
</tr>
</thead>
</table>

Microsoft SQL Server Space Analyst Tabs

The Space Analyst includes the following tabs:

<table>
<thead>
<tr>
<th>Storage Overview Tab</th>
<th>Files/File Groups Tab</th>
<th>Map Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases Tab</td>
<td>Objects Tab</td>
<td></td>
</tr>
</tbody>
</table>

IBM DB2 Space Analyst Tabs

The Space Analyst includes the following tabs:

<table>
<thead>
<tr>
<th>Database Overview Tab</th>
<th>Tables Spaces Tab</th>
<th>Tablespace Map Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables Tab</td>
<td>Indexes Tab</td>
<td></td>
</tr>
</tbody>
</table>

Sybase Space Analyst Tabs

The Space Analyst includes the following tabs:

<table>
<thead>
<tr>
<th>Storage Overview Tab</th>
<th>Database Devices</th>
<th>Databases Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables Tab</td>
<td>Indexes Tab</td>
<td></td>
</tr>
</tbody>
</table>

Database Overview Tab

The Database Overview Tab displays the storage overview for the DB2 datasource.

There are four distinct sections displayed on the first Database Overview Tab of Space Analyst:

- Database Overview
• Log Overview (DB2 Only)
• Tablespace Space Overview
• Tablespace Fragmentation Overview
• Tablespace Objects Overview
For more information, see Space Analyst Tabs.

Data Files Tab
The Data Files Tab displays datafile fragmentation information, and displays how much space in each datafile is assigned to tables, indexes, clusters, and rollback segments.
For more information, see Space Analyst Tabs.

Tablespace Map Tab
The Tablespace Map Tab allows for tablespace mapping.
For more information, see Space Analyst Tabs.

Tables Tab
The Tables Tab lets you identify tables with wasted space. The Space Analyst computes a block inefficiency ratio that quickly communicates the severity of the space problem. Space Analyst provides a summary count of all tables in a database that violate a customizable block inefficiency threshold, details on the tables, and the extent of wasted space.
For more information, see Space Analyst Tabs.

Indexes Tab
The Indexes Tab lets you select index reorganization candidates. The tab also offers the option to obtain deleted leaf row information for each index, although such data gathering can take some time, especially on large systems.
Space Analyst runs diagnostics against your database and graphically display a count of indexes that are found to contain various types of space problems.
For more information, see Space Analyst Tabs.

Tablespaces Tab
The Tablespaces Tab displays the Tablespace Container allocations and Database Partition Breakdown for each Tablespace.

There are three distinct sections displayed on the Tablespaces Tab of Space Analyst:
• Space Allocation by Container
• Space Allocation by Node
• Tablespaces
Using Space Analyst

Using Space Analyst covers how to use each component of the application. The Using chapter breaks each process and procedure into simple step-by-step instructions.

The Space Analyst includes the following tabs, by platform:

**Space Analyst for Oracle**

<table>
<thead>
<tr>
<th>Database Overview Tab</th>
<th>Datafiles Tab</th>
<th>Tables Tab</th>
<th>Indexes Tab</th>
<th>Tablespace Map Tab</th>
</tr>
</thead>
</table>

**Space Analyst for Microsoft SQL Server**

<table>
<thead>
<tr>
<th>Storage Overview Tab</th>
<th>Files/File Groups Tab</th>
<th>Map Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases Tab</td>
<td>Objects Tab</td>
<td></td>
</tr>
</tbody>
</table>

**Space Analyst for Sybase ASE**

The Space Analyst includes the following tabs:

<table>
<thead>
<tr>
<th>Storage Overview Tab</th>
<th>Database Devices Tab</th>
<th>Databases Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables Tab</td>
<td>Indexes Tab</td>
<td></td>
</tr>
</tbody>
</table>

**Space Analyst for IBM DB2 for Linux, Unix, and Windows**

The Space Analyst includes the following tabs:

<table>
<thead>
<tr>
<th>Storage Overview Tab</th>
<th>Datafiles Tab</th>
<th>Tablespace Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables Tab</td>
<td>Indexes Tab</td>
<td></td>
</tr>
</tbody>
</table>

**Space Analyst for Oracle and IBM DB2 for Linux, Unix, and Windows - Database Overview Tab**

The Database Overview Tab lets you view the global storage picture of your database.

The table below describes the options and functionality on the Database Overview Tab of Space Analyst for Oracle:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Overview</td>
<td>Shows how much space in KB is taken up in the database by various object types (tables, indexes, etc.) along with a pie chart that represents the percentage of each object type. You can switch between the default graphical view and a grid view.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> Objects owned by SYS and SYSTEM are not included in this breakdown.</td>
</tr>
<tr>
<td>Tablespace Space Overview</td>
<td>When in graphical mode, this section contains a bar chart graph that displays all tablespaces and how much total space each has allocated. When in grid mode, Space Analyst displays detailed information about all tablespaces (status, total, used, free, percent free, etc.)</td>
</tr>
<tr>
<td></td>
<td><strong>ORACLE ONLY:</strong> When in grid mode, Space Analyst displays tablespace status. You can switch between the default graphical view and a grid view.</td>
</tr>
</tbody>
</table>
The table below describes the options and functionality on the Database Overview Tab of Space Analyst for IBM DB2 for Linux, Unix, and Windows:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace Fragmentation Overview</td>
<td>Contains fragmentation information for all tablespaces and displays how badly fragmented a tablespace or datafile is. Higher scores for the tablespace’s fragmentation index indicate little or no fragmentation, while lower scores indicate fragmentation. You can switch between graphical and grid views of tablespace fragmentation. If fragmentation is present, you can view detailed object and fragmentation information for the tablespace in the tablespace map. You can correct fragmentation problems via Space Analyst's Reorganization Wizard. You can switch between the default graphical view and a grid view.</td>
</tr>
<tr>
<td>Tablespace Objects Overview</td>
<td>Contains a grid that displays how much space in each tablespace is assigned to tables, indexes, clusters, and rollback segments. ORACLE ONLY: When in grid mode, Space Analyst displays tablespace LOB data values.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Overview</td>
<td>Shows how much space in KB is taken up in the database by various object types (tables, indexes, etc.) along with a pie chart that represents the percentage of each object type. You can switch between the default graphical view and a grid view.</td>
</tr>
<tr>
<td>Log Space Overview</td>
<td>Displays all of the log options for the database: Total Used, Total Free, Block Log When Disk Full, Log File Size, User Exit, Primary Logs, Secondary Logs, Maximum Secondary Log Space Used, Maximum Total Log Space Used.</td>
</tr>
<tr>
<td>Tablespace Space Overview</td>
<td>When in graphical mode, this section contains a bar chart graph that displays all tablespaces and how much total space each has allocated. When in grid mode, displays detailed information about all tablespaces: Name, Type, State, Used Space (MB), Free Space (MB), Content Type (Any, Long, Temp, System temp), Number of containers. NOTE: You can switch between the default graphical view and a grid view.</td>
</tr>
<tr>
<td>Tablespace Objects Overview</td>
<td>Contains a grid that displays how much space in each tablespace is assigned to each object. Also displays tablespace LOB data values. Displays detailed information about all tablespace objects: Tablespace Name, Table - total space used by tables, Index - total space used by indexes, Extent Size - tablespace extent size, Page size - Tablespace page size.</td>
</tr>
</tbody>
</table>
Available Functionality
The following functionality is available in the Command menu, shortcut menu, and toolbar:

**NOTE:** Functionality differ by platform.

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Convert to Dictionary-Managed</th>
<th>Convert to Locally-Managed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract</td>
<td>Estimate Size</td>
<td>Map</td>
</tr>
</tbody>
</table>

Open Editor. For information on modifying database objects, see Object Editors.

Open Wizard | Reorganize | Update Statistics

| Tablespace Wizard for IBM DB2 for Linux, Unix, and Windows | Threshold Options |

For more information, see:

Correcting Tablespace 'Out of Space' Problems

Space Analyst Tabs

Space Analyst for Oracle and IBM DB2 for Linux, Unix, and Windows - Datafiles Tab
The Data Files Tab lets you view fragmentation and object placement information for data files.

The table below describes the options and functionality on the Datafiles Tab of the Space Analyst:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datafile Fragmentation</td>
<td>Displays fragmentation about datafiles. Higher scores for a datafile's fragmentation index indicate little or no fragmentation, while lower scores indicate fragmentation. If fragmentation is present, you can view detailed object and fragmentation information for the datafile's tablespace in the tablespace map. You can correct fragmentation problems via Space Analyst's Reorganization Wizard. <strong>NOTE:</strong> Datafiles for temporary tablespaces are not included in the Datafile Fragmentation grid.</td>
</tr>
<tr>
<td>Datafile Objects Overview</td>
<td>Contains a grid that displays how much space in each datafile is assigned to tables, indexes, clusters, and rollback segments.</td>
</tr>
</tbody>
</table>

**TIP:** To open a Tablespace Editor, right-click a datafile in any grid, and then select Open. To extract the tablespace DDL in an ISQL editor, right-click a datafile in any grid, and then select Extract.

Available Functionality
The following functionality is available in the Command menu, shortcut menu, and toolbar:

**NOTE:** Functionalists differ by platform.

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Convert to Dictionary-Managed</th>
<th>Convert to Locally-Managed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate Size</td>
<td>Extract</td>
<td>Map</td>
</tr>
</tbody>
</table>

Open Editor. For information on modifying database objects, see Object Editors.
Reorganize

Threshold Options

For more information, see Space Analyst Tabs.

Space Analyst for Oracle - Tablespace Map Tab

The Tablespace Map Tab lets you view a graphical layout regarding the physical placement of every object inside a selected tablespace. The tablespace map also helps identify fragmentation problems for a tablespace (honeycomb and bubble). An object grid is also present that lets you view detailed information about every object's storage properties.

The Space Info section contains information regarding the total, used, and free space (in MB) for the tablespace.

The View By section allows you to switch between two different tablespace map views:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Type</td>
<td>The normal tablespace map mode, which is color-coded by object type. Pressing the legend button shows the various colors assigned to the object types contained within the tablespace.</td>
</tr>
<tr>
<td>Extent Problems</td>
<td>When in this mode, the tablespace map highlights in red any object containing one or both of these object extent problems:</td>
</tr>
<tr>
<td></td>
<td>- Max extent problem - the object has reached or is near its maximum extent limit.</td>
</tr>
<tr>
<td></td>
<td>- Extent deficit - the object may not be able to allocate its next extent because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.</td>
</tr>
</tbody>
</table>

When in Object Type mode, the accompanying grid of the tablespace map contains the following data:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the object in owner.object name.partition name format.</td>
</tr>
<tr>
<td>Extents</td>
<td>The total number of allocated extents for the object.</td>
</tr>
<tr>
<td>Blocks</td>
<td>The total number of allocated blocks for the object.</td>
</tr>
<tr>
<td>Bytes</td>
<td>The total number of allocated bytes for the object.</td>
</tr>
<tr>
<td>Type</td>
<td>The object type (table, index, etc.).</td>
</tr>
<tr>
<td>Initial Extent</td>
<td>The initial space extent (in bytes) allocated to the object.</td>
</tr>
<tr>
<td>Next Extent</td>
<td>The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.</td>
</tr>
<tr>
<td>Percent Increase</td>
<td>The percent increase property for the object.</td>
</tr>
</tbody>
</table>

When in Extent Problems mode, the accompanying grid of the tablespace map contains the following data:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the object in owner.object name.partition name format.</td>
</tr>
<tr>
<td>Extents</td>
<td>The total number of allocated extents for the object.</td>
</tr>
<tr>
<td>Blocks</td>
<td>The total number of allocated blocks for the object.</td>
</tr>
<tr>
<td>Bytes</td>
<td>The total number of allocated bytes for the object.</td>
</tr>
<tr>
<td>Type</td>
<td>The object type (table, index, etc.).</td>
</tr>
</tbody>
</table>
When viewing a tablespace in the graphical map, there are two broad categories of fragmentation that you will want to be on the lookout for: free space honeycombs and free space bubbles. Honeycomb fragmentation is really not difficult to detect or handle and occurs when two free space extents in a tablespace reside next to one another. For example, rather than having one large free space extent of 6MB, a tablespace might have two extents that are next to each other (a honeycomb) that make up 3MB and 3MB. If an incoming object required a single 4MB extent allocation, Oracle would have to coalesce the honeycomb to make room for the object. The necessary coalesce operation would cause a small degree of performance degradation, however the situation would be resolved and no space-related bottleneck would remain.

Fragmentation bubbles, however, are another story and are the traditional reason DBAs have had to perform full tablespace reorganizations. Bubbles occur when objects are continuously added and dropped in a tablespace, with the end result being free space pockets (or bubbles) being interspersed between objects that remain in the tablespace. This can become problematic when, for example, an incoming object requires a single 5MB extent allocation, but there is not enough contiguous space in the tablespace to meet the demand. Even if the total amount of free space in the tablespace equals the required 5MB, it will do no good if that 5MB is made up of 5 non-contiguous blocks of free space. In this situation, Oracle has no alternative but to either extend the datafile to make room for the object (if the tablespace's datafile has the AUTOEXTEND property enabled), or return an error and deny the object entry into the tablespace.

Space Analyst makes it easy to recognize both of these bottleneck headaches. When the map is viewed in Object Type mode, pockets of green seen throughout the tablespace indicate bubble fragmentation, while two or more blocks of green segments viewed next to one another in the map indicate honeycomb fragmentation.

The Tablespace Map Tab includes the following buttons:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legend</td>
<td>Opens the Tablespace Map Legend.</td>
</tr>
<tr>
<td>Display</td>
<td>Opens the Display window. The Display window lets you specify the magnification level for the Tablespace Map.</td>
</tr>
<tr>
<td>Fit to Window</td>
<td>Fits the Tablespace Map to the window.</td>
</tr>
<tr>
<td>Full Screen</td>
<td>Fits the Tablespace Map to the full screen.</td>
</tr>
<tr>
<td>Print Map</td>
<td>Opens the Print dialog box to print the Tablespace Map.</td>
</tr>
</tbody>
</table>

### Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Extent</td>
<td>The initial space extent (in bytes) allocated to the object.</td>
</tr>
<tr>
<td>Next Extent</td>
<td>The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.</td>
</tr>
<tr>
<td>Percent Increase</td>
<td>The percent increase property for the object.</td>
</tr>
<tr>
<td>Max Extent Percent</td>
<td>A percentage indicating how close the object is to reaching its maximum extent limit. For example, a 100 indicates an object that has reached its maximum extent limit.</td>
</tr>
<tr>
<td>Extendibility</td>
<td>Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.</td>
</tr>
</tbody>
</table>
Space Analyst for Oracle, Sybase, and IBM DB2 for Linux, Unix, and Windows - Tables Tab

The Tables Tab displays diagnostic information regarding table space inefficiencies that may exist in your database.

The Table Reorganization Candidate Summary section uses a bar chart to count the various types of table space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the four distinct types of table space inefficiencies that Space Analyst computes for Oracle and Sybase:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Inefficiencies</td>
<td>These tables have wasted space that exceeds the allowable percentage of wasted space set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables with wasted space can take longer to scan and therefore contribute to slower response times.</td>
</tr>
<tr>
<td>Next Extent Problems</td>
<td>These tables cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Tables suffering from this problem may produce error conditions in the database if more data is added to them. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.</td>
</tr>
<tr>
<td>Chain Problems</td>
<td>These tables suffer from a chained or migrated row problem. The percentage of chained or migrated rows to overall rows contained in the table exceeds the allowable percentage of chained rows set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Because chained/migrated rows cause more I/O to occur when they are accessed, overall response times can be degraded when these tables are referenced in SQL queries.</td>
</tr>
<tr>
<td>Max Extent Problems</td>
<td>These tables are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables suffering from this problem may cause space-related errors to result if more data is added to them.</td>
</tr>
</tbody>
</table>

The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>The owner of the table.</td>
</tr>
<tr>
<td>Table Name</td>
<td>The name of the table.</td>
</tr>
<tr>
<td>Segment Type</td>
<td>Indicates if the object is a table, table partition, etc.</td>
</tr>
<tr>
<td>Table (KB)</td>
<td>The total allocated space of the table in KB.</td>
</tr>
<tr>
<td>Number of Rows</td>
<td>The number of rows that the table contains.</td>
</tr>
</tbody>
</table>

**NOTE:** This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
The table below describes the four distinct types of Reorganization Candidate Summary that Space Analyst computes for IBM DB2 for Linux, Unix, and Windows:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Blocks</td>
<td>The total number of blocks allocated to the table. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Empty Blocks</td>
<td>The number of empty blocks that the table contains. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Highwater Mark</td>
<td>The last block that Oracle will scan up to when searching the table for data. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Average Used Blocks</td>
<td>The average number of blocks that the table is using. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Block Inefficiency</td>
<td>A percentage that indicates how much wasted space (space that the table is not using) that the table contains. High percentages indicate tables that may be candidates for reorganization. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Chained Row Percent</td>
<td>A percentage that indicates how many rows in the table suffer from chaining or migration. High percentages indicate tables that may be candidates for reorganization. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Max Extent Percent</td>
<td>A percentage indicating how close an object is to reaching its maximum extent limit. High percentages indicate tables that may be candidates for reorganization.</td>
</tr>
<tr>
<td>Number of Extents</td>
<td>The number of extents that the table consumes.</td>
</tr>
<tr>
<td>Max Extents</td>
<td>The maximum extent limit for the table.</td>
</tr>
<tr>
<td>Extent Deficit</td>
<td>Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.</td>
</tr>
<tr>
<td>Next Extent</td>
<td>The next extent (in bytes) that the table will attempt to allocate when more space for the table is required.</td>
</tr>
<tr>
<td>Max Free Space</td>
<td>The largest contiguous block of free space that exists in the tablespace where the table resides.</td>
</tr>
<tr>
<td>Tablespace Name</td>
<td>The tablespace that contains the table.</td>
</tr>
</tbody>
</table>

IBM DB2 for Linux, Unix, and Windows

The table below describes the four distinct types of Reorganization Candidate Summary that Space Analyst computes for IBM DB2 for Linux, Unix, and Windows:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overflow Problems</td>
<td>Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).</td>
</tr>
<tr>
<td>Space Utilization</td>
<td>Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).</td>
</tr>
<tr>
<td>Page Utilization</td>
<td>Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).</td>
</tr>
<tr>
<td>Index Contention</td>
<td>Displays the number of tables where the index space is shared by the Tablespace. To resolve, use the Space Management Wizard to change the Index Tablespace in table definition.</td>
</tr>
</tbody>
</table>

Problem Description:
The table below describes the options and functionality on the Table Reorganization Candidate Details section:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Displays all Tablespaces (regular) in the database and the value [All].</td>
</tr>
<tr>
<td>Table Schema</td>
<td>Displays all schema in the database and the value [All].</td>
</tr>
<tr>
<td>Display</td>
<td>All Tables: Tables with Overflow Problems - Displays a list of all tables where overflow threshold is exceeded. Tables with Space Utilization problems - Displays a list of all tables where space utilization threshold is exceeded. Tables with Page Utilization problems - Displays a list of all tables where the page utilization threshold is exceeded. Tables with Index Contention - Displays a list of all tables where the table and indexes on the table share the same Tablespace.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refreshes the table grid with criteria selected in the above lists.</td>
</tr>
</tbody>
</table>

The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Schema</td>
<td>The schema name for the table.</td>
</tr>
<tr>
<td>Table Name</td>
<td>The name of the table.</td>
</tr>
<tr>
<td>Cardinality</td>
<td>The number of rows in the table.</td>
</tr>
<tr>
<td>Size</td>
<td>The size of Table in bytes.</td>
</tr>
<tr>
<td>Overflow</td>
<td>The number of overflow rows. Overflow rows can be created when rows are updated and the new rows contain more bytes than the old ones (VARCHAR fields), or when columns are added to existing tables.</td>
</tr>
<tr>
<td>Pages with Data</td>
<td>The number of allocated pages that contain data.</td>
</tr>
<tr>
<td>Total Pages</td>
<td>The total number of allocated pages.</td>
</tr>
<tr>
<td>Active Blocks</td>
<td>The number of Active Blocks in Tables with Multidimensional clustering).</td>
</tr>
<tr>
<td>Overflow %</td>
<td>Percentage of overflow rows. This should typical be a small value (5%).</td>
</tr>
<tr>
<td>Space Utilization%</td>
<td>A low space utilization is an indicator of excessive fragmentation (internal and external). This can occur when there are many pages with free space and/or many free pages. This should typically be &gt; 68%.</td>
</tr>
<tr>
<td>Page Utilization %</td>
<td>A low page utilization is an indicator of excessive external fragmentation. This value should typically be more than 80%.</td>
</tr>
<tr>
<td>Reorg</td>
<td>This is an indicator of how severe the fragmentation problem for the table is. --- means Overflow%, Space Utilization% and Page Utilization % are all with in the thresholds. <em>-</em> means that Overflow%, and Page Utilization % are outside the threshold but Space Utilization% is with in the threshold.</td>
</tr>
<tr>
<td>Tablespace</td>
<td>Tablespace where the table data resides.</td>
</tr>
<tr>
<td>Indexspace</td>
<td>Tablespace where the index data resides.</td>
</tr>
</tbody>
</table>

Available Functionality

**NOTE:** Functionality differ by platform.

The following functionality is available in the Command menu, shortcut menu, and toolbar:

| Analysis Report | Analyze | DBCC |
Space Analyst for Oracle - Tables Tab

The Tables Tab displays diagnostic information regarding table space inefficiencies that may exist in your database.

The Table Reorganization Candidate Summary section uses a bar chart to count the various types of table space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the four distinct types of table space inefficiencies that Space Analyst computes for Oracle and Sybase:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Inefficiencies</td>
<td>These tables have wasted space that exceeds the allowable percentage of wasted space set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables with wasted space can take longer to scan and therefore contribute to slower response times.</td>
</tr>
<tr>
<td>Next Extent Problems</td>
<td>These tables cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Tables suffering from this problem may produce error conditions in the database if more data is added to them. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.</td>
</tr>
<tr>
<td>Chain Problems</td>
<td>These tables suffer from a chained or migrated row problem. The percentage of chained or migrated rows to overall rows contained in the table exceeds the allowable percentage of chained rows set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Because chained/migrated rows cause more I/O to occur when they are accessed, overall response times can be degraded when these tables are referenced in SQL queries.</td>
</tr>
<tr>
<td>Max Extent Problems</td>
<td>These tables are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables suffering from this problem may cause space-related errors to result if more data is added to them.</td>
</tr>
</tbody>
</table>

The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>The owner of the table.</td>
</tr>
<tr>
<td>Table Name</td>
<td>The name of the table.</td>
</tr>
<tr>
<td>Segment Type</td>
<td>Indicates if the object is a table, table partition, etc.</td>
</tr>
<tr>
<td>Table (KB)</td>
<td>The total allocated space of the table in KB.</td>
</tr>
</tbody>
</table>
### Available Functionality

**NOTE:** Functionality differ by platform.

The following functionality is available in the Command menu, shortcut menu, and toolbar:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Rows</td>
<td>The number of rows that the table contains.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Number of Blocks</td>
<td>The total number of blocks allocated to the table.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Empty Blocks</td>
<td>The number of empty blocks that the table contains.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Highwater Mark</td>
<td>The last block that Oracle will scan up to when searching the table for data.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Average Used Blocks</td>
<td>The average number of blocks that the table is using.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Block Inefficiency</td>
<td>A percentage that indicates how much wasted space (space that the table is not using) that the table contains. High percentages indicate tables that may be candidates for reorganization.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Chained Row Percent</td>
<td>A percentage that indicates how many rows in the table suffer from chaining or migration.</td>
</tr>
<tr>
<td></td>
<td>High percentages indicate tables that may be candidates for reorganization.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Max Extent Percent</td>
<td>A percentage indicating how close an object is to reaching its maximum extent limit.</td>
</tr>
<tr>
<td></td>
<td>High percentages indicate tables that may be candidates for reorganization.</td>
</tr>
<tr>
<td>Number of Extents</td>
<td>The number of extents that the table consumes.</td>
</tr>
<tr>
<td>Max Extents</td>
<td>The maximum extent limit for the table.</td>
</tr>
<tr>
<td>Extent Deficit</td>
<td>Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.</td>
</tr>
<tr>
<td>Next Extent</td>
<td>The next extent (in bytes) that the table will attempt to allocate when more space for the table is required.</td>
</tr>
<tr>
<td>Max Free Space</td>
<td>The largest contiguous block of free space that exists in the tablespace where the table resides.</td>
</tr>
<tr>
<td>Tablespace Name</td>
<td>The tablespace that contains the table.</td>
</tr>
</tbody>
</table>
Space Analyst - Tables Tab

The Tables Tab displays diagnostic information regarding table space inefficiencies that may exist in your database.

The Table Reorganization Candidate Summary section uses a bar chart to count the various types of table space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the four distinct types of table space inefficiencies that Space Analyst computes for Sybase:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Inefficiencies</td>
<td>These tables have wasted space that exceeds the allowable percentage of wasted space set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables with wasted space can take longer to scan and therefore contribute to slower response times.</td>
</tr>
<tr>
<td>Next Extent Problems</td>
<td>These tables cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Tables suffering from this problem may produce error conditions in the database if more data is added to them. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.</td>
</tr>
<tr>
<td>Chain Problems</td>
<td>These tables suffer from a chained or migrated row problem. The percentage of chained or migrated rows contained in the table exceeds the allowable percentage of chained rows set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Because chained/migrated rows cause more I/O to occur when they are accessed, overall response times can be degraded when these tables are referenced in SQL queries.</td>
</tr>
<tr>
<td>Max Extent Problems</td>
<td>These tables are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables suffering from this problem may cause space-related errors to result if more data is added to them.</td>
</tr>
</tbody>
</table>

The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>The owner of the table.</td>
</tr>
<tr>
<td>Table Name</td>
<td>The name of the table.</td>
</tr>
<tr>
<td>Segment Type</td>
<td>Indicates if the object is a table, table partition, etc.</td>
</tr>
<tr>
<td>Table (KB)</td>
<td>The total allocated space of the table in KB.</td>
</tr>
<tr>
<td>Number of Rows</td>
<td>The number of rows that the table contains. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Number of Blocks</td>
<td>The total number of blocks allocated to the table. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
</tbody>
</table>

For more information, see Space Analyst Tabs.
The table below describes the four distinct types of Reorganization Candidate Summary that Space Analyst computes for IBM DB2 for Linux, Unix, and Windows:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overflow Problems</td>
<td>Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).</td>
</tr>
<tr>
<td>Space Utilization Problem</td>
<td>Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).</td>
</tr>
<tr>
<td>Page Utilization Problem</td>
<td>Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).</td>
</tr>
<tr>
<td>Index Contention</td>
<td>Displays the number of tables where the index space is shared by the Tablespace. To resolve, use the Space Management Wizard to change the Index Tablespace in table definition.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the Table Reorganization Candidate Details section:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Displays all Tablespaces (regular) in the database and the value [All].</td>
</tr>
<tr>
<td>Table Schema</td>
<td>Displays all schema in the database and the value [All].</td>
</tr>
</tbody>
</table>
The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>All Tables</td>
</tr>
<tr>
<td></td>
<td>Tables with Overflow Problems - Displays a list of all tables where overflow threshold is exceeded.</td>
</tr>
<tr>
<td></td>
<td>Tables with Space Utilization problems - Displays a list of all tables where space utilization threshold is exceeded.</td>
</tr>
<tr>
<td></td>
<td>Tables with Page Utilization problems - Displays a list of all tables where the page utilization threshold is exceeded.</td>
</tr>
<tr>
<td></td>
<td>Tables with Index Contention - Displays a list of all tables where the table and indexes on the table share the same Tablespace.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refreshes the table grid with criteria selected in the above lists.</td>
</tr>
</tbody>
</table>

The following functionality is available in the Command menu, shortcut menu, and toolbar:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis Report</td>
<td>Analyze</td>
</tr>
<tr>
<td>Estimate Size</td>
<td>Extract</td>
</tr>
<tr>
<td>New Table</td>
<td>Open Editor</td>
</tr>
</tbody>
</table>
Space Analyst for Sybase and IBM DB2 for Linux, Unix, and Windows - Indexes Tab

The Indexes Tab displays diagnostic information regarding index space inefficiencies that may exist in your database.

The Index Reorganization Candidate Summary section uses a bar chart to count the various types of index space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the three distinct types of table space inefficiencies that Space Analyst computes:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLevel</td>
<td>These indexes have extended beyond the allowable B-level size set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Indexes with extended levels can take longer to scan and can therefore contribute to slower response times.</td>
</tr>
<tr>
<td>Next Extent Problems</td>
<td>These indexes cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Indexes suffering from this problem may cause space-related errors if more data is added to their parent table. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.</td>
</tr>
<tr>
<td>Max Extent Problems</td>
<td>These indexes are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Indexes suffering from this problem may cause space-related errors to result if more data is added to their parent table.</td>
</tr>
</tbody>
</table>

The Index Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected indexes. The grid contains the following data:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Owner</td>
<td>The owner of the index.</td>
</tr>
<tr>
<td>Index Name</td>
<td>The name of the index.</td>
</tr>
<tr>
<td>Index Type</td>
<td>Indicates if the type of index (index, index partition, etc.)</td>
</tr>
<tr>
<td>Index (KB)</td>
<td>The total allocated space of the index in KB.</td>
</tr>
<tr>
<td>Number of Rows</td>
<td>The number of rows that the parent table contains. <strong>NOTE:</strong> This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Clustering Factor</td>
<td>Indicates the amount of order of the rows in the table based on the values of the index. If the value is near the number of blocks, then the table is very well ordered. If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.</td>
</tr>
<tr>
<td>Index Blevel</td>
<td>Indicates the level of the index. Numbers beyond 3-4 can indicate a fragmented index.</td>
</tr>
</tbody>
</table>
NOTE: The grid can also contain a column for deleted leaf rows if the Get Deleted Leaf Row Statistics option has been selected. Be aware that the calculation of this statistic can take a lot of time for databases with large numbers of indexes. Large numbers of deleted leaf rows can indicate an index in need of reorganization.

IBM DB2 for Linux, Unix, and Windows

The Index Reorganization Candidate Summary section uses a bar chart to represent the various types of index space inefficiencies in your database:

- Indexes with Cluster Ratio Problems
- Indexes with Used Space% problems
- Indexes with Balance Problems
- Indexes with Deleted RID problems
- Indexes with Empty Leaf Problems
- Indexes with Low Uniqueness

NOTE: Space Analyst uses the options you select in the Reorganization Wizard to compute each type of space inefficiency.

TIP: All of the problems in the Index Reorganization Candidate Summary section except low cardinality can be corrected with index reorganization. The Low cardinality is an indicator of an index where the columns on which it is defined have low level of uniqueness. Such indexes should either be dropped or changed to include columns that will make the index more unique.

The table below describes the options and functionality on the Index Reorganization Candidate Details section:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>Displays all Tablespaces (regular/large) in the database and the value [All].</td>
</tr>
<tr>
<td>Index Schema</td>
<td>Displays all schema in the database and the value [All].</td>
</tr>
</tbody>
</table>
The Index Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected indexes. The grid contains the following data:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>All Indexes&lt;br&gt;Indexes with Cluster Ratio Problems&lt;br&gt;Indexes with Used Space% problems&lt;br&gt;Indexes with Balance Problems&lt;br&gt;Indexes with Deleted RID problems&lt;br&gt;Indexes with Empty Leaf Problems&lt;br&gt;Indexes with Low Uniqueness</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refreshes the index grid with criteria selected in the above lists.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Schema</td>
<td>The schema name for the index.</td>
</tr>
<tr>
<td>Index Name</td>
<td>The name of the index.</td>
</tr>
<tr>
<td>Table Schema</td>
<td>The schema name for the base table</td>
</tr>
<tr>
<td>Table Name</td>
<td>The base table name.</td>
</tr>
<tr>
<td>Table Cardinality</td>
<td>The number of rows in the base table</td>
</tr>
<tr>
<td>Avg Key Size</td>
<td>The index key size, calculated from the average column length of all columns participating in the index.</td>
</tr>
<tr>
<td>Leaf</td>
<td>The total number of index leaf pages.</td>
</tr>
<tr>
<td>Pseudo Empty Leafs</td>
<td>The number of pseudo empty index leaf pages. A pseudo empty index leaf page is a page on which all the RIDs are marked as deleted, but have not been physically removed.</td>
</tr>
<tr>
<td>Pseudo Deleted RIDs</td>
<td>A RID that is marked deleted. This statistic reports pseudo deletes RIDs on leaf pages that are not pseudo empty. It does not include RIDs marked as deleted on leaf pages where all the RIDs are marked deleted.</td>
</tr>
<tr>
<td>Levels</td>
<td>The number of index levels.</td>
</tr>
<tr>
<td>Unique Entries</td>
<td>The number of unique index entries that are not marked deleted.</td>
</tr>
<tr>
<td>Page Size</td>
<td>The page size of the table space in which the table indexes reside, specified at the time of table creation.</td>
</tr>
<tr>
<td>% Free</td>
<td>Specifies the percentage of each index page to leave as free space, a value that is assigned when defining the index.</td>
</tr>
<tr>
<td>Cluster Ratio</td>
<td>The clustering ratio of an index should be greater than 80 percent. When multiple indexes are defined on one table, some of these indexes have a low cluster ratio. (The index sequence is not the same as the table sequence.) This cannot be avoided. Be sure to specify the most important index when reorganizing the table. The cluster ratio is usually not optimal for indexes that contain many duplicate keys and many entries.</td>
</tr>
<tr>
<td>Used Space%</td>
<td>Less than 50 percent of the space reserved for index entries should be empty.</td>
</tr>
<tr>
<td>Balance</td>
<td>To determine if recreating the index would result in a tree having fewer levels. This formula checks the ratio between the amount of space in an index tree that has one less level than the current tree, and the amount of space needed. If a tree with one less level could be created and still leave PCTFREE available, then a reorganization is recommended. The actual number of index entries should be more than 90% (or 100-PCTFREE) of the number of entries an NLEVELS-1 index tree can handle (only checked if NLEVELS&gt;1).</td>
</tr>
<tr>
<td>Deleted RIDs</td>
<td>The number of pseudo-deleted RIDs on non-pseudo-empty pages should be less than 20 percent.</td>
</tr>
</tbody>
</table>
Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

**NOTE:** Functionality differ by platform.

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Analyze</th>
<th>DBCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop Index</td>
<td>Extract</td>
<td>Estimate Size</td>
</tr>
<tr>
<td>Map</td>
<td>Index Wizard for IBM DB2 for Linux, Unix, and Windows</td>
<td>Open Editor</td>
</tr>
<tr>
<td>Open Wizard</td>
<td>Place</td>
<td>Quick Reorg</td>
</tr>
<tr>
<td>Reorganize</td>
<td>Threshold Options</td>
<td>Update Statistics</td>
</tr>
</tbody>
</table>

For more information, see:

- [Finding Space-Related Table Problems](#)
- [Space Analyst Tabs](#)

**Space Analyst for Oracle - Indexes Tab**

The Indexes Tab displays diagnostic information regarding index space inefficiencies that may exist in your database.

The Index Reorganization Candidate Summary section uses a bar chart to count the various types of index space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the three distinct types of table space inefficiencies that Space Analyst computes:
The Index Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected indexes. The grid contains the following data:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLevel</td>
<td>These indexes have extended beyond the allowable B-level size set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Indexes with extended levels can take longer to scan and can therefore contribute to slower response times.</td>
</tr>
<tr>
<td>Next Extent Problems</td>
<td>These indexes cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Indexes suffering from this problem may cause space-related errors if more data is added to their parent table. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.</td>
</tr>
<tr>
<td>Max Extent Problems</td>
<td>These indexes are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Indexes suffering from this problem may cause space-related errors to result if more data is added to their parent table.</td>
</tr>
</tbody>
</table>

The Index Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected indexes. The grid contains the following data:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Owner</td>
<td>The owner of the index.</td>
</tr>
<tr>
<td>Index Name</td>
<td>The name of the index.</td>
</tr>
<tr>
<td>Index Type</td>
<td>Indicates if the type of index (index, index partition, etc.)</td>
</tr>
<tr>
<td>Index (KB)</td>
<td>The total allocated space of the index in KB.</td>
</tr>
<tr>
<td>Number of Rows</td>
<td>The number of rows that the parent table contains.</td>
</tr>
<tr>
<td>Clustering Factor</td>
<td>Indicates the amount of order of the rows in the table based on the values of the index. If the value is near the number of blocks, then the table is very well ordered. If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.</td>
</tr>
<tr>
<td>Index Blevel</td>
<td>Indicates the level of the index. Numbers beyond 3-4 can indicate a fragmented index.</td>
</tr>
<tr>
<td>Number of Blocks</td>
<td>The total number of blocks allocated to the index. Note: this column may report NULL if the index has not been analyzed via the ANALYZE command or through the DBMS_STATS package.</td>
</tr>
<tr>
<td>Max Extent Percent</td>
<td>A percentage indicating how close an object is to reaching its maximum extent limit. High percentages indicate indexes that may be candidates for reorganization.</td>
</tr>
<tr>
<td>Number of Extents</td>
<td>The number of extents that the index consumes.</td>
</tr>
<tr>
<td>Max Extents</td>
<td>The maximum extent limit for the index.</td>
</tr>
<tr>
<td>Can Extent Space</td>
<td>Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.</td>
</tr>
<tr>
<td>Next Extent</td>
<td>The next extent (in bytes) that the index will attempt to allocate when more space for the index is required.</td>
</tr>
<tr>
<td>Max Free Space</td>
<td>The largest contiguous block of free space that exists in the tablespace where the index resides.</td>
</tr>
<tr>
<td>Tablespace Name</td>
<td>The tablespace that contains the index.</td>
</tr>
</tbody>
</table>
NOTE: The grid can also contain a column for deleted leaf rows if the Get Deleted Leaf Row Statistics option has been selected. Be aware that the calculation of this statistic can take a lot of time for databases with large numbers of indexes. Large numbers of deleted leaf rows can indicate an index in need of reorganization.

Available Functionality
The following functionality is available in the Command menu, shortcut menu, and toolbar:

NOTE: Functionality differ by platform.

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Analyze</th>
<th>DBCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop Index</td>
<td>Extract</td>
<td>Estimate Size</td>
</tr>
<tr>
<td>Map</td>
<td>Index Wizard for IBM DB2 for Linux, Unix, and Windows</td>
<td>Open Editor</td>
</tr>
<tr>
<td>Open Wizard</td>
<td>Place</td>
<td>Quick Reorg</td>
</tr>
<tr>
<td>Reorganize</td>
<td>Threshold Options</td>
<td>Update Statistics</td>
</tr>
</tbody>
</table>

For more information, see Space Analyst Tabs.

Space Analyst - Storage Overview Tab
The Storage Overview Tab displays an overview of the storage/space situation on the target server:

The table below describes the sections on the Storage Overview Tab of Space Analyst for Microsoft:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Overview</td>
<td>Overview of total database and log space allocated on the SQL Server, as well as a count of databases, files, and file groups.</td>
</tr>
<tr>
<td>Log Overview</td>
<td>Contains a graphical and grid control that works in the following way:</td>
</tr>
<tr>
<td></td>
<td>- When the graph is displayed, a bar chart is shown that lists the total percent used of all logs on the server.</td>
</tr>
<tr>
<td></td>
<td>- When the grid is displayed, detailed storage information for each database log is presented.</td>
</tr>
<tr>
<td>Database Overview</td>
<td>Contains a graphical and grid control that works in the following way:</td>
</tr>
<tr>
<td></td>
<td>- When the graph is displayed, a bar chart is shown that lists the total space usage of all databases on the server.</td>
</tr>
<tr>
<td></td>
<td>- When the grid is displayed, detailed storage information for each database is presented.</td>
</tr>
<tr>
<td>Server Disk Overview</td>
<td>Contains an overview of total free space, by physical server drive, of the server that houses SQL Server.</td>
</tr>
</tbody>
</table>

The table below describes the sections on the Storage Overview Tab of Space Analyst for Sybase:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Overview</td>
<td>Overview of total database and log space allocated on the Sybase Server, as well as a count of databases and devices.</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Log Overview</td>
<td>Contains a graphical and grid control that works in the following way:</td>
</tr>
<tr>
<td></td>
<td>- When the graph is displayed, a bar chart is shown that lists the total percent used of all logs on the server.</td>
</tr>
<tr>
<td></td>
<td>- When the grid is displayed, detailed storage information for each database log is presented.</td>
</tr>
<tr>
<td></td>
<td>Database name</td>
</tr>
<tr>
<td></td>
<td>Total log space</td>
</tr>
<tr>
<td></td>
<td>Free log space</td>
</tr>
<tr>
<td></td>
<td>Used log space</td>
</tr>
<tr>
<td></td>
<td>Log percent free</td>
</tr>
<tr>
<td>Database Overview</td>
<td>Contains a graphical and grid control that works in the following way:</td>
</tr>
<tr>
<td></td>
<td>- When the graph is displayed, a bar chart is shown that lists the total space usage of all databases on the server.</td>
</tr>
<tr>
<td></td>
<td>- When the grid is displayed, detailed storage information for each database is presented:</td>
</tr>
<tr>
<td></td>
<td>Database name</td>
</tr>
<tr>
<td></td>
<td>Total database space</td>
</tr>
<tr>
<td></td>
<td>Database free space</td>
</tr>
<tr>
<td></td>
<td>Database used space</td>
</tr>
<tr>
<td></td>
<td>Database percent free</td>
</tr>
<tr>
<td>Device Overview</td>
<td>Contains a graphical and grid control that works in the following way:</td>
</tr>
<tr>
<td></td>
<td>- When the graph is displayed, a bar chart is shown that lists the total space used (in MB) of all devices defined to the server.</td>
</tr>
<tr>
<td></td>
<td>- When the grid is displayed, detailed storage information for each database device is presented:</td>
</tr>
<tr>
<td></td>
<td>Device name</td>
</tr>
<tr>
<td></td>
<td>Physical name</td>
</tr>
<tr>
<td></td>
<td>Total space</td>
</tr>
<tr>
<td></td>
<td>Used space</td>
</tr>
<tr>
<td></td>
<td>Free space</td>
</tr>
<tr>
<td></td>
<td>Percent free</td>
</tr>
<tr>
<td></td>
<td>Percent used</td>
</tr>
</tbody>
</table>

Available Functionality
The following functionality is available in the Command menu, shortcut menu, and toolbar for Microsoft:

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Extract</th>
<th>Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Wizard</td>
<td>Object Details</td>
<td>Open Editor</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Wizard</td>
<td>Quick Space Tasks</td>
<td>Shrink Database</td>
</tr>
<tr>
<td>Threshold Options</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Available Functionality
The following functionality is available in the Command menu, shortcut menu, and toolbar for Sybase:

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Extract</th>
<th>Index Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Space Analyst for Microsoft - File Groups/Files Tab
The Files/File Groups Tab displays a graph and grid for filegroups on the server and a grid for files.

**TIP:** You can toggle between the graphs and grid views.

The File Groups/Files Tab reports space usage on SQL Server by file group and by file.

When the File Groups radio button is pressed, Space Analyst presents a graphical layout of file group space usage across all databases. For large servers, not all file groups will initially be shown, but the graph is scrollable so you can click the graph and scroll to see all file groups on the chart. When in grid mode, more detailed information is shown for each file group including its auto-growth properties, how many files make up the group, the amounts (in MB) taken up by tables and indexes, and more.

When the Files radio button is pressed, detailed information is displayed for each individual file for every database on the server.

**Available Functionality**
The following functionality is available in the Command menu, shortcut menu, and toolbar:

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Extract</th>
<th>Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Wizard for Microsoft SQL Server</td>
<td>Object Details</td>
<td>Open Editor</td>
</tr>
<tr>
<td>Open Wizard</td>
<td>Quick Space Tasks</td>
<td>Shrink Database</td>
</tr>
</tbody>
</table>

For more information, see Space Analyst Tabs.

**Space Analyst for Microsoft - Map Tab**
The Map Tab displays a graphical file map that displays extent placement for objects within database files.

The Map Tab lets you view a graphical layout regarding the physical placement of every object inside a selected database and file. The map helps identify allocated and unused extents throughout a database file. An object grid is also present that lets you view detailed information about every object's storage usage in that file. Keep in mind that an object can span more than one file.

The Space Info section at the top contains information regarding the total number of extents allocated to a database file, a count of extents that contain table data, a count of extents that contain index data, and a count of extents that contain text data.

Databases are comprised of logical pages, each of which is a fixed 8KB in size. Note that the 8KB page is also a unit of I/O, which affects performance and locking. Extents are collections of database pages. An extent is made up of eight, 8K pages and is therefore 64KB in size. Extents are the graphical "blocks" displayed in the map.
Extents come in two flavors - uniform and mixed. Uniform extents ascribe all eight database pages to a single object, whereas mixed extents have the potential to hold multiple objects. With mixed extents, SQL Server tries to preserve space by not letting very small objects take up an entire extent and therefore waste a lot of space. By placing your mouse pointer over an extent “block” in the map, Space Analyst will present a pop-up dialog displaying what object(s) is/are contained in that extent.

The table below describes the options and functionality on the Map Tab of Space Analyst:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Info</td>
<td>Displays total extent count, extents with tables count, extents with indexes count, and extents with text count. Note that an extent can contain more than one object and more than one object type.</td>
</tr>
<tr>
<td>Map</td>
<td>Displays object extent data from beginning to end of the database or file. All blocks in the map are the same size. A Microsoft SQL Server extent is 64KB in size or 8, 8K pages. Purple - Indicate extents that have been allocated to objects (used), and extents that are free. Green - Free extents.</td>
</tr>
<tr>
<td>Owner</td>
<td>Displays table owner.</td>
</tr>
<tr>
<td>Table name</td>
<td>Displays table name.</td>
</tr>
<tr>
<td>Object type</td>
<td>Displays object type.</td>
</tr>
<tr>
<td>Index name</td>
<td>Displays index name.</td>
</tr>
<tr>
<td>Pages used</td>
<td>Displays pages used.</td>
</tr>
<tr>
<td>Pages allocated</td>
<td>Displays pages allocated.</td>
</tr>
<tr>
<td>In extents</td>
<td>How many extents each object is in.</td>
</tr>
<tr>
<td>Table id</td>
<td>Displays table id.</td>
</tr>
<tr>
<td>Index id</td>
<td>Displays index id.</td>
</tr>
</tbody>
</table>

The Tablespace Map Tab includes the following buttons:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legend</td>
<td>Opens the Tablespace Map Legend.</td>
</tr>
<tr>
<td>Display</td>
<td>Opens the Display window. The Display window lets you specifies the magnification level for the Tablespace Map.</td>
</tr>
<tr>
<td>Fit to Window</td>
<td>Fits the Tablespace Map to the window.</td>
</tr>
<tr>
<td>Full Screen</td>
<td>Fits the Tablespace Map to the full screen.</td>
</tr>
<tr>
<td>Print Map</td>
<td>Opens the Print dialog box to print the Tablespace Map.</td>
</tr>
</tbody>
</table>

Available Functionality
The following functionality is available in the Command menu, shortcut menu, and toolbar:

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Extract</th>
<th>Database Wizard for Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Details</td>
<td>Open Editor</td>
<td>Open Wizard</td>
</tr>
<tr>
<td>Quick Space Tasks</td>
<td>Shrink Database</td>
<td>Threshold Options</td>
</tr>
</tbody>
</table>

For more information, see Space Analyst Tabs.
Space Analyst for Microsoft - Databases Tab

The Databases Tab lets you analyze databases.

**TIP:** You can toggle between the graphs and grid views.

The Databases Tab displays information regarding space usage across all databases, potential space-related bottlenecks that are threatening databases, and recent database backup information.

When in graph form, the Database Summary section displays a graphical view of table, index, and free space usage (in MB) across all databases. For large servers, not all databases will initially be shown, but the graph is scrollable so you can click the graph and scroll to see all databases on the chart. When in grid form, the Database Summary section displays much more detail regarding database and log usage, object counts, and more.

The Database Bottlenecks section provides a graphical count of various space-related bottlenecks that are threatening the availability of your databases (some of which are controlled by Space Analyst's customizable thresholds). When in grid form, each database is listed along with information that relays whether or not a potential bottleneck exists for that database.

The Database Backups section details the most recent backup for each database and the size (in MB) of the backup.

**Available Functionality**

The following functionality is available in the Command menu, shortcut menu, and toolbar:

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Extract</th>
<th>Map</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database Wizard for Microsoft SQL Server</strong></td>
<td>Object Details</td>
<td>Open Editor</td>
</tr>
<tr>
<td>Open Wizard</td>
<td>Quick Space Tasks</td>
<td>Shrink Database</td>
</tr>
<tr>
<td>Threshold Options</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For more information, see [Space Analyst Tabs](#).

---

Space Analyst for Microsoft - Objects Tab

The Objects Tab displays database objects, including tables and indexes.

You can choose whether to view table or index information by clicking either the Tables or Indexes ratio buttons. You can also choose whether to view general space information for tables and indexes or reorganization metrics by selecting the General Space or Space Diagnostics radio buttons.

**NOTE:** To obtain space diagnostic information for each object, Space Analyst uses the SQL Server DBCC SHOWCONTIG command. This command may take a while to run on large databases. Also, the command is not as efficient in terms of performance on version 7.0 of SQL Server as it is on SQL Server 2000, so understand that running space diagnostics on most versions of SQL Server will take some time.

The **Object summary** presents one of two views:

- When the **General Space** radio button is pressed, a graph is presented that visually displays that amount of allocated space (in KB) for tables, indexes, and free space. For large servers, not all objects for a database will initially be visible on the chart, but the graph is scrollable so you can click the graph and scroll to see all database objects on the chart.

- When the **Space Diagnostics** radio button is pressed, Space Analyst presents a count of all objects (tables or indexes) that exceed your customizable thresholds for various forms of fragmentation of other space-related inefficiency. See below for a thorough discussion of these types of space problems.
The **Object Details** grid presents one of two views:

- **When the General Space radio button is pressed**, general space usage for tables or indexes is shown. Base object metadata (object name, type, partitions, etc.) is shown along with the amounts of total, used, and free space (in KB) for the object. The percent of space used for the object is also shown along with the percent of space in the database that the object consumes.

- **When the Space Diagnostics radio button is pressed**, Space Analyst presents fragmentation and other space-related inefficiency information for all objects in the selected database. Note that definitions for all the various forms of object fragmentation and space inefficiencies follow later in the help.

The table below describes the options and functionality on the Objects Tab of Space Analyst:

<table>
<thead>
<tr>
<th>View</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Lets you switch between databases.</td>
</tr>
<tr>
<td>Tables</td>
<td>Lets you switch between viewing space for tables and indexes. Displays object details when you select.</td>
</tr>
<tr>
<td>Indexes</td>
<td>Lets you switch between viewing space for tables and indexes. Displays object details when you select.</td>
</tr>
<tr>
<td>General Space View</td>
<td>Displays general space information for tables and indexes in the selected database. Displays object details when you select.</td>
</tr>
<tr>
<td>Space Diagnostic View</td>
<td>Displays fragmentation and space information to help the DBA decide if a reorg is necessary. Displays object details when you select.</td>
</tr>
<tr>
<td>File Group</td>
<td>Lets you view object data for all filegroups for the database or only a selected filegroup.</td>
</tr>
<tr>
<td>Owner</td>
<td>Lets you view object data for all table owners or only a selected owner.</td>
</tr>
<tr>
<td>Display</td>
<td>Lets you view space diagnostic information for tables or indexes.</td>
</tr>
</tbody>
</table>

**Available Functionality**

The following functionality is available in the Command menu, shortcut menu, and toolbar:

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Extract</th>
<th>Database Wizard for Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Editor</td>
<td>Open Wizard</td>
<td>Quick Space Tasks</td>
</tr>
<tr>
<td>Threshold Options</td>
<td>Update Statistics</td>
<td></td>
</tr>
</tbody>
</table>

For more information, see [Space Analyst Tabs](#).

**Space Analyst for Sybase ASE - Database Devices Tab**

The Database Devices Tab displays database device information.

**TIP:** You can toggle between graph and grid views.
The table below describes the sections on the Database Devices Tab of Space Analyst:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Summary</td>
<td>Contains summary information for all Sybase database devices. Displays the amount of data, log, data and log, and free space amounts (in MB).</td>
</tr>
<tr>
<td>Device Details</td>
<td>Displays detailed information for selected devices. Lets you click in a bar chart and populate the lower section of the tab with that device's information.</td>
</tr>
<tr>
<td>Database Fragments</td>
<td>Displays database fragments.</td>
</tr>
</tbody>
</table>

Available Functionality
The following functionality is available in the Command menu, shortcut menu, and toolbar:

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Disk Resize</th>
<th>Extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Details</td>
<td>Move Log</td>
<td>Database Wizard for Sybase ASE</td>
</tr>
<tr>
<td>Database Device Wizard for Sybase ASE</td>
<td>Segment Wizard for Sybase ASE</td>
<td>Open Editor</td>
</tr>
<tr>
<td>Open Wizard</td>
<td>Quick Space Tasks</td>
<td>Reinit</td>
</tr>
<tr>
<td>Refit</td>
<td>Table Details</td>
<td>Threshold Options</td>
</tr>
</tbody>
</table>

For more information, see Space Analyst Tabs.

Space Analyst for Sybase ASE - Databases Tab
The Databases Tab lets you analyze databases.

TIP: You can toggle between graph and grid views.

The table below describes the sections on the Databases Tab of Space Analyst:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Summary</td>
<td>Contains chart that displays database used and free space, along with log used and free space information for databases. The summary section totals up the total database, log, table, and index space. Also displays a summary count of all user tables and indexes.</td>
</tr>
<tr>
<td>Database Details</td>
<td>Displays detailed information for selected databases. Lets you select whether the graph shows space utilization or object fragmentation scores.</td>
</tr>
</tbody>
</table>

Available Functionality
The following functionality is available in the Command menu, shortcut menu, and toolbar:

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Disk Resize</th>
<th>Extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Details</td>
<td>Move Log</td>
<td>Database Wizard for Sybase ASE</td>
</tr>
<tr>
<td>Database Device Wizard for Sybase ASE</td>
<td>Segment Wizard for Sybase ASE</td>
<td>Open Editor</td>
</tr>
<tr>
<td>Open Wizard</td>
<td>Quick Space Tasks</td>
<td>Reinit</td>
</tr>
<tr>
<td>Refit</td>
<td>Table Details</td>
<td>Threshold Options</td>
</tr>
</tbody>
</table>
For more information, see Space Analyst Tabs.

Space Analyst for IBM DB2 for Linux, Unix, and Windows - Tablespace Tab
The Tablespace Tab lets displays tablespace space allocation and tables.

Available Functionality
The following functionality is available in the Command menu, shortcut menu, and toolbar:

NOTE: Functionality differ by platform.

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Extract</th>
<th>Open Editor - see Object Editors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Wizard</td>
<td>Reorganize</td>
<td>Update Statistics</td>
</tr>
<tr>
<td>Tablespace Wizard for IBM DB2 for Linux, Unix, and Windows</td>
<td>Threshold Options</td>
<td></td>
</tr>
</tbody>
</table>

For more information, see Space Analyst Tabs.

Space Analyst for Functionality
Space Analyst offers functionality that help you analyze space. The Space Analyst support the following functionality:

NOTE: Functionality differ by platform.

<table>
<thead>
<tr>
<th>Analysis Report</th>
<th>Convert to Dictionary Managed</th>
<th>Convert to Locally Managed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Resize</td>
<td>DBCC</td>
<td>Drop Index</td>
</tr>
<tr>
<td>Estimate Size</td>
<td>Extract</td>
<td>Index Details</td>
</tr>
<tr>
<td>Map</td>
<td>Move Log</td>
<td>Database Wizard for Sybase ASE</td>
</tr>
<tr>
<td>Database Device Wizard for Sybase ASE</td>
<td>Segment Wizard for Sybase ASE</td>
<td>Table Wizard for IBM DB2 for Linux, Unix, and Windows</td>
</tr>
<tr>
<td>Tablespace Wizard for IBM DB2 for Linux, Unix, and Windows</td>
<td>Open Editor</td>
<td>Open Wizard</td>
</tr>
<tr>
<td>Options</td>
<td>Place</td>
<td>Quick Reorg</td>
</tr>
<tr>
<td>Quick Space Tasks</td>
<td>Refit</td>
<td>Reinit</td>
</tr>
<tr>
<td>Reorganize</td>
<td>Shrink Database</td>
<td>Table Details</td>
</tr>
<tr>
<td>Update Statistics</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Space Analyst - Analysis Report
The Analysis Report dialog box lets you build an HTML analysis report for the selected tablespace. You can run reports on a recurring basis and dynamically alter any saved thresholds that flag space problems in the tablespace being analyzed.

For Oracle, the report includes sections for particular reorganization needs, including:

- Fragmentation index for the tablespace
• Tables with high block inefficiencies
• Tables or indexes with next extent space deficit problems
• Tables or indexes with maximum extent limit problems
• Tables or indexes with percent increase greater than zero
• Tables with chained/migrated row problems
• Indexes with B-tree depth problems

For Microsoft, the report contains different sections that highlight particular reorganization needs, including:

• Free space levels for the database and log
• Tables with forwarded record problems
• Indexes with average fragmentation problems
• Tables or indexes with extent fragmentation problems
• Tables or indexes with logical fragmentation problems
• Tables or indexes with scan density problems
• Tables or indexes with average page density problems

For Sybase, the report contains different sections that highlight particular reorganization needs, including:

• Free space levels for the database and log
• Tables with forwarded row problems
• Tables with empty page problems
• Tables with deleted row problems
• Tables with low fragmentation scores
• Indexes with unused space problems
• Indexes with empty leaf problems
• Indexes with height/level problems
• Indexes with low fragmentation scores

For IBM DB2 for Linux, Unix, and Windows, the report contains different sections that highlight particular reorganization needs, including:

• Tables with overflow problems
• Tables with space utilization problems
• Tables with page utilization problems
• Tables with index space contention
• Indexes with cluster ratio problems
• Indexes with used space% problems
• Indexes with balance problems
• Indexes with deleted RID problems
• Indexes with empty leaf problems
• Indexes with low uniqueness
• Free levels of tablespace and log space

To open the Analysis Report dialog box:
1 Open the Space Analyst.
2 On the Command menu, click the Analysis Report.

For all platforms, browse to or type a report title and make your desired selections from the drop-down lists. After the report executes, you can print it or save it.

To print the report:
1 Click Print from the Analysis Report toolbar.

To save the report:
1 Select Save from the File menu on the main toolbar.

Space Analyst - Convert to Locally-Managed

NOTE: The Convert to Locally-Managed dialog box is for Oracle 8.1.6 or later.

The Convert to Locally-Managed dialog box lets you convert a dictionary-managed tablespace to locally-managed. Objects in locally-managed tablespaces have unlimited extents; they will never fail to extend due to an imposed maximum extent limit.

Important Notes
There are a few restrictions on these conversion procedures (for example, Oracle9i UNDO tablespaces currently can't be converted, etc.), so you should check the Oracle documentation for the specifics on performing a conversion.

The Convert to Locally-Managed option is only available if the selected tablespace(s) are:

• ONLINE
• In Read/Write mode
• Dictionary managed
• Not UNDO tablespaces
• Not Temporary tablespaces
• Not the SYSTEM tablespace

The table below describes the options and functionality on the Convert to Locally-Managed dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespaces to be Converted</td>
<td>Displays the tablespaces to be converted to locally-managed.</td>
</tr>
<tr>
<td>Conversion Options</td>
<td>OPTIONAL: Lets you type allocation unit and relative file number information.</td>
</tr>
</tbody>
</table>
Completing the Convert to Locally-Managed Dialog Box for Space Analyst

To complete this dialog box, do the following:

1. Select **Convert to Locally-Managed**.

   OR

   Right-click the target object, and then select **Convert to Locally-Managed**.

   The Space Analyst opens the dialog box.

2. Do one of the following:

   For more information, see [Convert to Locally-Managed](#).

   **NOTE:** The Convert to Dictionary-Managed functionality is for Oracle 9i only.

   The Convert to Dictionary-Managed dialog box lets you convert a locally-managed tablespace to dictionary.

   **Important Notes**
   The Convert to Dictionary-Managed dialog box is only available if the selected tablespace(s) are:

   - ONLINE
   - In Read/Write mode
   - Locally managed
   - Not UNDO tablespaces
   - Not Temporary tablespaces
   - Not the SYSTEM tablespace

   The table below describes the options and functionality on the Convert to Dictionary-Managed dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespaces to be Converted</td>
<td>Displays the tablespaces to be converted to dictionary-managed.</td>
</tr>
</tbody>
</table>
Completing the Dictionary-Managed Dialog Box for Space Analyst
To complete this dialog box, do the following:

1. Select **Convert to Dictionary-Managed**.
   
   OR
   
   Right-click the target object, and then select **Convert to Dictionary-Managed**.
   
   The Space Analyst opens the dialog box.

2. Do one of the following:
   - **Open the Preview dialog box**.
   - **Open a scheduling application**.
   - **Execute the task**.

For more information, see [Convert to Dictionary-Managed](#).

### Space Analyst - Extract
The Extract functionality lets you extract the database DDL in an ISQL editor.

To extract the database DDL in an ISQL editor, do the following:

1. Click the **Extract DDL** button.
   
   OR
   
   Select **Extract**.
   
   OR
   
   Right-click the target object, and then select **Extract**.
   
   Space Analyst extract the database DDL in an ISQL editor.

### Space Analyst - Map
The Map functionality opens the [Tablespace Map Tab](#) or [Map Tab](#) of the Space Analyst for the selected index.

### Important Notes
None

For more information, see [Opening the Tablespace Map Tab](#).
Opening the Tablespace Map Tab for Space Analyst

To open the Tablespace Map or Map Tab, do the following:

1. Select **Map**.

   OR

   Right-click the target object, and then select **Map**.

Space Analyst opens the Tablespace Map Tab or Map Tab.

For more information, see **Map**.

---

**Space Analyst - Open Editor**

The Open functionality lets you modify objects in the object editors.

To open the object editor, do the following:

1. In any grid, select an object.

2. Click the **Open Editor** button.

   OR

   Select **Open**.

   OR

   Right-click the target object, and then select **Open**.

Space Analyst opens the object editor.

---

**Space Management Wizard**

The Space Management Wizard can create two specialized space management jobs that allow you to proactively eliminate space threats before they affect your database:

You can create a "reorganization analyst job" that allows you to specifically select various databases and/or objects and then set reorganization thresholds that determine when a reorganization will occur. Once constructed, you can schedule these jobs to run as often as you’d like. During execution, these jobs will interrogate your chosen database/objects and determine if a reorganization needs to occur by applying your customized reorganization thresholds. If any objects are found to need reorganization, they are dynamically reorganized using any customized options you specify.

You can build space maintenance jobs that will perform functions such as check the validity of your databases and filegroups, repair any found minor corruption, and eliminate unused space in your databases. You can also set the job up to notify you if any problems are found during job execution.

---

**Creating or Editing a Space Management Job**

The Space Management Wizard lets you create space reorganization or maintenance jobs without knowing the code. After creation, jobs can be reused as is, or edited.
To open the Space Management Wizard

1. Start the Space Analyst and click the **Open Wizard** button on the Space Analyst toolbar.

   OR

   Select **Open Wizard** from the Space Analyst menu.

The sections below display the fields you may find creating or editing a space management job. Be aware that the sections you see, and the order in which they appear in the wizard vary depending on the selections you make and the database platform.

<table>
<thead>
<tr>
<th>New or Used Task</th>
<th>Type of Space Management Task</th>
<th>Database/Objects to Be Part of the Reorganization or Maintenance Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Reorganization</td>
<td>Save the Operation Executive Preferences for a Reorganization Operation</td>
<td>Global Customizations</td>
</tr>
<tr>
<td>Specific Object Customization</td>
<td>Threshold Criteria for Object Reorganization</td>
<td>Integrity Check Options</td>
</tr>
<tr>
<td>Integrity Check for Specific Database, Objects</td>
<td>Unused Space Options</td>
<td>Problem Notification</td>
</tr>
</tbody>
</table>

**New or Used Task**

Here you choose whether to create a new space management task or whether to use a previously saved task. If you want to reuse a previously created job, you can type or browse to find the previously saved task definition file (*.sptsk).

**Type of Space Management Task**

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create standard reorganization job</td>
<td>Select to build a reorganization job for a database or the objects you specify.</td>
</tr>
<tr>
<td>Create reorganization analyst job</td>
<td>Select to create and deploy a reorganization job that runs periodically to detect and automatically correct space inefficiencies.</td>
</tr>
<tr>
<td>Create space maintenance job</td>
<td>Select to build a space maintenance job that validates database structures and performs storage maintenance tasks.</td>
</tr>
</tbody>
</table>

**Database/Objects to Be Part of the Reorganization or Maintenance Task**

Choose the databases and/or objects you want to include in the job you are creating. Simply move items from the tree in the right pane to the left pane by clicking the right arrow. You can amend your selections using the left arrow. The selection is limited to entire databases for a space maintenance job.
Custom Reorganization

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform no customizations</td>
<td>When you choose this option, all default object and database attributes are used.</td>
</tr>
<tr>
<td>Perform global customizations</td>
<td>This option gives you the chance to define options that will be globally applied to all objects being reorganized.</td>
</tr>
<tr>
<td>Perform specific object customizations</td>
<td>Here you individualize options for specific objects being reorganized.</td>
</tr>
</tbody>
</table>

Save the Operation

If you want to save the operation, you can use the default name supplied or type another in the upper panel. If you don't want to save the operation, remove the default name. Browse to the location where you want to save the file. The job as it was created appears in the read-only preview panel. To make changes, you must go back and change your selections or open the job in an ISQL window prior to execution.

Execution Preferences for a Reorganization Operation

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open the reorganization script in an ISQL editor.</td>
<td>The reorganization script will be generated and then opened in a new ISQL editing session.</td>
</tr>
<tr>
<td>Perform reorganization now</td>
<td>The reorganization is run immediately with results presented afterward in an ISQL window.</td>
</tr>
<tr>
<td>Schedule the reorganization script for later execution.</td>
<td>The reorganization script will be generated and then scheduled to run at a specified date/time.</td>
</tr>
</tbody>
</table>
## Global Customizations

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relocate to file group</td>
<td>Select to perform an extended alter and move the table(s) to the requested file group using the ON keyword in the table creation DDL.</td>
</tr>
<tr>
<td>Perform sorted reorg using table’s clustered index</td>
<td>Select to rebuild of any table’s clustered index.</td>
</tr>
<tr>
<td>Perform online reorganization where possible</td>
<td>MICROSOFT SQL SERVER 2000 OR LATER ONLY: Select to use the DBCC INDEXDEFRAG command to perform an online rebuild of every table’s index(es).</td>
</tr>
<tr>
<td>View reorganization output messages</td>
<td>Select for the WITH NO_INFOMSGS clause of any DBCC command to not be used.</td>
</tr>
<tr>
<td>Defragment tables in DROP/CREATE fashion</td>
<td>Select to defragment tables in DROP/CREATE fashion.</td>
</tr>
<tr>
<td>Relocate to file group</td>
<td>Select for DROP_EXISTING clause instead of doing a drop and then create.</td>
</tr>
<tr>
<td>Fill Factor</td>
<td>Select to specify a new fill factor for the DBCC DBREINDEX command or a create/drop of an existing index.</td>
</tr>
<tr>
<td>Index Reorganization</td>
<td>Rebuild: This is the default (ALTER INDEX...REBUILD for MS SQL 2005; MS SQL Server 2000 and 2005 display DBCC DBREINDEX). Drop/Create For MS SQL Server 2005 or later: Reorganize</td>
</tr>
<tr>
<td>Perform online reorganization where possible</td>
<td>MICROSOFT SQL SERVER 2000 OR LATER ONLY: Select to use the DBCC INDEXDEFRAG command to perform an online rebuild of every index.</td>
</tr>
<tr>
<td>View reorganization output messages</td>
<td>Select for the WITH NO_INFOMSGS clause of any DBCC command to not be used.</td>
</tr>
</tbody>
</table>

## Specific Object Customization

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select</td>
<td>Drop-down list lets you switch between indexes and tables and see the particular objects for each.</td>
</tr>
<tr>
<td>Relocate to file group</td>
<td>Select a clause instead of doing a drop and then create.</td>
</tr>
<tr>
<td>Fill Factor</td>
<td>For indexes, select to specify a new fill factor.</td>
</tr>
<tr>
<td>Perform sorted reorganization using table’s clustered index</td>
<td>For tables, select to perform sorted reorganization using table’s clustered index.</td>
</tr>
<tr>
<td>Index reorganization</td>
<td>Rebuild: This is the default. Drop/Create Reorganize</td>
</tr>
<tr>
<td>Perform online reorganization where possible</td>
<td>MICROSOFT SQL SERVER 2000 OR LATER ONLY: Select to perform an online rebuild of every index.</td>
</tr>
<tr>
<td>View reorganization output messages</td>
<td>Select for the WITH NO_INFOMSGS clause of any DBCC command to not be used.</td>
</tr>
<tr>
<td>Defragment tables in DROP/CREATE fashion</td>
<td>For tables, select to defragment tables in DROP/CREATE fashion.</td>
</tr>
</tbody>
</table>
Threshold Criteria for Object Reorganization

**NOTE:** The table combines criteria for MS SQL Server 2000 and 2005.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table:</strong> Average Page Density</td>
<td>Default is less than or equal to 50.</td>
</tr>
<tr>
<td>Scan Density Percent</td>
<td>Default is less than or equal to 0.</td>
</tr>
<tr>
<td>Logical Fragmentation</td>
<td>Greater than or equal to 25 is the default value.</td>
</tr>
<tr>
<td>Extent Fragmentation</td>
<td>Default is greater than or equal to 50.</td>
</tr>
<tr>
<td>Forwarded Record Percent</td>
<td>Default is greater than or equal to 50.</td>
</tr>
<tr>
<td><strong>Index:</strong> Average Fragmentation</td>
<td>Default is greater than or equal to 25.</td>
</tr>
<tr>
<td>Scan Density Percent</td>
<td>Default is less than or equal to 0.</td>
</tr>
<tr>
<td>Logical Fragmentation</td>
<td>Greater than or equal to 25 is the default value.</td>
</tr>
<tr>
<td>Extent Fragmentation</td>
<td>Default is greater than or equal to 50.</td>
</tr>
<tr>
<td>Index Tree Depth</td>
<td>Default is greater than or equal to 3.</td>
</tr>
<tr>
<td>Average Page Density</td>
<td>Default is less than or equal to 50.</td>
</tr>
</tbody>
</table>

**NOTE:** All defaults are editable. Also, all current metrics for these criteria can be found on the Objects tab >Space Diagnostics view. The default thresholds are visible when you choose Threshold Options from the Space Analyst drop-down menu.

Integrity Check Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not perform any integrity checks</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>Check databases</td>
<td>Checks the database integrity for database(s) you identified earlier.</td>
</tr>
<tr>
<td>Check file groups</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>Check specific database objects</td>
<td>In a later step you are able to select the particular objects.</td>
</tr>
<tr>
<td>Include index checks for all tables</td>
<td>This is the default and isn’t actually optional.</td>
</tr>
<tr>
<td>Attempt to repair minor problems</td>
<td>Enabled as a default for database or database objects integrity checks. You can unselect.</td>
</tr>
<tr>
<td>Include check for database catalogs</td>
<td>Enabled as a default for database integrity checks. You can unselect.</td>
</tr>
<tr>
<td>Correct data dictionary inaccuracies</td>
<td>Enabled as a default for database integrity checks. You can unselect.</td>
</tr>
<tr>
<td>Check data purity</td>
<td>Verifies that columns only have data in the correct ranges, and so on.</td>
</tr>
</tbody>
</table>

Integrity Check for Specific Database Objects

Here you select objects for the database(s) you have included in this maintenance job.
Unused Space Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not eliminate unused space</td>
<td>Self-explanatory</td>
</tr>
<tr>
<td>Eliminate unused space for databases</td>
<td>Select to only truncate space for databases larger than a specific size (in MB).</td>
</tr>
<tr>
<td>Only eliminate space when database is larger than</td>
<td>Available if you selected Eliminate unused space for databases option. Lets you specify maximum database size in MB.</td>
</tr>
<tr>
<td>Amount of free space to leave for database</td>
<td>Lets you specify how much free space is to remain after a space truncation.</td>
</tr>
<tr>
<td>Eliminate unused space for specific database tables</td>
<td>Self-explanatory.</td>
</tr>
</tbody>
</table>

Problem Notification

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No, do not notify me</td>
<td>Select to not have Space Analyst notify you of any errors.</td>
</tr>
<tr>
<td>Yes, notify me in the following ways.</td>
<td>Select to have Space Analyst notify you of any errors and then type an e-mail or net send address.</td>
</tr>
</tbody>
</table>

Space Analyst - Threshold Options

The Space Analyst Reorganization Thresholds Options dialog box lets you customize the various reorganization thresholds that can be used to flag storage and data objects that need to be reorganized. The thresholds you specify will apply to the Space Analysis Reports.

Important Notes

None

The table below describes the options and functionality on the Space Analyst Thresholds Options dialog box for IBM DB2 for Linux, Unix, and Windows:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Reorganization Thresholds</td>
<td>The table thresholds apply to the displays used in tab 4 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the tablespace analysis report.</td>
</tr>
<tr>
<td>Index Reorganization Thresholds</td>
<td>The index thresholds apply to the displays used in tab 5 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the tablespace analysis report.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the Space Analyst Thresholds Options dialog box for Oracle:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace and Datafile Fragmentation Thresholds</td>
<td>The tablespace and datafile thresholds apply to the tablespace and datafile fragmentation index. The default for each is 50 - meaning, flag any tablespace or datafile whose fragmentation index is less than 50. They are also used for the tablespace analysis report.</td>
</tr>
<tr>
<td>Table Reorganization Thresholds</td>
<td>The table thresholds apply to the displays used in tab 4 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the tablespace analysis report.</td>
</tr>
</tbody>
</table>
The table below describes the options and functionality on the Space Analyst Thresholds Options dialog box for Microsoft SQL Server (excluding SQL Server 2005) and Sybase:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Reorganization Thresholds</td>
<td>The index thresholds apply to the displays used in tab 5 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the tablespace analysis report.</td>
</tr>
<tr>
<td><strong>Database and Log Space Thresholds</strong></td>
<td>The database and log space thresholds apply to the database and log fragmentation index. The default for each is 80 - meaning, flag any database or log whose fragmentation index is less than 80. The database and log space thresholds apply to any grid that shows free space percent for those entities.</td>
</tr>
<tr>
<td><strong>Table Reorganization Thresholds</strong></td>
<td>The table thresholds apply to the displays used in tab 5 of the Space Analyst and the procedures that are run to present the various customized displays. They are also used for the database analysis report.</td>
</tr>
<tr>
<td><strong>Index Reorganization Thresholds</strong></td>
<td>The index thresholds apply to the displays used in tab 5 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the database analysis report.</td>
</tr>
</tbody>
</table>

The table below describes the options and functionality on the Space Analyst Thresholds Options dialog box for SQL Server 2005:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database and Log Space Thresholds</td>
<td>The database and log space thresholds apply to the database and log fragmentation index. The default for each is 80 - meaning, flag any database or log whose fragmentation index is less than 80. The database and log space thresholds apply to any grid that shows free space percent for those entities.</td>
</tr>
<tr>
<td>Table Reorganization Thresholds</td>
<td>The table thresholds apply to the displays used in tab 5 of the Space Analyst and the procedures that are run to present the various customized displays. They are also used for the database analysis report.</td>
</tr>
<tr>
<td>Index Reorganization Thresholds</td>
<td>The index thresholds apply to the displays used in tab 5 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the database analysis report.</td>
</tr>
<tr>
<td>Fragmentation Analysis Mode</td>
<td>Detailed is the default. The other options are Sampled and Limited. This option serves as a parameter that's passed to the space diagnostic procedures use on the Objects tab for the reorganization summary and the detail views for indexes and tables.</td>
</tr>
</tbody>
</table>

**Completing the Space Analyst Thresholds Options Dialog Box for Space Analyst**

To complete this dialog box, do the following:

1. On the **Analysts Toolbar**, click the **Options** button.
   
   The Space Analyst opens the dialog box.

2. Review thresholds.

3. Click **OK**.
Space Analyst for Oracle Tutorial

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result from poor space management. Running inside the Embarcadero DBArtisan console, Space Analyst helps you identify and correct storage problems by supplying powerful analysis facilities and corrective utilities and that will satisfy advanced experts, yet are easy enough to use for database novices.

About This Space Analyst for Oracle Tutorial

This tutorial helps you get started with Embarcadero Space Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Space Analyst. You will have learned how to get a summary view of storage for your database as well as how to identify potential space-related threats. Finally, you’ll use the reorganization wizard to correct any space problems that you find.

This tutorial is divided into 5 sections.

Session 1: Getting Started with Space Analyst

Session 2: Obtaining a Global View of Storage

Session 3: Identifying Storage Structure Problems

Session 4: Pinpointing Object Performance Inefficiencies

Session 5: Correcting Space-Related Problems

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see Additional Evaluation Resources.

Session 1: Getting Started with Space Analyst for Oracle

Download and Install

You can obtain the latest version of Space Analyst by downloading DBArtisan from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Space Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Space Analyst.

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Space Analyst for Oracle Overview

Once you are connected to an Oracle database, you can open Space Analyst from within DBArtisan two ways:

1. On the Analyst menu, select Space Analyst.

   **NOTE:** You do not need to install anything for Space Analyst to work.
Session 2: Obtaining a Global View of Storage Using Space Analyst for Oracle

When it opens, Space Analyst shows a complete overview of the storage situation of a particular database or database server. The first tab of Space Analyst displays all pertinent global statistics regarding storage structure usage throughout a database.

There are four distinct sections displayed on the first **Database Overview Tab** of Space Analyst:

- **Database Overview**
- **Tablespace Space Overview**
- **Tablespace Fragmentation Overview**
- **Tablespace Objects Overview**

**Space Analyst for Oracle - Database Overview**

The Database Overview displays:

- The total count of all tablespaces, datafiles and temporary files.
- The percentage breakdown of the various object types in a database. Positioning the mouse pointer over the graph causes it to display the actual percentage amounts for each object type.
- The amount of space in kilobytes that each object type consumes is also shown. This section helps you get a clear picture of exactly how much space the various object types in your database are taking up.

  **TIP:** If you want to see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the ‘toggle display’ option.

**Space Analyst for Oracle - Tablespace Space Overview**

The Tablespace Space Overview section displays summary information regarding every tablespace in the database. When invoked in graph form, this section shows how much total space is currently allocated to each tablespace. When shown in grid form, this section shows not only the total space allocated for each tablespace, but also includes the used, free, and percent free amounts, along with an indicator that shows whether any of the tablespace’s underlying data or temp files can automatically grow if necessary.

  **TIP:** To see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the ‘toggle display’ option. To return to the graph, just perform the same actions over the grid.

When this section is in grid form, you can also perform a number of actions on a tablespace like open it in a DBArtisan editor, extract it’s DDL, convert it to either local or dictionary-managed, and more. For example, to open a tablespace in one of DBArtisan’s object editors, do the following:

1. In the grid, select a tablespace.
2. Right-click and select **Open**.

   **OR**

   Select a tablespace and click the **Open** toolbar button.

   DBArtisan opens a Tablespace Editor.
Space Analyst for Oracle - Fragmentation Overview
The fragmentation overview provides a quick glance into how badly fragmented a tablespace is. Space Analyst computes a fragmentation index for every tablespace that can be used to quickly tell if fragmentation problems are plaguing a tablespace. A 100 percent reading indicates no fragmentation, while lower scores indicate the presence of fragmentation.

To see the percentage of fragmentation for each tablespace in the graph, do the following:
1. Position the pointer over each bar in the chart.

To see a grid-style display of the information shown in the graph, do the following:
1. Position the pointer over the graph.
2. Right-click and select Toggle Display.

Space Analyst for Oracle - Tablespace Objects Overview
The tablespace objects overview provides a window into the types of objects that each tablespace contains. This view is useful because it helps you quickly understand if undesired objects are present in a tablespace.

NOTE: There is no graphical view available for this section.

Session 3: Identifying Storage Structure Problems Using Space Analyst for Oracle
Space Analyst can help you quickly find storage structure problems that threaten the availability of your database.

Pinpointing ‘Out of Space’ Problems Using Space Analyst for Oracle
To quickly tell if any of the tablespaces in your database are about to run out of free space, do the following:
1. Select the Database Overview Tab.
2. In the Tablespace Space Overview section, position the pointer over the graph.
3. Right-click and select Toggle Display.
4. Examine the Percent Free columns of each tablespace along with the Autoextend column. If a tablespace is nearing the end of free space and cannot automatically grow, take action to prevent an out-of-space error.

Locating Fragmentation Issues Using Space Analyst for Oracle
There are a number of facilities within Space Analyst that helps you understand storage structure fragmentation.

To get summary information regarding tablespace fragmentation, do the following:
1. Select the Database Overview Tab.
2. In the Tablespace Fragmentation section, position the pointer over the graph.
3. Right-click and select Toggle Display.
The tablespace fragmentation grid scores each tablespace in terms of fragmentation levels and also present data regarding the number of free chunks in a tablespace along with the largest free chunk size. Tablespaces with poor fragmentation ratings are highlighted in red. Note that you can sort the grid from worst to best tablespaces by clicking on the Fragmentation Index column header in the grid.

To get the same information for datafiles, do the following:

1. Select the Data Files Tab.

   The datafile fragmentation grid scores each datafile in terms of fragmentation levels and also present data regarding the number of free chunks in the datafile along with the largest free chunk size. Note that temp files are not shown in this display. Datafiles with poor fragmentation ratings are highlighted in red.

To determine the type of fragmentation for a tablespace (bubble or honeycomb), do the following:

1. Select a tablespace or datafile that appears to have fragmentation issues.
2. Right-click and select Map.

   Space Analyst displays a graphical rendering of the tablespace on the Tablespace Map Tab.

Bubble fragmentation can be confirmed by the presence of many free space fragments being found throughout the tablespace. Honeycomb fragmentation can be identified by seeing free space fragments adjacent to one another in the map.

   NOTE: Large tablespace maps can be read easier if the “Full Screen” option is chosen. The standard view may be returned from full screen by clicking on the small full screen dialog box that appears at the upper left side of the display.

Note that object extent fragmentation problems can also be identified in Space Analyst’s tablespace map. To see if any table or index in a designated tablespace suffers from extent fragmentation, do the following:

1. Select the Tablespace Map Tab.
2. Select a tablespace.
3. In the View By section, select the Extent Problems option.

   Space Analyst reconstructs the tablespace map and accompanying left-hand object grid to reveal any objects that have extent problems. All objects with extent problems are color-coded in red for easy identification. Objects colored in green indicate that they do not suffer from any extent problems.

Using Analysis Reports in Space Analyst for Oracle

You can run HTML-formatted reports that pinpoint all major storage-related problems for a particular tablespace and the objects that are contained within it. To produce an Analysis Report, do the following:


   Space Analyst opens the Analysis Report dialog box.
2. In Report File Name and Report Title you can change the location and report title of the report.
3. In Category, select the types of space problems you would like to investigate. The default option is All.
4. In Tablespace, select the tablespace you would like to analyze.
5. Click Execute.

   Space Analyst creates a frame-based Analysis report for the tablespace.
Session 4: Pinpointing Object Performance Inefficiencies Using Space Analyst for Oracle

Space Analyst can greatly accelerate the process of locating objects in your database that are degrading overall response times because of their current storage/space situation.

Finding Space-Related Table Problems with Space Analyst for Oracle

To quickly determine if any tables in your database may be contributing to performance slowdowns, do the following:

1. In DBArtisan connect to the database you would like to check.
2. Open Space Analyst.
3. Select the Tables Tab.

Space Analyst runs diagnostics against your database and graphically display counts of tables that are found to contain various types of space problems.

The graph at the top of the Tables Tab shows the overall count of problem tables, but you can also view the details for each type of problem by using the Details grid at the bottom of the tab. For example, if Space Analyst identifies tables that have block inefficiencies, to find the actual tables and view details regarding the identified space problem, do the following in the Details grid:

1. In the Tablespace list, select All.
2. In the Table Owner list, select All.
3. In the Display list, select Tables with inefficient block usage.
4. Click Execute.

Space Analyst displays the actual tables that suffer from block inefficiencies along with their demographic details. To take corrective action on any object, select one or more tables and click the right mouse button to bring up a menu of possible actions.

Finding Space-Related Index Problems with Space Analyst for Oracle

To quickly determine if any indexes in your database may be contributing to performance slowdowns by following these steps:

1. In DBArtisan connect to the database you would like to check.
2. Open Space Analyst.
3. Select the Indexes Tab.

The graph at the top of the Indexes Tab shows the overall counts of problem indexes, but you can also view the details for each type of problem by using the Details grid at the bottom of the tab. For example, if Space Analyst identifies indexes that have next extent problems, then to find the actual indexes and view details regarding the space problem, do the following in the Details grid:

1. In the Tablespace list, select All.
2. In the Index Owner list, select All.
3. In the Display list, select Indexes with NEXT_EXTENT space deficits.
4 Click **Execute**.

Space Analyst displays the actual indexes that suffer from next extent space problems along with their demographic details. If you wanted to take corrective action on any object, select one or more indexes and click the right mouse button to bring up a menu of possible actions.

**NOTE:** Another critical statistic in determining index problems is deleted leaf rows. To display deleted leaf row statistics in the Details grid, do the following:

1 On the **Command** toolbar, select **Get Deleted Leaf Row Statistics**.

---

**Session 5: Correcting Space-Related Problems Using Space Analyst for Oracle**

After you have identified space-related bottlenecks in your database with Space Analyst, you can use a number of different tools and facilities within the product to eliminate each storage problems.

**Correcting Storage Structure ‘Out of Space’ Problems for Space Analyst for Oracle**

Space Analyst can help you quickly remedy tablespaces that are running out of room by:

- Resizing an existing tablespace to have more room.
- Adding a new datafile to an existing tablespace.
- Setting the Autoextend attribute for one or more of a tablespace’s datafiles to allow for automatic growth.

To manage these functions, do the following:

1 Select the **Database Overview Tab**.
2 In the **Tablespace Space Overview** section, right-click the graph, and then select **Toggle Display**.
3 Select the tablespace that is running low on free space.
4 Select the **Open** toolbar button or the **Open** shortcut menu option.

DBArtisan opens a Tablespace Editor.

In the Tablespace Editor, you can remedy the tablespace problem three ways:

- To resize an existing tablespace, open the tablespace’s datafile in the editor (using the **Edit** button) and change the datafile’s size to a new, larger size. Then close the editor and select the **Alter** toolbar button to make the change permanent to the database.

- To add a new datafile to the tablespace, select the **Add** button in the Datafile Tab of the tablespace editor, and specify the new datafile’s attributes. Once complete, close the Add Datafile dialog and select the **Alter** toolbar button to make the change permanent to the database.

- To set Autoextend for one or more of a tablespace’s datafiles, open each datafile in the datafile editor by selecting the **Edit** button. Once in the Modify Datafile dialog, select the option for **Auto Extend** and specify other attributes as needed. When complete, close the dialog and select the **Alter** toolbar button to make the change permanent to the database.

---

**Correcting Storage Structure Fragmentation Issues Using Space Analyst for**
Oracle

Space Analyst can help you quickly fix tablespaces that are suffering from fragmentation by:

- Coalescing adjacent free space fragments (honeycombs) in a tablespace
- Reorganizing a tablespace to eliminate free space bubbles
- Converting a dictionary-managed tablespace to locally-managed

To coalesce free space fragments in a tablespace, follow these steps:

1. On the Database Overview Tab, select any tablespace that appears to be suffering from fragmentation problems.
2. Select the Open toolbar button or the Open shortcut menu option.
   DBArtisan opens a Tablespace Editor.
3. To view fragmentation details for the tablespace, click the Space Tab of the Tablespace Editor.
4. To combine adjacent free space pockets into singular free space chunks, click the Coalesce button.

   **NOTE:** The fragmentation score of the tablespace does not change after a coalesce operation, then the tablespace likely suffers from bubble fragmentation. This can be verified by viewing the tables map for the tablespace.

To reorganize any tablespace suffering from bubble fragmentation, do the following:

1. On the toolbar, click the Open Wizard button.
   Space Analyst opens the Reorganization Wizard.
2. Select the New Reorganization option, and then click Next to continue.
   Space Analyst opens the next panel of the Reorganization Wizard.
3. Select Reorganize a Tablespacem, and then click Next to continue.
   Space Analyst opens the next panel of the Reorganization Wizard.
4. Select one or more tablespaces you would like to reorganize, and then click Next to continue.
   Space Analyst opens the next panel of the Reorganization Wizard.
5. Select a scratch tablespace that temporarily holds the reorganized tablespace’s objects.
6. Select the option to perform a space analysis so you can be assured that the scratch tablespace has enough room to temporarily hold all of the reorganized tablespace’s objects.
7. If the selected scratch tablespace does not contain enough free space, select another tablespace to serve as the scratch tablespace.
8. Click Next to continue.
   Space Analyst opens the next panel of the Reorganization Wizard.
9. You can choose to perform no customizations for the reorganization, or you can select various global or object-level options to customize your reorganization.
10. Click Finish.
   Space Analyst builds a reorganization script to reorganize your selected tablespace. Note that while the script is being constructed, you can still perform other work in Space Analyst or DBArtisan. Just minimize the script build window and continue to do other work in DBArtisan. You can check back whenever you’d like to see if the script is ready.
After the reorganization script is complete, you can choose to view/run the script in a DBArtisan ISQL window, schedule the script for later execution, or immediately execute the reorganization.

If you would like to convert a dictionary-managed tablespace to locally-managed, you can do one of the following:

- Create a new locally-managed tablespace in DBArtisan and then use the Reorganization Wizard to permanently relocate all objects from the dictionary-managed tablespace to the new locally-managed tablespace. This is the cleanest method for converting a dictionary-managed tablespace to locally-managed.

- Choose the option in Space Analyst that allows a quick conversion of an existing dictionary-managed tablespace to a locally-managed tablespace. This option is only available for Oracle versions 8.1.6 or later. This option does not defragment an existing dictionary-managed tablespace, but converts it so that all space management is conducted through the tablespace’s datafiles instead of the data dictionary and ensure that any new objects added to the tablespace conform to the tablespace’s storage attributes.

Correcting Space-Related Object Problems for Space Analyst for Oracle

Space Analyst can help you quickly fix objects that are experiencing space problems by:

- Modifying various storage-related properties for an object.
- Reorganizing the object to eliminate space inefficiencies.

Storage properties that are affecting the availability of an object (for example, the next extent or maximum extent values for a table or index) can easily be altered by using the integration that exists between Space Analyst and DBArtisan. For example, to modify a table’s next extent value, do the following:

1. On the Tablespace Map Tab or the Table Details grid on the Tables Tab, select a table you would like to alter.
2. Select the Open toolbar button or the Open shortcut menu option.
   
   DBArtisan opens a Tablespace Editor.
3. Select the Storage Tab.
4. Change the numeric value for the table’s next extent.

   **NOTE:** You cannot change next extent values for tables that exist in locally-managed tablespaces.

5. On the toolbar, click the Alter button.

   DBArtisan displays the DDL that change the table’s next extent value.

You can easily reorganize one or more objects with the Reorganization Wizard. To reorganize any table or index that in any grid, do the following:

1. Select the object(s).
2. On the toolbar, click the Reorganization Wizard button, or right-click and select Reorganize.

   Space Analyst opens the Reorganization Wizard.

You can customize object reorganizations by using either the global or specific object customization options in the Reorganization Wizard. For example, you can perform online reorganizations of indexes or index-organized tables.

Space Analyst’s Reorganization Wizard can also intelligently identify objects that need to be reorganized. For example, to reorganize tables in your database that have chained/migrated rows, do the following:

1. On the toolbar, click the Reorg Wizard button.

   Space Analyst opens the Reorganization Wizard.
2 Select the **New Reorganization** option, and then click **Next** to continue.

Space Analyst opens the next panel of the Reorganization Wizard.

3 Select **Repair tables with chained/migrate rows**, and then click **Next** to continue.

Space Analyst runs diagnostics against your database to find any table or table partition that suffers from chained rows. Space Analyst displays the objects under either the Tables or Table Partition(s) node in the Reorg Wizard explorer tree. You can then select one or all table objects and continue with the reorganization process.
Space Analyst for Microsoft SQL Server Tutorial

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result from poor space management. Running inside the Embarcadero DBArtisan console, Space Analyst helps you identify and correct storage problems by supplying powerful analysis facilities and corrective utilities and that will satisfy advanced experts, yet are easy enough to use for database novices.

Space Analyst for Microsoft Product Benefits
Embarcadero Space Analyst allows the database administrator to quickly ascertain if space problems are threatening the availability or performance of their key databases. If storage problems are found, Space Analyst provides powerful but easy to use space management wizards and utilities that can accomplish simple or complex reorganizations of all or parts of a database.

About this Space Analyst for Microsoft Tutorial
This tutorial helps you get started with Embarcadero Space Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Space Analyst. You will have learned how to get a summary view of storage for your database as well as how to identify potential space-related threats. Finally, you’ll use the reorganization wizard to correct any space problems that you find.

This tutorial is divided into 5 sections.

Session 1: Getting Started with Space Analyst
Session 2: Obtaining a Global View of Storage
Session 3: Identifying Storage Structure Problems
Session 4: Pinpointing Object Performance Inefficiencies
Session 5: Correcting Space-Related Problems

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see Additional Evaluation Resources.

Session 1: Getting Started with Space Analyst for Microsoft

Download and Install
You can obtain the latest version of Space Analyst by downloading DBArtisan from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Space Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Space Analyst.

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.
Space Analyst for Microsoft Overview

Once you are connected to an Microsoft SQL Server database, you can open Space Analyst from within DBArtisan two ways:

1. On the Analyst menu, select Space Analyst.

   NOTE: You do not need to install anything for Space Analyst to work.

Session 2: Obtaining a Global View of Storage Using Space Analyst for Microsoft

When first invoked, Space Analyst displays a complete overview of the storage situation of a particular SQL Server. The first tab of Space Analyst can be used to see all pertinent global statistics regarding storage structure usage throughout a server.

There are four distinct sections displayed on the first tab (labeled Storage Overview) of Space Analyst:

- Server Overview
- Database Overview
- Log Overview
- Server Disk Overview

Space Analyst for Microsoft - Server Overview

The total count of all databases, file groups, and files are shown in this section. You can also see the percentage breakdown of the database vs. log space on the server. Positioning the mouse pointer over the graph causes it to display the actual percentage amounts for storage type.

   TIP: If you want to see a grid-style display of the information shown in most graphs, position your mouse over the graph, click the right mouse button and select the 'toggle display' option.

Space Analyst for Microsoft - Database Overview

Summary information regarding every database on the server is present in the database overview section. When invoked in graph form, this section shows how much total space is used by each database. When shown in grid form, this section shows not only the total space used for each database, but also includes the total allocated, free, and percent free amounts, along with an indicator that shows whether any of the database's underlying datafiles can automatically grow if necessary.

   TIP: To see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the 'toggle display' option. To return to the graph, just perform the same actions over the grid.

When this section is in grid form, you can also perform a number of actions on a database like open it in a DBArtisan editor, extract its DDL, shrink it, and more. For example, if you want to open a database in DBArtisan's database manager, do the following:

1. In the grid, select a database.

2. Right-click and select Open.

   OR
3 Select a database and click the **Open** toolbar button. 

**DBArtisan** opens the Database Manager.

---

**Space Analyst for Microsoft - Log Overview**

The log overview provides a quick glance into database transaction log usage. When invoked in graph form, this section shows the percentage of used log space per database. When shown in grid form, this section shows not only the total percentage of transaction log space used for each database, but also includes the total allocated and free amounts (in MB), along with an indicator that shows whether any of the database's transaction logs underlying log files can automatically grow if necessary.

**Space Analyst for Microsoft - Server Disk Overview**

The server disk overview provides a window into how much free space exists on the actual server machine. In graph form, this section shows the amount of free space (in MB) across each of the server's drives. When in grid form, it shows free space per drive, but also includes the amount of allocated SQL Server data and log space per drive.

---

**Session 3: Identifying Storage Structure Problems in Space Analyst for Microsoft**

Space Analyst can help you quickly find storage structure problems that threaten the availability of your database.

**Pinpointing ‘Out of Space’ Problems Using Space Analyst for Microsoft**

To quickly tell if any of the tablespaces in your database are about to run out of free space, do the following:

1. Select the **Storage Overview Tab**.
2. In the **Database Overview** and **Log Overview** sections, position the pointer over the graph.
3. Right-click and select **Toggle Display**.

Examine the Percent DB Used columns of each database and transaction log along with the DB Can Grow column. If a database or transaction log is nearing the end of free space and cannot automatically grow, then action should be taken to prevent an out-of-space error.

You can also obtain graphical views of databases and pertinent storage property information by viewing the Databases Tab of Space Analyst. You can visually get a handle on the amounts of table, index, and free space of your databases and quickly view critical storage property facts such as the number of databases that cannot automatically grow in size if needed.

All the graphs can be toggled to grids by using the right mouse menu option of Toggle Display. When in grid mode, very detailed information is available concerning database and transaction log space. Note that the graph is scrollable so you may have to scroll to see all your databases.

You can perform a number of actions on a database, such as shrinking it or invoking database-related DBCC operations by using either the right mouse menu or the Command toolbar button (the yellow arrow).
Viewing Filegroup and File Details Using Space Analyst for Microsoft
You can also view space usage at the filegroup and file levels. To get summary and detailed information for filegroups and files, follow these steps:

1. Invoke Space Analyst and view the File Groups/Files Tab. The default presentation is a graphical overview of filegroups. Note that the graph is scrollable so you may have to scroll to see all your filegroups.

2. To view detailed information for filegroups, right-click and select Toggle Display to view the grid.

3. To view file information instead of filegroup metrics, select Files.

When viewing filegroup and file information, you can perform a number of actions on selected filegroups and files such as extracting their DDL or shrinking the parent database or selected file. All options are available by using either the shortcut menu or Command menu.

Using Analysis Reports in Space Analyst for Microsoft
You can run HTML-formatted reports that pinpoint all major storage-related problems for a particular tablespace and the objects that are contained within it. To produce an Analysis Report, do the following:


2. In Report File Name and Report Title you can change the location and report title of the report.

3. In Category, select the types of space problems you would like to investigate. The default option is All.

4. In Database, select the database you would like to analyze.

5. In File Group, select the filegroup level.

6. Click Execute. Space Analyst creates a frame-based Analysis report for the database.

Session 4: Pinpointing Object Performance Inefficiencies Using Space Analyst for Microsoft
Space Analyst can greatly accelerate the process of locating objects in your databases that are degrading overall response times because of their current inefficient storage usage.

Finding Space-Related Table Problems with Space Analyst for Microsoft
To quickly determine if any tables in your database may be contributing to performance slowdowns, do the following:

1. Open Space Analyst.

2. Select the Objects Tab.

3. In Object Details, select the target object.

By default, the General Space button is selected. This view will show you a graphical layout of table used and free space for the database. Note that the graph is scrollable so you can scroll to see all the tables for your database.

The Object details grid in the lower area of the tab can be used to view detailed storage information for every table in the selected database.
To troubleshoot table storage issues in your database, select the Space Diagnostics radio button. This option will interrogate your database and will provide you with a graphical count of the number of tables that are experiencing specific types of storage inefficiencies. Space Analyst uses predefined thresholds (which you can customize by selecting the Threshold Options toolbar button) to determine how many and what objects are suffering from storage inefficiencies.

The table below describes the types of table-related space problems shown in the bar graph found on the Objects Tab:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan Density</td>
<td>This is computed by dividing the optimum number of extent switches by the actual number of extent switches. This percentage can be misleading, however, if the object spans more than one database file and should not be considered in such an event. Values close to 100% are best.</td>
</tr>
<tr>
<td>Logical Fragmentation</td>
<td>This metric represents the percentage of pages that are out of physical order. High values (greater than 25%) for this metric may mean that index scan performance is not what it could be. Note that this statistic is meaningless for tables without clustered indexes (heap tables).</td>
</tr>
<tr>
<td>Extent Fragmentation</td>
<td>This highlights any gaps (lack of contiguous order) in extents and indicates the presence of external fragmentation. Percentages of 25-50% or more are not a favorable reading.</td>
</tr>
<tr>
<td>Average Page Density</td>
<td>This metric represents the fullness of the data or index pages, and is an indicator of internal fragmentation. Remember that the more full a data or index page is, the less I/O needed by SQL Server when performing scans of tables or indexes. High values are good here, with anything below 50% being suspect. Low values for this metric often indicate the presence of page splitting. Keep in mind, however, that internal fragmentation isn't necessarily bad in OLTP environments where large FILLFACTORs are often specified. Also understand that small objects (those with little or no data) will likely show low readings because they simply don't consume enough space in a database page. You can ignore these objects.</td>
</tr>
<tr>
<td>Forwarded Records</td>
<td>SQL Server will move rows in a table under certain conditions. One situation might arise when you update a row in a table that has a variable-length column to a larger size that will no longer fit on its original page. Another situation would be if SQL Server moves a row when the clustered index column changes. Forwarded records can reduce performance at times because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself. But when does row forwarding become a problem? For example, just because a table has one hundred forwarded rows, does that mean a performance problem exists? The answer is: it depends. If the table has one hundred rows in it, then yes, a problem exists - 100% of the table suffers from forwarded rows. If the table has three million rows, then the forwarded row situation is likely not causing much fuss in the database.</td>
</tr>
</tbody>
</table>
The graph at the top of the Objects Tab shows the overall count of problem tables, but you can also view the details for each type of problem by using the Details grid at the bottom of the tab. For example, if Space Analyst identifies tables that have forwarded records, then to find the actual tables and view details regarding the identified space problem, do the following:

1. In the File Group list, select All.
2. In the Owner list, select All.
3. In the Display list, select Tables with forwarded records problems.
4. Click Refresh.

You would then see the actual tables that suffer from forwarded records along with their demographic details. If you wanted to take corrective action on any object, you could highlight one or more tables and click the right mouse button to bring up a menu of possible actions.

Session 5: Correcting Space-Related Problems Using Space Analyst for Microsoft

After you have identified space-related bottlenecks in your database with Space Analyst, you can use a number of different tools and facilities within the product to eliminate each storage problems.

Correcting Storage Structure ‘Out of Space’ Problems Using Space Analyst for Microsoft

Space Analyst can help you quickly remedy databases and transaction logs that are running out of room by:

- Resizing an existing database or transaction log to have more room.
- Adding a new filegroup or file to an existing database or transaction log.
- Setting the Autogrowth attribute for one or more of a database’s or transaction log’s files to allow for automatic growth.

To manage these functions, do the following:

1. Select the Databases Tab.
2. In the Database Summary section, right-click the graph, and then select Toggle Display.
3. Select the database that is running low on free space.
4. Select the Open toolbar button or the Open shortcut menu option.
   
   DBArtisan opens the Database Manager.

In the Database Manager, you can remedy the database problem in one of three ways:

- To resize an existing database or transaction log, open one of the database’s or log’s files in the editor (using the Edit button) and change the file’s size to a new, larger size. Then close the editor and select the Alter toolbar button to make the change permanent to the database.
- To add a new file or file group to the database or transaction log (note: you can’t add new filegroups to a transaction log, only files), select the Add button in the editor, and specify the new filegroup or file’s attributes. Once complete, close the Add Datafile dialog and select the Alter toolbar button to make the change permanent to the database.
• To set Autogrowth for one or more of a database's or transaction log's files, open each file in the file editor by selecting the file and then pressing the Edit button. Once in the Modify File dialog, select the option for Allow Growth and specify other attributes as needed. When complete, close the dialog and select the Alter toolbar button to make the change permanent to the database.

Removing Wasted Space in Databases or Files Using Space Analyst for Microsoft

Space Analyst can help you quickly eliminate unused space in a database or database file. The unused space can be released back to the operating system for other use. To remove unused space from a database, do the following:

1. To shrink an entire database, right-click a database in any of Space Analyst's grids, and then select Shrink Database.
   DBArtisan opens the Shrink File dialog box.

2. To shrink a specific file in a database, select the Filegroups/Files Tab and then view files by selecting the Files radio button.

3. Right-click a file and select Shrink File.
   DBArtisan opens the Shrink File dialog box.

4. Select desired options.

5. Click Execute to make the change permanent.

Correcting Space-Related Object Problems for Space Analyst for Microsoft

Once you identify specific objects that need storage corrections, you can use Space Analyst to help you quickly resolve your found issues. The Space Management wizard inside of Space Analyst can be used to build reorganization and space maintenance jobs that run right the first time. All jobs can either be immediately run or scheduled for later execution.

To create a standard database or object reorganization job, do the following:

1. Click the Open Wizard button.

2. Select New Space Management Task.

3. Click Next.

4. Select Create standard reorganization job.

5. Click Next.

6. Select the databases and objects you would like to reorganize from the Explorer tree on the left hand side of the wizard. You can reorganize full databases, selected schema accounts, or selected objects.

7. Click Next.

8. The next panel allows you to customize the reorganization. You can either set options at the global level (affects all objects) or specific object level. Using the options, you can relocate objects to different filegroups, perform online index rebuilds (for SQL Server 2000 or later), and specify other customizations.

9. Click Next.

10. The next panel builds your reorganization script and allows you to preview the operation. Click Next when ready.

11. Click Next.
The final panel allows you to open the reorganization script in a SQL editor, immediately run it, or schedule it for later execution.

Proactive Space Management Using Space Analyst for Microsoft

If you have many databases to take care of, then you might want to relieve the burden of performing detailed storage analysis on every database, each of which may or may not require attention. Fortunately, Space Analyst makes it easy to create smart space management jobs that can automate this task for you.

You can create two specialized space management jobs that allow you to proactively eliminate space threats before they affect your database:

You can create a "reorganization analyst job" that allows you to specifically select various databases and/or objects and then set reorganization thresholds that determine when a reorganization will occur. Once constructed, you can schedule these jobs to run as often as you'd like. During execution, these jobs will interrogate your chosen database/objects and determine if a reorganization needs to occur by applying your customized reorganization thresholds. If any objects are found to need reorganization, they are dynamically reorganized using any customized options you specify.

You can build space maintenance jobs that will perform functions such as check the validity of your databases and filegroups, repair any found minor corruption, and eliminate unused space in your databases. You can also set the job up to notify you if any problems are found during job execution.

These "smart" space maintenance jobs allow you to truly "set and forget" the task of monitoring and reorganizing the space of your critical databases.
Space Analyst for Sybase ASE Tutorial

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result from poor space management. Running inside the Embarcadero DBArtisan console, Space Analyst helps you identify and correct storage problems by supplying powerful analysis facilities and corrective utilities and that will satisfy advanced experts, yet are easy enough to use for database novices.

Space Analyst for Sybase ASE Product Benefits

Embarcadero Space Analyst allows the database administrator to quickly ascertain if space problems are threatening the availability or performance of their key databases. If storage problems are found, Space Analyst provides powerful but easy to use space management wizards and utilities that can accomplish simple or complex reorganizations of all or parts of a database.

About this Space Analyst for Sybase ASE Tutorial

This tutorial helps you get started with Embarcadero Space Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Space Analyst. You will have learned how to get a summary view of storage for your database as well as how to identify potential space-related threats. Finally, you’ll use the reorganization wizard to correct any space problems that you find.

This tutorial is divided into 4 sections.

Session 1: Getting Started with Space Analyst

Session 2: Obtaining a Global View of Storage

Session 3: Identifying Storage Structure Problems

Session 4: Correcting Space-Related Problems

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see Additional Evaluation Resources.

Session 1: Getting Started with Space Analyst for Sybase ASE

Download and Install

You can obtain the latest version of Space Analyst by downloading DBArtisan from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Space Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Space Analyst.

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Space Analyst for Sybase ASE Overview

The graphic below illustrates all the main elements of the DBArtisan User Interface.
Once you are connected to an Sybase ASE database, you can open Space Analyst from within DBArtisan two ways:

1. On the Analyst menu, select Space Analyst.

   **NOTE:** You do not need to install anything for Space Analyst to work.

### Session 2: Obtaining a Global View of Storage Using Space Analyst for Sybase ASE

When first invoked, Space Analyst displays a complete overview of the storage situation of a particular server. The first tab of Space Analyst can be used to see all pertinent global statistics regarding storage structure usage throughout a server.

There are four distinct sections displayed on the Storage Overview Tab of Space Analyst:

- **Server Overview**
- **Log Overview**
- **Database Overview**
- **Device Overview**

#### Space Analyst for Sybase ASE - Server Overview

The total count of all databases, file groups, and files are shown in this section. You can also see the percentage breakdown of the database vs. log space on the server. Positioning the mouse pointer over the graph causes it to display the actual percentage amounts for storage type.

   **TIP:** If you want to see a grid-style display of the information shown in most graphs, position your mouse over the graph, click the right mouse button and select the 'toggle display' option.

#### Space Analyst for Sybase ASE - Log Overview

The log overview provides a quick glance into database transaction log usage. When invoked in graph form, this section shows the percentage of used log space per database. When shown in grid form, this section shows not only the total percentage of transaction log space used for each database, but also includes the total allocated and free amounts (in MB), along with an indicator that shows whether any of the database's transaction logs underlying log files can automatically grow if necessary.

#### Space Analyst for Sybase ASE - Database Overview

Summary information regarding every database on the server is present in the database overview section. When invoked in graph form, this section shows how much total space is used by each database. When shown in grid form, this section shows not only the total space used for each database, but also includes the total allocated, free, and percent free amounts, along with an indicator that shows whether any of the database's underlying datafiles can automatically grow if necessary.

   **TIP:** To see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the ‘toggle display’ option. To return to the graph, just perform the same actions over the grid.
When this section is in grid form, you can also perform a number of actions on a database like open it in a DBArtisan editor, extract its DDL, shrink it, and more. For example, if you want to open a database in DBArtisan's database manager, do the following:

1. Select a database, right-click, and select **Open**.
   OR
   Select a database and click the **Open** toolbar button.
   DBArtisan opens the Database Manager.

**Space Analyst for Sybase ASE - Device Overview**

The device overview section contains a graphical and grid control that works in the following way:

- When the graph is displayed, a bar chart is shown that lists the total space used (in MB) of all devices defined to the server.
- When the grid is displayed, detailed storage information for each database device is presented:
  - Device name
  - Physical name
  - Total space
  - Used space
  - Free space
  - Percent free
  - Percent used

**Session 3: Identifying Storage Structure Problems in Space Analyst for Sybase ASE**

Space Analyst can help you quickly find storage structure problems that threaten the availability of your database.

**Pinpointing ‘Out of Space’ Problems Using Space Analyst for Sybase ASE**

To quickly tell if any of the tablespaces in your database are about to run out of free space, do the following:

1. Select the **Storage Overview Tab**.
2. In the **Database Overview** and **Log Overview** sections, position the pointer over the graph.
3. Right-click and select **Toggle Display**.

Examine the Database Percent Free columns of each database and Log Percent Free for each transaction log. If a database or transaction log is nearing the end of free space and cannot automatically grow, then action should be taken to prevent an out-of-space error.

You can also obtain graphical views of databases and pertinent storage property information by viewing the Databases Tab of Space Analyst. You can visually get a handle on the amounts of table, index, and free space of your databases and quickly view critical storage property facts such as the number of databases that cannot automatically grow in size if needed.
All the graphs can be toggled to grids by using the right mouse menu option of 'Toggle Display'. When in grid mode, very detailed information is available concerning database and transaction log space. Note that the graph is scrollable so you may have to scroll to see all your databases.

Using Analysis Reports in Space Analyst for Sybase ASE
You can run HTML-formatted reports that pinpoint all major storage-related problems for a particular tablespace and the objects that are contained within it. To produce an Analysis Report, do the following:


   Space Analyst opens the Analysis Report dialog box.

2. In Report File Name and Report Title you can change the location and report title of the report.

3. In Category, select the types of space problems you would like to investigate. The default option is All.

4. In Database, select the database you would like to analyze.

5. In File Group, select the filegroup level.

6. Click Execute.

   Space Analyst creates a frame-based Analysis report for the database.

Session 4: Correcting Space-Related Problems Using Space Analyst for Sybase ASE
After you have identified space-related bottlenecks in your database with Space Analyst, you can use a number of different tools and facilities within the product to eliminate each storage problems.

Correcting Storage Structure ‘Out of Space’ Problems Using Space Analyst for Sybase ASE
Space Analyst can help you quickly remedy databases and transaction logs that are running out of room by:

- Resizing an existing database or transaction log to have more room.
- Adding a new filegroup or file to an existing database or transaction log.
- Setting the Autogrowth attribute for one or more of a database's or transaction log's files to allow for automatic growth.

To manage these functions, do the following:

1. Select the Storage Overview Tab.

2. In the Database Overview or Log Overview section, right-click the graph, and then select Toggle Display.

3. Select the database that is running low on free space.

4. Select the Open toolbar button or the Open shortcut menu option.

   DBArtisan opens the Database Manager.

In the Database Manager, you can remedy the database problem in one of three ways:
• To resize an existing database or transaction log, open one of the database’s or log’s files in the editor (using the Edit button) and change the file’s size to a new, larger size. Then close the editor and select the Alter toolbar button to make the change permanent to the database.

• To add a new file or file group to the database or transaction log (note: you can’t add new filegroups to a transaction log, only files), select the Add button in the editor, and specify the new filegroup or file’s attributes. Once complete, close the Add Datafile dialog and select the Alter toolbar button to make the change permanent to the database.

• To set Autogrowth for one or more of a database’s or transaction log’s files, open each file in the file editor by selecting the file and then pressing the Edit button. Once in the Modify File dialog, select the option for Allow Growth and specify other attributes as needed. When complete, close the dialog and select the Alter toolbar button to make the change permanent to the database.

Correcting Space-Related Object Problems for Space Analyst for Sybase ASE

Once you identify specific objects that need storage corrections, you can use Space Analyst to help you quickly resolve your found issues. The Space Management wizard inside of Space Analyst can be used to build reorganization and space maintenance jobs that run right the first time. All jobs can either be immediately run or scheduled for later execution.

To create a standard database or object reorganization job, do the following:

1. Click the Open Wizard button.
2. Select New Space Management Task.
3. Click Next.
4. Select Create standard reorganization job.
5. Click Next.
6. Select the databases and objects you would like to reorganize from the Explorer tree on the left hand side of the wizard. You can reorganize full databases, selected schema accounts, or selected objects.
7. Click Next.
8. The next panel allows you to customize the reorganization. You can either set options at the global level (affects all objects) or specific object level.
9. Click Next.
10. The next panel builds your reorganization script and allows you to preview the operation. Click Next when ready.
11. Click Next.

The final panel allows you to open the reorganization script in a SQL editor, immediately run it, or schedule it for later execution.

Proactive Space Management Using Space Analyst for Sybase ASE

If you have many databases to take care of, then you might want to relieve the burden of performing detailed storage analysis on every database, each of which may or may not require attention. Fortunately, Space Analyst makes it easy to create smart space management jobs that can automate this task for you.

You can create two specialized space management jobs that allow you to proactively eliminate space threats before they affect your database:
You can create a "reorganization analyst job" that allows you to specifically select various databases and/or objects and then set reorganization thresholds that determine when a reorganization will occur. Once constructed, you can schedule these jobs to run as often as you'd like. During execution, these jobs will interrogate your chosen database/objects and determine if a reorganization needs to occur by applying your customized reorganization thresholds. If any objects are found to need reorganization, they are dynamically reorganized using any customized options you specify.

You can build space maintenance jobs that will perform functions such as check the validity of your databases and filegroups, repair any found minor corruption, and eliminate unused space in your databases. You can also set the job up to notify you if any problems are found during job execution.

These "smart" space maintenance jobs allow you to truly "set and forget" the task of monitoring and reorganizing the space of your critical databases.
Space Analyst for IBM DB2 for Linux, Unix, and Windows Tutorial

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result from poor space management. Running inside the Embarcadero DBArtisan console, Space Analyst helps you identify and correct storage problems by supplying powerful analysis facilities and corrective utilities and that will satisfy advanced experts, yet are easy enough to use for database novices.

Space Analyst for IBM DB2 for Linux, Unix, and Windows Product Benefits

Embarcadero Space Analyst allows the database administrator to quickly ascertain if space problems are threatening the availability or performance of their key databases. If storage problems are found, Space Analyst provides powerful but easy to use space management wizards and utilities that can accomplish simple or complex reorganizations of all or parts of a database.

About this Space Analyst for IBM DB2 for Linux, Unix, and Windows Tutorial

This tutorial helps you get started with Embarcadero Space Analyst. After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Space Analyst. You will have learned how to get a summary view of storage for your database as well as how to identify potential space-related threats. Finally, you'll use the reorganization wizard to correct any space problems that you find.

This tutorial is divided into 5 sections.

Session 1: Getting Started with Space Analyst
Session 2: Obtaining a Global View of Storage
Session 3: Identifying Storage Structure Problems
Session 4: Pinpointing SQL Server Performance Inefficiencies
Session 5: Correcting Space-Related Problems

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see Additional Evaluation Resources.

Session 1: Getting Started with Space Analyst for IBM DB2 for Linux, Unix, and Windows

Download and Install

You can obtain the latest version of Space Analyst by downloading DBArtisan from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Space Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Space Analyst.

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.
Space Analyst for IBM DB2 for Linux, Unix, and Windows Overview

Once you are connected to an IBM DB2 for Linux, Unix, and Windows database, you can open Space Analyst from within DBArtisan two ways:

1. On the Analyst menu, select Space Analyst.

   **NOTE:** You do not need to install anything for Space Analyst to work.

Session 2: Obtaining a Global View of Storage Using Space Analyst for IBM DB2 for Linux, Unix, and Windows

When it opens, Space Analyst shows a complete overview of the storage situation of a particular database or database server. The first tab of Space Analyst displays all pertinent global statistics regarding storage structure usage throughout a database.

There are four distinct sections displayed on the first **Storage Overview Tab** of Space Analyst:

- **Storage Overview**
- **Tablespace Space Overview**
- **Log Overview**
- **Tablespace Objects Overview**

Space Analyst for IBM DB2 for Linux, Unix, and Windows - Storage Overview

The breakdown of space usage of the allocated tablespace space, by index, table and temp data is shown in this section. You can also see the total number of tablespaces and the number SMS and DMS spaces here.

   **NOTE:** Make sure your database object statistics are up to date. Space analyst uses these statistics in the various graphs and grids.

Space Analyst for IBM DB2 for Linux, Unix, and Windows - Tablespace Space Overview

The Tablespace Space Overview section displays summary information regarding every tablespace on the database. When invoked in graph form, this section shows how much total space is allocated to each tablespace. When shown in grid form, this section shows not only the total allocated for each tablespace, but also includes information such as used space, free space, state, number of containers etc.

   **TIP:** To see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the ‘toggle display’ option. To return to the graph, just perform the same actions over the grid.

When this section is in grid form, you can also perform a number of actions on a tablespace like open it in a DBArtisan editor, extract its DDL, update its object statistics and more. For example, if you want to update the statistics on all tables in a tablespace, you would perform the following:

1. Select a tablespace in a grid.
2. Right-click the tablespace, and select **Update Statistics**.

   **OR**

3. On the **Space Analyst** toolbar click the **Update Statistics**.
Space Analyst for IBM DB2 for Linux, Unix, and Windows - Log Overview
The log overview provides a quick glance into database transaction log usage. This section shows the percentage of used and free log space for the database, the various database configuration parameters associated with the log space and the high watermarks for the log space usage.

Space Analyst for IBM DB2 for Linux, Unix, and Windows - Tablespace Objects Overview
The tablespace objects overview section provides an overview of how much space is used by the tables and indexes on each tablespace.

Session 3: Identifying Storage Structure Problems
Space Analyst can help you quickly find storage structure problems that threaten the availability of your databases.

Pinpointing 'Out of Space' Problems
You can quickly tell if your transaction logs are about to run out of free space by following these steps:

1. Open Space Analyst.
2. View the Database and Log Overview sections on the Storage Overview Tab.
3. Examine the free and used space metrics in the pie chart. If the transaction log is nearing the end of free space, then action should be taken to prevent a log full error.

You can also obtain graphical views of tablespaces and pertinent storage property information by viewing the Tablespaces tab of Space Analyst. You can visually get a handle on the amount of used and free space of your tablespaces and quickly view critical storage property facts such as the number of containers for each tablespace, the space allocated for each container, the usage of tablespace space on each partition in a multi-partition database.

All the graphs can be toggled to grids by using the right mouse menu option of 'Toggle Display'. When in grid mode, very detailed information is available concerning the selected tablespace.

NOTE: You can perform further corrective actions on a tablespace, by opening it in the tablespace editor by using either the right mouse menu or the Command toolbar button (the yellow arrow).

Using Analysis Reports
You can run HTML-formatted reports that pinpoint all major storage-related problems for a particular database and the objects that are contained within it. To produce an Analysis Report, follow these steps:

2. If you want to change the location and report title of the report, edit the predefined entries for the Report File Name and Report Title.
3. Select the types of space problems you would like identified on the report by using the Category list.
4. Click Execute.

Space Analyst creates a frame-based Analysis report for the database you indicated.
Session 4: Pinpointing SQL Server Performance Inefficiencies

Space Analyst can greatly accelerate the process of locating objects in your database that are degrading overall response times because of their current inefficient storage usage.

Finding Space-Related Table Problems

You can quickly tell if any tables in a tablespace that may be contributing to performance slowdowns by following these steps:

1. Open the Space Analyst and select the Tables Tab.
2. Select the tablespace you would like to interrogate in the Tablespace list.
3. Click the Refresh button.

You will see a graphical count of the number of tables that are experiencing specific types of storage inefficiencies in the upper area of the tab. The table reorganization candidate details grid in the lower area of the tab can be used to view detailed storage information for every table in the selected tablespace. Space Analyst uses predefined thresholds (which you can customize by selecting the Threshold Options toolbar button) to determine how many and what tables are suffering from storage inefficiencies.

The types of table-related space problems shown in the bar graph found on the objects tab include:

- **Overflow Problems** - Overflows occur in tables when overflows occur when rows are updated with more bytes than the original row (VARCHAR fields etc.). Overflows also occur when the new columns are added to existing tables. Since the overflow data will not be stored contiguously with the other data, it causes access inefficiencies.

- **Space Utilization Problems** - Normally, DB2 reclaims any space freed up by deletes, in subsequent insert/update operations. However, if a DB2 table has the 'APPEND' mode set to on and a large amount of data is deleted from the table DB2 will not reclaim freed space. Also when some types of large load operations are performed, the table may be left with a large amount of unused space.

- **Page Utilization Problems** - As the number of empty pages increases, the need for table reorganization also increases. Empty pages waste space and reduce the table scan efficiency, since empty pages are read into the buffer pool for a table scan wasting bufferpool memory.

- **Index Contention** - Unless the user explicitly separates the index and table tablespaces, DB2 tables and indexes are placed on the same tablespace. This can become a bottleneck with heavily accessed tables using index scans since data and index accesses will compete for the IO device resources.

The graph at the top of the Tables tab shows the overall count of problem tables, but you can also view the details for each type of problem by using the Details grid at the bottom of the tab. For example, if Space Analyst identifies tables that have overflow records, then to find the actual tables and view details regarding the identified space problem, you would perform the following:

1. Select All from the Tablespace list in the Details grid.
2. Select All from the Table Schema list in the Details grid.
3. Select Tables with Overflow Problems from the Display list in the Details grid.
4. Click Refresh.

You would then see the actual tables that suffer from overflowed records along with their demographic details. If you wanted to take corrective action on any object, you could highlight one or more tables and click the right mouse button to bring up a menu of possible actions.
Finding Space-Related Index Problems
You can quickly tell if any indexes in your database may be contributing to performance slowdowns by following the exact same steps as outlined above for tables with one exception: You want to select the Indexes tab instead of the Tables tab. The overview graph and index details sections of the Indexes tab work in the exact same manner for indexes as they do for tables.

Session 5: Correcting Space-Related Problems
Once you have identified space-related bottlenecks in your database with Space Analyst, you can use a number of different tools and facilities within the product to eliminate each storage headache.

Correcting Tablespace 'Out of Space' Problems
Space Analyst can help you quickly remedy tablespaces that are running out of room by:
- Resizing existing tablespace containers to have more room.
- Adding new containers to an existing tablespace.
- Moving Objects from a near full tablespace to less utilized tablespace

Resizing and adding of tablespace containers can be managed by performing the following steps:
1. Open Space Analyst and select the Database Overview Tab.
2. Select the graph in Tablespace Overview section and change to the grid display by selecting the right mouse menu option toggle display.
3. Select the DMS tablespace that is running low on free space and either select the Open toolbar option or the Open right mouse menu option.

This opens the Tablespace in DBArtisan's tablespace editor. Once you are in the editor, you can remedy the tablespace problem in one of two ways:
- To resize an existing container, select the container from the grid and click the Edit button and extend/resize the container to a new, larger size. Then close the editor and select the Alter toolbar button to make the change permanent to the tablespace.
- To add a new container to the tablespace, select the Add button in the editor, and specify the new container attributes. Once complete, close the Add Container dialog and select the Alter toolbar button to make the change permanent to the database.

Correcting Space-Related Object Problems
Once you identify specific objects that need storage corrections, you can use Space Analyst to help you quickly resolve your found issues. The Space Management wizard inside of Space Analyst can be used to build reorganization and space maintenance jobs that run right the first time. All jobs can either be immediately run or scheduled for later execution.

To create a standard database or object reorganization job, follow these steps:
1. On the toolbar, click the Space Management.
2. Select New Space Management Task.
3. Click Next.
4. Select Create standard reorganization job.
5 Click Next.

6 Select the objects you would like to reorganize from the Explorer tree on the left hand side of the wizard. Note that you can reorganize full tablespaces, selected schemas or selected tables.

7 Click Next.

8 The next few panels let you customize the reorganization. You can choose to either set options at the global level (affects all objects) or specific object level. Using the options, you can reorganize tables and indexes, perform LOB and Long data reorganizations and specify other customizations.

9 The second to last panel builds your reorganization script and allows you to preview the operation.

10 Click Next.

11 The final panel allows you to execute the reorganization script immediately, or schedule it for later execution.

**Proactive Space Management**

If you have many tablespace/schemas to take care of, then you might want to relieve the burden of performing detailed storage analysis on every tablespace or schema, each of which may or may not require attention. Fortunately, Space Analyst makes it easy to create smart space management jobs that can automate this task for you.

You can create a "reorganization analyst job" that allows you to specifically select various tablespaces/ schemas or objects and then set reorganization thresholds that determine when reorganization will occur. Once constructed, you can schedule these jobs to run as often as you'd like. During execution, these jobs will interrogate your chosen database/objects and determine if a reorganization needs to occur by applying your customized reorganization thresholds. If any objects are found to need reorganization, they are dynamically reorganized using any customized options you specify. Optionally, this job can also automatically update the statistics for the selected objects before and after reorganization and rebind any packages that depend on the reorganized objects.

These "autonomic" space maintenance jobs allow you to truly "set and forget" the task of monitoring and reorganizing the space of your critical databases.
Space Analyst Expert Guide

IBM DB2 fragmentation can be broken down into two basic types:

- **External Fragmentation** - this situation exists when indexes have a logical order, based on their key value, which does not match the actual physical order inside the database file that houses the index. When an ordered scan is performed on an index, I/O performance is enhanced when the physical order of the database pages matches the logical order of the index. This is because the disk head can scan in one direction instead of moving back and forth to obtain the needed information.

- **Internal Fragmentation** - this situation exists when tables and indexes aren't utilizing space as efficiently as they should. The amount of wasted space in the database pages artificially inflates the size of the table or index and causes more I/O to be performed than would otherwise be needed if the object were compacted.

Keep in mind that fragmentation is not a performance factor at all once data reaches the IBM DB2 memory caches.

Related Topics

- [Environments That Benefit From Fragmentation Elimination](#)
- [Diagnosing Fragmentation](#)
- "I've Reorganized, But It Didn't Help!"
- [Forwarded Rows](#)

**Environments That Benefit From Fragmentation Elimination**

One important point must be understood regarding fragmentation: not every situation benefits from fragmentation removal. Before you invest a lot of time and effort into diagnosing and attempting the removal of fragmentation, you first need to see if jumping through hoops will actually improve your database's availability and performance.

At the highest level, the environments that benefit the most from fragmentation removal are read-intensive databases where large indexes are being frequently scanned. There are a couple of reasons why this is the case.

In most OLTP environments, data retrieval tends to be mostly selective, which negates most of the bad effects of external fragmentation. OLTP environments also often benefit from internal fragmentation because it's smart to leave room on index pages that are being added to and updated frequently. Free index page space helps avoid the dreaded page split where a page is divided in two due to the fact that incoming clustered index data can't logically fit on a needed page, so IBM DB2 makes room by creating two index pages out of one. Administrators oftentimes specify a low FILLFACTOR for their indexes to create internal fragmentation and hopefully avoid page splits.

Data warehousing or databases with many resource-intensive/scanning queries are another matter. These environments will likely benefit from fragmentation removal. One reason is that the indexes tend to be larger, and therefore are not cached by IBM DB2 as easily as small indexes. Remember that fragmentation has no effect on performance once the data is nestled safely in IBM DB2's memory caches.

Another reason why these environments benefit from fragmentation removal is the effect fragmentation has on IBM DB2's read ahead manager. The read ahead manager helps queries that perform large scans by scanning index pages (and data pages) that it believes will be read and placing them into memory before they are actually needed. Naturally, this process can reduce overall scan times because data read in memory is many times faster than when that same data must be retrieved from disk. However, fragmentation affects how well the read ahead manager works. The read ahead manager will dynamically adjust the size of I/O it performs based on the actual physical order of the data and index pages on the server. When little external fragmentation is present, the read ahead manager can work very efficiently because it can read larger blocks at a time, whereas excessive fragmentation causes the read ahead manager to read smaller blocks of data. The end result when fragmentation is present is less overall throughput.
SQL execution patterns are something else to examine. Queries that perform large index scans are the ones that should see performance improvements when fragmentation is removed in a database. In DBArtisan, you can visually set I/O statistics on in the ISQL facility and examine queries to determine the amount of physical and logical read activity. Queries that show physical and read ahead I/O may see their response times drop when fragmentation has been removed.

Diagnosing Fragmentation

If you have an environment that will benefit from reducing fragmentation, how do you go about recognizing it? There are a number of metrics and factors to consider before you can intelligently start creating reorganization plans.

When diagnosing fragmentation, pay particular attention to the following metrics:

- **Extent Fragmentation** - this highlights any gaps (lack of contiguous order) in extents and indicates the presence of external fragmentation. Percentages of 25-50% or more are not a favorable reading.

- **Logical Fragmentation** - this metric represents the percentage of pages that are out of physical order. High values (greater than 25%) for this metric may mean that index scan performance is not what it could be. Note that this statistic is meaningless for tables without clustered indexes (heap tables).

- **Average Page Density** - this metric represents the fullness of the data or index pages, and is an indicator of internal fragmentation. Remember that the more full a data or index page is, the less I/O needed by IBM DB2 when performing scans of tables or indexes. High values are good here, with anything below 50% being suspect. Low values for this metric often indicate the presence of page splitting. Keep in mind, however, that internal fragmentation isn't necessarily bad in OLTP environments where large FILLFACTORs are often specified. Also understand that small objects (those with little or no data) will likely show low readings because they simply don't consume enough space in a database page. You can ignore these objects.

- **Extent Switches** - in a perfect world, this will equal # of object extents - 1. Higher values (values many times higher than an object's extent count) can indicate external fragmentation.

- **Scan Density** - this is computed by dividing the optimum number of extent switches by the actual number of extent switches. This percentage can be misleading, however, if the object spans more than one database file and should not be considered in such an event. Values close to 100% are best.

As you can see, you can't just blindly diagnose fragmentation in IBM DB2, but instead have to evaluate a number of factors.

"I've Reorganized, But It Didn't Help!"

Sometimes, a DBA will reorganize a table or index and no measurable benefits are noticed afterwards. The same or worse fragmentation metrics are observed for the objects in question. What could have gone wrong?

When attempting the reorganization of either tables or indexes, you have to understand that only objects that are of a significant size will show improvement from a reorganization. IBM DB2 for Linux, Unix, and Windows has stated that objects with less than 1,000 pages will oftentimes offer no performance improvements whatsoever from being reorganized.

Other points to keep in mind include:

- Tables without indexes receive zero benefit (naturally) from running either a DBCC DBREINDEX or a DBCC INDEXDEFRAG. The only way to reorganize a table without indexes is to create a clustered index on the table and then immediately drop it.

- Heap tables should not be diagnosed as having fragmentation through the logical scan fragmentation measure. This metric is meaningless for such tables.
• As mentioned above, small objects seldom benefit from reorganizations. For example, the page density for a table with only 10 rows in it will likely not be high, and reorganizing it won’t change things one bit. You need to keep such things in mind when determining whether to reorganize an object or not. Space Analyst keeps you informed of not only reorganization metrics but object metrics (extent size, row counts, etc.) as well so you can make smart reorganization choices.

• Lack of contiguous free space can result in all fragmentation not being eliminated. Without large amounts of contiguous free space, reorganization operations may have to reuse other areas of space within the database files. The end result is oftentimes an index being reorganized, but still having an amount of logical scan fragmentation.

Forwarded Rows
IBM DB2 will move rows in a table under certain conditions. One situation might arise when you update a row in a table that has a variable-length column to a larger size that will no longer fit on its original page. Another situation would be if IBM DB2 moves a row when the clustered index column changes. When IBM DB2 creates a forwarding pointer, the pointer remains in place unless one of two things happens. The first is when a row shrinks enough to move back to its original location. The second is when the entire database shrinks. When a database file shrinks, IBM DB2 will reassign the row identifiers, which are used as the row locators, so the shrink process never generates forwarded rows.

Forwarded records can reduce performance at times because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself. But when does row forwarding become a problem? For example, just because a table has one hundred forwarded rows, does that mean a performance problem exists? The answer is: it depends. If the table has one hundred rows in it, then yes, a problem exists - 100% of the table suffers from forwarded rows. If the table has three million rows, then the forwarded row situation is likely not causing much fuss in the database.
Backup Analyst

Welcome to the Embarcadero Backup Analyst. Backup Analyst increases the efficiency of performing and managing database backups. Backup Analyst can significantly improve operational efficiency and produce significant savings for your organization. A key capability of Backup Analyst for SQL Server is its ability to compress database backup images significantly. Because it can compress backup images by up to 90 percent or more, it will reduce your storage costs. The Backup Analyst Backup Wizard and Backup Analyst Recovery Wizard make backing up and recovering SQL Server databases straightforward and easy.

NOTE: Backup Analyst does not have to be run from within DBArtisan. It is fully supported via scripting and command line functions so you can use the Backup Operations and Restore Operations anywhere.

The table below describes the major sections of Help for the Backup Analyst.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Provides information about installing, opening, and licensing Backup Analyst.</td>
</tr>
<tr>
<td>Application Basics</td>
<td>Provides conceptual information about Backup Analyst’s product design and navigation.</td>
</tr>
<tr>
<td>Using Backup Analyst for Microsoft SQL Server</td>
<td>Shows how to use each component of the application. Using Backup Analyst breaks each process and procedure into simple step-by-step instructions.</td>
</tr>
<tr>
<td>Backup Analyst for Microsoft Command Line Reference</td>
<td>Describes the wide range of operations that Backup Analyst supports for backing up and restoring SQL Server databases using the command line interface.</td>
</tr>
</tbody>
</table>
Welcome to Backup Analyst

Establishing a reliable backup and recovery plan for production databases is perhaps the most important responsibility of any database administrator (DBA). Indeed, many DBAs find themselves spending a large share of their time implementing, refining and monitoring backup operations to safeguard the availability of critical database applications.

A number of major trends suggest that managing database backups will become a more pressing issue for professional DBAs over time because corporate databases are growing so quickly. Industry analysts estimate that corporate data is doubling every 2 - 2.5 years. As databases become larger, they become progressively harder to backup and recover as time windows become tighter and the risk of failed backups increases. Every DBA knows that it is far harder to backup a 500 GB database than one that is only 50 GB in size. In addition, DBAs find themselves managing more database applications and, therefore, more database backups. In the quest to compete more effectively, companies are automating more business processes and adopting a wider variety of applications, which means that more production databases get deployed in the enterprise. Finally, organizations have renewed their focus on controlling IT costs and raising staff productivity. The confluence of these factors has thrust DBAs into the uncomfortable position of managing a larger number of critical databases, each of which is growing rapidly in size, with limited incremental resources.

Backup Analyst increases the efficiency of performing and managing database backups. Backup Analyst for SQL Server can significantly improve operational efficiency and produce significant savings for your organization. A key capability of Backup Analyst for SQL Server is its ability to compress database backup images significantly. Because it can compress backup images by up to 90% or even more, it will reduce your storage costs. In addition, Backup Analyst for SQL Server can significantly cut the time required to perform backup and recovery operations and the network bandwidth consumed by them. As a result, DBA teams can achieve substantially greater productivity as well as better operational results in managing backup operations, freeing them to spend time on more strategic tasks.

**Reliable** - Backup Analyst for SQL Server extends the native functionality for performing backup and recovery in SQL Server to incorporate compression and encryption. Backup Analyst is tightly integrated with SQL Server and leverages native functionality of the database wherever possible in order to maximize product reliability and to minimize its footprint.

**Fast and Efficient** - Customers will find that they can reduce backup file sizes and total processing time substantially by using Backup Analyst for SQL Server. In fact, they will find that the gains are proportionally better as database backups get bigger. Backup Analyst for SQL Server utilizes highly efficient and reliable compression algorithms in order to achieve the best balance of compression ratio vs. compression processing time.

**Scalable** - Customers can deploy Backup Analyst for SQL Server to a large number of SQL Server instances very efficiently. The product consumes very little computing resource and works very efficiently with large database servers.

**Manageable** - Backup Analyst for SQL Server builds on the strength of Embarcadero's DBArtisan family of database administration products. From the DBArtisan Workbench, users will find that it is easy to manage the disaster recovery plans for even a large number of SQL Servers.

**Economical** - Because of its reliability, efficiency, scalability and ease of management, customers will find that Backup Analyst for SQL Server yields significant and tangible savings in storage costs, bandwidth requirements, processing time and manpower.

For more information, see [Backup Analyst Functionality](#).

### Backup Analyst Functionality

The Backup Analyst includes the following features:

- Backup/Recovery Management
• **Backup Wizard** that offers:

  • Increased backup speed for large databases
  • Backup compression
  • Backup encryption
  • Support for full, differential, file/file group, transaction log, table, and indexed view backups
  • Scheduling

• **Recovery Wizard** that offers:

  • Increased backup speed for large databases
  • Backup compression
  • Backup encryption
  • Support for full, differential, file/file group, transaction log, table, and indexed view backups
  • Scheduling

• Notifications and scheduling via SQL Agent

• **Reports**

• **Alerts**

• Basic database/object maintenance functions including:

  • **Database Wizard for Microsoft SQL Server**
  • **Open**
  • **Extract**
  • **Update Statistics**
  • **DBCC**

---

**Backup Analyst System Requirements**

The basic requirements for installing and using Backup Analyst for SQL Server are quite modest. Essentially, you must be running a supported version of SQL Server and have a small amount of disk space on the SQL Server machine to store the backup agent code.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| Supported Versions of SQL Server | SQL Server 2005  
SQL Server 2000  
SQL Server 2000 64-bit (forthcoming)  
Yukon (forthcoming) |
| Disk Space               | 5 MB                                                                                                                                 |
| Windows Security         | Privileges to install software on the SQL Server machine                                                                                   |
| SQL Server Security      | Must login to the SQL Server using a database account that is either system administrator ("sa") or is a member of the 'sysadmin' fixed server role. |

For more information, see:

• **Backup Analyst Installation**
What You Should Know Before Installation

When you install the Analyst, several Embarcadero-generated stored objects are pushed onto the database server. These objects are as follows:

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>embt_ba_create_alerts</td>
</tr>
<tr>
<td>embt_ba_create_embt_backupinfo</td>
</tr>
<tr>
<td>embt_ba_create_job_categories</td>
</tr>
<tr>
<td>embt_ba_list_backupdevices</td>
</tr>
<tr>
<td>embt_ba_list_db_filegroups</td>
</tr>
<tr>
<td>sp_BackupRestoreDrop</td>
</tr>
<tr>
<td>sp_BaBackupRestoreInstall</td>
</tr>
<tr>
<td>sp_embt_ba_database_backup_history</td>
</tr>
<tr>
<td>sp_embt_ba_database_backup_history_overview</td>
</tr>
<tr>
<td>sp_embt_ba_database_drill_down</td>
</tr>
<tr>
<td>sp_embt_ba_database_file_backup_history</td>
</tr>
<tr>
<td>sp_embt_ba_database_file_backup_history_overview</td>
</tr>
<tr>
<td>sp_embt_ba_filegroup_summary</td>
</tr>
<tr>
<td>sp_embt_ba_file_summary</td>
</tr>
<tr>
<td>sp_embt_ba_list_database_status</td>
</tr>
<tr>
<td>sp_embt_ba_root_db_summary</td>
</tr>
<tr>
<td>sp_embt_ba_table_summary</td>
</tr>
<tr>
<td>sp_embt_ba_trans_log_summary</td>
</tr>
<tr>
<td>sp_embt_ba_version</td>
</tr>
<tr>
<td>sp_execBackupRestore</td>
</tr>
</tbody>
</table>

Backup Analyst Installation

Installing Backup Analyst for SQL Server should only take several minutes for each database instance. The installation requires you to follow a simple setup wizard that walks you through the process of installing and registering the backup agents on an instance of SQL Server.

The Backup Analyst for SQL Server requires two components: a console that installs as part of the DBArtisan 8.0 and higher client, and a set of EMBT proprietary extended stored procedures that are installed on the target SQL Server server.

**NOTE:** The client-side console can only be used against those SQL Servers that are licensed to run the proprietary extended stored procedures. A Backup Analyst license applies to one SQL Server, but it covers all databases running on that server.

**CAUTION:** Backup Analyst server components must be installed by the SQL Server service account owner who has administrative rights on the server machine.

In order to complete the wizard, you must:
- Know the name of the target SQL Server instance.
- **Have sufficient Windows privileges to install software on the SQL Server machine.** You must have local admin rights on the server machine.

**NOTE:** The Backup Analyst is installed and started on the SQL Server machine. You do not need to install DBArtisan on the server machine, just the Backup Analyst component.

- Obtain a login and password to that SQL Server instance with sufficient privileges to install the agents (see Security Requirements).
Installing Backup Analyst
To install Backup Analyst, do the following:

1. Go to **Program Files\Embarcadero\DBA851**, and copy the **BackupAnalyst** folder.
   
   The BackupAnalyst folder includes the Backup Analyst components necessary for the server-side installation of Backup Analyst.

2. Go to **Program Files\Microsoft SQL Server** on the machine that hosts the target SQL Server instance and paste the **BackupAnalyst** folder in the target SQL Server machine folder.

3. In **Program Files\Microsoft SQL Server\BackupAnalyst**, double-click the **setup.exe** icon.
   
   DBArtisan opens the Embarcadero Backup Analyst InstallShield Wizard that guides you through the installation of Backup Analyst. You

4. Complete the wizard panels, and then click **Install**.
   
   DBArtisan installs the Backup Analyst engine on the SQL Server machine(s) that you will be testing against.

DBArtisan also performs the following tasks:

- Copies extended stored procedure DLLs and dependencies to target SQL Server directory
- Creates extended stored procedures into any target database
- Creates a standard stored procedures used by the product into any target database
- Creates a backup device if needed
- Creates a new role that controls security
- Grants a new role to all system administrator logins
- Creates a new system job categories (optional)
- Creates new alerts (optional)

**NOTE:** When you first install an evaluation copy of DBArtisan and the Backup Analyst engine, you can use the tools for 14 days. After that time, a permanent license is needed.

Uninstalling Backup Analyst
To uninstall Backup Analyst, do the following:

1. In **Program Files\Microsoft SQL Server\BackupAnalyst**, double-click the **setup.exe** icon.
   
   DBArtisan opens the Embarcadero Backup Analyst InstallShield Wizard.

2. Select **Remove**.

3. Click **Next**.

4. Select **Remove**.

   The wizard uninstalls all Backup Analyst dlls, exes, and scripts.

5. Click **Finish**.
Repairing Backup Analyst
The InstallShield Wizard lets you repair installation errors in BackUp Analyst. Fixes missing or corrupt files, shortcuts, and registry entries.

1. In Program Files\Microsoft SQL Server\BackupAnalyst, double-click the setup.exe icon.

DBArtisan opens the Embarcadero Backup Analyst InstallShield Wizard.

2. Select Repair.

3. Click Next.

4. Select Install.

The wizard reinstalls Backup Analyst.

5. Click Finish.

Starting the SQL Agent
To use Backup Analyst, your SQL Server SQL Agent should be running on the machine where you are installing the components. You can start it via the Services window.

1. Select Start, Settings, Control Panel, Administrative Tools, and then Services.

2. Right-click SQLSERVERAGENT, and select Start.

Opening Backup Analyst
To open the Backup Analyst, do the following:

1. On the Analyst menu, click Backup Analyst.

DBArtisan opens the Backup/Recovery Manager.

Licensing Backup Analyst
Backup Analyst includes a 30-day trial period in which you can use the application before you must license it. To use Backup Analyst after 30 days, you must obtain a Workstation license for your machine. To continue using the application without interruption, we recommend that you license it as soon as possible.

You can request Backup Analyst Workstation license from Embarcadero Technologies by e-mail or fax.

To add the new license to Backup Analyst, do the following:

1. Go to Program Files/Embarcadero/ BackupAnalyst, and then paste the license.txt into the folder.

   **TIP:** To check the status of your Backup Analyst license, click the Server Information button on the Backup Analyst tool bar.

For more information, see Contacting Embarcadero Technologies Technical Support.

Single-User Mode
Backup Analyst works with single-user mode databases with the following restrictions:

- While in single-user mode, Backup Analyst does not let you Backup or Restore a master database using the UI.
• While in single-user mode, use the command line interface to Backup or Restore a master database.
Backup Analyst Application Basics

Backup Analyst includes the following features:

- Backup/Recovery Management
- Backup Wizard that offers:
  - Increased backup speed for large databases
  - Backup compression
  - Backup encryption
  - Support for full, differential, file/file group, transaction log, table, and indexed view backups
  - Scheduling
- Recovery Wizard that offers:
  - Increased backup speed for large databases
  - Backup compression
  - Backup encryption
  - Support for full, differential, file/file group, transaction log, table, and indexed view backups
  - Scheduling
- Notifications and scheduling via SQL Agent
- Reports
- Alerts
- Basic database/object maintenance functions including:
  - Database Wizard for Microsoft SQL Server
  - Open
  - Extract
  - Update Statistics
  - DBCC

Backup Analyst Product Design

The product leverages agent technologies that extend the native backup and restore functionality of SQL Server to include compression and encryption. These agents, which are installed on each supported instance of SQL Server, have a very small footprint, requiring only several megabytes of storage. In order to integrate the extended functionality tightly with SQL Server, Backup Analyst registers these agents as extended stored procedures that can be called from inside the database.

The parameters passed to the extended stored procedures closely follow the patterns established by the native SQL Server backup and restore commands. As a result, users should find the product easy to learn and to integrate with established backup routines. The subsections that follow document the arguments for calling these extended stored procedures.

User Interface

Backup Analyst may be run within DBArtisan, Embarcadero's award-winning, cross-platform database administration tool. Customers using Backup Analyst through DBArtisan will find a rich interface that provides a central console to manage all their backup and recovery needs.
The **Backup Analyst Backup Wizard** and **Backup Analyst Recovery Wizard** make backing up and recovering SQL Server databases straightforward and easy. The **Backup Analyst Alerts Wizard** lets you create alerts and notifications so no backup or recovery problem goes overlooked. All backup and recovery jobs can be filtered, managed, and tracked in one place. And informative backup and recovery reports can be created and scheduled to run in an automated fashion so everyone is kept informed of backup operations.

**NOTE:** Backup Analyst does not have to be run from within DBArtisan. It is fully supported via scripting and command line functions so you can use the **Backup Operations** and **Restore Operations** anywhere.

### Backup Analyst Toolbar and Menus

**Toolbar:**

![Backup Analyst Toolbar](image)

**Main Toolbar Menu**

![Backup Analyst Main Toolbar Menu](image)
Command Menu

The Backup Analyst Toolbar and menus help you perform the most common operations in Backup Analyst like opening a Backup Wizard and Recovery Wizard, creating reports, and using DBArtisan functionalities including DBCC and Update Statistics.

Backup/Recovery Management

Backup/Recovery management provides instant access to major backup and recovery needs, and key database objects. The Backup/Recovery management facilities also display the current status and events of objects, jobs, and other items.

Backup/Recovery management includes the following components:

<table>
<thead>
<tr>
<th>Backup Analyst Explorer</th>
<th>Backup Analyst Server Information Panel</th>
<th>Backup Analyst Toolbar</th>
</tr>
</thead>
</table>

Backup Analyst Explorer

The Backup Analyst Explorer lets you navigate the objects of your SQL Server as they relate to backup and recovery. For example, you can manage databases, files and filegroups, transaction logs, and tables, as well as system objects like backup devices, the SQL Server Agent, jobs, and alerts/notifications.

The Backup Analyst Explorer includes the following nodes:

- Server Node
- Databases Node
- Individual Database Nodes
- File Groups Node (and subnodes)
Server Node
The SQL Server name is at the top of the Server Node. The default landing page is displayed when this node is selected.

The upper portion of the landing page displays the current status of various backup/recovery activities including the number of backups and recoveries running, databases needing backup, and job successes and failures. The lower pane displays the summary of the last 'N' backups for all databases on the server (where 'N' is an option defined in the global options dialog box).

Databases Node
On the Explorer tree, green database nodes indicate the database is online. A red node indicates the database is offline. When you select the databases node, the right pane displays the Database Summary of backups. You see when each database was created, its online status, size, the log size, the last backup date and type, compressed and uncompressed sizes, and whether or not backup/restore jobs are actively running.

Individual Database Nodes
When you select a specific database's node in the Explorer, Backup Analyst displays its database properties in the right pane. The top portion displays basic metadata regarding the database in graphic format, while the bottom displays a backup history for the database.

File Groups Node
When you select the Filegroups node in the Explorer, Backup Analyst displays both the File Group Summary and the Filegroup/File Backup Summary in the right pane. The File Group summary identifies each group (for example, Log and Primary groups) and their capacities to grow, their sizes, and other helpful information. The File Backup summary gives you specific information about backups including the backup type, the run time, and so forth. Additional summaries are provided for subordinate nodes as you click them:

NOTE: The list that follows are the default nodes. You can add nodes and subnodes with the Database Wizard for Microsoft SQL Server.

- Primary,
- Files,
- Tables

Transaction Log
The Transaction Log node, when selected, graphically displays transaction log properties in the right pane. The graph enables you to see how much physical space is used and how much is available. You also see a transaction log summary that includes information about when a backup job started and finished, along with its expiration date and other pertinent operation details.
Functions available from the Backup Analyst Databases Explorer Nodes include:

<table>
<thead>
<tr>
<th>Backup Wizard</th>
<th>Recovery Wizard</th>
<th>Database Wizard for Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Extract</td>
<td>Report</td>
</tr>
<tr>
<td>Update Statistics</td>
<td>DBCC</td>
<td></td>
</tr>
</tbody>
</table>

These functions are all available from the Backup Analyst toolbar’s command menu.

Backup/Recovery Management Node
When you select the Backup/Recovery Management node in the Explorer, Backup Analyst displays default landing page for the server, which is the same information you see when you select the server node.

Available Functionality
- Start SQL Server Agent - starts the SQL Server Agent
- Stop SQL Server Agent - stops the SQL Server Agent

Select the Explorer Backup/Recovery Management node and use the command menu to effect these functions.

Backup Devices Node
When you select the Backup Devices node in the explorer, Backup Analyst displays the Backup Devices Summary in the right pane.

Available Functionality

<table>
<thead>
<tr>
<th>Backup Device Wizard for Microsoft SQL Server</th>
<th>Open</th>
<th>Extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>View Contents</td>
<td></td>
</tr>
</tbody>
</table>

Backup Analyst Server Information Pane
The Backup Analyst Server Information window displays information comparable to what you see when you open the About window for DBArtisan. You see the Backup Analyst version number along with pertinent license information.

You can open this information panel from the Backup Analyst drop-down menu on the main toolbar or by clicking on the Backup Analyst toolbar.
Using Backup Analyst for Microsoft SQL Server

Using Backup Analyst for Microsoft SQL Server covers how to use each component of the application. The Using chapter breaks each process and procedure into simple step-by-step instructions.

NOTE: Backup Analyst is fully contained within DBArtisan. Backup Analyst can be open while you work with other components of DBArtisan.

Backup Analyst for Microsoft SQL Server Functionality

Backup Analyst lets you create SQL Server backup jobs. Backup Analyst extends standard SQL Server backup functionality by providing specialized features that offer backup file compression that results in storage savings, as well as security encryption that supplies special protection for sensitive database information.

The Backup Analyst supports the following functionality:

- **Backup Wizard**
- **Recovery Wizard**
- **Alert Wizard**
- **Options**
- **Report**
- **Server Free Space**

Backup Analyst Backup Wizard

Implementing a plan to back up critical databases is perhaps the most important job for any database administrator. Without a sound disaster recovery plan, your company has no safeguards against hardware and software failures.

The Backup Analyst's Backup Wizard lets you backup your databases without knowing the underlying syntax. It provides support for backing up databases as well as transaction logs, including commands to stripe the backup across multiple devices.

The Backup Analyst Backup Wizard lets you:

- Create a new backup operation.
- Re-use a previously defined operation.
- Specify the target server, database, and type of backup operation.
- Specify backup device type.
- Specify how to report the progress of the backup operation.
To Open the Backup Analyst's Backup Wizard:

1. With the Backup Analyst running:
   - On the Backup Analyst Explorer, right-click the database/node you want to back up and select Backup Wizard.
   - OR
   - From the Backup Analyst drop-down menu, select Backup Wizard.
   - OR
   - Select Backup Wizard from the Backup Analyst toolbar.

In the following sections, you'll find all the fields you'll encounter as you create or edit a backup operation:

Job Category Wizard

Database Backup Wizard

**Job Category Wizard**

The Job Category Wizard panels appear ONLY when you are performing a backup operation for a MS SQL datasource. The job categories are relevant to the Backup Analyst and are described below:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Backup</td>
<td>Any backup job which is a full database, differential database, file and file group, transaction log or table backup which does not include the compression or encryption options.</td>
</tr>
<tr>
<td>Advanced Backup</td>
<td>Any backup job which is a full database, differential database, file and file group, transaction log or table backup which includes at least one of the compression or encryption options.</td>
</tr>
<tr>
<td>Recovery</td>
<td>Any recovery job created by Backup Analyst.</td>
</tr>
</tbody>
</table>
| Would you like to create the 16 popular SQL Server Alerts for common Backup and Recovery Errors? | Yes - Opens the Alerts wizard.  
No - Opens the Database Backup Wizard. |

**Backup Wizard**

The first panel of the Database Backup Wizard welcomes you to the Backup Wizard.

Subsequent panels of the wizard are as follows:

<table>
<thead>
<tr>
<th>Playback or New Operation</th>
<th>Database Selection</th>
<th>Backup Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Style</td>
<td>Backup Location</td>
<td></td>
</tr>
<tr>
<td>Backup Options</td>
<td>Backup Operation Summary</td>
<td>Backup Operation Execution Selection</td>
</tr>
</tbody>
</table>
**Playback or New Operation**

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Dump Operation</td>
<td>This initiates the process of creating a new backup operation.</td>
</tr>
<tr>
<td>Previous Dump Operation</td>
<td>If you want to replay a backup operation, when you select the radio button, the browse function is enabled so you can find the .DBDump file you want to re-use.</td>
</tr>
</tbody>
</table>

**Database Selection**

Here you select the database you want to backup from a drop-down list and name the backup operation. You can keep the default name for the operation or replace it with one you type. The job description is optional and you can accept the default or write another.

**Backup Type**

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Database</td>
<td>Backs up the entire database. The size of the database is listed in parentheses.</td>
</tr>
<tr>
<td>Differential Database</td>
<td>Records all modifications made to a database after the last database backup, not just the changes made after the last differential database backup.</td>
</tr>
<tr>
<td>File and File Group</td>
<td>Select to specify a file or file group.</td>
</tr>
<tr>
<td>Transaction Log</td>
<td><strong>Full Truncate</strong>: Backs up the inactive portion of the transaction log. The size of the transaction log is listed in parentheses.</td>
</tr>
<tr>
<td></td>
<td><strong>With No Log</strong>: Removes the inactive portion of the transaction log without making a backup of it. You should only use this command when you run out of room in the database and cannot issue DUMP TRAN WITH TRUNCATE_ONLY.</td>
</tr>
<tr>
<td></td>
<td><strong>Full No Truncate</strong>: Makes it possible to dump a transaction log even if the database is inaccessible. You can use this option when the data portion of your database is damaged and the transaction log resides on a separate device.</td>
</tr>
<tr>
<td></td>
<td><strong>With Truncate Only</strong>: Removes the inactive portion of the transaction log without making a backup of it. You should use this command prior to backing up a database.</td>
</tr>
</tbody>
</table>

**Backup Style**

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Analyst Backup</td>
<td>Select to use the Backup Analyst for your backup, assuming the Backup Analyst is installed on the database you are backing up. Also lets you compress the backup and apply an encrypted password.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE</strong>: 128-bit encryption is supported</td>
</tr>
<tr>
<td>Standard SQL Server Backup</td>
<td>Select to continue the Database Backup Wizard.</td>
</tr>
</tbody>
</table>
Backup Location

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Device</td>
<td>Depending on what is available to you, you can choose to backup to a disk, floppy, pipe, or tape device.</td>
</tr>
<tr>
<td>Disk Directory and Name</td>
<td>Lets you specify a backup location on your machine or click the browse button and select a location.</td>
</tr>
<tr>
<td>View Contents</td>
<td>Click to open the Backup Media Contents Dialog Box.</td>
</tr>
</tbody>
</table>

Backup and Media Information

Here, for MS SQL, you specify the backup set name, backup description, backup set password, media name, media description, and media name password.

Backup Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expire Date</td>
<td>The date when the backup set expires and can be overwritten. If supplied as a variable (@takeover), this date is specified as either a string constant (@takeover = date), as a variable of character string data type (except for the text or text data types), a small datetime, or datetime variable, and must follow the configured system datetime format.</td>
</tr>
<tr>
<td>Retain Days</td>
<td>The number of days that must elapse before this backup media set can be overwritten. If supplied as a variable (@days_var), it must be specified as an integer.</td>
</tr>
<tr>
<td>Skip backup set expiration and name checking before overwriting</td>
<td>Microsoft SQL Server will skip the set expiration and name checking before overwriting.</td>
</tr>
<tr>
<td>Overwrite all backup sets on the device</td>
<td>Microsoft SQL Server will overwrite the contents of the device.</td>
</tr>
<tr>
<td>Format the entire media before backup operation</td>
<td>Will format the entire media before the backup operation starts.</td>
</tr>
<tr>
<td>Read and verify the integrity after the backup after finish</td>
<td>After the operation completes, the integrity of the backup is verified.</td>
</tr>
<tr>
<td>Backup Threads</td>
<td>0 is the default. Zero enables the server default value.</td>
</tr>
</tbody>
</table>

Backup Operation Summary

Here you indicate whether or not you want to save the operation, and if so, where the file should be saved. Also displayed are the elements of the function you are creating. You can go back to earlier panels if you want to make changes before the job runs.
Backup Operation Execution Selection

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open the backup script in an ISQL editor</td>
<td>The backup script will be opened into a new ISQL session.</td>
</tr>
<tr>
<td>Perform the backup now</td>
<td>The backup will be executed immediately.</td>
</tr>
<tr>
<td>Schedule the backup for later execution</td>
<td>The backup will be scheduled to run at a specified date/time.</td>
</tr>
</tbody>
</table>

Results Summary
If you selected to execute the job immediately, you see a summary of the results of the operation.

For more information, see:

Executing a Backup Script in the ISQL Window

Executing a Backup Script in the ISQL Window
To execute the actual command DBArtisan uses to backup your database in the ISQL Window, do the following:

1. On Backup Operation Execution Selection, click Open.
   
   DBArtisan opens the ISQL Window, which is used to execute SQL and Transact-SQL commands. The ISQL Window displays the procedure call that Backup Analyst makes to back up SQL Server databases, complete with the parameters needed to perform the backup.

2. To execute your backup, on the ISQL Editor toolbar, click Execute.
   
   DBArtisan completes your backup and displays a message "Procedure Succeeded" on the Results tab of the ISQL Editor.

Backup Analyst Recovery Wizard
To both validate your back ups and practice the disaster recovery process, you should regularly restore your database dumps. If possible, every day, you should restore your databases on a separate server to enhance system availability. In this manner, you can maintain a warm backup of your database guarding against downtime from system failure.

The Backup Analyst Recovery Wizard lets you:

- Create a new load operation.
- Reuse a previously defined operation.
- Specify the target server, database, and type of restore operation.
- Specify the recovery completion state.
- Specify how to report the progress of the load operation.
- Specify which file to load from the device.
- Specify whether to perform safety checks before replacing the database.
- Save the definition of the dump operation for future re-use.
Completing the Backup Analyst Recovery Wizard
1. On the Backup/Recovery Manager, right-click the target node.
2. Select Recovery Wizard.
   DBArtisan opens the Recovery Wizard.
3. Complete the wizard panels, and then click OK.
   DBArtisan recovers the database.

For more information, see Database Recovery Wizard.

Backup Analyst Alerts Wizard
The Alert Wizard lets you create and configure popular SQL Server Alerts for backup and recovery operations. Alerts inform the user about actions that need to be taken or about events that are occurring and that do not require any action. For example, there is an alert that indicates a backup is in progress. Once the backup is complete, the alert is cleared.

To create backup alerts:
1. Connect to a SQL Server database.
2. From the Utilities menu, select Database Backup.
   The Job Categories wizard opens and on the second panel of this wizard, if you elect to create Alerts, the Alert Wizard opens.
   OR
1. Connect to a SQL Server database and open the Backup Analyst.
2. From the Backup Analyst drop-down menu on the toolbar, select Alerts Wizard.
   OR
   From the Command menu on the toolbar, select Alerts Wizard.

   **NOTE:** If an alert already exists and is mapped to one of the existing SQL Server Error numbers, the alert will not be created.

   **NOTE:** Be aware that when you work with the Alerts, Operators, or Jobs features and click Help, you connect directly to SQL Server Books online. Embarcadero Technologies is not responsible for the contents of the SQL Server Books materials.

SQL Server Agent reads the application log and compares events to alerts that you defined. When SQL Server Agent finds a match, it fires an alert. DBArtisan lets you define alerts to respond to SQL Server events by notifying operators, forwarding the event to another server, or executing a job. These administrative tasks runs automatically if the SQL Server Agent service has been started.

When you select the Alerts node on the Database Explorer, DBArtisan displays all alerts for the server in the right pane of the Explorer window. The column headings in the right pane of the Explorer window provide important information about alerts. These headings are:

- Name
- Enabled
- Error
- Severity
• Last Occurred
• E-mail
• Panel
• Net Send
• Count

You can use the shortcut menu to access functionality associated with alerts. Shortcut menus let you:

• View Properties
• Create a New Alert
• Drop an Alert
• Find Object
• Select All
• Unselect All

NOTE: On the shortcut menu, when you click New, DBArtisan opens the Microsoft SQL Server Agent Alert Properties dialog box where you can enter the new alert information.

Backup Analyst Alert Wizard - Panel 1
The first panel of the wizard displays a message regarding the purpose of the wizard.

Backup Analyst Alert Wizard - Panel 2
This panel lets you select the alerts you would like to create for backup jobs.

NOTE: If there are alerts already defined using any of the SQL Server Errors, the wizard will not create the alert.

Backup Analyst Alert Wizard - Panel 3
This panel lets you select the alerts you would like to create for recovery jobs.

NOTE: If there are alerts already defined using any of the SQL Server Errors, the wizard will not create the alert.

Backup Analyst Alert Wizard - Panel 4
This panel confirms the creation of the alerts.

Backup Analyst Options
The Backup Analyst options enable you to specify some global settings for queries pertaining to your backup jobs.

To Open the Backup Analyst Options dialog box:
1 With the Backup Analyst open:
   Click the Options icon on the Backup Analyst toolbar,
   OR
   Select Options from the Backup Analyst drop-down menu.
The table below describes the options and functionality for the Backup Analyst Options dialog box:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grace period (days) for gap in backups</td>
<td>Lets you specify the query limit for when a database needs a backup (for example, database 'A' hasn't been backed up in 7 days).</td>
</tr>
<tr>
<td>Show last... backups</td>
<td>Lets you specify how much backup history to show (for example, show the last 100 backups) for all backup display grids.</td>
</tr>
<tr>
<td>Days to check for successful/failed jobs</td>
<td>Lets you specify the query limit as to how many days to look back to report on succeeded and failed jobs in the main landing page.</td>
</tr>
</tbody>
</table>

Backup Analyst Report
The Backup Analyst Report Wizard lets you:

- Create a new report or re-execute a previously saved report.
- Select the report to create and select the time frame for report data.
- Select the object(s) to be used for the report.
- Select options for the report.
- Save the report definition.
- Preview and then execute or schedule the report.

Important Notes
None

Completing the Backup Analyst for Microsoft Report Wizard
To complete the Backup Analyst Report Wizard, do the following:

1. On the Backup/Recovery Manager, right-click the target node.
2. Select Report.  
   DBArtisan opens the Report Wizard.
3. Complete the wizard panels, and then click OK.  
   DBArtisan creates the report(s).

Backup Analyst Report Wizard - Welcome
The first panel of the Backup Analyst Report Wizard welcomes you to the wizard.

For more information, see Backup Analyst Report Wizard.

Backup Analyst Report Wizard - Panel 2
The table below describes the options and functionality on this panel of the Backup Analyst Report Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Backup Analysis Report</td>
<td>Select to create a new analysis report.</td>
</tr>
</tbody>
</table>
For more information, see Backup Analyst Report Wizard.

Backup Analyst Report Wizard - Time
The table below describes the options and functionality on this panel of the Backup Analyst Report Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>Lets you select type of report:</td>
</tr>
<tr>
<td></td>
<td>Database Backup History - Provides information regarding all backups over the specified time period.</td>
</tr>
<tr>
<td></td>
<td>Database Restore History - Provides information regarding all restores over the specified time period.</td>
</tr>
<tr>
<td></td>
<td>Database File Backup Details - Provides information regarding the files backed up for a database over the specified time period.</td>
</tr>
<tr>
<td></td>
<td>Databases in Need of Backup - Provides any database that has not been backed up over the specified time period, along with a number of days since the last backup occurred.</td>
</tr>
<tr>
<td></td>
<td>Objects Modified in Transaction Log - Displays the objects and page counts that have been modified since the last transaction log backup.</td>
</tr>
<tr>
<td>Time period sample</td>
<td>Lets you enter a sampling time period.</td>
</tr>
<tr>
<td>Databases</td>
<td>Lets you select databases for the report.</td>
</tr>
</tbody>
</table>

For more information, see Backup Analyst Report Wizard.

Backup Analyst Report Wizard - Save
The table below describes the options and functionality on this panel of the Backup Analyst Report Wizard.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report File Name</td>
<td>Lets you type a report file name. Or click the browse button and locate an existing file.</td>
</tr>
<tr>
<td>Report Title</td>
<td>Lets you type a report title.</td>
</tr>
<tr>
<td>Save Operation</td>
<td>Lets you type a definition file name. Or click the browse button and locate an existing file.</td>
</tr>
<tr>
<td>Execute Now</td>
<td>Generates report immediately.</td>
</tr>
<tr>
<td>Schedule</td>
<td>The report will be scheduled to run at a specified date/time.</td>
</tr>
</tbody>
</table>

For more information, see Backup Analyst Report Wizard.

Server Free Space Monitor
The Server Free Space Monitor displays the total amount of physical free space available on the hard drives of a server. This information helps you ensure there is enough physical space for any new backup devices you may need for local storage of hot backups.
To Open the Free Space Monitor:

1. With the Backup Analyst open, select a datasource from the Explorer.
2. Choose **Server Free Space** from the **Command** drop-down menu.
   - OR
   - Right-click in the right pane and choose **Server Free Space**.
Backup Analyst Command Line Reference

If you do not want to use the Backup Analyst GUI, or if you want to wrap up your database backup and recovery into larger scripts, all of Backup Analyst's functionality is also available via an ISQL window within DBArtisan or from the command line using ETSBackupConsole.exe.

For specific command line operations and examples, see:

Backup Operations
Restore Operations
ETSBackupConsole.exe

For descriptions of all of the parameters for Backup and Restore operations, see:

Description of Backup Parameters
Description of Restore Parameters

Backup Operations
Backup Analyst supports a wide array of options for implementing a disaster recovery plan on SQL Server 2000 and 2005. Specifically, the product allows you to perform a:

• Full database backup
• Incremental database backup
• Transaction log backup
• Backup of specific filegroups or files
• Dynamically naming a backup job

The following sections detail the syntax for executing each of these operations:

• Backing Up an Entire Database
• Backing Up a Database Incrementally
• Backing Up a Database Transaction Log
• Backing Up Specific File Groups
• Backing Up Specific Files
• Dynamic Naming of a Backup Job
• Description of Backup Parameters

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

Backing Up an Entire Database
The syntax for backing up an entire database is as follows:

EXEC xp_embt_backup
   @Action = '{database}',
   @Database = {string},
@BackupFile = {string},
@Compress = {Boolean},
@DBLoginId = {string},
@DBLoginPassword = {string},
@DBPassword = {string},
@DBInstanceName = {string},
@BackupThreads = {long},
@ExpirationDate = {string | date},
@RetainDays = {long},
@Initialize = {boolean},
@MediaDescription = {string},
@MediaName = {string},
@MediaPassword = {string},   - SQL Server 2000 only
@BackupSetName = {string},
@BackupSetDescription = {string},
@BackupSetPassword = {string},   - SQL Server 2000 only
@Restart = {boolean},
@PercentCompleteNotification = {long},
@LogLevel = {long}

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

**Example**

Backing up the entire pubs database:

```
EXEC xp_embt_backup
    @Action = 'database',
    @Database = 'pubs',
    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat'
```

For more information, see [Backup Operations](#).

**Backing Up a Database Incrementally**

The syntax for backing up a database incrementally is as follows:

```
EXEC xp_embt_backup
    @Action = {'incremental'},
    @Database = {string},
    @BackupFile = {string},
```
@Compress = {Boolean},
@DBLoginId = {string},
@DBLoginPassword = {string},
@DBPassword = {string},
@DBInstanceName = {string},
@BackupThreads = {long},
@ExpirationDate = {string | date},
@RetainDays = {long},
@Initialize = {boolean},
@MediaDescription = {string},
@MediaName = {string},
@MediaPassword = {string},  - SQL Server 2000 only
@BackupSetName = {string},
@BackupSetDescription = {string},
@BackupSetPassword = {string},  - SQL Server 2000 only
@Restart = {boolean},
@PercentCompleteNotification = {long},
@LogLevel = {long}

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

**Example**
Performing a differential backup of the pubs database:

```sql
EXEC xp_embt_backup
    @Action = 'incremental',
    @Database = 'pubs',
    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat'
```

For more information, see [Backup Operations](#).  

**Backing Up a Database Transaction Log**
The syntax for backing up a transaction log is as follows:

```sql
EXEC xp_embt_backup
    @Action = ('log'),
    @Database = {string},
    @BackupFile = {string},
    @Compress = {Boolean},
```
@DBLoginId = 'sa',
@DBLoginPassword = 'password',
@DBPassword = {string},
@DBInstanceName = {string},
@BackupThreads = {long},
@ExpirationDate = {string | date},
@RetainDays = {long},
@Initialize = {boolean},
@MediaDescription = {string},
@MediaName = {string},
@MediaPassword = {string},  - SQL Server 2000 only
@BackupSetName = {string},
@BackupSetDescription = {string},
@BackupSetPassword = {string},  - SQL Server 2000 only
@TruncateLog = {'NoTruncate'},
@Restart = {boolean},
@PercentCompleteNotification = {long},
@LogLevel = {long}

NOTE: Mandatory parameters to an extended stored procedure are shown in bold text.

Example
Backing up of the transaction log of the pubs database:

EXEC xp_embt_backup
    @Action = 'log',
    @Database = 'pubs',
    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat',
    @TruncateLog = 'NoOption'

For more information, see Backup Operations.

Backing Up Specific File Groups
The syntax for backing up a database filegroup is as follows:

EXEC xp_embt_backup
    @Action = 'files',
    @Database = {string},
    @BackupFile = {string},
@DatabaseFileGroups = {string},
@Compress = {Boolean},
@DBLoginId = 'sa',
@DBLoginPassword = 'password',
@DBPassword = {string},
@DBInstanceName = {string},
@BackupThreads = {long},
@ExpirationDate = {string | date},
@RetainDays = {long},
@Initialize = {boolean},
@mediaDescription = {string},
@mediaName = {string},
@mediaPassword = {string}, SQL Server 2000 only
@BackupSetName = {string},
@BackupSetDescription = {string},
@BackupSetPassword = {string}, SQL Server 2000 only
@Restart = {boolean},
@PercentCompleteNotification = {long},
@LogLevel = {long}

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

**Example**
Backing up the primary filegroup of the pubs database:

EXEC xp_embt_backup
   @Action = 'files',
   @Database = 'pubs',
   @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat',
   @DatabaseFileGroups = 'PRIMARY'

For more information, see [Backup Operations](#).

**Backing Up Specific Files**
The syntax for backing up a database file is as follows:

EXEC xp_embt_backup
   @Action = ('files'),
   @Database = {string},
@BackupFile = {string},
@DatabaseFiles = {string},
@Compress = {Boolean},
@DBLoginId = 'sa',
@DBLoginPassword = 'password',
@DBPassword = {string},
@DBInstanceName = {string},
@BackupThreads = {long},
@ExpirationDate = {string | date},
@RetainDays = {long},
@Initialize = {boolean},
@MediaDescription = {string},
@MediaName = {string},
@MediaPassword = {string},- SQL Server 2000 only
@BackupSetName = {string},
@BackupSetDescription = {string},
@BackupSetPassword = {string},- SQL Server 2000 only
@Restart = {boolean},
@PercentCompleteNotification = {long},
@LogLevel = {long}

Example
Backing up the primary data file of the pubs database:

EXEC xp_embt_backup
  @Action = 'files',
  @Database = 'pubs',
  @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat',
  @DatabaseFiles = 'c:\Program Files\Microsoft SQL Server\MSSQL\data\pubs.mdf'

For more information, see Backup Operations.

Dynamic Naming of a Backup Job
You can easily give a backup job a dynamic name, that is, a time/date stamp by following the script in the example that follows:

declare @sql varchar(500)
select @sql = 'master..xp_embt_backup @DATABASE=N"PUBS",
@ACTION=N"database",
@BACKUPFILE=N"d:\PUBS_' + rtrim(convert(char(15), getdate(),112)) + '_' + rtrim(replace(convert(char(15),
getdate(), 08),':','')) + '_embt.bak",
@DBLOGINID=N"sa",
@DBLOGINPASSWORD=N"password",
@JOBNAME=N"EMBT_PUBS_080405_083301",
@OPERATIONSTYLE=N"0",
@BACKUPSETDESCRIPTION=N"Backup Description",
@BACKUPSETNAME=N"Backup Set Name",
@JOBDESCRIPTION=N"PUBS Backup Job on 8/4/2005 8:33:01 AM"

exec (@sql)

### Description of Backup Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Datatype</th>
<th>Required</th>
<th>Description</th>
<th>Microsoft SQL Server Backup Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>@Action</td>
<td>String</td>
<td>Always</td>
<td>Valid values: database</td>
<td>files</td>
</tr>
<tr>
<td>@Database</td>
<td>String</td>
<td>Always</td>
<td>The database that is the target of the backup operation.</td>
<td>Database Name</td>
</tr>
<tr>
<td>@BackupFile</td>
<td>String</td>
<td>Always</td>
<td>The location and name of the target backup file.</td>
<td>Backup Device/File</td>
</tr>
<tr>
<td>@Compress</td>
<td>Boolean</td>
<td>Optional</td>
<td>Whether or not to apply compression to the backup. The default is set to true (1).</td>
<td></td>
</tr>
<tr>
<td>@DBLoginid</td>
<td>String</td>
<td>Optional</td>
<td>The Login ID for accessing the SQL Server and performing the backup. If omitted, then Backup Analyst will use the login of the owner of the extended stored procedure.</td>
<td>N/A</td>
</tr>
<tr>
<td>@DBLoginPassword</td>
<td>String</td>
<td>Optional</td>
<td>The password for the login. If omitted, then Backup Analyst will use the password of the owner of the extended stored procedure.</td>
<td>N/A</td>
</tr>
<tr>
<td>@DBPassword</td>
<td>String</td>
<td>Optional</td>
<td>Encryption key used to obfuscate password.</td>
<td>N/A</td>
</tr>
<tr>
<td>@DBInstanceName</td>
<td>String</td>
<td>Optional</td>
<td>It there are multiple instances of SQL Server installed on the same machine, then you should specify the instance name using this parameter. If none is specified, then 'local' is assumed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Parameter</td>
<td>Datatype</td>
<td>Required</td>
<td>Description</td>
<td>Microsoft SQL Server Parameter</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>@BackupThreads</td>
<td>Long</td>
<td>Optional</td>
<td>Number of processors to dedicate to the backup process.</td>
<td>N/A</td>
</tr>
<tr>
<td>@ExpirationDate</td>
<td>Date or String</td>
<td>Optional</td>
<td>Specifies the date when the backup set expires and can be overwritten.</td>
<td>EXPIREDATE</td>
</tr>
<tr>
<td>@RetainDays</td>
<td>Long</td>
<td>Optional</td>
<td>Specifies the number of days that must elapse before this backup media set can be overwritten.</td>
<td>RETAINDDAYS</td>
</tr>
<tr>
<td>@Initialize</td>
<td>Boolean</td>
<td>Optional</td>
<td>Specifies whether or not a backup should overwrite or be appended to any existing backup sets. If overwritten, the media header is preserved.</td>
<td>INIT</td>
</tr>
<tr>
<td>@MediaDescription</td>
<td>String</td>
<td>Optional</td>
<td>Specifies information that helps to identify a backup set.</td>
<td>MEDIADESCRIPTION</td>
</tr>
<tr>
<td>@MediaName</td>
<td>String</td>
<td>Optional</td>
<td>Specifies the media name, a maximum of 128 characters, for the entire backup media set. If MEDIANAME is specified, it must match the previously specified media name already existing on the backup volume(s). If not specified or if the SKIP option is specified, there is no verification check of the media name.</td>
<td>MEDIANAME</td>
</tr>
<tr>
<td>@MediaPassword</td>
<td>String</td>
<td>Optional</td>
<td>SQL 2000 Only Sets the password for the media set.</td>
<td>PASSWORD</td>
</tr>
<tr>
<td>@BackupSetDescription</td>
<td>String</td>
<td>Optional</td>
<td>Specifies the free-form text describing the backup set. The string can have a maximum of 255 characters.</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>@BackupSetName</td>
<td>String</td>
<td>Optional</td>
<td>Specifies the name of the backup set. Names can have a maximum of 128 characters.</td>
<td>NAME</td>
</tr>
<tr>
<td>@BackupSetPassword</td>
<td>String</td>
<td>Optional</td>
<td>SQL 2000 Only Sets the password for the backup set. If a password is defined for the backup set, the password must be supplied to perform any restore operation from the backup set.</td>
<td>PASSWORD</td>
</tr>
<tr>
<td>@Restart</td>
<td>Boolean</td>
<td>Optional</td>
<td>Specifies that SQL Server restarts an interrupted backup operation. The RESTART option saves time because it restarts the backup operation at the point it was interrupted.</td>
<td>RESTART</td>
</tr>
<tr>
<td>@PercentCompleteNotifica</td>
<td>Long</td>
<td>Optional</td>
<td>Provides information on the progress of a backup or restore operation.</td>
<td>STATS</td>
</tr>
<tr>
<td>@LogLevel</td>
<td>String</td>
<td>Optional</td>
<td>error</td>
<td>info</td>
</tr>
<tr>
<td>@DatabaseFileGroups</td>
<td>String</td>
<td>When backing up file groups Names one or more filegroups to include in the database backup.</td>
<td>FILEGROUPS</td>
<td></td>
</tr>
</tbody>
</table>
Restore Operations
Backup Analyst supports a wide array of options for implementing a disaster recovery plan on Microsoft SQL 2000 and 2005. Specifically, the product allows you to:

- Restore a full database
- Restore a partial database
- Restore a transaction log
- Restore specific filegroups or files
- Verify the integrity of a backup
- Retrieve a summary of the contents of a backup
- Retrieve the header information for a backup
- Retrieve the label for a backup

The following sections detail the syntax for executing each of these operations.

- [Restoring a Complete Database](#)
- [Restoring a Database to a Different Location](#)
- [Restoring a Database Transaction Log](#)
- [Restoring Specific File Groups](#)
• Restoring Specific Files
• Retrieving the List of Databases and Log Files in a Backup Set
• Retrieving Backup Header Information
• Retrieving a Backup Label
• Verifying the Integrity of Backup Media
• Description of Restore Parameters

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

---

### Restoring a Complete Database

The syntax for restoring an entire database is as follows:

```sql
EXEC xp_embt_restore
    @Action = {'database'},
    @Database = {string},
    @BackupFile = {string},
    @DBLoginId = {string},
    @DBLoginPassword = {string},
    @FileNumber = {long},
    @EncryptionPassword = {string},
    @DBInstanceName = {string},
    @MediaName = {string},
    @MediaPassword = {string}, - SQL Server 2000 only
    @RelocateFiles = {multi-string},
    @KeepReplication = {boolean}, - SQL Server 2000 only
    @BackupSetName = {string},
    @BackupSetPassword = {string}, - SQL Server 2000 only
    @ReplaceDatabase = {string},
    @Restart = {boolean},
    @PercentCompleteNotification = {long},
    @LogLevel = {long},
    @StandbyFiles = {string}
```

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

---

**Example**

Restoring the entire pubs database:

```sql
EXEC xp_embt_restore
```
@Action = 'database',
@Database = 'pubs',
@BackupFile = 'C:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat'

For more information, see Restore Operations.

Restoring a Database to a Different Location
The syntax for restoring and moving a database is as follows:

EXEC xp_embt_restore
    @Action = {'database'},
    @Database = {string},
    @BackupFile = {string},
    @ReplaceDatabase = {string},
    @RelocateFiles = {multi-string},
    @DBLoginId = {string},
    @DBLoginPassword = {string},
    @FileNumber = {long},
    @EncryptionPassword = {string},
    @DBInstanceName = {string},
    @MediaName = {string},
    @MediaPassword = {string}, - SQL Server 2000 only
    @BackupSetName = {string},
    @BackupSetPassword = {string}, - SQL Server 2000 only
    @Restart = {boolean},
    @PercentCompleteNotification = {long},
    @LogLevel = {long},
    @StandbyFiles = {string}

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

Example
Restoring and moving the pubs database with different file locations (located in drive d, not c):

EXEC xp_embt_restore
    @Action = 'database',
    @Database = 'pubs',
    @BackupFile = 'C:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat',
    @ReplaceDatabase = 1,
@RelocateFiles = ['[pubs],[d:\mssql\data\pubs.mdf],[pubs_log],[d:\mssql\data\pubs_log.ldf]'

For more information, see Restore Operations.

Restoring Part of a Database
The syntax for restoring part of a database is as follows:

EXEC xp_embt_restore
    @Action = {'database'},
    @Database = {string},
    @BackupFile = {string},
    @DBLoginId = {string},
    @DBLoginPassword = {string},
    @FileNumber = {long},
    @EncryptionPassword = {string},
    @DBInstanceName = {string},
    @MediaName = {string},
    @MediaPassword = {string}, - SQL Server 2000 only
    @RelocateFiles = {multi-string},
    @BackupSetName = {string},
    @BackupSetPassword = {string}, - SQL Server 2000 only
    @ReplaceDatabase = {string},
    @Restart = {boolean},
    @PercentCompleteNotification = {long},
    @LogLevel = {long},
    @StandbyFiles = {string}

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

Example
Performing a differential restore of the pubs database:

EXEC xp_embt_restore
    @Action = 'database',
    @Database = 'pubs',
    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat'

For more information, see Restore Operations.
Restoring a Database Transaction Log

The syntax for restoring a database transaction log is as follows:

```
EXEC xp_embt_restore
    @Action = '{log}',
    @Database = {string},
    @BackupFile = {string},
    @DBLoginId = {string},
    @DBLoginPassword = {string},
    @LastRestore = {boolean},
    @ToPointInTime = {string},
    @FileNumber = {long},
    @EncryptionPassword = {string},
    @DBInstanceName = {string},
    @MediaName = {string},
    @MediaPassword = {string} - SQL Server 2000 only
    @RelocateFiles = {multi-string},
    @KeepReplication = {boolean} - SQL Server 2000 only
    @BackupSetName = {string},
    @BackupSetPassword = {string}, - SQL Server 2000 only
    @Restart = {boolean},
    @PercentCompleteNotification = {long},
    @LogLevel = {long},
    @StandbyFiles = {string}
```

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

**Example**

Restoring the transaction log of the pubs database:

```
EXEC xp_embt_restore
    @Action = 'log',
    @Database = 'pubs',
    @BackupFile = 'C:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat',
    @LastRestore = false
```

For more information, see [Restore Operations](#).
Restoring Specific File Groups
The syntax for restoring specific database file groups is as follows:

EXEC xp_embt_restore
    @Action = 'files',
    @Database = {string},
    @BackupFile = {string},
    @DatabaseFileGroups = {string},
    @DBLoginId = {string},
    @DBLoginPassword = {string},
    @FileNumber = {long},
    @EncryptionPassword = {string},
    @DBInstanceName = {string},
    @MediaName = {string},
    @MediaPassword = {string}, - SQL Server 2000 only
    @RelocateFiles = {multi-string},
    @BackupSetName = {string},
    @BackupSetPassword = {string}, - SQL Server 2000 only
    @ReplaceDatabase = {string},
    @Restart = {boolean},
    @PercentCompleteNotification = {long},
    @LogLevel = {long},
    @StandbyFiles = {string}

NOTE: Mandatory parameters to an extended stored procedure are shown in bold text.

Example
Restoring the primary filegroup of the pubs database:

EXEC xp_embt_restore
    @Action = 'files',
    @Database = 'pubs',
    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat',
    @DatabaseFileGroups = 'PRIMARY'

For more information, see Restore Operations.

Restoring Specific Files
The syntax for restoring specific database files is as follows:
EXEC xp_embt_restore
    @Action = {files'},
    @Database = {string},
    @BackupFile = {string},
    @DatabaseFiles = {string},
    @DBLoginId = {string},
    @DBLoginPassword = {string},
    @FileNumber = {long},
    @EncryptionPassword = {string},
    @DBInstanceName = {string},
    @MediaName = {string},
    @MediaPassword = {string}, - SQL Server 2000 only
    @RelocateFiles = {multi-string},
    @BackupSetName = {string},
    @BackupSetPassword = {string}, - SQL Server 2000 only
    @ReplaceDatabase = {string},
    @Restart = {boolean},
    @PercentCompleteNotification = {long},
    @LogLevel = {long},
    @StandbyFiles = {string}

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

**Example**
Restoring the primary data file of the pubs database:

EXEC xp_embt_restore
    @Action = 'files',
    @Database = 'pubs',
    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat',
    @DatabaseFiles = 'c:\Program Files\Microsoft SQL Server\MSSQL\data\pubs.mdf'

For more information, see Restore Operations.

**Retrieving the List of Databases and Log Files in a Backup Set**
The syntax for retrieving the list of databases and log files contained in a backup set is as follows:

EXEC xp_embt_restore_filelistonly
    @BackupFile = {string},
@EncryptionPassword = {string}

NOTE: Mandatory parameters to an extended stored procedure are shown in bold text.

Example
EXEC xp_embt_restore_filelistonly
    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat'

For more information, see Restore Operations.

Retrieving Backup Header Information
The syntax for retrieving the backup header information is as follows:

EXEC xp_embt_restore_headeronly
    @BackupFile = {string},
    @EncryptionPassword = {string}

NOTE: Mandatory parameters to an extended stored procedure are shown in bold text.

Example
EXEC xp_embt_restore_headeronly
    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat'

For more information, see Restore Operations.

Retrieving a Backup Label
The syntax for retrieving the backup label is as follows:

EXEC xp_embt_restore_labelonly
    @BackupFile = {string},
    @EncryptionPassword = {string}

NOTE: Mandatory parameters to an extended stored procedure are shown in bold text.

Example
EXEC xp_embt_restore_labelonly
    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat'

For more information, see Restore Operations.

Verifying the Integrity of Backup Media
The syntax for verifying the integrity of backup media is as follows:

EXEC xp_embt_restore_verifyonly
    @BackupFile = {string},
@EncryptionPassword = {string}

**NOTE:** Mandatory parameters to an extended stored procedure are shown in bold text.

**Example**

EXEC xp_embt_restore_verifyonly

    @BackupFile = 'c:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\pubs.dat'

For more information, see [Restore Operations](#).

### Description of Restore Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Datatype</th>
<th>Required</th>
<th>Description</th>
<th>Microsoft SQL Server Backup Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>@Action</td>
<td>String</td>
<td>Always</td>
<td>Valid values: database</td>
<td>files</td>
</tr>
<tr>
<td>@Database</td>
<td>String</td>
<td>Always</td>
<td>The database that is the target of the backup operation.</td>
<td>Database Name</td>
</tr>
<tr>
<td>@BackupFile</td>
<td>String</td>
<td>Always</td>
<td>The location and name of the target backup file.</td>
<td>Backup Device/File</td>
</tr>
<tr>
<td>@DBLoginId</td>
<td>String</td>
<td>Optional</td>
<td>The Login ID for accessing the SQL Server and performing the backup. If omitted, then Backup Analyst will use the login of the owner of the extended stored procedure.</td>
<td>N/A</td>
</tr>
<tr>
<td>@DBLoginPassword</td>
<td>String</td>
<td>Optional</td>
<td>The password for the login. If omitted, then Backup Analyst will use the password of the owner of the extended stored procedure.</td>
<td>N/A</td>
</tr>
<tr>
<td>@DBPassword</td>
<td>String</td>
<td>Optional</td>
<td>Encryption key used to obfuscate password.</td>
<td>N/A</td>
</tr>
<tr>
<td>@DBInstanceName</td>
<td>String</td>
<td>Optional</td>
<td>It there are multiple instances of SQL Server installed on the same machine, then you should specify the instance name using this parameter. If none is specified, then 'local' is assumed.</td>
<td>N/A</td>
</tr>
<tr>
<td>@FileNumber</td>
<td>Long</td>
<td>Optional</td>
<td>Identifies the backup set to be restored. For example, a file_number of 1 indicates the first backup set on the backup medium and a file_number of 2 indicates the second backup set.</td>
<td>FILE</td>
</tr>
<tr>
<td>@MediaName</td>
<td>String</td>
<td>Optional</td>
<td>Specifies the media name, a maximum of 128 characters, for the entire backup media set. If MEDIANAME is specified, it must match the previously specified media name already existing on the backup volume(s). If not specified or if the SKIP option is specified, there is no verification check of the media name.</td>
<td>MEDIANAME</td>
</tr>
<tr>
<td>Parameter</td>
<td>Datatype</td>
<td>Required</td>
<td>Description</td>
<td>Microsoft SQL Server Backup Parameter</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>@MediaPassword</td>
<td>String</td>
<td>Optional</td>
<td>Sets the password for the media set.</td>
<td>PASSWORD</td>
</tr>
<tr>
<td>@RelocateFiles</td>
<td>Multi-string</td>
<td>Optional</td>
<td>Specifies that the given logical_file_name should be moved to operating_system_file_name. By default, the logical_file_name is restored to its original location. If the RESTORE statement is used to copy a database to the same or different server, the MOVE option may be needed to relocate the database files and to avoid collisions with existing files. Each logical file in the database can be specified in different MOVE statements.</td>
<td>MOVE</td>
</tr>
<tr>
<td>@KeepReplication</td>
<td>Boolean</td>
<td>Optional</td>
<td>Instructs the restore operation to preserve replication settings when restoring a published database to a server other than that on which it was created.</td>
<td>KEEP_REPLICATION</td>
</tr>
<tr>
<td>@BackupSetName</td>
<td>String</td>
<td>Optional</td>
<td>Specifies the name of the backup set. Names can have a maximum of 128 characters.</td>
<td>NAME</td>
</tr>
<tr>
<td>@BackupSetPassword</td>
<td>String</td>
<td>Optional</td>
<td>Sets the password for the backup set. If a password is defined for the backup set, the password must be supplied to perform any restore operation from the backup set.</td>
<td>PASSWORD</td>
</tr>
<tr>
<td>@ReplaceDatabase</td>
<td>Boolean</td>
<td>Optional</td>
<td>Specifies that SQL Server should create the specified database and its related files even if another database already exists with the same name. In such a case, the existing database is deleted. When the REPLACE option is not specified, a safety check occurs (which prevents overwriting a different database by accident).</td>
<td>REPLACE</td>
</tr>
<tr>
<td>@Restart</td>
<td>Boolean</td>
<td>Optional</td>
<td>Specifies that SQL Server restarts an interrupted backup operation. The RESTART option saves time because it restarts the backup operation at the point it was interrupted.</td>
<td>RESTART</td>
</tr>
<tr>
<td>@PercentCompleteNotification</td>
<td>Long</td>
<td>Optional</td>
<td>Provides information on the progress of a backup or restore operation.</td>
<td>STATS</td>
</tr>
<tr>
<td>@LogLevel</td>
<td>String</td>
<td>Optional</td>
<td>error</td>
<td>info</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Indicates the level of logging to occur during execution. The default is set to 'info'.</td>
<td></td>
</tr>
<tr>
<td>@DatabaseFileGroups</td>
<td>String</td>
<td>When backing up file groups</td>
<td>Names one or more filegroups to include in the database backup.</td>
<td>FILEGROUPS</td>
</tr>
<tr>
<td>@DatabaseFiles</td>
<td>String</td>
<td>When backing up files</td>
<td>Names one or more files to include in the database backup.</td>
<td>FILES</td>
</tr>
<tr>
<td>Parameter</td>
<td>Datatype</td>
<td>Required</td>
<td>Description</td>
<td>Microsoft SQL Server Backup Parameter</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
<td>-------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>@StandbyFiles</td>
<td>String</td>
<td>Optional</td>
<td>Specifies the name of an undo file used as part of the backup strategy</td>
<td>STANDBY</td>
</tr>
<tr>
<td>@ToPointInTime</td>
<td>String</td>
<td>When restoring transaction logs</td>
<td>Sets an endpoint for restoring a transaction log</td>
<td>N/A</td>
</tr>
<tr>
<td>@LastRestore</td>
<td>Boolean</td>
<td>When restoring transaction logs</td>
<td>Specifies whether the transaction log is the last in a chain of log backups</td>
<td>N/A</td>
</tr>
<tr>
<td>@LoadHistory</td>
<td>Boolean</td>
<td>Optional</td>
<td>Specifies whether or not to verify the integrity of the backup</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Datatype</th>
<th>Required</th>
<th>Description</th>
<th>SQL Server Backup Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>@BackupFile</td>
<td>String</td>
<td>Yes</td>
<td>Location and name of the target backup file.</td>
<td>Backup Device/File</td>
</tr>
<tr>
<td>@EncryptionPassword</td>
<td>String</td>
<td>Optional</td>
<td>Encryption key used to obfuscate password.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Datatype</th>
<th>Required</th>
<th>Description</th>
<th>SQL Server Backup Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>@BackupFile</td>
<td>String</td>
<td>Yes</td>
<td>Location and name of the target backup file.</td>
<td>Backup Device/File</td>
</tr>
<tr>
<td>@EncryptionPassword</td>
<td>String</td>
<td>Optional</td>
<td>Encryption key used to obfuscate password.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Datatype</th>
<th>Required</th>
<th>Description</th>
<th>SQL Server Backup Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>@BackupFile</td>
<td>String</td>
<td>Yes</td>
<td>Location and name of the target backup file.</td>
<td>Backup Device/File</td>
</tr>
<tr>
<td>@EncryptionPassword</td>
<td>String</td>
<td>Optional</td>
<td>Encryption key used to obfuscate password.</td>
<td>N/A</td>
</tr>
</tbody>
</table>
ETSBackupConsole.exe

From the command line, the product allows you to perform a:

- Full database backup
- Incremental database backup
- Transaction log backup
- Backup of specific filegroups or files
- Restore a full database
- Restore a transaction log
- Restore specific filegroups or files
- Verify the integrity of a backup
- Retrieve a summary of the contents of a backup
- Retrieve the header information for a backup
- Retrieve the label for a backup

The syntax for backing up or restoring using ETSBackupConsole.exe follows. Mandatory parameters are shown in bold. All other parameters are optional.

**NOTE:** In addition to {'backup_database'}, action parameters available for backup operations are: {'backup_incremental'}, {'backup_log'}, {'backup_files'}, {'backup_database'}, and for restore operations are: {'restore_files'}, {'restore_log'}, {'restore_filelistonly'}, {'restore_labelonly'}, {'restore_headeronly'}, and {'restore_verifyonly'}.

```
@Action = {'backup_database'},
@Database = {string},
@BackupFile = {string},
@DBLoginId = {string},
@DBLoginPassword = {string},
@DBInstanceName = {string},
@BackupThreads = {long},
```

For more information, see [Restore Operations](#).
@BackupSetName = {string},
@BackupSetDescription = {string},
@Log Level = {string},
@Compress = {boolean},
@DatabaseFileGroups = {multi-string},
@DatabaseFiles = {multi-string},
@ExpirationDate = {string},
@FormatMedia = {Boolean},
@Initialize = {Boolean},
@MediaDescription = {string},
@MediaName = {string},
@MediaPassword = {string} - SQL Server 2000 only
@BackupSetPassword = {string}, - SQL Server 2000 only
@Restart = {boolean},
@SkipTapeHeader = {boolean},
@TruncateLog = {string},
@UnloadTapeAfter = {string},
@NoRewind = {boolean}, SQL Server 2000 only
@PercentCompleteNotification = {long},
@RetainDays = {long}

Example:
Etsbackupconsole.exe
  @action=backup_database
  @Database=[pubs]
  "@backupfile=c:\temp\pubs.embt.bak"
  @EncryptionPassword=PASSWORD
Index

Symbols
% Privileged Time 947, 1109, 1226, 1367
% User Time 947, 1110, 1227, 1367
%FREE index partition setting 202
(VPXSEQT setting 172
A
abort tran on log full 186
Access Methods
capacity/growth statistics 810
account lock time 232
accounting
CPU and I.O usage 209
accounts
locked status 274
user identification 274
action, triggers 265
Activate Logging 404
Active Rollback Ratio 929
Active Threads 40
Active Transactions setting 240
activity levels
rollback segments 239
Ad Hoc SQL Hit Ratio 1022
Add 224
Add Cluster Column 407
Add Datafile 408
Add Partition 405
adding removing partitions 216
adding/deleting, tables 250
Additional resources 8
adsm_mgmtclass 184
adsm_nadename 184
adsm_owner 184
adsm_password 184
Advanced View 781
After 68
After setting 266
Alarm Log
Organizing 837
Aliases 107
Creating for IBM DB2 282
aliases
adding comments 170
creating for IBM DB2 for OS/390 and z/OS 306
creating for Sybase ASE 389
object editor functions 170
see also objects, database
transferring with database ownership 184
viewing DDL/SQL 170
viewing definition 170
Alignment index partition setting 202, 228
Alignment unique key partition setting 269
Allocating Extent 409
Allow Bit Data setting 271
allow nulls by default 186, 390
Allow Parallel setting 196
Allow Reverse Scans index setting 200
Alter statements
migration/extraction options 608
analysis reports, Capacity Analyst 821
Analyze 411
Analyze Code 539
Analyzing
Index Partitions 411
ANSI Labels 171
ANSI null default 185, 335
ANSI nulls 185, 335
ANSI padding 185, 335
ANSI warnings 185, 335
Append Data setting 250
Append, materialized query tables 212
appl_ctl_heap_sz 184
applheapz 184
Apps Waiting on Locks 1258
ar 834
arithabort 185
As Locator function setting 195
Asc Sort, indexes 199
Assisting Parallel Sequential setting 172
association, file 75
attributes
structured types 245
Audit table setting 250
auto allocation 263
auto close 185, 335
Auto Commit Status 540
auto create statistics 185, 335
auto format
DDL extract options 70
auto identity 186, 390
auto shrink 185, 335
auto update statistics 185, 335
auto-commit options 75
automatic segment management 263
auto-reload, options 73
autorestart 184
auto-save, files 73
autostart, event monitors 191
Available Paged Memory 944, 1106, 1223, 1363
Available Physical Memory 943, 1106, 1223, 1363
Available Swap Memory 944, 1106, 1223, 1364
Average Disk Queue Length 942, 1104, 1221, 1361
Average Size package setting 219
avg_appls 184
B
Backing Up Databases 645
Backing-Up Microsoft SQL Servers 645
Backup
Compress 649
backup
capacity/growth statistics 809
growth reports 821
Backup Analyst 1465
Alert Creation Wizard 1481
Backup Analyst Toolbar 1472
Backup Operation Syntax 1486
Entire Database 1486
File Groups 1489
Incremental 1487
Specific Files 1490
Transaction Log 1488
Backup Parameters 1492
Backup/Recovery Manager 1473
Command Line 1486
Encryption 1478
Explorer 1473
Any Database Node 1474
Backup Directories Node 1475
Backup Recovery Management Node 1475
Databases Node 1474
Filegroups Node 1474
Server Node 1474
Functionality 1476
Information Pane 1475
Installation 1467
License Status 1469
Licensing 1469
Managing more backups 1465
Opening 1469
Options 1482
Product Design 1471
Recovery Wizard 1480
Removing 1468
Repairing 1469
Report 1483
Restore Operation Syntax 1494
Backup Header Information 1501
Backup Label 1501
Backup Media Integrity 1501
Complete Database 1495
Different Location 1496
File Groups 1499
Files 1499
List of Databases and Log Files in Backup Set 1500
Partial 1497
Transaction Log 1498
Restore Parameters 1502
Server Free Space Monitor 1484
Single-User Mode 1469
Starting the SQL Agent 1469
Transaction Log Node 1474
Uninstalling 1468
User Interface 1471
Using 1476
Backup Control File 666
Backup Control Files 665
Backup Device Wizard for Microsoft SQL Server 336
Backup Devices 117
Creating for Microsoft SQL Server 336

Backup devices
ANSI Labels 171
device type 171
object editor functions 171
physical name 171
see also objects, database
viewing DDL/SQL 171
Backup Wizard 645
backup_pending 184
Backups Tab 1098
Backup Detail 1099
Database Backup Summary 1099
Log Backup Summary Backup 1099
batch files
Java drop/load 76
Batch Mode execution 69
BCP Options 643
BCP Utility 606
Before setting 266
Begin Transaction ON/OFF 540
Bind Data Cache 414
Bind Package 414
Bind Plan 417
bind size 79
Bind Time package setting 219
Binder package setting 218
binding
packages 219
bindings
objects to caches 178
table columns 251
bodies
package status/size 221
Bookmarks 54, 556
Clearing 557
Goto 556
Moving 556
Buffer Cache Hit Ratio 1021
Buffer Cache Tab 885, 1038
Buffer Cache Hit Ratio 885, 1038
Buffer Cache Utilization 886
Cached Blocks by Tablespace 886
Cached Tables 886
Page Activity 1039
Buffer Manager
capacity/growth statistics 810
Buffer Partition
capacity/growth statistics 810
Buffer Pool Wizard
Panel 1 for IBM DB2 283
Panel 2 for IBM DB2 283
Panel 3 for IBM DB2 283
Buffer Pool Wizard for IBM DB2 283
Buffer Pools
Creating for IBM DB2 283
Editing for IBM DB2 for OS/390 and z/OS 172
Buffer Pools Tab 888
Buffer Pool Hit Ratio 888
Buffer Pool Memory Configuration 889
Buffer Pool Object Assignments 889
Buffered Inset package setting 218
Bufferpools 107
bufferpools
  and event monitors 190
  indexes 200
  managing for data caches 178
  managing nodegroups assigned 172
  object editor functions 172
  partitions 172
  primary key setting 226
  privileges 172
  see also objects, database
  storage 172
  tablespaces default 260
  tablespaces 261
  viewing DDL/SQL 172
Buffersize setting, event monitors 191
buffpage 184
Building a Query
  Statements
    UPDATE 702
  by default
table columns 251
C
  Cache Efficiency 957, 1119, 1236, 1377
  Cache option
    materialized view logs 216
  Cache setting 250
  cache setting, clusters 175
  cache settings
    data caches 177
  Cache Size setting 244
  cache, table columns 251
  Capacity Analyst
    analysis reports 821
    charts/reports 817
    cloning collections 813
    collection summary reports 821
    creating collections 812
    deleting collections 814
    editing collections 813
    Installation 792
    License Manager 797
    Licensing 797
    manually updating collections 814
    purging old data 816
    scheduling collection updates 814
    SQL Server Embarcadero Stored Objects 789
  Capacity Analyst for Oracle
    Tutorial 798
  Cascade primary key setting 226
  Cascade unique key setting 268
  Cascading Permissions 735
case
  object identifiers 75
  Castout setting 172
catalogcache_sz 184
  CCSID database setting 187
  CCSID setting 250
  CCSID table setting 271
  Chained Row Fetch Ratio 929
  Change Category 421
  Change Container Size 421
  Change Password 422
  Change Status 423
  character set, client 68
  character strings. See strings
  Check Constraints 109
    Changing Constraint Status 423, 524
    Creating for IBM DB2 440
    Creating for IBM DB2 for OS/390 and z/OS 440
    Creating for Microsoft SQL Server 440
    Creating for Oracle 440
    Creating for Sybase ASE 440
    Modifying for IBM DB2 440
  check constraints
    adding comments 173
    enabling/disabling 173
    modifying definition 173
    object editor functions 173
    replication setting 173
    see also objects, database
    viewing DDL/SQL 173
  Check Data 424
  Check Flag table setting 250
  Check Index 426
  check key constraints
    binding messages to 272
  Checking Syntax 539
  Checkpoint 428
  chngpgs_thresh 184
  Claimers status 260
  Clean Buffer Grab Rate 1140
  Clone Partition 428
  Clone Table 429
  Close index setting 200
  Close primary key setting 226
  Close Rule tablespace setting 261
  Cluster Factor setting 201, 225, 270
  Cluster Factor, primary keys 225
  Cluster Ration setting 201, 225, 270
  Cluster Wizard for Oracle 351
  Clustered setting 226
    unique keys 268
  clustered setting
    indexes 200
Clusters 109
  Analyzing 411
  Creating for Oracle 351
clusters
  and materialized views 214
  capacity/growth statistics 811
  cluster type, hash/index 174
  data block storage 175
  extent details 176
  extents 175
  freelists 176
  key size 174
  object editor functions 174
  performance settings 175
  see also objects, database
  space utilization 176
  storage settings 175
  table details 176
  viewing DDL/SQL 174
Coalesce 429
Code Formatting 539
Code Page package setting 218
Code Size setting 197, 231
  packages 221
Code Workbench 730
  Creating Auto Replace Shortcuts 733
  Exporting Settings 734
  Importing Settings 734
  Modifying Auto Replace Shortcuts 733
  Setting Column Auto Lookup 733
  Tutorial 732
codepage 184
codeset 184
collate_info 184
Collation Compatible Server setting 234
Collecting Column Statistics 533
Collecting column statistics 533
Collecting Distribution Statistics for Tables 533
Collecting Extended Index Statistics 533
Collecting extended index statistics 533
Collecting sample statistics 534
Collecting Sample Statistics for Indexes 533
Collection Analysis Viewer
  adding metrics 819
  introduced 817
  opening 818
  viewing options 818
collections, Capacity Analyst
  analysis viewer 817
  analyst reports 821
  cloning 813
  creating 812
  deleting 814
  editing 813
  manually updating 814
  purging older data 816
  scheduling updates 814
  summary reports 821
colors
  Query Builder options 80
  Column Auto Lookup 733
  Column Rename 497
columns 250
  cluster key 174
  cluster key details 176
  displaying for view 275
  filtering for materialized view logs 215
  managing in indexes 199
  managing in unique keys 267
  managing for materialized query tables 210
  modifying in foreign keys 193
  of a user dataytpe 271
  properties, table 250
  specifying in primary keys 225
command history 74
Command Line
  ETSBackupConsole.exe 1505
comments
  adding to aliases 170
  adding to check constraints 173
  adding to columns 251
  adding to databases 210
  adding to functions 194
  adding to indexes 198
  adding to nodegroups 216
  adding to packages 218
  adding to primary keys 224
  adding to procedures 228
  adding to tables 248
  adding to tablespace 257
  adding to triggers 264
  adding to unique keys 266
  adding to user datatypes 270
  adding to views 274
  for table columns 251
Compatible level 185
Compatible Level setting 185
Compile 429
composite limit 232
Compress Backup 649
Compress setting 250
computed values, table columns 251
concat null yields null 185, 335
conditions
  event monitors 191
Configuration Tab 1099
confirmation dialogs
  data editor options 69
  connect string, database links 182
  connect time 232
Connection Editor 433
connections
  and event monitors 190
  lock options 75
  options/preferences 68
Consistency Token setting 188
Consistent Reads 903
Constraint State setting 193
constraints
  binding messages to 272
  viewing associated objects 242
Contacting Embarcadero Technologies Technical Support 16
containers
  adding to tablespaces 258
  modifying for tablespaces 258
Context Switches/Sec 948, 1110, 1227, 1368
Control Files 110
control page 253
Convert to Partitioned 434
Copy index setting 200
Copy Object Names 435
Copy primary key setting 226
Copy Read Hits % 957, 1120, 1236, 1377
Copy to Copy 435
country setting 184
CPU
  login usage 209
CPU Busy 1023
CPU per call 232
CPU per session 232
CPU Tab 947, 1109, 1226, 1366
  CPU Events 948, 1110, 1227, 1367
  CPU Events Context Switches/Sec 948, 1110, 1227, 1368
  CPU Events Interrupts/Sec 948, 1110, 1227, 1367
  CPU Events System Calls/Sec 948, 1110, 1227, 1368
  CPU Utilization 947, 1109, 1226, 1366
  CPU Utilization % Privileged Time 947, 1109, 1226, 1367
  CPU Utilization % User Time 947, 1110, 1227, 1367
  Load Averages 949, 1111, 1228, 1368
  Processor Queue Length 948, 1111, 1228, 1368
CPU Utilization 947, 1109, 1226, 1366
Create Alias 438
Create Insert Statements 438
Create Like 439
Create Synonym 441
Create User Message Text 441
Created setting packages 221
Creating
  Space Management Job 1425
  Users 584
Creating a Primary Key
  MySQL 582
Creating a Recovery Operation 650
Creating a Unique Key
  MySQL 582
Creating an Index
  MySQL 582
Creating Objects 279
Creating or Editing
  MySQL Functions 580
  MySQL Tables 589
Creating or Editing Foreign Keys
  MySQL 578
Criteria Tab 1127
Current Object Blocks 1024
Current SQL 1009
Current SQL Transactions 1009
CurrentData package setting 219
cursor Blocking package setting 218
cursor close on commit 185, 335
Cycle When Reach Max/Min setting 244
cycle, table columns 251
D
Data
  Importing from external source 721
data
    migration/extraction options 608
    usage/distribution in segments 242
Data Access Server setting 234
Data Blks/Key setting 201, 270
Data Blks/Key, primary keys 225
Data Block Storage
  unique keys 268
data block storage 175, 214
Data Block Storage settings
  materialized view logs
  tablespaces
  materialized view logs 215
Data Cache Hit Activity 1151
Data Cache Wizard for Sybase ASE 391
Data Caches 115
Creating for Sybase ASE 391
data caches
  binding strategies 178
  cache settings 177
  managing bufferpools in 178
  object binding 178
  object editor functions 177
  overhead memory 177
  run size 177
  see also objects, database
  status 177
  viewing DDL/SQL 177
data capture option 212
data dictionary
  viewing usage permission 75
Data Editor 456
  Calendar 725
  Data Editor Filter 724
  Date Functions 725
  Date/Time Format Builder 725
  Edit Window 723
  Editing Date and Time Functions 725
  Editing Date/Time 725
  Modifying Criteria 724
  SQL Window 723
  Time Functions 725
  Toolbar 723
data editor
  execution mode 69
  options/preferences 66
Data Editor Filter 724
Data Export
  Data Export Wizard 655
Data Export Wizard 655
Panel 1 655
Panel 2 656
Panel 3 656
Panel 4 657
Panel 5 658
Panel 6 658
Panel 7 659

Data Export Wizard for IBM DB2 for Windows/Unix
Panel 1 626
Panel 2 626
Panel 3 628
Panel 4 628
Panel 5 628
Panel 6 628
Panel 7 628
Starting 629

Data Import 660
Data Import Wizard 660
Panel 1 660
Panel 2 661
Panel 3 661
Panel 4 662
Panel 5 663
Panel 6 663
Panel 7 664
Panel 8 664

Data Import Wizard for IBM DB2 for Windows/Unix
Panel 1 629
Panel 2 630
Panel 3 630
Panel 4 631
Panel 5 631
Panel 6 631
Panel 7 632
Starting 629

Data Load 629
Data Load Wizard 629
Panel 1 642
Panel 2 642
Panel 3 642
Panel 4 642
Panel 5 643
Panel 6 644

Data Map Hits % 957, 1120, 1237, 1377
data pages 253

Data Pages setting 202, 227, 269

Data Unload 639
Data Unload Wizard
Panel 1 639
Panel 2 640
Panel 3 640
Panel 4 640
Panel 5 641
Starting 639

Database adding comments 210
configurations/options settings 184
Database Backup Wizard 646
Starting 645

Database Device Wizard for Sybase ASE 389

Database Devices 110
Creating for Sybase ASE 389
Refitting 495
database devices
capacity/growth statistics 809
database fragments 180
object editor functions 179, 182
see also objects, database
size details 180
space, used/free 180
viewing DDL/SQL 179

Database Devices Editor for Microsoft SQL Server
Information Tab 179

Database Devices Editor for Sybase
Information Tab 179

Database Heap Utilization 1255

Database I/O Tab 1059
Database Bytes Read/Write Summary 1059
Database I/O Detail 1059
Database Read/Write Summary 1059

Database Link Wizard for Oracle 353

Database Links 111
Creating for Oracle 353
database links
connect string 182
public setting 182
see also objects, database
user/password 182
viewing DDL/SQL 182
database links, synonyms 247

Database Low on Space 1149
Database Manager 644
Opening 645

Database Monitor 611
All Locks Tab 619
Blocking Locks Tab 619
Database I/O Summary Tab 622
File I/O Summary Tab 623
Last 50 Backups Tab 622
Locks Tab 617
Processes 616
Opening 612
System I/O Summary Tab 622
Top SQL Cache Tab 622
Top Twenty SQL Tab 623
User I/O Summary Tab 622
Viewing Session Details 615
Wait Events Tab 623
Waits Tab 622

Database Object Finder 103
Database partition Group table setting 255
Database Restore Wizard
Last Panel 654
Starting 651

Database Search
Results 681

Database Search Wizard 680
Panel 1 680
Panel 2 681

Database Wizard for IBM DB2 284
database_consistent 184
INDEX ▶

database_level 184
Databases 112
   Attach 451
   Backing Up 645
   Checkpoint 428
   Coalescing 429
   DBCC 444
   Detach 451
   MySQL 586
   Restoring 650
   Setting Online/Offline 520
databases
   and event monitors 190
   basic properties 186
   binding to datacaches 178
   capacity/growth statistics 809
   creating for IBM DB2 for OS/390 and z/OS 307
   creating for Microsoft SQL Server 334
   creating for Sybase ASE 390
   growth reports 821
   login accounts on 208
   modifying database fragments 183
   modifying file details 183
   modifying tablespaces 183
   object dependencies 183
   object editor functions 182
   privileges 183
   publishing HTML reports 609
   see also objects, database
   setting options 185
   space usage 186
   specifying log details 184
   transaction log details 187
   viewing DDL/SQL 183
Databases Editor for Sybase
   Options Tab 185
Databases Low on Space 1027
Databases Performance Category View
   Statistics 1190
Datafile Fragmentation 923
Datafile I/O Tab 906
datafiles
   details/managing for tablespaces 262
DataLink Options 443
dataset handling, tablespaces 261
Datasource
   Edit Registration 86
   Register 86
   Datasource Catalog 91
   Sharing 91
Datasource Explorer 33
   Blocking Locks 39
   Configuration Node 46
   Connections 50
   Databases Node 35
   Databases Node 44
   DDF 51
   Displaying 52
   Image Copies 38
   Instance Node 36–37
   Lock Node 38
   Options 34
   Organizing 51
   Parameters 51
   Performance Node 37
   Plan 44
   Processes Node 38
   Server Node 45
   Storage Node 50
   Subsystem Node 50
   Traces Node 42
datasource explorer
   options/preferences 66, 70
Datasource Groups 104
   Changing 92
   Modifying 99
   Removing 99
Datasource Landing Page 93
Datasource Lock 539, 567
Datasource Overviews 93
Datasources 90
   Auto-Discover 92
   Connecting 93
   Discovering 92, 94
   Editing 96
   Managing 95
   Modifying 94
   MySQL 577
   Properties 96
   Registering 103
   Retaining After Uninstalling DBArtisan 15
   Selecting 97
datasources
   HTML reports 609
   JDBC driver 89
   options/preferences 66
datatype
   user-defined, definition 270
datatypes
   columns 251
   columns in views 275
   columns of 271
   function parameters 194–195, 231
   sequence 244
   structured type attributes 245
   synonyms 247
date formats
   data editor 69
   Date Time Format package setting 218
   db chaining 185, 335
   DB Protocol package setting 219
Decimal Point setting 188
default binding
  table columns 251
Default Schema package setting 218
Default Tablespace user setting 274
default to local cursor 185, 335
Default Wizard for Sybase ASE 391
Defaults 116
  Creating for Sybase ASE 391
defaults
  binding, datatypes 271
  creating for Microsoft SQL Server 335
database device 180
DDL expression 189
filegroup, in database 184
modifying value 189
object dependencies 189
object editor functions 189
  see also objects, database
Defer Prepare package setting 219
Deferable primary key setting 226
Deferrable/Deferred unique key setting 268
Deferred primary key setting 226
Deferred Write Threshold setting 172
DEFINE index partition setting 202
Definer package setting 218
Degree package setting 218–219
degrees setting, clusters 175
degrees setting 216
Delayed Disk I/Os 1170
Delete Cascade setting 193
Delete Rule setting 193
delimited identifiers 68
density setting 201, 225, 270
dependencies
  databases 183
dependencies
  debugger options 72
defaults 189
  extended procedures 192
  functions 194
  libraries 204
  materialized views 213, 215
  migration/extraction options 608
  packages 218
  plans 222
  procedures 228
  rules 240
  sequences 243
  synonyms 247
  tables 248
  triggers 264
  users 273
  views 274
Deterministic setting 197, 230
Device I/O Contention 1143
device number, database device 180
device type 171
  backup/dump device 171
devices
  container settings 259
  file names 184
growth reports 821
dft_degree 184
dft_extent_sz 184
dft_loadrec_ses 184
dft_prefetch_sz 184
dft_queryopt 184
dft_refresh_age 184
dft_sqlmathwarm 184
Dialog Box 7
  Grant Privileges To 736
  Revoke Privileges From 737
Dialog Boxes 61
  Completing 279
dictionary extent management 263
dimension, table support 254
dir_obj_name 184
Directories 117
  Creating for Oracle 353
directories
  object editor functions 189
  Oracle utilities 79
  oracle utilities 69
  path and SQL 190
  privileges 190
  see also objects, database
Directory Wizard for Oracle 353
Dirty Read Requests 1151
Dirty Read Restarts 1152
Disable Job Queues 453
discover setting 184
Discussion Groups 8
Disk Resize 455
disk Time 941, 1103, 1220, 1360
Display
  Active Windows Dialog 62
  Customizing 61
  Full Screen Mode 62
Distinct Keys setting 201, 225, 269–270
Distinct Keys, primary keys 225
Distributed Queries 129
Distribution Server setting 234
dl_expint 184
dl_num_copies 184
dl_time_drop 184
dl_token 184
dl_upper 184
dilchkt ime 184
DILs, extended procedures 192
Documentation 8
Drop 455
drop
  segments 242
DROP statements
  DDL extract options 71
drop, Java 76
dropping
  columns, materialized query tables 210
dump device 171
Dump Device Wizard for Microsoft SQL Server 336
Dump Device Wizard for Sybase ASE 392
Dump Devices 117
Creating for Microsoft SQL Server 336
Creating for Sybase ASE 392
dump utility, MySQL 79
duplicate keys
indexes 200
DWQT setting 172
dyn_query_mgmt 184
Dynamic Rules package setting 219
Dynamic setting 204
Dynamic Sizing settings 240
E
Edit Data 456
Edit Node Connection 36
Edit registration
Datasource 86
Editing
Space Management Job 1425
Users 584
Editing Result Sets 570
Adding a Column 573
Adding a Row 572
Columns 571
Copying 571
Cutting 570
Deleting a Column 573
Deleting a Row 573
Formatting Result Sets 574
Inserting a Column 573
Inserting a Row 572
Pasting 571
Resizing Columns 574
Resizing Rows 573
Rows 571
Sorting Data 574
Editing SQL Scripts 550
Changing Case 554
Commenting Code 554
Copying and Pasting a Column 555
Copying and Pasting Text 555
Cutting and Pasting a Column 556
Cutting and Pasting Text 555
Pasting SQL Statements 551
Pasting SQL Syntax 550
Selecting a Block of Text 553
Selecting Text 553
Setting Bookmarks 556
Uncommenting Code 554
Editor
JDBC Driver 89
Editors 162
eeditors
options/preferences 73
EditProcedure table setting 250
eemail. See also SMTP
Embarcadero SQL Debugger for IBM DB2 740
Embarcadero SQL Debugger for IBM DB2 for Windows/Unix
Basics Interface 749
DDL Editor 749
Features 749
Functionality 752–753
Break 756
Breakpoints 754–755
Close 756
Go 755
Input Parameters 752
Restart 756
Run to Cursor 754
Step Into 753
Step Out 753
Step Over 754
Interface
Call Stack Window 751
Dependency Tree Window 751
Variables Window 750
Watch Window 750
Options 748
Requirements 741
Using 757
Debugging an SQL Script 758
Opening a Debugging Session 757
Embarcadero SQL Debugger for Sybase 740
Features 740
Functionality 752
Break 756
Breakpoints 754–755
Close 756
Go 755
Restart 756
Run to Cursor 754
Step Into 753
Step Out 753
Step Over 754
Stop 756
Interface 749
Call Stack Window 751
DDL Editor Window 749
Dependency Tree Window 751
Variables Window 750
Watch Window 750
Options 748
Using 757
Debugging an SQL Script 758
Opening a Debugging Session 757
Embarcadero Technologies Technical Support 16
Contacting 16
EMEA
Technical Support 8
emulation 74
Enable Job Queues 453
Enable Query Rewrite materialized view setting 213
Enabled primary key setting 226
EnableQueryOptimization setting
queries
optimization for views 275
Enabling/Disabling
materialized query tables 211
unique keys 268
enabling/disabling
check constraints 174
Encoding package setting 219
Encoding Scheme database setting 187
Encoding Scheme tablespace setting 261
encoding, Java 76
Encryption
Backup Analyst 1478
Enforcing Index settings 226
Engines Tab 1182
Enterprise Management Console 52
Erase index setting 201
Erase primary key setting 226
Erase Rule tablespace setting 261
Error 457
Error Log Tab 1096
Error Log Content Analysis 1096
Error Log Volume Analysis 1096
SQL Server Error Log 1096
Error Size setting 197, 231
packages 221
Errors in Current Log 1026
Errors materialized view setting 213
Estimate Size 457
estore_seg_sz 184
ETBackupConsole.exe 1505
ETStart/ETEnd tags 461, 607
Event Monitor Wizard 286
Event Monitors 118
event monitors
autostart option 190
deadlock monitoring options 190
event type/condition 191
file attributes 191
object editor functions 190
partitions 190
see also objects, database
specifying evnt type 190
target type 191
viewing DDL/SQL 190
write buffer settings 191
events
firing triggers 265
monitoring 190
trigger 265
Excel 2000 569
Execute 458
Executing a Recovery Script in the ISQL Window 654
Executing Procedures 539
Executing Scripts 557
Cancelling a Query 559
Compilation Error Tab 559
Executing from Command History 559
Executing Partial Scripts 558
Plan Facility 561
Query Options 561
Script Execution Facility 558
Step Execution Facility 560
Token Matching Capability 566
Expected Row Size setting 250
Explain Level/Snapshot/Mode setting 218
Explain package setting 219
Explicit Bind Time package setting 218
Explicit Permissions 735
export
default directory 69
utility directory 79
extend
segments 242
Extended Procedures 119
extended procedures
creating for Microsoft SQL Server 337
creating for Sybase ASE 392
DLL 192
dll code 192
library name 192
object dependencies 192
object editor functions 191
privileges 192
procedure name 192
see also objects, database
viewing DDL/SQL 192
Extended Procedures Editor for Microsoft SQL Server
Definition Tab 192
Extended Procedures Editor for Sybase
Definition Tab 192
Extended Storage setting 172
Extends setting 240
Extent Analysis Tab 924
Extent Size table tablespacesetting 255
extents
clusters 176
index settings 201
LOB column usage 255
materialized view logs 215
primary key settings 227
primary keys 225
table usage 252
tablespace 263
unique keys 268, 270
extents, clusters 175
extents, materialized views 214
External Action setting 197, 230
External Name setting 196, 230
External Name, user 274
Externally setting 237
Extract 460
extraction
DDL options 70
extraction, schema 604
failed login attempts 232
FAQs 8
Fenced setting 196, 230
File Attributes setting, event monitors 191
File Factor setting 226
File Group setting 200, 226, 268
File Group Tab 1068
File Growth Rate setting 184, 187
File Tab 1060
   File Bytes Read/Write Summary 1061
   File I/O Detail 1061
   File Read/Write Summary 1060
Filegroup setting 250
Filegroups
   in indexes 201
   in filegroups
      growth reports 821
files
   assigning for databases 183
   association, SQL editor 75
   auto-saving 73
   containers, details 259
   data editor types 69
   database details 184
   growth reports 821
   licensing 77
   SQL logging options
      Logging, SQL truncating 78
   tracking 73
Files Tab 1069
filgroups
   capacity/growth statistics 809
Fill Factor setting 200–201, 226, 268
Fill Rows with Default Value setting 251
Filter 462
Filtering Results 462
Final Call setting 197
First Key index setting 201
First Key primary key setting 225
First n Keys setting 201, 225, 270
first page 253
first virtual page, database device 180
Floating Output Window View 64
Flush Cache 463
Flush Options
   MySQL 577
fonts
   data editor options 69
   editor options 74
   Query Builder options 80
For Data table setting 271
foreign key constraints
   binding messages to 272
Foreign Keys 119
   Changing Constraint Status 423, 524
   Creating for IBM DB2 289
   MySQL 578
foreign keys
   creating for IBM DB2 for OS/390 and z/OS 308
   creating for Microsoft SQL Server 337
   creating for Oracle 354
   creating for Sybase ASE 392
   modifying columns 193
   modifying properties 193
   object editor functions 192
   see also objects, database
   viewing DDL/SQL 192
Foreign Keys Editor for Sybase
   Columns Tab 193
Format 539
Formatting Code 539
Formatting Result Sets 574
   Setting Border Properties 575
   Setting Colors 576
   Setting Fonts 575
Forums 8
fragmentation
   growth reports 822
fragments, database 180
   assigning for databases 183
Free Disk Space 945, 1108, 1225, 1365
Free Packages 463
Free Page 201
Free Page primary key setting 226
free pages 243
Free Plans 464
Free Shared Pool Percent 880
free space
   database devices 180
freelists
   clusters 176
   in indexes 201, 253
   in unique keys 268
   primary key settings 227
FREETABLE index partition setting 202
Full Keys index setting 201
Full Keys primary key setting 225
Full Keys setting 201, 225, 270
Full Text Indexing settings 250
function
   triggers, path 265
Function Path package setting 218
Function-Based index setting 200
Functions 119
   Compiling 431
   MySQL 580
functions
- adding comments 194
  - Creating for IBM DB2 290
  - creating for IBM DB2 for OS/390 and z/OS 308
  - creating for Microsoft SQL Server 338
  - creating for Oracle 355
- definition/basic settings 195–196
- extract options 461, 607
- igrate options 607
- modifying body 195
- object dependencies 194
- object editor functions 194
- parameters/return parameters 194
- password verification 232
- privileges 194
- see also objects, database
- status/size 197
- synonyms to 247
- viewing DDL/SQL 194

G
- GBP Cache index setting 201
- GBP Cache primary key setting 226
- GBPCACHE index partition setting 202
- Generate Numbers in Order setting 244
- Generate Reports 510
- Gets setting 240
- global database statistics 809
- globally setting 237
- grace period, password 232
- Grant Roles 464
- Graph Types 838
  - Changing 838
- grid properties, results window 75
- Group Member database setting 186
- Group Member package setting 219
- Group Wizard for IBM DB2 292
- Group Wizard for Sybase ASE 393
- Groups 121
  - Creating for Sybase ASE 393
  - groups
    - adding/removing members 198
    - assigning users 274
    - object editor functions 197
    - privileges 197
    - see also objects, database
    - viewing DDL/SQL 198

H
- hash composite 202, 228, 269
- hash details
  - clusters 174
- headers
  - package status/size 221
  - packages 220–221
- heap size, JDBC 76
- Help
  - Contents Tab 15
  - Favorites Tab 15
  - Index Tab 15
  - Search Tab 15
  - RIVILEGES_TAB 191
  - HIDD_GEN_UIR_PACKAGE_BODY_EDITOR_DDL_TAB 217
  - HIDD_GEN_UIR_PACKAGE_EDITOR_PRIVILEGES_TAB 217
  - HIDD_SQS_SSU_DEFAULT_EDITOR_BINDINGS_TAB 189
  - HIDD_SYB_SYU_DEFAULT_EDITOR_BINDINGS_TAB 189
  - High Watermark setting 240
  - highlighting. See style
  - Hiperpool Sequential Steal Threshold setting 172
  - Hiperpool Size setting 172
  - Histogram Statistics 413
  - history, command 74
  - Home
    - Sybase
      - I/O Analysis 1145
  - Home View Statistics
    - DB2 1252
      - Accesses to Overflowed Records 1261
      - Application Activity Analysis 1268
      - Applications Currently Executing 1269
      - Average Lock Wait Time 1259
      - Bottleneck Analysis 1258
      - Buffer Pool Hit Ratio 1253
      - Catalog Cache Hit Ratio 1253
      - DMS Space 1267
      - DMS Tablespaces Low on Space 1263
      - Dynamic SQL 1265
      - Failed SQL 1265
      - Hash Join Overflows 1262
      - Idle Applications 1269
      - Inaccessible Containers 1263
      - Internal Rebinds 1266
      - Internal Rollbacks Due to Deadlocks 1266
      - Key Ratio Analysis 1252
      - Lock Escalations 1260
      - Lock Timeouts 1260
      - Log Pages I/O 1261
      - Log Space 1267
      - Node with Smallest Log 1263
      - Package Cache Hit Ratio 1254
      - Page Reorganizations 1262
      - Shared Workspace Hit Ratio 1255
      - Sort Overflows 1262
      - SQL Analysis 1264
      - Static SQL 1265
      - Storage Analysis 1266
      - Synchronous I/O 1261
      - Tablespaces in Abnormal State 1264
      - Total SMS Space Used 1267
      - Units of Work 1265
      - Workload Analysis 1268
  - Sybase 1137
  - Hot Objects Tab 935
  - Hot Code 936
  - Hot Tables 935
  - HPSEQT setting 172
HTML reports, default directories 73
schema/database/object reports 609
template directory 73
hysteresis value 243

I/O Busy 1023
I/O Details 950, 1113, 1230, 1370
I/O Errors 1146
I/O Performance Category View
Statistics 1167, 1287
I/O settings
tablespaces 260
I/O Tab 950, 1112, 1229, 1370
Average Disk Queue 951, 1114, 1231, 1371
Bytes per I/O Operation 951, 1113, 1230, 1371
Disk I/O Time 952, 1114, 1231, 1372
I/O Details 950, 1113, 1230, 1370
Total Disk Queue 951, 1114, 1231, 1371
Transfer Rate 951, 1113, 1230, 1370
I/O usage, logins 209
Identity Column
table columns 251
Identity Gap setting 250
identity in nonunique index 186, 390
idle time 232
Ignore Duplicate Key index setting 200
Ignore Duplicate Rows index setting 200, 394
Image Copies 121
images
columns, binding to caches 178
Impact Analysis 469
import
default directory 69
utility directory 79
import utility, MySQL 79
Importing Data 721
Inactive Threads 41
Include Longvvar Columns setting 212
Increment By
table columns 251
Index Buffer Pool database setting 187
Index Level setting 201, 225, 270
Index Level, primary keys 225
Index Page Ratio setting 202, 227, 269
Index Scans Tab 1182
Index Tablespace setting 211
Index Type index setting 200
Indexes 122
Analyzing 411
DBCC 446
Dealocating Unused Space 449
MySQL 581
Partitions 124
Rebuilding 492
Reorganizing 503
indexes
adding comments 198
binding to datacaches 178
cache binding strategies 178
capacity/growth statistics 809
cards settings 200
Creating for IBM DB2 292
creating for IBM DB2 295
creating for IBM DB2 for OS/390 and z/OS 310
creating for Microsoft SQL Server 338
creating for Oracle 355
creating for Sybase ASE 393
level trend 822
managing in tables 251
managing space 200
managing columns 199
object editor functions 198
page and row statistics 202
partitioning 202
primary key columns 225
privileges 198
properties 200
see also objects, database statistics settings 200
storage settings 201
usage/distribution in segments 242
viewing associated objects 242
viewing DDL/SQL 198
Indexes Editor for Microsoft SQL Server
Storage Tab 201
Indexes Editor for Sybase
Storage Tab 201
indexrec 184
indexsort 184
Indoubt Threads 40
Inherit Special Registers setting 197, 230
Initial Extent
primary key settings 227
Initial Extent setting 201, 215, 239, 253, 263, 268
initial extent, clusters 175
initial extent, materialized views 214
Initial Transactions
primary key settings 227
Initial Transactions setting 201, 215, 239, 253, 263, 268
initial transactions, clusters 175, 214
input parameters 231
Insert Partition 405
Instance 126
Accessing 36
Configuration 37
Displaying Version Information 37
Instance Configuration 623
Instance Parameter Configuration
Changing 669
Extracting 669
Generating Report 669
Instance Parameters 37
instances setting
clusters 175
Instances setting 216
Interrupts/Sec 942, 948, 1105, 1110, 1222, 1227, 1362,
Intra-partition package setting 218
Invalid Objects Tab 936
Global Summary 938
Invalid Objects Detail 937
Invalid Objects Summary 936
IOT properties, tables 254
IP addresses
Performance Center 79
Isolation Level package setting 218
ISQL Editor
Shortcut Keys 538
ISQL editor
options/preferences 66, 75
ISQL Window
Column Auto Lookup 733
Using Auto Replace Shortcuts 733
ISQL Window Status Bar 540
IsValid index setting 200
Java
load/drop options 76
options/preferences 66
Java Classes 127
Java Resources 127
Java Sources 127
jConnect 76
JDBC
options/preferences 66, 76
JDBC driver 89
JDBC Driver Editor 89
Job Queue Activity Tab 913
Access Pattern Detail Tab 914
Job Queue Summary 913
Problem Jobs 914
Running Jobs Detail 914
Job Queue Wizard 358
Job Queues 127
job queues
enabling/disabling jobs 203
modifying the schedule 203
modifying What Parameter 203
object editor functions 203
see also objects, database
viewing DDL/SQL 203
joins
  type/index Query Builder restrictions 80
K
Katakana Charset package setting 219
Katakana charset setting 188
Keep Dynamic package setting 188
Kerberos authentication 87
key size, clusters 174
Keyboard Commands 57
Keyboard Emulation Mode 540
Killing Processes 616
Knowledge Base 8
Label table setting 250
Language
  functions 196, 230
language
  logins, associated with 207–208
  user messages 271
Language Level package setting 218
Language package setting 219
Language setting 188
Large I/O Hit Rate 1140, 1154
Last Bind Time package setting 218
Last Hour Response Time
Oracle 10g 996
Last Modified setting 197
packages 221
last virtual page, database device 180
Latches
  capacity/growth statistics 810
Leaf Blks/Key setting 201, 270
Leaf Blks/Key, primary keys 225
Leaf Blocks setting 201, 270
Leaf Blocks, primary keys 225
Leaf Pages setting 201, 225, 270
Libraries 128
  Creating for Oracle 359
libraries
  dynamic/status settings 204
  modify path/file name 204
  object dependencies 204
  object editor functions 204
  privileges 204
  see also objects, database
  viewing DDL/SQL 204
Library Cache Tab 893
  Library Cache Hit Ratio 893
  Library Cache Performance 894
Pinned Objects 895
library name, extended procedures 192
Library Wizard for Oracle 359
Licensing 12
  Registration Wizard 12
licensing
  options/preferences 66, 77
Licensing DBArtisan 12
lifetime, password 232
Limit Key Values index partition setting 202
line numbers 74
Linked Servers 128
linked servers
  creating for Microsoft SQL Server 340
  managing logins 206
  object editor functions 204
  privileges 204
  see also objects, database
Linked Servers Editor for Microsoft SQL Server
  Information Tab 205
list composite 202, 228, 269
Live Mode execution 69
Load Average 941, 1103, 1220, 1361
load data
default directory 69
Load Data Utility Wizard for DB2 OS390 632
Load Java 469
load, Java 76
LOB columns
space usage
tables
LOB column space usage 255
local extent management 263
Local Server setting 234–235
Locality index partition setting 202, 228
Locality unique key partition setting 269
Lock
Datasource 539, 567
Lock Isolation package setting 219
Lock List Utilization 1256
Lock Scheme setting 250
Lock Size setting 250
Lock Size tablespace setting 261
Lock Table 471
lock time, account 232
Locked Objects 933
locking
connection options 75
locklist 184
Locks 617
capacity/growth statistics 810
Locks status 260
Locks Tab 1086
Locks 1087
Locks by Lock Type 1088
Locks/Blocking Locks Tab 985
All Locks 985
Blocking Locks 986
Locksize, materialized query tables 212
locktimeout 184
Log Cache Hit Ratio 1022
Log Cache Tab 1042
Database Log Cache Ratio 1042
Log Cache Details 1042
Log Cache Hit Ratio 1042
Log File
Truncating 625
Log Space Utilization 1256
Log Storage Definition 471
log_retain_status 184
logbufsz 184
Logfile 624
Activating Logging 624
Opening 624
Options 624
Truncating 625
Logfile Options 624
logfilsize 184
Logged Initially setting 212
logging
capacity/growth statistics 809
for tablespaces 261
index setting 200
licensing, options 77
materialized query table options 212
materialized view logs 215
216
primary key index constraints 226
tablespace options 262
unique keys 268
Logging setting 250
Logging, SQL
options, SQL 78
logging, SQL
options/preferences 66
loghead 184
Logical Changes 903
Logical I/O 903
Logical Reads 903
logical reads 232
Login Wizard for Microsoft SQL Server 341
Login Wizard for Sybase ASE 394
LogIndexBuild setting 250
Logins 129
Creating for Microsoft SQL Server 341
Creating for Sybase ASE 394
logins
accounting statistics 209
basic information for 207–208
extracting with user 70
locked/unlocked 208
managing roles 209
object editor functions 207
password settings 208, 238
profile limits 232
see also objects, database
status 208
to linked servers 206
user/database account 208
viewing DDL/SQL 207
loghead 184
logprimary 184
logretain 184
logs
specifying for databases 184
Logs Low on Space 1027
Sybase 1149
logsecond 184
Long Tablespace setting 211
LPL status 260
M
Mailing Result Sets 569
Make Unique setting 201, 225, 269–270
manual segment management 263
Map
Tablespace 263
map, tablespace 263
Master View materialized view setting 213
Match Full setting 193
Materialized Query Table
Creating 293
Materialized Query Tables 131
materialized query tables
managing columns 210
object editor functions 209
page/row statistics 212
partition columns 211
performance settings 211
privileges 210
query optimization 211
refresh details 211
see also objects, database
tablespace settings 211
Materialized setting 251
Materialized View Log Wizard for Oracle 364
Materialized View Logs 132
Creating for Oracle 364
materialized view logs
cache setting 215
column filtering 215
data block storage 215
extents 215
log table/replication 215
logging option 215
object dependencies 215
object editor functions 214
parallel query option 215
performance settings 215
see also objects, database
viewing DDL/SQL 215
Materialized View Wizard for Oracle 360
Materialized Views 131
Compiling 432
Creating for Oracle 360
Synonyms 441
materialized views
datablock storage 214
extents 214
object dependencies 213
object editor functions 212
privileges 212
properties 213
queries 214
see also objects, database
storage settings 214
viewing DDL/SQ L 213
materialized views placement 214
Max Extents setting 201
Max Extents, primary keys 225
Max Row Size setting 227, 269
Max Rows Per Page setting 227, 269
Max Size of Non-Leaf Index Row setting 227, 269
Max Transactions
primary key settings 227
Max Transactions setting 201, 215, 253, 268
maxapps 184
maxflop 184
Maximum Extents
primary key settings 227
maximum extents 175
Maximum Extents setting 201, 215, 239, 253, 263, 268
Maximum Extents unique key setting 270
maximum extents
materialized views 214
Maximum Failed Logins setting 238
Maximum Rows Per Page setting 226, 250, 268
maximum transactions, clusters 175, 214
maxlocks 184
MDL Read Hits % 957, 1120, 1237, 1377
members
redo log group 233
Memory Drill Down 885
Memory Manage
capacity/growth statistics 810
Memory Performance Category View
Statistics 1150, 1269
Memory Tab 952, 1115, 1232, 1372
Buffer Size 956, 1119, 1236, 1376
Cache Efficiency 957, 1119, 1236, 1377
Cache Efficiency Copy Read Hits % 957, 1120, 1236, 1377
Cache Efficiency Data Map Hits % 957, 1120, 1237, 1377
Cache Efficiency MDL Read Hits % 957, 1120, 1237, 1377
Cache Efficiency Pin Read Hits % 957, 1120, 1237, 1377
Cache Size 956, 1119, 1236, 1376
Memory Available 955, 1117, 1235, 1375
Page Replacements 956, 1119, 1236, 1376
Page Replacements Clock Algorithm Scans
(Pages/sec) 956, 1119, 1236, 1376
Page Replacements Memory Freed Algorithm Scans
(Pages/sec) 956, 1119, 1236, 1376
Paging Activity 953, 1115, 1232, 1373
Paging Activity Blocks Output/Sec 953, 1115, 1232, 1373
Paging Activity Paged In 953, 1116, 1233, 1373
Paging Activity Paged Out 954, 1116, 1233, 1374
Paging Activity Pages Input/Sec 954, 1118, 1233, 1374
Paging Activity Pages Output/Sec 954, 1117, 1234, 1374
Menu Animations 56
Menu Contents 56
Menu Shadows 56
Menus 53
Main Menu 54
merge publish 185, 335
Message Confidentiality setting 235
Message Integrity setting 235
methods
managing for structured types 246
structured types 246
Microsoft SQL Server Agent 46
Migrate 472
migration
options/preferences 66
migration, schema 605
mincommit 184
Minimum Extent Size setting 263
Minimum Extents
primary key settings 227
minimum extents 175
materialized views 214
Minimum Extents setting 201, 215, 239, 253, 263, 268
Minimum Password Length setting 238
Minimum Percent Used setting 268
Minimum Rows Per Page setting 227, 269
Mixed Character Set package setting 219
Mixed Character Set setting 188
Mode materialized view setting 213
Modify Cluster Column 407
Modify Datafile 408
Modify Partition 405
Modify Recovery 479
Modify Statistics 479
Modify User Message Text 441
Move Log 481
MRU Replacement Strategy setting 200, 250, 394
Multi-node Bound package setting 218
multipage_alloc 184
Multithreading 52
Mutual Authentication setting 235
MySQL
Creating a Primary Key 582
Creating a Unique Key 582
Creating an Index 582
Creating or Editing a Backup Operation 596
Creating or Editing Foreign Keys 578
Creating or Editing Functions 580
Database Backup 595
Datasources 577
Flush Options 577
Foreign Keys 578
Full-Text Index 582
Functions 580
Indexes 581
Primary Keys 581
Session Variable Options 561
Spatial Index 582
Tables 588
Unique Keys 581
Users and Privileges 583
utilities options/preferences 66
utility options/preferences 79
MySQL Databases 586
Creating or Editing 587
MySQL Tables
Creating or Editing 589
mysqldump 79
mysqlimport 79
N
Network Details 1122
Network Name setting 235
Network Output Queue Length 943, 1105, 1222, 1363
Network Password Encryptions setting 235
Network Queue Length 943, 1105, 1222, 1363
Network Tab 959, 1122–123, 1239, 1379
Inbound Packet Errors 962, 1124, 1242, 1382
Network Details 959, 1239, 1379
Outbound Packet Errors 961, 1124, 1241, 1381
Packet Collisions 960, 1123, 1240, 1380
Packet Discards 961, 1124, 1241, 1381
Packet Discards Inbound Packets Discarded 961, 1124, 1241, 1381
Packet Discards Outbound Packets Discarded 961, 1124, 1241, 1381
Packet Errors 961, 1124, 1241, 1381
Transmission Queue Length 960, 1123, 1240, 1380
Transmission Rate 960, 1240, 1380
Transmission Rate (Bytes) 960, 1123, 1240, 1380
Transmission Rate Packets Received/Sec 960, 1123, 1240, 1380
Transmission Rate Packets Sent/Sec 960, 1123, 1240, 1380
Transmission Received Rate (KB/Sec) 960, 1123, 1240, 1380
Transmission Sent Rate (KB/Sec) 960, 1123, 1240, 1380
New Datasource
Register 86
newlogpath 184
Next Extent
primary key settings 227
Next Extent setting 201, 215, 239, 253, 263, 268
next extent, clusters 175
next extent, materialized views 214
nextactive 184
no chkpt on recovery 186, 390
no free space acctg 186, 390
No Sort index setting 200
No Sort unique key setting 268
No. of Extents, primary keys 225
Node Connection 36
Node Group Wizard for IBM DB2 295
Node Groups
Creating for IBM DB2 295
Nodegroups 133	nodegroups 216
adding comments 216
assigned to bufferpools 172
object editor functions 216
see also objects, database
tablespace default 260
viewing DDL/SQL 216
Norely primary key setting 226
Not For Replication setting 193
NoValidate index constraint status 226
nulls
allowable in view columns 275
num_db_backups 184
num_estore_segs 184
num_freqvalues 184
num_io_cleaners 184
num_iocleaners 184
numQuantiles 184
Number of Extents unique key setting 270
Number of Logins 946, 1109, 1226, 1366
Number of Processes 946, 1109, 1226, 1366
numeric roundabout 185, 335
numsegs 184

O
OAM Page Ratio setting 202, 227, 269
OBID table setting 250
object editors
aliases 170
backup devices 171
bufferpools 172
check constraints 173
clusters 174
data caches 177
database devices 179, 182
databases 182
defaults 189
directories 189
event monitors 190
extended procedures 191
foreign keys 192
functions 194
groups 197
indexes 198
job queues 203
libraries 204
linked servers 204
logins 207
materialized query tables 209
materialized view logs 214
materialized views 212
max per open operation 75
nodegroups 216
outlines 217
package bodies 217
packages 217
plans 221
primary keys 224
procedures 228
profiles 232
remote servers 234
rollback segments 238
rules 236, 240
schemas 240
segments 241, 243
stogroups 244
structured types 245
synonyms 246
tables 247
tablespaces 257
triggers 264
unique keys 266
user datatypes 270
user messages 271
users 272
views 274
XXX 233
Object Status trigger setting 265
Object Types
Creating for Oracle 384
Objects
Creating 279

Objects Summary Tab 938
Objects Tab 1070
Database Object Detail 1071
Database Object Summary 1070
objects, database
DDL extract options 71
extraction 604
growth reports 821
migration 605
publishing HTML reports 609
offline 185, 335
Offline Databases 1026
OMF
Oracle Managed Files 101
Open Control Files 482
Operative package setting 219
Optimal Size setting 239
Optimization Hint package setting 219
options
database settings 184
databases 185
options (application) 66
Oracle
global database statistics 809
utilities options/preferences 66
Oracle Job Queues 133
OS View
Processor 941, 1103, 1220, 1360
Processor Speed 941, 1103, 1220, 1360
Outlines 134
outlines
creating for Oracle 366
object editor functions 217
see also objects, database
output parameters 231
Output Window
Configuring 63
Displaying 64
Docking 64
Hiding 64
Messages
Copying 65
Undocking 64
output window
maximum entries 75
Overflow Rows setting 75
Overhead setting 260
Overview Tab 1094
Database Object Analysis 1094
Database Size Analysis 1094
Database Summary 1094
Database User Analysis 1094
owner
transferring databases between 184

P
Package Bodies 136
Compiling 432
package bodies
object editor functions 217
see also objects, database
viewing DDL/SQL 217
Package Size package setting 219
Package Source package setting 219
Package Wizard for Oracle 366
Packages 134
  Compiling 432
  Creating for Oracle 366
  Rebinding 490
  Synonyms 441
packages
  adding comments 218
  bind parameters 219
  body specification 221
  body/header status and size 221
  copy statements to SQL editor 220
  header specification 220
  managing definition settings 218
  modifying properties 219
  object dependencies 218
  object editor functions 217
  see also objects, database
  synonyms to 247
  viewing DDL/SQL 218
packet size 68
Pad Index setting 200–201
Page Faults/Sec 942, 1105, 1222, 1362
Page Fetch Pairs setting 201, 225
page settings
  tablespaces 260
Page Size setting 172
Page Size table tablespace setting 255
page statistics
  materialized query tables 212
Page Stealing Algorithm setting 172
Page Writes package setting 219
Paged Memory Used 941, 1104, 1221, 1361
Pages Input/Sec 954, 1116, 1233, 1374
Pages Output/Sec 954, 1117, 1234, 1374
Pages Reserved setting 227, 269
Pages Used setting 227
Paging Activity 953, 1115, 1232, 1373
Parallel Degree setting 250
Parallel Execution index settings 200
Parallel Instances setting 250
Parallel Query Option 200
  materialized view logs 216
parallel query options 175
Parallel Sequential setting 172
Parameter Style setting 197, 230
Parameters
  Extracting Server Configuration 94
parameters
  functions 194
  managing for procedures 231
Parsed Size setting 197, 231
packages 221
Partition Type index setting 202, 228
Partition Type unique key partition setting 269
Partition Upper Bound 484
Partitioned Columns 483
Partitioning 379
  Composite 379
  Hash 379
  List 379
  Range 379
partitions
  adding to nogroup 216
  bufferpools 172
  capacity/growth statistics 809
  enabling/disabling in tables 253
  for materialized query tables 212
  locking in tablespaces 261
  managing for indexes 202
  raw, containers 259
  tablespaces 260
  unique keys 268
Password Expiration setting 238
Passwords
  Changing 422
passwords
  assigning to users 274
  database links, associated 182
  expired status 274
  linked server logins 206
  logins, associated with 208, 238
  profile limits 232
  role authentication 237
path
  trigger functions 265
paths
  directories 190
  pckcachesz 184
PDS Name package setting 219
Percent Free index setting 201
Percent Free primary key setting 226
Percent Free setting 201, 253, 268
percent free, extents 175, 214
Percent Free/,Used setting 215
Percent Increase setting 201, 215, 253, 263, 268
percent increase, extents 175
percent increase, materialized views extents 214
percent used, extents 175, 214
performance
  tablespaces 259
Performance Analyst
Copy Template 847
Datasource Thresholds Maintenance 845
License Manager 831
Licensing 831
Opening 830
OS Page 1100, 1358
  Available Paged Memory 1106, 1223, 1363
  Available Physical Memory 1106, 1223, 1363
  Available Swap Memory 1106, 1223, 1364
  Average Disk Queue Length 1104, 1221, 1361
  Bottleneck Analysis 1102, 1219, 1359
  Disk Analysis 1102, 1219, 1360
  Disk Time 1103, 1220, 1360
  Free Disk Space 1108, 1225, 1365
  Interrupts/Sec 1105, 1222, 1362
  Key Resource Usage 1101, 1218, 1359
  Load Average 1103, 1220, 1361
  Memory Analysis 1102, 1219, 1359
  Network Output Queue Length 1105, 1222, 1363
  Network Queue Length 1105, 1222, 1363
  Number of Logins 1109, 1226, 1366
  Number of Processes 1109, 1226, 1366
  OS Detail 1109–1117, 1119–1124, 1226–1242, 1366–1368, 1370–1382
  Page Faults/Sec 1105, 1222, 1362
  Paged Memory Used 1104, 1221, 1361
  Processor Queue Length 1105, 1222, 1362
  Processor Time 1103, 1220, 1360
  Swap Memory Used 1104, 1221, 1361
  Top CPU Process 1108, 1225, 1365
  Top I/O Process 1108, 1225, 1366
  Top Memory Process 1108, 1225
  Total Disk Space 1107, 1224, 1365
  Total Paged Memory 1107, 1224, 1364
  Total Physical Memory 1107, 1224, 1364
  Total Swap Memory 1107, 1224, 1364
  Used Disk Space 1107, 1224, 1364
  Workload Analyst 1103, 1220, 1360
OS Page Statistics 1218
Product Design 833
Save 842
SelectDatasource(s) 846
Threshold Properties 843
Threshold Template Maintenance 845
Threshold Template Properties 844
Threshold Templates 844
Thresholds 843
Top Sessions 847
Top SQL Page 1125
  Criteria Tab 1127
  SQL Details Tab 1127
  Summary Tab 1125
Turning off Alarm Sound 835

Performance Analyst for Microsoft SQL Server
Tutorial 1011
Open Transactions 1003
Overview Tab 1003
SQL Tab 1009
Space Page 915
  Autoextend Datafiles at Extend Limit 920
  Bottleneck Analysis 916
  Objects at Maximum Extents 919
  Objects with No Room to Expand 919
  Offline Tablespaces 920
  Redo Logs 917
  Space Analysis 915
  Space Detail 920–926
  Space Drill Down 920
  SQL Analysis 917
  Top Fragmented Datafiles 918
  Top Space Hogs 917
  User Space Analysis 917
Statistics
  Memory Page 880
  Users Page 966
    Active Sessions 972
    Bottleneck Analysis 967
    CPU/Parse Ratio 969
    Inactive Sessions 972
    Key User Analysis 966
    Memory Sort Ratio 971
    Open Cursors 972
    Parse/Execute Ratio 970
    Sessions Blocked 974
    Sessions in Disk Sorts 974
    Sessions Waiting 973
    Sessions with Enqueue Waits 976
    SQL Analysis 967
    Top Bottleneck Events 973
    Transaction Analysis 969
    Users Detail 976–978, 980–982, 984–986
    Users Drill Down 976
    Workload Analysis 969
Performance Analyst Microsoft SQL Server Expert Guide 1020
Performance Analyst's
OS Page
  Top Memory Process 1365
Performance Analysts
OS Page
  OS Detail 1376
Performance Center 730
  client option 79
  connectivity options 79
  Opening within DBArtisan 730
  options/preferences 66
  Web Client 730
performance center
  identifiers 79
  options/preferences 79
performance settings
  clusters 175
Permissions
  Cascading 735
  Explicit 735
  Using Roles to Grant 735
  Using Roles to Revoke 736
permissions
  DBA/data dictionary 75
  Permissions Management 735
PGA Tab 896
  PGA Overview 896
  PGA Session Detail 898
  PGA Workarea Summary 898
PGSTEAL setting 172
Physical File Name setting 184
Physical I/O 902
physical name 171
  backup/dump device 171
physical name, database device 180
Physical Reads 902
Physical Writes 902
Piece Size index setting 200
Piece Size primary key setting 226
Pin Read Hits % 957, 1120, 1237, 1377
PIPE event monitor target 191
PL/SQL Debugger 740
  Debugging a Sample Script 766
    Sample Script 1 768
    Sample Script 2 771
  Debugging a SQL Script 758
  Debugging Sample Script 2 771
  Exclusion List 748
Functionality 752
  Breakpoints 755
  Close 756
  Go 755
  Input Parameters 752
  Pause 756
  Restart 756
  Run to Cursor 754
  Step Into 753
  Step Out 753
  Step Over 754
  Stop 756
Interface 749
  Call Stack Window 751
  Dependency Tree Window 751
  Variables Window 750
  Watch Window 750
Options 748
Using 757
  Opening a Debugging Session 757
PL/SQL Profiler
  Functionality 777
    Clear Profile Table 780
    Flush 778
    Run Detail 779
    Run Summary 779
    Start 778
    Stop 781
    Unit Detail 780–781
    Unit Summary 780
  Requirements 776
  Setting Up 776
  Using 782
    Sample Session 783
Place 485
plans
  object dependencies 222
  object editor functions 221
  privileges 222
  see also objects, database
port numbers
  Performance Center 79
precision
  function parameters 195
  precision, function parameters 195, 231
Precompile Timestamp package setting 219
Precompile Timestamp setting 188
preference (application) 66
pre-fetch options 72
Prefetch setting 201
Prefetch Size setting 260
Prefetch Strategy setting 200, 250, 394
Preview 486
Preview Create 485
Primary Keys 137
  Changing Constraint Status 423, 524
  Creating for IBM DB2 295
  MySQL 581
primary keys
  adding comments 224
  card settings 225
  creating for IBM DB2 for OS/390 and z/OS 312
  creating for Oracle 367
  creating for Sybase ASE 395
  extents settings 226
  freelist settings 226
  managing storage 226
  object editor functions 224
  page/row statistics 227
  sattribute settings 225
  see also objects, database
  setting properties 225
  sort option
    Asc Sort setting 225
  specifying columns 225
  sstatistics settings 225
  storagesetting 225
  viewing DDL/SQL 224
Primary Space Allocation 226
Primary Space Allocation index setting 201
PRIQTY index partition setting 202
private SGA 232
Privileges
  Deny Privileges From 738
  Grant Privileges 736
  Revoke Privileges To 737
privileges
  bufferpools 172
  databases 183
  directories 190
  extended procedures 192
  functions 194
  groups 197
  indexes 198
  libraries 204
  linked servers 204
  materialized views 212
  MQTs 210
  plans 222
  procedures 228
  profiles 232
  roles 236
  schemas 241
  sequences 243
  stogroups 244
  tables 248
  users 272
  views 274
Privileges and Users
  MySQL 583
Procedure Cache Tab 1040
  Procedure Cache Hit Ratio 1040
  Procedure/SQL Cache Details 1041
  Procedure/SQL Cache Hit Summary 1041
Procedure Plan Hit Ratio 1021
Procedure Reads from Disk 1153
Procedure Requests 1153
Procedure Wizard
  Panel 1 for Microsoft SQL Server 343
Procedures 137
  Compiling 432
  Creating for Oracle 369
  Recompiling 429
  Synonyms 441
procedures
  adding comments 228
  creating for IBM DB2 296
  creating for IBM DB2 for OS/390 and z/OS 313
  creating for Microsoft SQL Server 342
  creating for Oracle 369
  creating for Sybase ASE 396
  extract options 461, 607
  managing input/output parameters 231
  migration options 607
  modifying SQL definition 230
  object dependencies 228
  object editor functions 228
  privileges 228
  see also objects, database
  synonyms to 247
  threshold-associated 243
  viewing DDL/SQL 228
Processes Tab 949, 1111, 1228, 1368
Processor Queue Length 943, 948, 1105, 1111, 1222, 1228,
  1362, 1368
Processor Time 940, 1103, 1220, 1360
Profile Wizard for Oracle 369
Profiles 138
  Creating for Oracle 369
profiles
  assigning to users 274
  assigning/unassigning users 232
  modifying general limits 232
  modifying login limits 232
  modifying password limits 232
  modifying session limits 232
  modifying time limits 232
  object editor functions 232
  privileges 232
  see also objects, database
profiler
  options/preferences 72
  public, database device 182
Publication Server setting 234
published 185, 335
Publisher/Subscriber setting 234
Q
  Qualifier package setting 219
queries
  builder options 79
  materialized views 214
  optimization, MQTs 211
Query Builder
  Auto Join options 80
  code generation options 80
  display/style options 80
  execution options 80
  syntax check options 80
Query Builder 421
  Adding a Join 710
  Auto Join Properties 694
  Auto Joins 712
  Auto Layout 711
  Building a CREATE VIEW Statement 704
  Building a DELETE Statement 703
  Building a Query 700
  Building a SELECT Statement 701
  Building an INSERT Statement 702
  Building an UPDATE Statement 702
  Column Properties 695
  Creating a Clause 712
  Creating a GROUP BY Clause 716
  Creating a HAVING Clause 717
  Creating a WHERE Clause 712
  Creating an AND Clause 713
  Creating an ORDER BY Clause 715
  Design 690
  Display Options 694
  DML Tab 691
  Editing Joins 710
  Execution Properties 694
  Explorer 690
  General Properties 694
  Getting Started 698
  Inserting an AND or OR Clause 714
  Joins 708
    Editing 710
    Inner 708
    Left Outer 709
    Properties 696
    Self 709
  Keyboard Commands 693
  Moving Additional Tables to the SQL Diagram Pane 706
  Moving Columns in the SQL Statement Tree 718
  Moving Tables 706
  Moving Tables and Columns in the SQL Statement Tree 718
  Moving Tables and Views 706
  Moving Tables in the SQL Statement Tree 718
  Opening Queries 719
  Removing a Join 710
  Results Tab 692
  Saving Queries 719
  Saving Queries as XML 719
  Schema Change Detection 720
  Selecting a Statement 699
  Selecting ALL Columns 707
  Selecting an Instance or Database 698
  Selecting Columns 699
  Selecting Columns in the SQL Diagram Pane 700
  Selecting Columns in the SQL Statement Tree 700
  Selecting DISTINCT Columns 708
  Selecting Tables 705
  Setting Column Properties 695
  Setting options/preferences 79
  SQL Diagram Pane 691
  SQL Statement Pane 691
  SQL Tab 691

Statement Properties 694
  Subqueries 718
  Syntax Checker 719
  Table Properties 695
  Table Shortcut Menus 692
  Tables in the SQL Diagram Pane 705
  Tables/Views Tab 691
  Toolbar 692
  Using 697
  Using Data Editor with Query Builder 720
  View Properties 695
  Working with Tables in the SQL Diagram Pane 705
  Workspace Window Options 690
query builder
  options/preferences 66
Query Optimization package setting 218
Query Options 561
Query Plan Facility 561
query rewrite, materialized views 213
Quiesce Database 487
Quiesce Instance 487
Quiesce Tablespaces 488
quotas, user 260
quoted identifier 185, 335
quoted identifiers 68
R
RAC Detail View 987
  Bottleneck Analysis 988
  Conflicts 989
  Instance Analysis 988
  Key Ratio Analysis 987
  Locks 993
    DLM Lock Activity 993
    Latch Detail 993
    Lock Convert Summary 993
    PCM Lock Summary 993
  Memory 990
    Data Dictionary Cache Analysis 992
    Key Ratio Analysis 990
    Library Cache Analysis 992
    Memory Analysis 991
  Overview 987
  Ping Activity 992
    Ping by Datafile 992
    Ping by Instance 992
    Ping by Object 992
  Top Rac Related Waits 989
  Workload Analysis 990
range-hash/list composite 202, 228, 269
Raw Partition/Drive settings 259
read only 185–186, 335, 390
reads, logical 232
Reassign by Category 489
Rebind Packages 490
Rebind Plans 490
Rebuild Index 491
Rebuild Outlines 494
rec_his_retentn 184
Recompile 494
Recovery Status setting 260
Recovery Wizard 650
recursive triggers 185, 335
Redistribute 494
Redo Log Group Wizard for Oracle 371
Redo Log Groups 139
redo log groups
object editor functions 233
see also objects, database
viewing DDL/SQL 233
working with members 233
Redo Writes 902
Reducing the size of Microsoft SQL Server Restores 650
REF CURSOR
Display in ISQL Window 538
Refit 495
Refresh Method materialized view setting 213
Refresh Table 496
refresh, materialized query tables 211
Register
Datasource 86
Registration Wizard 12
Reinstalling DBArtisan 15
Release Bound package setting 219
release setting 184
Rely primary key setting 226
Rely unique key setting 268
Remote Server Wizard for Microsoft SQL Server 343
Remote Server Wizard for Sybase ASE 396
Remote Servers 139
Creating for Microsoft SQL Server 343
Creating for Sybase ASE 396
remote servers
identification settings 234–235
object editor functions 234
see also objects, database
viewing DDL/SQL 234
Remote Servers Editor for Microsoft SQL Server
Information Tab 234
Logins Tab 235
Remote Servers Editor for Sybase
Logins Tab 235
Reoptvar package setting 219
Reorganize 498
Reorganize Tablespace 503
Repairing Backup Analyst 1469
Replacement Strategy setting 201, 253
replication
check constraints 174
Replication Tab 1100
Replication Agent Details 1100
Report 510
reports
Capacity Analyst 821
HTML (schema/databases/objects) 609
Reserve Page Gap setting 200–201, 226, 250, 394
Reserved Pages setting 202
Resize 573
Resource Cost 670
Resource Limit Wizard
Last Panel 675
Panel 2 673
Panel 3 674
Starting 673
Resource Limits 671
Adding 673
Dropping 676
Editing 675
Editor 675
Named Time Ranges 674
Scopes 675
Search Parameters 672
Setting Time Ranges 674
Showing 672
Types 675
Resource Release package setting 219
Response Time
Bottleneck Detail 1000
Bottleneck History 1001
Bottleneck Summary 1000
Current Workload 995
Historical Session Analysis 1002
Historical Session Analysis Details 1002
Last Hour Detail 997
Session Bottleneck Detail 1001
Session Bottleneck Summary 1001
Session Time Detail 999
Session Wait History 1002
System Bottlenecks Tab 1000
System Time Detail 998
System Time Summary 997
Response Time Analysis 995
Active Time Tab 997
Historical Session Analysis Tab 1002
Session Bottlenecks Tab 1001
Wait Histogram View 1002
Response Time Tab 995
Response Times
Current 996
Restart 511
restore_pending 184
Restoring Databases 650
Restoring Microsoft SQL Servers 650
RestrictDrop setting 250
RestrictDrop table setting 250
Result Sets
Closing 569
Editing 570
Formatting 574
Mailing 569
Saving 569
Saving as XML 569
Results Editor 567
  Closing a Result Set 570
  Closing a Result Window 569
  Configuring 567
  Editing Result Sets 570
  Exporting Data 568
  Mailing a Result Set 569
  Opening Result Files 543
  Read-only Mode 568
  Saving Results 569
  Status Bar 568
results window
  grid options 75
  options/preferences 66
results, SQL
  format options 81
  window/file/set options 80
return parameters, functions 194
reuse max, password 232
reuse time, password 232
Reverse index setting 200
Reverse primary key setting 226
reverse scans, indexes 200
Reverse unique key setting 268
Revoke Role 511
Right Outer Joins 709
Role Locked setting 238
Role Wizard for Microsoft SQL Server 344
Role Wizard for Oracle 372
Role Wizard for Sybase ASE 397
Roles 140
  Creating for Microsoft SQL Server 344
  Creating for Oracle 372
  Creating for Sybase ASE 397
roles
assigning to logins 209
assigning to users 273
granting/revoking 236
object editor functions 236
password/authentication 237
privileges 236
see also objects, database
viewing DDL/SQL 236
working with defaults for logins 209
Roles Editor for Microsoft SQL Server
  Users Tab 237
Roles Editor for Oracle
  Users Tab 236
Roles Editor for Sybase
  Logins Tab 238
  Roles Tab 238
Rollback Contention Ratio 929
Rollback I/O Tab 907
  Active Rollback Details 908
  Rollback I/O 907
  Session Rollback Activity 908
Rollback Segment Wizard for Oracle 373
Rollback Segments 142
  Creating for Oracle 373
rollback segments
  activity levels 239
  associated tablespaces 238
  capacity/growth statistics 809
  enabled/disabled status 238
  extent settings 239
  object editor functions 238
  see also objects, database
  storage settings 239
  viewing DDL/SQL 238
rollbacks, materialized views 213
Row Compression setting 250
Row firing, triggers 266
Row Movement setting 250
Row Organization setting 250
row statistics
  materialized query tables 212
rows
  growth reports 821
  table details 252
RPC Out Server setting 234
rule binding
  table columns 251
Rules 143
rules
  binding, datatypes 271
  creating for Microsoft SQL Server 345
  creating for Sybase ASE 398
  object dependencies 240
  object editor functions 240
  see also objects, database
run size, bufferpool/data cache 179
Runstats Index 512
S
scale
  function parameter 195
Scheduling
  ISQL Window Scripts 684
  Microsoft Task Scheduler 685
Schema 517
schema
  copying 435
  extraction 604
  migration 605
  publishing HTML reports 609
  see also objects, database
schema extraction
  default directories 73
Schema Path package setting 219
schemas
  object editor functions 240
  privileges 241
  viewing DDL/SQL 241
Scratchpad settings 196
Script Execution Facility
  Using 682
Script Files 541
  Inserting Files 542
Secondary Space Allocation 226
Secondary Space Allocation index setting 201
SECQTY index partition setting 202
Secure Socket Layer encryption 89
Security Label setting 251
Security Mechanism setting 235
Security Model A/B setting 235
Security Policy setting 250
segdetect 184
segment management, extents 263
Segment Name setting 200, 226, 250, 268, 394
Segment Wizard for Sybase ASE 399
Segments 144
Creating for Sybase ASE 399
specifying for index 201
segments
dropping 242
extending 242
LOB column usage 255
object editor functions 241, 243
see also objects, database
thresholds 243
usage/space distribution 242
viewing associated object 242
viewing DDL/SQL 240, 242
Segments Editor for Sybase
Space Tab 242
Select * From 518
select into/bulkcopy/plsql 185–186, 335, 390
Selecting Columns 162
Selective Partition Locking tablespace setting 261
Sequence Wizard for Oracle 374
Sequences 144
Creating for Oracle 374
Synonyms 441
sequences
object dependencies 243
parameters/definition/datatype 244
privileges 243
see also objects, database
synonyms to 247
viewing DDL/SQL 243
Sequential Pages setting 201, 225, 270
Sequential Steal setting 172
Server Components
Installation for OS/390 and z/OS 323
Server Idle 1023
Server Messages
Printing 65
Saving 64
Server Space Tab 1071
Servers
Configuring 101
Disconnecting 94
Extracting Configuration Parameters 94
Generating a Report on Configuration 97
servers
connection options/preferences 68
Session Detail View
Oracle 1003
Session Details 615
Session Files
Executing 103
Session Information 671
Session Recording 101
Session Variable Options
ISQL Window 561
Session Waits Tab 982, 1084
Current Waits 984
Historical Waits 982
Session Wait Percent by Time Waited 1086
Session Wait Percent by Wait Type 1086
Session Waits 1084
sessions
defining profile limits 232
sessions Per User setting 232
Set Online/Offline 520
Set Tablespace Quota 521
Set UNDO Tablespace 522
Setting Access Levels While Updating Statistics 533
Setting Scope Level 613
Shortcut Menus 57
Shrink 523
Shrinks setting 240
single user 185–186, 335, 390
size 177
SMTP
encoding 82
notification identifiers
encoding
SMTP email 82
options/preferences 66, 82
Snapshot Logs 145
Snapshot Wizard for Oracle 375
Snapshots 145
Creating for Oracle 375
Synonyms 441
snapshots
synonyms to 247
softmax 184
Sort Activity Tab 910
Current Disk Sorts 912
Session Sort Detail 911
Sort in Tempdb setting 200–201
sort, index setting 200
sortheap 184
sorting
index option 199
Source Size setting 197
packages 221
Source Size setting 231
space
database devices 180
Space Analyst
Analysis Report 1420
Convert to Dictionary-Managed 1423
Convert to Locally-Managed 1422
Database Devices Tab 1418
Database Overview Tab 1393
Databases Tab 1417, 1419
Datafiles Tab 1394, 1397
Extract 1424
Files/Filegroups Tab 1415
Functionality 1420
Help 1390
Indexes Tab 1394, 1408, 1411
License Manager 1389
Licensing 1388
LOB data 1395
Map 1424
Map Tab 1415
Objects Tab 1417
Open 1425
Opening 1388
Options 1430
options/preferences 66
options/preferences/defaults 83
Product Design 1393
Reports 1420
Storage Overview Tab 1413
Tables Tab 1394, 1400, 1403
Tablespace Map Tab 1394, 1398
Tablespace Status 1395
Technical Support 1391
Thresholds Options 1430
Space Analyst Expert Guide 1461
Space Analyst for IBM DB2 for Open Systems
Tutorial 1455
Space Analyst for IBM DB2 for Windows/Unix
Tutorial 1455
Space Analyst for Microsoft
Diagnosing Fragmentation 1462
Forwarded Rows 1463
Space Analyst for Microsoft SQL Server
Tutorial 1441
Space Analyst for Oracle
Tutorial 1432
Space Analyst for Sybase ASE
Correcting Space-Related Problems 1452
Identifying Storage Structure Problems 1451
Obtaining a Global View of Storage 1450
Pinpointing Out of Space Problems 1451
Proactive Space Management 1453
Tutorial 1449
Using Analysis Reports 1452
Space Detail View
Sybase 1186, 1313
Space Fragmentation Tab 923
Tablespace Fragmentation 923
Space Management Wizard 1425
Space Performance Category View
Statistics 1183, 1304
Space Tab 958, 1121, 1237, 1378
Device Detail 959, 1122, 1239, 1379
Device Summary 958, 1121, 1238, 1378
space usage
databases/transaction logs 186
tables 252
space utilization
clusters 176
spacing. See style
special registers 197, 230
SQL
executed during migration 607
execution options 75
materialized views 214
procedures 230
script directory 73
See also debugger, SQL
viewing for aliases 170
viewing for backup devices 171
viewing for bufferpools 172
viewing for check constraints 173
viewing for clusters 174
viewing for data caches 177
viewing for database devices 179
viewing for database links 182
viewing for databases 183
viewing for event monitors 190
viewing for extended procedures 192
viewing for foreign keys 192
viewing for functions 194
viewing for groups 198
viewing for indexes 198
viewing for job queues 203
viewing for libraries 204
viewing for logins 207
viewing for materialized view logs 215
viewing for materialized views 213
viewing for nodegroups 216
viewing for package bodies 217
viewing for packages 218
viewing for primary keys 224
viewing for procedures 228
viewing for redo log groups 233
viewing for remote servers 234
viewing for roles 236
viewing for rollback segments 238
viewing for schemas 241
viewing for segments 240, 242
viewing for sequences 243
viewing for stogroups 244
viewing for structured types 245
viewing for tables 248
viewing for tablespaces 257
viewing for triggers 264
viewing for unique keys 266
viewing for user datatypes 270
viewing for user messages 271
sql
logging options 78
SQL * Loader
directory 79
INDEX

SQL * Loader Utility 606
SQL Access Level setting 196, 230
SQL Agent Tab 1098
  Job/Alert Detail 1098
  SQL Server Alert Summary 1098
  SQL Server Job Summary 1098
SQL Details Tab 1127
SQL Editor 538
  Closing a Query Window Tab 547
  Closing a Script 549
  Closing the Error List 550
  Editing a Script 550
  Executing Scripts 557
  Goto 546
  ISQL Windows 539
  Navigating 546
  Opening DDL Editors 540
  Opening Script Files 541
  Opening SQL Windows 539
  Regular Expressions 545
  Renaming a Query Window Tab 547
  Saving a Script 549
  Windows 542
    Splitting 543
SQL Error package setting 219
SQL Logging 624
SQL Scripting 538
  DDL Editor 540
  Results Editor 567
  SQL Editor 538
SQL SELECT 606
SQL Server
  Clusters 99
    Creating or Editing a Backup Operation 645
SQL Warn/Math Warn package setting 218
SQL Windows 539
  Find and Replace 543
  Replacing 544
  Searching 543
SQL. See also results, SQL
SQL/DDL
  displaying for a view 275
SSL encryption 89
Start Database 524
  start with
  table columns 251
stat_heap_sz 184
Statement firing, triggers 266
statements
  and event monitors 190
statistics
  CPU/IO usage, logins 209
statistics
  page and row, indexes 202
  primary keys, page/row 227
  unique keys, page/row 269
Statistics Recompute index setting 200
Status 524
  functions 197
  packages 221
status
  data caches 177
  libraries 204
  procedures 231
  rollback segments 238
  triggers 265
Status package setting 218
Stay Resident procedure setting 229
Step Execution Facility 560
stmtheap 185
Stogroup Wizard for IBM DB2 for OS/390 and z/OS 315
Stogroups 146
  Creating for IBM DB2 for OS/390 and z/OS 315
stogroups
  object editor functions 244
  privileges 244
  see also objects, database
  viewing DDL/SQL 244
Stop Database 526
Storage
  Database Devices 110
storage
  migration/extraction options 608
Storage Analysis
  Home 859
Storage Group database setting 187
Storage Group index setting 201
Storage Group primary key setting 226
storage settings
  bufferpools 172
  indexes 201
  rollback segments 239
Stored Procedure Build Option
  dialog box 229
STORNAME index partition setting 202
String Delimiter package setting 219
String Delimiter setting 188
stripe sets, containers 259
Structured Types 146
structured types
  attributes 245
  managing methods 246
  object editor functions 245
  providing method bodies 246
  see also objects, database
  viewing DDL/SQL 245
style
  auto-indent 74
  editor colours/ fonts
    color, editors 74
  ISQL window 67
  syntax highlighting options 74
  tabs to spaces 74
Subpartition type index setting 202, 228
Subpartition type unique key partition setting 269
subscribed 185, 335
Subscription Server setting 234
Summary Definition 527
summary reports, Capacity Analyst 821
Summary Tab 1125
Suspect Databases 1025
Swap Memory Used 941, 1104, 1221, 1361
Switch Online 527
Sybase
performance statistics 809
Sybase Expert Guide 1136
Synonym Wizard for IBM DB2 for OS/390 and z/OS 316
Synonyms 147
  Creating for IBM DB2 for OS/390 and z/OS 316
  synonyms
    creating for Oracle 375
    definition/object details 247
    object dependencies 247
    object editor functions 246
    see also objects, database
syntax
  checking, Query Builder 80
Syntax Check 539
syntax checking 75
SysEntries package setting 219
System I/O Tab 1051
  AWE I/O 1054
  I/O Activity 1054
  Physical I/O 1051
  Space I/O 1053
System Required setting 201, 225, 269
System Waits Tab 978, 1082
  System Waits 978, 1082
  Wait Percent by Time Waited 981, 1084
  Wait Percent by Total Waits 980, 1083
T
  TABLE event monitor target 191
Table I/O Tab 910
Table Lock Escalations 1025
Table Status setting 250
Table Tablespace setting 211
Table Type setting 250
Table Wizard for IBM DB2 for OS/390 and z/OS 316
Tables 148
  Analyzing 411
  Creating for IBM DB2 for OS/390 and z/OS 316
  DBCC 445
  Dealloacting Unused Space 449
  Histograms 411
  Moving Tables in the SQL Diagram Pane 706
  MySQL 588
  Partitioning for Oracle 379
  Reorganizing 501
tables
  adding a comment 248
  adding/deleting columns 250
  and event monitors 190
  basic properties 250
  binding to datacaches 178
  bufferpool 253
  cache binding strategies 178
  cache strategy 250
  capacity/growth statistics 809
  creating for IBM DB2 299
  creating for Microsoft SQL Server 345
  creating for Oracle 376
  creating for Sybase ASE 399
datablock storage 253
  extents 252–253
  freelists 253
  full text indexing 250
  growth reports 821
  managing dimensions 254
  managing indexes 251
  managing IOT properties 254
  managing tablespaces 255
  modifying column properties 250
  object dependencies 248
  object editor functions 247
  partitioning/unpartitioning 253
  privileges 248
  row information 252
  row locking strategy 250
  see also objects, database
  space usage 252
  synonyms to 247
  used in a cluster 176
  viewing associated segments 242
  viewing DDL/SQL 248
Tables Editor for Microsoft SQL Server
  Constraints Tab 252
Tables Editor for Sybase
  Constraints Tab 252
Tablespace Buffer Pool database setting 187
Tablespace Detail Tab 921
  Datafile Detail 921
  Object Space Tab 922
  Tablespace Object Summary 921
Tablespace Fragmentation 923
Tablespace Growth Tab 922
Tablespace I/O Tab 905
Tablespace Wizard for IBM DB2 300
Tablespace Wizard for IBM DB2 for OS/390 and z/OS 317
Tablespaces 151
  and materialized views 214
  Changing Status 423
  Creating for IBM DB2 300
  Creating for IBM DB2 for OS/390 and z/OS 317
  for materialized query tables 211
  Quiescing 488
tablespaces
  adding comments 257
  adding containers 258
  and event monitors 190
  assigning to databases 183
  bufferpool/nodegroup 259
  capacity/growth statistics 809
  containers, modifying 258
  creating for Oracle 380
  default/temporary, user 274
  dropped table settings 259
  extent details 262
  growth/failure prediction reports 821
  in indexes 201, 253
  in unique keys 268
  logging options 262
  managing datafiles 262
  managing in tables 255
  map 263
  object editor functions 257
  page settings 259
  partitions, modifying 260
  primary key settings 227
  properties 261
  rollback segments 238
  see also objects, database
  segment management 262
  status 260
  user quotas 260
  viewing DDL/SQL 257
  viewing stored objects 262
  tablespaces/I/O setting 259
.tabs
  to spaces 74
  Target setting, event monitors 191
  Task Scheduler 685
Technical Requirements 10
  Database 11
  Hardware 10
  Operating System 10
Technical Support 8, 16
  Contacting
    E-mail 17
    Phone 16
  E-mail 1392
  Phone 1392
  Web Site 1391
temporary tables
  automatic creation/deletion, extract 461, 607
temporary tables, migration options 607
Temporary Tablespace user setting 274
text
  columns
    binding to caches 178
    encoding scheme, tablespaces 261
Text Image Filegroup setting 250
Text In Row settings 250
Threadsafe function setting 196, 230
thresholds
  segments 243
time formats
  data editor 69
  Time out inactive connections setting 235
time-based settings
  profile limits 232
timeouts
  debugger 72
  server connection/query 68
  Timestamp package setting 219
timing setting, triggers 266
  Toggle 538
  Token Matching Capability 566
  Toolbars 55
  Moving 56
toolbars
  options/preferences 73
Tools
  Code Workbench 730
  Data Editor 722
  Database Search 680
  Embarcadero Products 729
  File Execution Facility 682
  Find in Files 680
  Performance Center Integration 730
  Query Builder 689
  Scheduling 684
  Script Execution Facility 681
  SQL Logfile 624
  Visual Difference 686
Tools Menu
  Customizing 59
    Specifying an Argument for a Command 60
  Top CPU Process 946, 1108, 1225, 1365
  Top I/O Process 946, 1108, 1225, 1366
  Top Memory Process 945, 1108, 1225, 1365
  Top Processes 871
  Top Sessions 676
  Top Sessions Tab 976, 1080
    Top CPU Process 977, 1081
    Top I/O Process 976
    Top Memory Process 977, 1080
    Top Sessions 977, 1081
  Top SQL Statistics
    Sybase 1242
  Top System Bottlenecks 1024
torn page detection 185, 335
  Total Disk Space 945, 1107, 1224, 1365
  Total Free Space 1029
  Total Keys setting 201, 225, 269
  Total Paged Memory 944, 1107, 1224, 1364
  Total Pages Modified setting 202, 227, 269
  Total Physical Memory 944, 1107, 1224, 1364
  Total Rows Modified Since Last setting 227, 269
  Total Sections package setting 218
Total Server Reads
  Sybase 1145
  Total Server Writes
  Sybase 1146
  Total Swap Memory 944, 1107, 1224, 1364
Total Used 1029
Transaction Log
Moving 481
Transaction Log Writes 1146
transaction logs
adding/deleting/editing 187
file details 187
space usage 186
transactions
and event monitors 190
Transfer Rate setting 260
Transmission Rate 1123
Triggers 152
triggers
adding comments 264
before/after timing 265
creating for IBM DB2 301
creating for IBM DB2 for OS/390 and z/OS 318
creating for Microsoft SQL Server 347
creating for Oracle 382
creating for Sybase ASE 400
events 265
firing event 265
function path 265
modifying action 265
modifying CREATE TRIGGER statement 264
object dependencies 264
object editor functions 264
row trigger information 265
see also objects, database
status 265
timing 265
type 265
update columns 265
viewing DDL/SQL 264
trunc log on chkpt 185–186, 335, 390
Truncate 528
Truncating
Logfile 625
Tables 528
T-SQL Debugger for Microsoft 740
Basics
Interface 749
Requirements 743
Client 746
Server 743
Features 740, 749
Functionality 752–753
Break 756
Breakpoints 754–755
Close 756
Go 755
Input Parameters 752
Restart 756
Run to Cursor 754
Step Into 753
Step Out 753
Step Over 754
Stop 756
Interface
Call Stack Window 751
DDL Editor 749
Dependency Tree Window 751
Variables Window 750
Watch Window 750
Options 748
Using 757
Debugging a Sample Script 759
Debugging an SQL Script 758
Opening a Debugging Session 757
Txn Log Writes 1146
Type Bodies 154
Compiling 433
Types 153
Compiling 433
U
UI
Setting Appearance 56
Setting Classic Microsoft Office 97 style 56
Setting Microsoft Office 2003 style 56
Setting Microsoft Office XP style 56
uniform allocation 263
Uninstalling DBArtisan 15
unique auto_identity index setting 390
Unique Keys 154
MySQL 581
unique keys
adding comments 266
clustered 267
creating for IBM DB2 for OS/390 and z/OS 319
creating for Microsoft SQL Server 348
creating for Oracle 384
creating for Sybase ASE 401
data block storage 268
enabling/disabling 267
extents 268
freelists 268
managing columns 267
object editor functions 266
page/row statistics 269
partitions 268
properties 267
see also objects, database
space allocations
space allocations
unique keys 269
viewing DDL/SQL 266
unique setting
indexes 200
Unlimited Max Size setting 184
Unlimited Max Size setting 187
unload data
default directory 69
Unquiesce Database 531
Unquiesce Instance 532
update columns, triggers 266
Update Rule setting 193
Update Statistics 533
Updating Table Statistics 533
Upgrading DBArtisan 15
USE statements
DDL extract options 70
Use statements
migration/extraction options 608
Use System Value tablespace setting 261
Use tablespace status 260
Used Disk Space 945, 1107, 1224, 1364
Used Pages setting 202, 269
used space
database devices 180
user allocation 263
User Datatypes 155
user datatypes
adding comments 270
columns of 271
creating for IBM DB2 304
creating for IBM DB2 for OS/390 and z/OS 321
creating for Microsoft SQL Server 349
creating for Sybase ASE 402
object editor functions 270
see also objects, database
viewing DDL/SQL 270
user datatypes
base definition 270
User I/O Tab 1054
Lock Activity 1057
SQL Activity 1056
Temporary Object Activity 1058
User Object Activity 1054
User Message Wizard for Microsoft SQL Server 348
User Message Wizard for Sybase ASE 403
User Messages 156
Creating for Microsoft SQL Server 348
Creating for Sybase ASE 403
user messages
binding to constraints 272
editing content/language 271
object editor functions 271
see also objects, database
viewing DDL/SQL 271
User Space Tab 925
Top Space User Accounts 926
user_exit_status 185
userexit 185
Users 157
Creating or Editing 584
SCHEMA 158
SQLID 157
users
assigning profiles 274
assigning roles 273
assigning to groups 274
assigning/unassigning, profiles 232
creating for IBM DB2 305
creating for Microsoft SQL Server 349
creating for Oracle 386
creating for Sybase ASE 403
database links, associated 182
extracting logins with 70
group membership 198
identification details 273
logins to linked servers 206
logins, database 208
modifying properties 273
object dependencies 273
object editor functions 272
privileges 272
see also objects, database
tablespace quotas 260
Users and Privileges
MySQL 583
Users Detail View
Sybase 1209, 1331
Users Editor for Microsoft SQL Server
Object Tab 273
Users Editor for Sybase
Objects Tab 273
Users Performance Category View
Statistics 1200, 1322
Using the Performance Center Web Client Within
DBArtisan 730
util_heap_sz 185
Utilities 602
Archive Activity 664
Chained Row Analysis 666
Data Export Wizard for IBM DB2 for Windows/Unix 625
Data Export Wizard for Oracle 655
Data Import Wizard for IBM DB2 for Windows/Unix 629
Data Import Wizard for Oracle 660
Data Load Wizard for Microsoft SQL Server and Sybase
ASE 641
Data Unload Wizard for Microsoft SQL Server and
Sybase ASE 639
Database Backup Wizard 645
Database Monitor 611
Database Restore Wizard 650
Free Space Deficit 667
Instance Parameter Configuration 669
Logfile 624
Maximum Extent Limits 668
Pinned Code 667
Resource Limits 671
Session Information 671
Unloading Data 639
V
Valid package setting 219
Validate index constraint status 226
Validate package setting 219
Validate unique key setting 268
ValidProc table setting 250
VCAT Catalog index setting 201
VCAT Catalog primary key setting 226
VCAT index partition setting 202
verify function, password 232
Version setting 188
Vertical Deferred Write Threshold setting 172
View Contents 537
View Wizard for Oracle 388
Views 160
  Compiling 433
views
  adding comments 274
  creating for IBM DB2 305
  creating for IBM DB2 for OS/390 and z/OS 322
  creating for Microsoft SQL Server 350
  creating for Oracle 388
  creating for Sybase ASE 404
  displaying for columns 275
  displaying SQL/DDL for 275
  object dependencies 274
  object editor functions 274
  privileges 274
  see also objects, database
  synonyms to 247
views query optimization 275
Virtual Log Files Tab 1070
Visual Difference
  Comparing Files 687
  Navigating 687
  Opening 686
  Options 689
  Printing 688
  Searching 688
Volatile setting 250
Volatile table setting 250
Volatile, materialized query tables 212
VPPSEQT setting 172
VPSEQT setting 172
VPSIZE setting 172
VPTYPE setting 172
W
wait events
  capacity/growth statistics 809
Waits setting 240
Warnings
  options/preferences 66
warnings
  Query Builder options 80
  setting 83
wash size 179
WEPR status 260
WHAT parameter 203
Width setting, user datatypes 270
Windows 939, 1101, 1218, 1358
    Navigating 63
Windows 2000 11
With Details 191
Wizard
    Database Backup 646
    Space Management 1425
wizards, default directories 73
Wraps setting 240
Write Buffer setting, event monitors 191
Write Mode setting, event monitors 191
Writes setting 240
X
XML 569
    Extracting Data 461