



Product Documentation

DB Optimizer™ XE and DB Optimizer™

Quick Start Guide

Version 2.5.1

Published August 2010

© 2010 Embarcadero Technologies, Inc. Embarcadero, the Embarcadero Technologies logos, and all other Embarcadero Technologies product or service names are trademarks or registered trademarks of Embarcadero Technologies, Inc. All other trademarks are property of their respective owners.

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, CodeGear™ RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.

August 12, 2010

CORPORATE HEADQUARTERS

100 CALIFORNIA STREET
12TH FLOOR
SAN FRANCISCO, CALIFORNIA
94111 USA

EMEA HEADQUARTERS

YORK HOUSE
18 YORK ROAD
MAIDENHEAD, BERKSHIRE
SL6 1SF, UNITED KINGDOM

ASIA-PACIFIC HEADQUARTERS

L7. 313 LA TROBE STREET
MELBOURNE VIC 3000
AUSTRALIA

Contents

Introducing Embarcadero DB Optimizer.	5
Major Interface Components	5
New Features	6
New Features in DB Optimizer 2.5	6
New Features in 2.5.1	6
Limitations of the Trial Version.	8
Limitations of DB Optimizer XE Developer	8
Additional Product Information.	8
Installation	9
Technical Requirements	9
Installation Issues	10
Installing DB Optimizer	11
Installing DBOptimizer XE	13
Getting Started with DB Optimizer	25
Start DB Optimizer	25
Overview of the User Interface	26
Profiling a Data Source	31
Tuning SQL Statements	36
SQL Code Assist and Execution	50
Code Extraction	52
Code Highlighting	53
Automatic Error Detection	53
Code Complete	55
Hyperlinks	55
Code Formatting	56
Code Folding	57
Code Quality Checks	57
SQL Execution	58
Configuring SQL Execution Parameters	59

INTRODUCING EMBARCADERO DB OPTIMIZER

Embarcadero DB Optimizer simplifies SQL optimization and development with many features for improving productivity and reducing errors. A rich SQL IDE with statement tuning, data source profiling, code completion, real-time error checking, code formatting, and sophisticated object validation tools helps streamline coding tasks. DB Optimizer's user interface helps improve overall productivity with integrated development, monitoring, and tuning components. DB Optimizer offers native support for IBM® DB2® for LUW, Oracle®, Microsoft® SQL Server and Sybase® as well as JDBC support for other DBMS. It is available as a stand-alone application, or as an Eclipse plug-in. DB Optimizer is structured and composed into three main interface parts. This design provides a comprehensive workflow that enables development, query analysis, and tuning capabilities. This workflow, in turn, leads to more efficient task management in terms of time and efficiency, overall.

MAJOR INTERFACE COMPONENTS

The three major interface components of DB Optimizer are as follows:

SQL Profiler: Provides continuous data source monitoring that builds a statistical model or profile, of the specified data source, and highlights top SQL, event, and session activity. This component is used to locate and diagnose problematic SQL code and event-based bottlenecks via its graphical interface, which is used to identify problem areas and drill down to individual, problematic statements. Additionally, Profiler enables the investigation of execution and wait time event details for individual stored routines. Profiling Details of the SQL Profiler have been expanded to show Session Details for Sybase, and SQL Server and the SQL that ran in the selected session for Sybase, SQL Server, and DB2.

SQL Tuner: Provides an easy and optimal way to discover efficient paths for queries that may not be performing as quickly or as efficiently as they could be. Tuner enables the optimization of poorly performing SQL code through the detection and modification of execution paths in data retrieval through hint injections. Users are supplied with a list of possible cases generated by Tuner, and can select and update a statement with the most efficient path to reduce load and improve efficiency, overall. The Analysis tab of the tuner provides a graphical diagram of an SQL query to show how the tables in the query would be joined to satisfy the query and also provides suggestions for the creation of indexes that may increase query performance.

SQL Editor: Simplifies SQL development by utilizing features that improve productivity and reduce errors. It provides a rich interface that offers code completion, real-time error caching, code formatting, and sophisticated object validation tools. In context with SQL Profiler and SQL Tuner, it provides an interface for viewing and editing SQL files and database packages, as accessed through Profiler and Tuner functionality.

NEW FEATURES

NEW FEATURES IN DB OPTIMIZER 2.5

Improvements to Profiling:

- New Procedures tab for SQL Server and Sybase platforms enables you to examine procedures and their effect on database efficiency.
- Top SQL tab now provides average execution time, Average Elapsed (sec) statistics for SQL Server, DB2, and Oracle platforms.
- Top Events tab now provides average wait time, Avg. Per Wait (sec) statistics for Oracle events.
- Improved SQL text collection by querying the database cache more often and more intelligently thereby reducing the occurrences of UNKNOWN statements.
- Max CPU and Max Engine count can now be specified for cases when the database platform does not report how many CPUs or engines are available and for when you want to see the performance gain or hit created by increasing or decreasing the number of CPUs or engines.
- New Profiling Repository available for Oracle platforms offers the ability to gather and store more profiling information that is then readily available for analysis from the Data Source Explorer.
- Performance optimization enables the Profiler to better support longer sessions and more heavily loaded servers.

Improvements to Tuning:

- New transformations or query rewrites available for Oracle data sources offer more tuning alternatives.
- Extended support for statements with bind variables on SQL Server and Sybase.
- Filter ratios, and table size and table join size optionally added to Visual SQL Tuning diagrams. Collecting this data may be time-consuming for large databases.

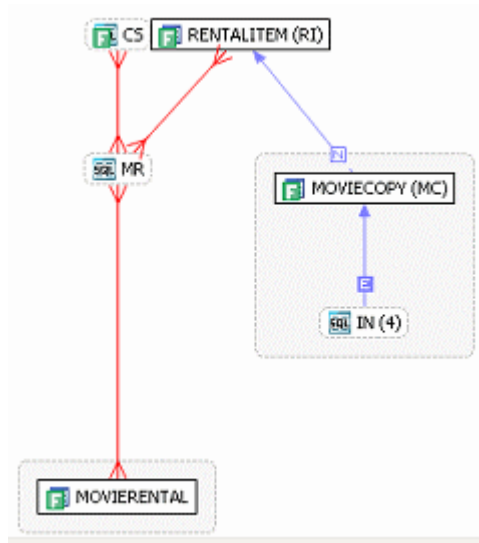
NEW FEATURES IN 2.5.1

Operating System Support

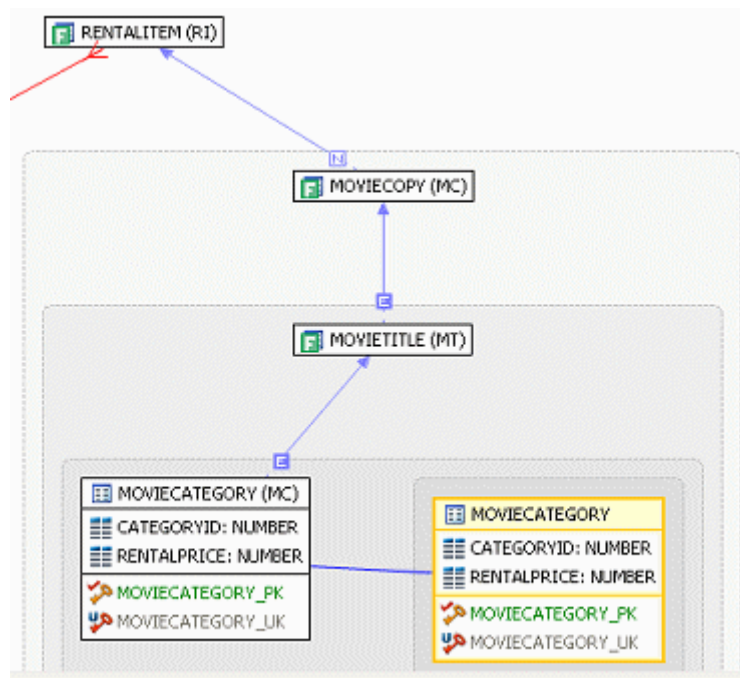
DB Optimizer platform support has been extended to include Windows XP x64, Windows Vista x64, Windows 7 x32, and Windows 7 x64.

Improvements to Tuning

- EXISTS, NOT EXISTS, IN, NOT IN annotation support in Visual SQL T.uning



- Visual SQL Tuning support for sub-queries as expandable nodes.



- Complete state persistence when saving tuning editor.

LIMITATIONS OF THE TRIAL VERSION

The following limitations apply to the trial version of DB Optimizer. These limitations are removed once the product has been purchased.

- Command line profiling and tuning are not available.
- There are a limited number of profiling sessions you can run.
- Profiling is limited in that you will not be able to see the captured SQL text of the data source profiled.
- Only the Ad Hoc SQL tab is available for specifying tuning sources. You cannot tune database objects, SQL files or the Oracle SGA.
- There are a limited number of statement extractions from the profiler to the tuner.
- In the tuning editor the following case actions are disabled: Clone Case, Edit Case, and Create Custom Case.

LIMITATIONS OF DB OPTIMIZER XE DEVELOPER

The following features are not supported on DB Optimizer XE Developer:

- Visual SQL Tuning.
- Hint analysis via SQL hint injection for Oracle datasources.
- Saving profile sessions to an Oracle data source repository.

ADDITIONAL PRODUCT INFORMATION

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](#)
- [Embarcadero Documentation Wiki](#)
- [Online Demos](#)
- [Technical Papers](#)
- [Discussion Forums](#)

INSTALLATION

This section will help you ensure your environment meets the technical requirements and helps you install DB Optimizer or DB Optimizer XE. It is comprised of the following topics:

- [Technical Requirements](#) on page 9
- [Installation Issues](#) on page 10
- [Installing DB Optimizer](#) on page 11
- [Installing DBOptimizer XE](#) on page 13

TECHNICAL REQUIREMENTS

Before installing Embarcadero® DB Optimizer™, verify that your environment meets the following requirements.

Hardware Requirements

The following are the minimum hardware requirements:

- Pentium 4-Level Processor
- 1024 MB of memory
- 500 MB of disk space
- 1024 x 768 screen resolution

Operating System Requirements

DB Optimizer supports the following operating systems:

- Microsoft Windows 7, 32 and 64 bit
- Microsoft Windows Server 2008, 32 and 64 bit
- Microsoft Windows Vista, 32 and 64 bit
- Microsoft Windows XP, 32 and 64 bit
- Microsoft Windows Server 2003
- Red Hat Linux 5.0, x86-32, GTK 2
- SuSe Linux Enterprise Server (x86) GTK+ 2.x

DBMS Support

DB Optimizer supports the following platforms:

- Generic JDBC
- IBM DB2 LUW 8.0 - 9.5
- Microsoft SQL Server 2000, 2005, and 2008
- Oracle 8i - 11g
- Sybase 12.5 - 15.0.1

Eclipse Support

DB Optimizer can be installed as a standalone application (RCP installation) or as an Eclipse plug-in (plug-in installation).

The Eclipse plug-in version of DB Optimizer requires Eclipse version 3.3 or higher and Sun Java Standard Edition 5.0 Update 11 or later in addition to regular system requirements. Before installing the plug-in version of DB Optimizer, ensure that Eclipse and Java are installed on your machine. You can download Eclipse from the following Web site: <http://www.eclipse.org/downloads>.

TOOLCLOUD REQUIREMENTS

Minimum requirements:

- Intel Pentium III 1.2 GHz processor or higher
- 1 GB of RAM
- 100 MB of hard disk space for the ToolCloud
- At least 8GB is recommended for the repository of initial and future product downloads
- High resolution monitor (1024 x 768), 16-bit display or greater
- Microsoft Windows XP Professional (x86-32, Win32), Microsoft Windows Server 2003, or Microsoft Windows Vista
- Internet Explorer 7 or later, or Mozilla Firefox
- Java Runtime Environment, version 1.5 (provided by the installer, if you do not already have it installed)
- The ToolCloud requires ports 8005 and 9001 for Tomcat and other internal processes.

INSTALLATION ISSUES

DB Optimizer should be installed in a different directory than previous versions of DB Optimizer. If DB Optimizer is installed in the same directory, the following error will result in at application start up: "Unable to read workbench state. Workbench UI layout will be reset."

If a previous version is already installed on your machine, either change the default Workspace or delete all files from the Workspace directory. The Workspace directory location appears whenever you start Optimizer, and can be redefined from this dialog box. If the same Workspace is indicated for the new version as was indicated previously, the following error will result: "Cannot restore workbench layout".

If DB/Optimizer is already running, you can also modify the Workspace directory by choosing **File > Switch Workspace > Other** from the Main Menu.

INSTALLING DB OPTIMIZER

To install DB Optimizer, run the installer and follow the prompts provided by the Installation Wizard. The notes below provide information on parts of the installation process that are not self-explanatory. When the installation is complete, it is advised that you review the ReadMe file to check if it contains any known issues or advisory notes that will affect your installation of the product.

LICENSING DB OPTIMIZER

DB Optimizer supports the following types of licenses:

- **Trial licenses:** A license for a 14-day, full-featured trial version of the product. The trial license must be registered before you can use the product. For more information, see "[Registering a Trial or Workstation License at Application Startup](#)" on page 11.
- **Workstation or Network Named licenses:** A license or set of licenses that is tied to a particular workstation. The product can only be used on that workstation. For more information, see "[Registering a Workstation License after Application Startup](#)" on page 12.
- **Concurrent Licenses:** If you are using ToolCloud and DB Optimizer XE you can also have concurrent licenses which are floating licenses that allow a specific number of users to concurrently use a license, without tying the license to a specific workstation serial number. For more information, see [Installing DBOptimizer XE](#) on page 13.

If you have problems registering, see "[Registering by Phone](#)" on page 12.

Registering a Trial or Workstation License at Application Startup

Shortly after downloading a trial version of an Embarcadero product, you should receive an e-mail from Embarcadero with a serial number, which must be registered as follows. These instructions also apply if you have purchased a product and received a serial number prior to running the application for the first time, or if the trial period has expired.

- 1 Start the application.

The **Embarcadero License Registration** dialog appears.

- 2 Copy the serial number from the e-mail, and then paste it into the **Serial Number** field.

- 3 Enter your Embarcadero Developer Network (EDN) account credentials in the **Login or Email** and **Password** fields.

If you have not previously created an EDN account, or have forgotten your password, click [I need to create ... or I've lost my password](#).

- 4 Click **Register**.

The activation file should be downloaded and installed automatically. If this does not happen, click the **Trouble Connecting? Try Web Registration** link and follow the prompts. If you still have problems, see "[Registering by Phone](#)" on page 12.

Registering a Workstation License after Application Startup

The following instructions assume that you have received a workstation license by e-mail and that you currently have a valid trial license. If you did not install a trial version or if the trial period has expired, follow the instructions in "[Registering a Trial or Workstation License at Application Startup](#)" on page 11 instead.

- 1 Click **Help > Launch License Manager**.
- 2 In the **License Manager** dialog, click **Serial > Add**.
- 3 Copy the serial number from the e-mail, paste it into the **Add Serial Number** dialog, and then click **OK**.
- 4 Expand **Unregistered Serial Numbers**, right-click the serial number just added, and then click **Register**.

A registration dialog appears.

NOTE: The **Registration Code** box shows a machine-specific identifier required with other registration methods.

- 5 Ensure that **Register using Online Registration** is selected.
- 6 Provide Developer Network credentials in the **Login or Email** and **Password** fields.
- 7 Click **Register**.

The activation file should download and install automatically. If this does not happen, click the **Trouble Connecting? Try Web Registration** link and follow the prompts. If you still have problems, see "[Registering by Phone](#)" on page 12.

Registering by Phone

If you have problems with either of the above procedures, you can register licenses by phone. You must provide the following information:

- Developer Network credentials
- The registration code displayed in the Embarcadero License Registration dialog that appears when you start an unlicensed application

- The product base license serial number
- The license serial numbers for any additional features you have purchased.

For North America, Latin America, and Asia Pacific: call (415) 834-3131 option 2, and then follow the prompts. The hours are Monday to Friday, 6:00 A.M. to 6:00 P.M. Pacific time.

For Europe, Africa, and the Middle East: Call +44 (0)1628-684 494. The hours are Monday to Friday, 9 A.M. to 5:30 P.M. U.K. time.

Shortly after phoning in, you will receive an e-mail containing an activation file. Then do the following:

- 1 Save the activation file to the desktop or to a scratch directory such as `c:\temp`.
- 2 From within the application, click **Help > Launch License Manager**.
- 3 Click **License > Import**.
- 4 Navigate to the directory where you saved the activation file, and then double-click the activation file.
- 5 Click the **Import** button to import the activation file, and then click **Finish**.

INSTALLING DBOPTIMIZER XE

This guide tells you how to install Embarcadero ToolCloud, including the Embarcadero License Center (ELC), and get them up and running for use with DB Optimizer XE. It assumes you are familiar with Embarcadero XE product descriptions.

For the latest update of this document, see http://docs.embarcadero.com/products/all_access/.

- [Eclipse Support](#) on page 10
- [STEP 1 - OBTAINING THE INSTALLER](#) on page 14
- [STEP 2 - INSTALLING THE TOOLCLOUD AND ELC](#) on page 14
- [STEP 3 - LOGGING INTO THE TOOLCLOUD ADMIN CONSOLE](#) on page 16
- [STEP 4 - SETTING UP LICENSES IN THE TOOLCLOUD](#) on page 17
- [STEP 5 - NAMED OR CONCURRENT LICENSES?](#) on page 17
- [STEP 6 - SETTING UP NAMED NETWORK USERS IN THE TOOLCLOUD](#) on page 18
- [STEP 7 - SETTING UP NAMED NETWORK USERS IN THE ELC](#) on page 19
- [STEP 8 - DOWNLOADING XE APPLICATIONS TO THE PRODUCT REPOSITORY](#) on page 19
- [STEP 9 - DISTRIBUTING THE TOOLBOX TO PRODUCT USERS](#) on page 20
- [STEP 10 - CONFIGURING PRODUCT CATALOG UPDATES](#) on page 22

- [STEP 11 - CHANGING THE TOOLCLOUD ADMINISTRATOR PASSWORD](#) on page 22

NOTE: For complete information on using the ToolCloud, see the ToolCloud Administration Guide, Embarcadero License Center Administration Guide, and ToolBox User Guide, available from the Start menu or your Embarcadero representative. For wiki FAQs, see http://docwiki.embarcadero.com/DBOptimizer/en/Main_Page.

STEP 1 - OBTAINING THE INSTALLER

In most situations, you will receive an email containing instructions including how to download the installer.

STEP 2 - INSTALLING THE TOOLCLOUD AND ELC

- 1 Quit all programs.
- 2 Launch the ToolCloud installer and click **Next**.
- 3 Read the terms of the license agreement, and if you agree select **I Agree** and click **Next**.
- 4 If you do not have Embarcadero License Center (ELC) installed on your machine, click **Next**. Otherwise, clear the box and click **Next**.

- 5 The ELC installer opens. Click **Next**.
 - a Select **I accept the terms of the License Agreement** and click **Next**.
 - b Accept the default installation folder and click **Next**.
 - c Accept the default **Typical** installation and click **Next**.
 - d If there is no recent Java VM already installed, install the Java VM specifically for ELC. Otherwise, select **Choose a Java VM** already installed on this system, and click **Search**. Select a pre-existing VM from the list and click **Next**. *Note:* The host name of the Java VM is the name used when creating .slip license files for Embarcadero products.
 - e Accept the default Program Group for the icons and click **Next**.
 - f Review the Summary and click **Install**.
 - g When the installer prompts for configuration, click **Next**.
 - h Accept the check box to enable logging in ELC and click **Next**.
 - i Select the check box to run the ELC as a Windows service and click **Next**.
 - j The installer displays the Server Host Name and Server IP Address, which are later used by the license hosting site to create configuration files to manage licenses. Write down this information for later use and click **Next**.
 - k Select the check box to start the ELC and click **Next**.
 - l You want to configure the Reporting Engine, so click **Next**.
 - m The final ELC installation screen displays the directory where the application was installed. Note this for future reference. Click **Done**. The ToolCloud installer returns.
- 6 Accept the default installation directory and click **Next**.
- 7 Provide a new valid listening port for network communication. Write down the port number for later use.

NOTE: Do not use 8005 or 9001. The ToolCloud automatically uses ports 8005 and 9001 for Tomcat and other internal processes.
- 8 The ToolCloud distributes products through a shared repository created during installation. The Installer grants read-only access rights to everyone in order to provide a shared network directory from which products can be distributed. Normally you should accept the default and click **Next**. If there are specific restrictions for creating share directories, you can choose an existing shared directory.
- 9 The installer is ready to install. Click **Next**.
- 10 The installer prompts you to enable automatic updates. **Do not enable updates at this time.** You can configure this option later. Click **Next**.
- 11 Click **Next** to install the ToolCloud. Click **Finish** to complete.

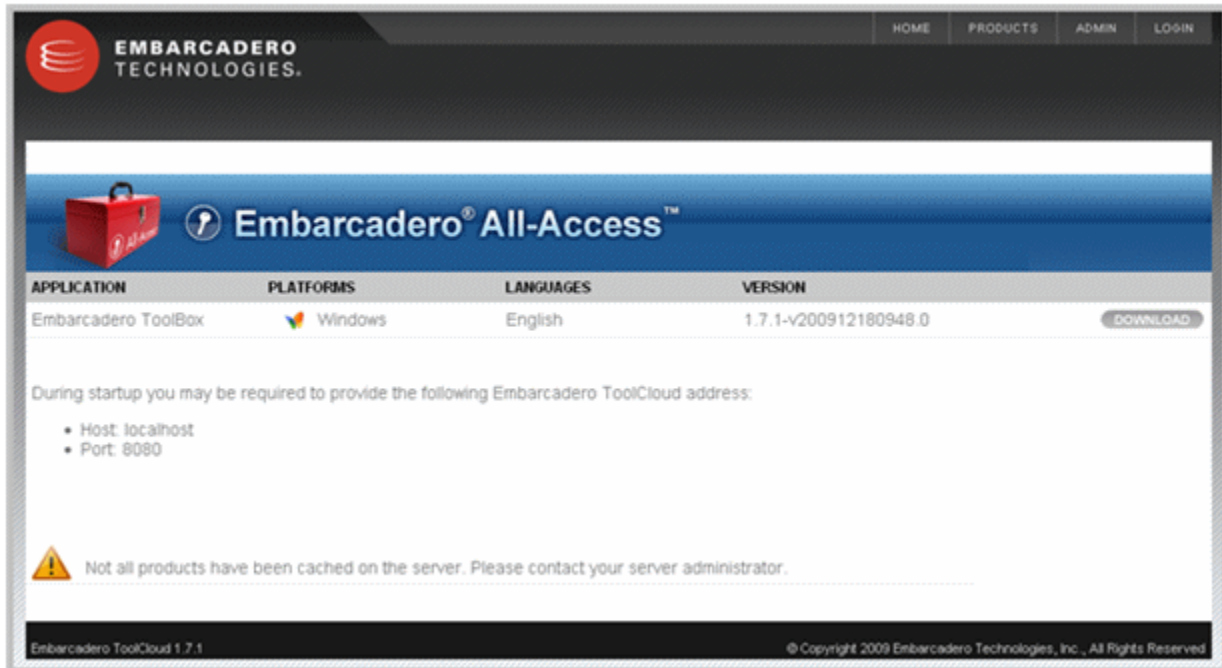
STEP 3 - LOGGING INTO THE TOOLCLOUD ADMIN CONSOLE

- 1 From the machine onto which you installed the ToolCloud, open a browser and go to the URL:

`http://localhost/aaserver`

If you gave it a port number other than 80 during installation, use:

`http://localhost:<port_number>/aaserver.`



- 2 At the top of the ToolCloud **HOME** page, click **ADMIN**.
- 3 Enter your login credentials. By default, the initial login is "admin" and the password is "admin".
- 4 Click **Login**. The **ADMIN** page opens.

STEP 4 - SETTING UP LICENSES IN THE TOOLCLOUD

- 1 In the **LICENSES** menu of the **ADMIN** page, click **Retrieve Licenses**.

Retrieve License

* Login Name

* Password

Certificate Number(s)

ELC Host Address

Leaving this field empty will retrieve all network licenses assigned to your account and associate them with this machine. If you have been assigned multiple license certificates and would like to enable only a subset on this server, then you may enter one or more comma-separated Certificate numbers, to retrieve just those licenses.

In case the server host name cannot be resolved to an IP address on your company network, enter either a fully qualified address (e.g. hostname.hostdomain.com) or an IP address (e.g. 123.123.12.21). This address will be used by clients to send licensing requests to the server.

- 2 Enter your login ID and password, as supplied to you by Embarcadero.
- 3 Enter the license certificate number, as supplied to you by Embarcadero.
- 4 Enter the ELC host address name or IP number, as you previously noted in Step 2.
- 5 Click **Retrieve**.

The license or licenses will be registered with the ToolCloud and the ELC.

Current Licenses

Default level Custom

License retrieval finished without any errors

License Title	Certificate	Server Name	File Location	Expire Date	
DBArtisan XE Professional with Analyst Pack	86971	SFWDELL01	named_86971.slp	03/31/2010	

STEP 5 - NAMED OR CONCURRENT LICENSES?

If you have named network licenses, proceed to Steps 6 and 7. You must set up users in both the ToolCloud and the ELC.

If you have concurrent network licenses, you do not need to set up specific users. When a user connects to the ToolCloud from the ToolBox, the ToolCloud will provide access to its concurrent licenses. You can skip to Step 8.

STEP 6 - SETTING UP NAMED NETWORK USERS IN THE TOOLCLOUD

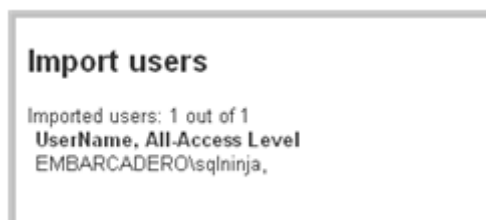
For named network licenses, you add a user or users to the ToolCloud.

- 1 On the **ADMIN** page, click **Import Users**.
- 2 Select **Create user**.



The screenshot shows the 'Create user' dialog box. At the top, it says 'Create user' with a default level of 'Custom'. There are two radio buttons: 'Create user' (selected) and 'Import the users from a comma separated text file (CSV)'. Under 'Create user', there are input fields for 'Domain' and 'Username', and a dropdown for 'All-Access Level' set to 'None'. A 'Create User' button is below these fields. Under the 'Import' option, there is a text input field and a 'Browse...' button. Below that, it states the expected format for a line in the file is 'Username[, AllAccessLevel]'. It also notes that the AllAccessLevel part is optional and lists possible options: BRONZE, SILVER, GOLD, PLATINUM, NONE, HIGHEST, SMALLEST. An 'Import' button is at the bottom.

- 3 Enter a network domain and user name.
- 4 If you are using All-Access, select an All-Access level to match the type of license you want to provide to the user. If you select the default, **None**, then the default access level specified in the **Current Licenses** page will apply.
- 5 Click **Create User**. The ToolCloud displays the user you have created.



The screenshot shows the 'Import users' confirmation dialog box. It states 'Imported users: 1 out of 1'. Below this, it shows a table with two columns: 'UserName' and 'All-Access Level'. The first row contains the values 'EMBARCADERO\sqlninja,' and an empty cell.

UserName	All-Access Level
EMBARCADERO\sqlninja,	

STEP 7 - SETTING UP NAMED NETWORK USERS IN THE ELC

For named network licenses, you need to add the exact same user or users to the ELC as you added to the ToolCloud in Step 6.

- 1 On the **ADMIN** page, click **Embarcadero License Center**.
- 2 Enter your login credentials. By default, the initial login is "admin" and the password is "admin".
- 3 Select **Named Users**.
- 4 Click **Add entry**.

The screenshot shows the 'Named Users' interface with a table of columns: User Name, Host Name, Product Title, Product Id, Sku Id, Version, and Offline usage. An 'Add new entry' dialog box is open in the foreground, containing the following fields and values:

- User name: sqlninja
- Host name: ANY
- Product title: DBArtisan XE Professional (dropdown menu)
- Product id: 14002
- Sku id: 64
- Version: 0
- Offline usage: 1 day (dropdown menu)

At the bottom of the dialog are 'Add' and 'Cancel' buttons. Below the dialog, in the main interface, are buttons for 'Add entry', 'Save to file', 'Undo changes', and 'Import from LDAP'.

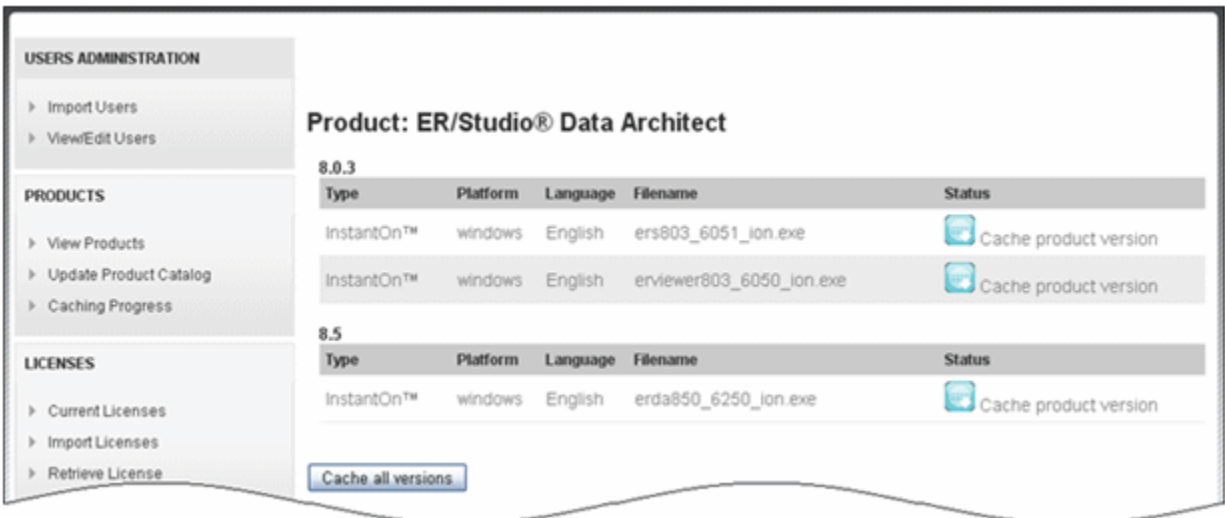
- 5 Enter a user name that matches one in Step 5.
- 6 Click **Add**.
- 7 Click **Save to file**.

STEP 8 - DOWNLOADING XE APPLICATIONS TO THE

PRODUCT REPOSITORY

In the ToolCloud, the **ADMIN** page lists products along with their status.

- 1 Click the name of the product you want. The **Product:...** page opens.
- 2 Click **Cache Product version** or **Cache all versions**.



STEP 9 - DISTRIBUTING THE TOOLBOX TO PRODUCT USERS

Product users access software from the ToolCloud using the ToolBox. Have them follow these instructions, and substitute the correct values for the ToolCloud host name and port:

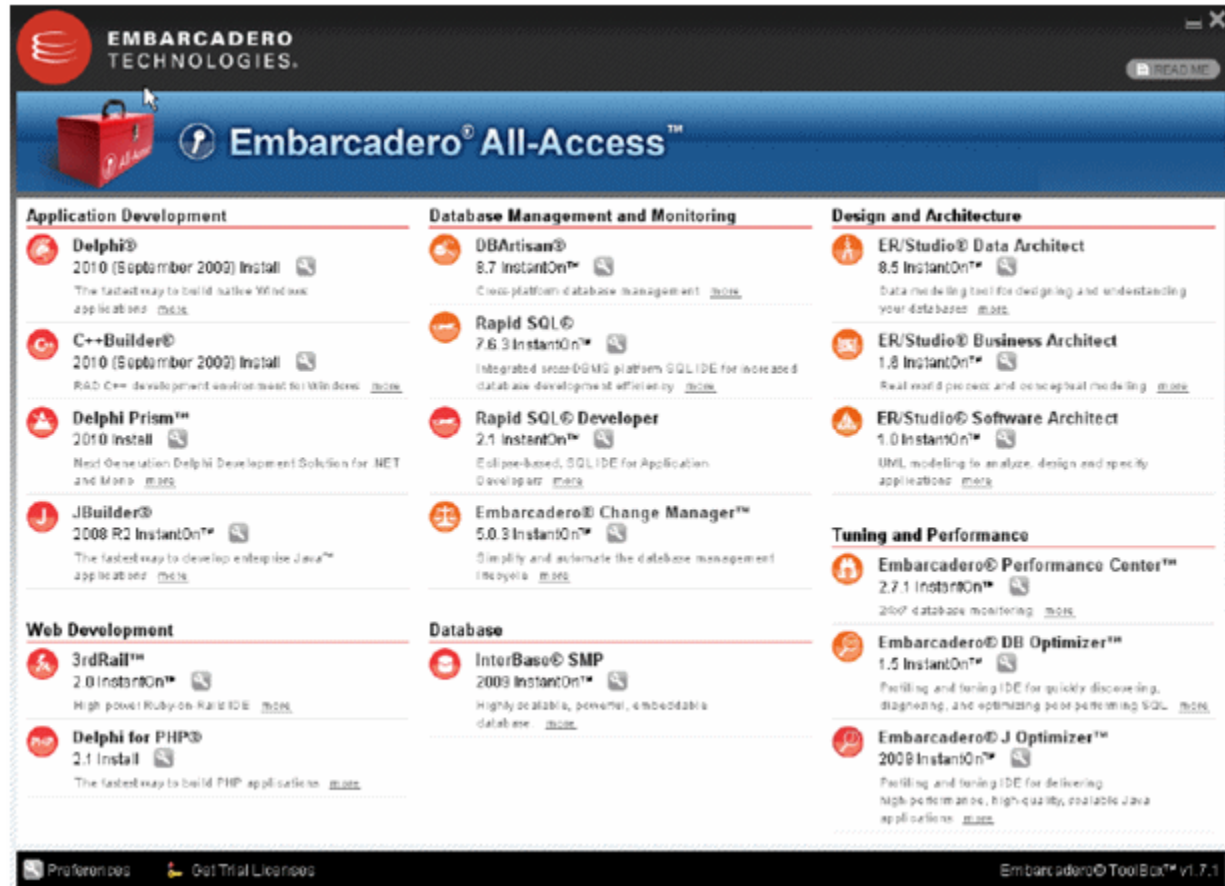
- 1 Connect to the ToolCloud at:

```
http://<serverhostname>:<portnumber>/aaserver
```
- 2 The ToolCloud **HOME** page lists the ToolBox. Click **Download**.
- 3 Save the ToolBox InstantOn to disk; the location depends on your browser setting.

- After the copy completes, click **Yes** to launch the ToolBox. Subsequent launches will originate from the local machine.

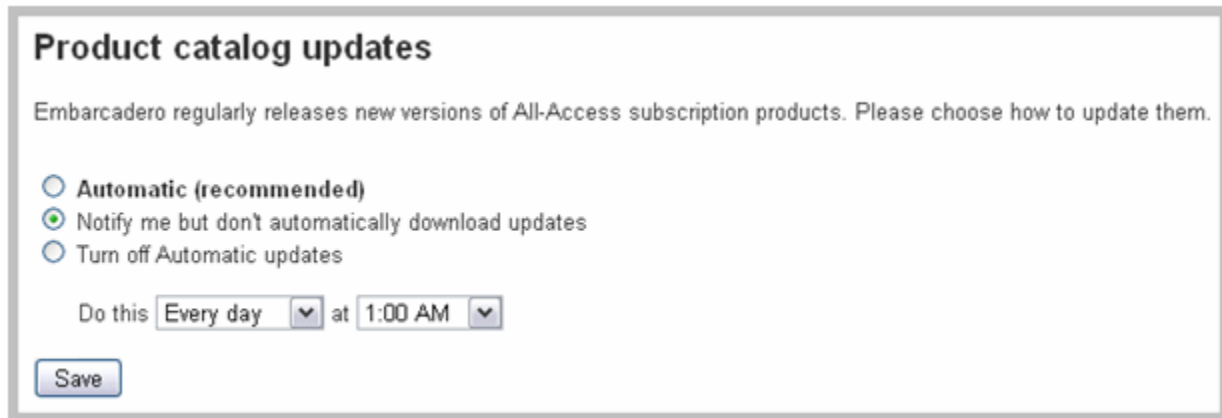
Products that are not available are grayed out in the ToolBox interface.

Click the wrench icon to configure the InstantOn launch and installation options.



STEP 10 - CONFIGURING PRODUCT CATALOG UPDATES

Set the ToolCloud behavior for obtaining product updates from Embarcadero.



- 1 In the **PRODUCTS** menu of the ToolCloud **ADMIN** page, click **Update Product Catalog**.
- 2 Select the option you want for getting updates.
- 3 If you select **Automatic** or **Notify...**, specify the frequency and time for notifications.
- 4 Click **Save**. If you selected **Automatic** or **Notify...**, the **NOTIFICATIONS** button will display the number of new messages in red text. Click **NOTIFICATIONS** to see the messages.

STEP 11 - CHANGING THE TOOLCLOUD ADMINISTRATOR PASSWORD

Most commonly, you will have internal policies about software passwords. Using your company guidelines, change the Administrator password.

- 1 At the top of the ToolCloud **HOME** page, click **CHANGE PASSWORD**.
- 2 Enter the new password.
- 3 Click **Save**.

NOTE: Do not forget the Administrator password, as the system cannot recover it for you. Write it down and store it in a secure location.

CLICK-AND-RUN WITH EMBARCADERO® INSTANTON™

In addition to the standard installation executables for ToolCloud products, Embarcadero provides InstantOn versions of each product. InstantOn installers provide click-and-run functionality for a product, to simplify deployment and enable side-by-side versioning. InstantOn executables are also ideal in locked-down desktop environments because they do not affect system files or system registry settings.

InstantOn files use a technique called application virtualization to run within their own space without sharing DLLs or system-wide settings that could conflict with other applications. Application data, including configuration and license information, is stored locally on the hard disk, separate from the executable itself. InstantOn executables run on the local machine and are launched from either the local hard drive or the network using the ToolBox.

In a ToolCloud-centered environment, also called network mode, InstantOn applications are provisioned and licensed by the ToolCloud (leveraging the Embarcadero License Center) and then accessed using the ToolBox. In this scenario, the ToolCloud can obtain software updates from Embarcadero servers and make them available to the ToolBox.

In a ToolBox-centered environment, also called workstation mode, the ToolBox downloads products directly from Embarcadero. Users license the products using the Embarcadero Developer Network (EDN). See the All-Access ToolBox User Guide for more information.

Using InstantOn executables on a shared ToolCloud avoids the need to keep multiple machines updated with new versions. By keeping the ToolCloud version up to date, users will automatically gain access to the latest versions of the products.

InstantOn executables launched from the ToolCloud benefit from application streaming, because the application will begin to open before the product has been fully downloaded. However, a user who needs frequent access to an application should run the file locally, which is typically faster than running it from the ToolCloud.

GETTING STARTED WITH DB OPTIMIZER

The following tutorial sessions enable you to get started using DB Optimizer. Their purpose is to provide you with the foundation needed to fully utilize the features and benefits of the products, and apply them to your own enterprise. This section contains information on how to register data sources from your enterprise, and how to profile, analyze, and tune SQL statements, sessions, and events for the purpose of optimizing the efficiency of your data sources, as well as identify and prevent system bottlenecks and other wait-related issues.

This guide is comprised of the following sections:

- [Start DB Optimizer](#) on page 25
- [Overview of the User Interface](#) on page 26
- [Working with Data Source Explorer](#) on page 27
- [Profiling a Data Source](#) on page 31
- [Tuning SQL Statements](#) on page 36
- [SQL Code Assist and Execution](#) on page 50

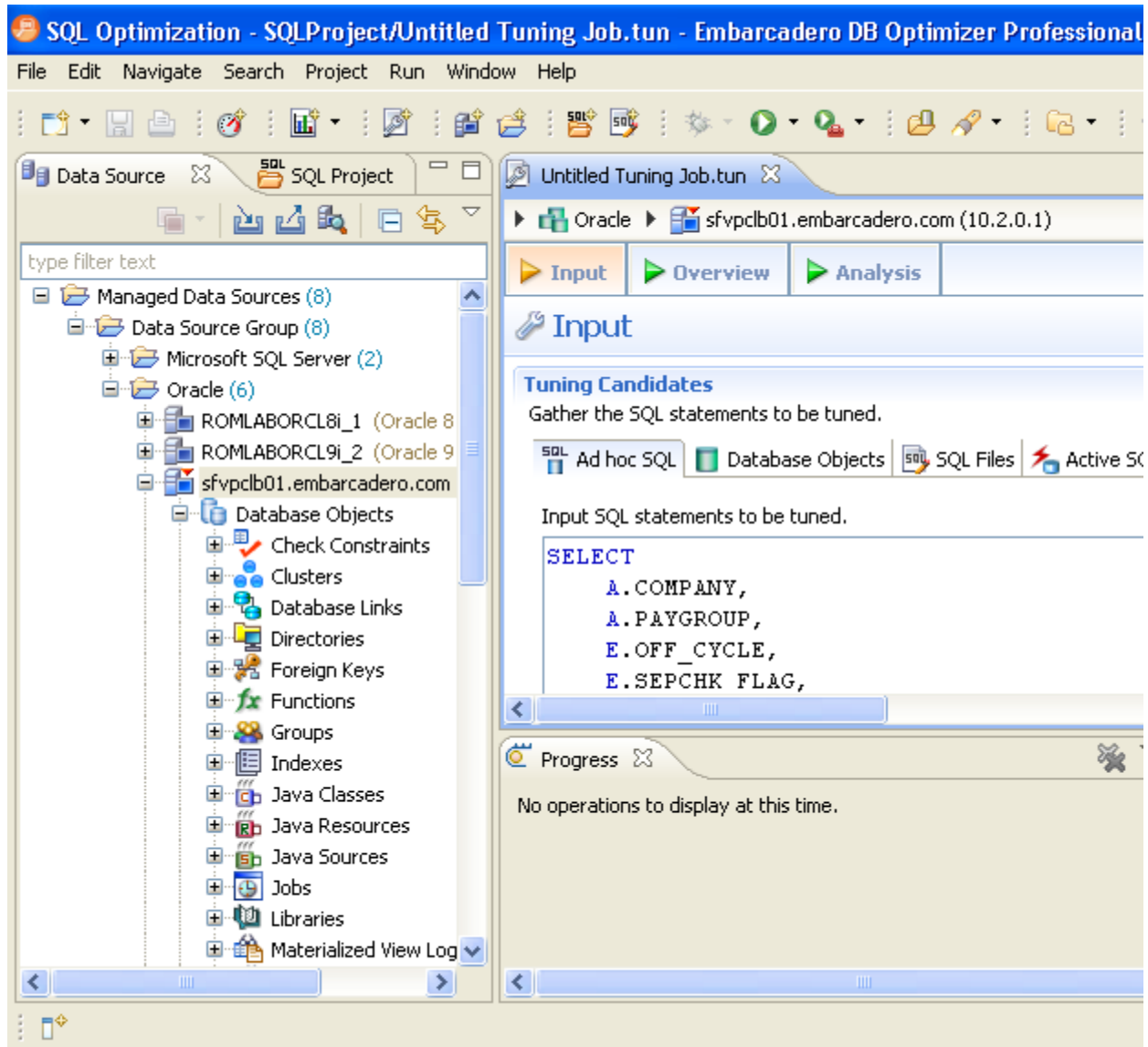
START DB OPTIMIZER

- From the Desktop, choose the Windows Start menu and then select **Programs > Embarcadero > Embarcadero DB Optimizer X.X > Embarcadero DB Optimizer X.X**.

NOTE: Once you have started the application, you can select **Help** from the Menu Bar to find additional resources that complement and build upon the features and tasks presented in this guide.

OVERVIEW OF THE USER INTERFACE

The DB Optimizer application environment is known as the **Workbench** that provides an interface through which you manage data sources and analyze and tune statements. The Workbench is composed of common interface elements that provide tools to accomplish these tasks. These objects provide a uniform system for working with tuning jobs, profile sessions, and data sources.



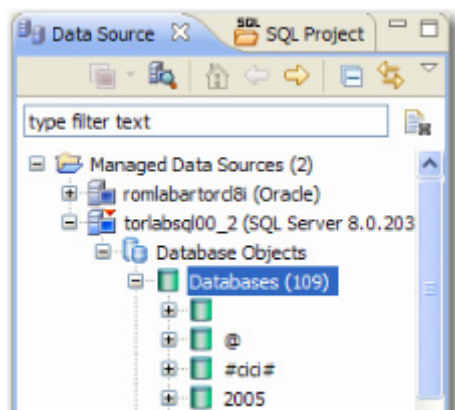
The Workbench provides an environment to build profiling sessions and tune query statements.

The **Workbench Space** is composed of Views, Editors, and the Menu Bar and Command Toolbar. Views and Editors enable you to perform DB Optimizer functions and work tasks, as well as manage resources.

- **Views** are used to navigate a hierarchy of information, open Editors or display the properties of various application elements. For example, the Data Source Explorer view provides a tree of all data sources in the environment and the comparison jobs associated with each. You can launch these jobs directly from Data Source Explorer, modify the connection properties of data sources, or create and edit configuration archives from this View.
- **Editors** are used to access DB Optimizer functionality. For example, the SQL Editor provides a means to view, edit, and save SQL code. Editors are differentiated from Views in that they operate on an individual level rather than a supportive one. The SQL Profiler and SQL Tuner are also considered to be Editors.
- The **Menu Bar** and **Command Toolbar** contain commands that execute various functional aspects of the application such as launching Views and Editors, navigation commands, and environment preference settings. The Toolbar contains icons that represent specific menu commands.

WORKING WITH DATA SOURCE EXPLORER

Data Source Explorer provides a tree view of all registered data sources. It breaks down the components of each data source and categorizes databases and corresponding database objects by object type and underlying SQL code. This feature provides you with a view of the contents of data sources in your enterprise in an easily navigated and cataloged interface.

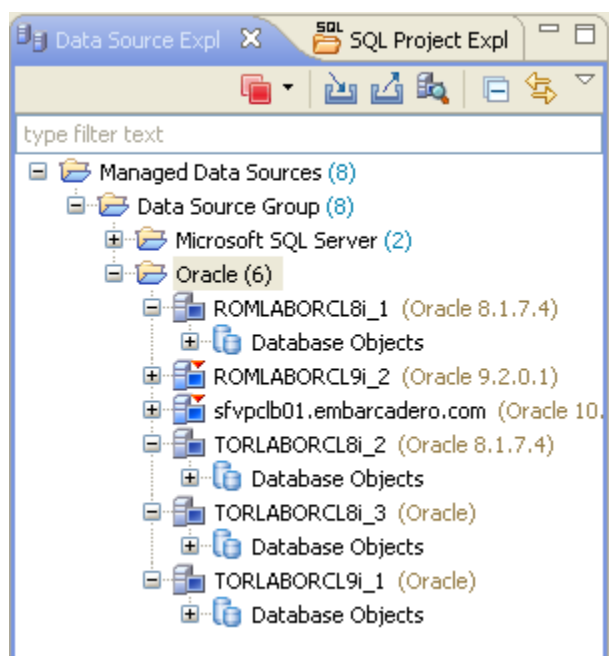


Data Source Explorer sorts databases and database objects by category within DB Optimizer.

If data sources are particularly large or complex, or you are only developing specific objects, you can apply database object filters on the view in Data Source Explorer.

ADDING DATA SOURCES

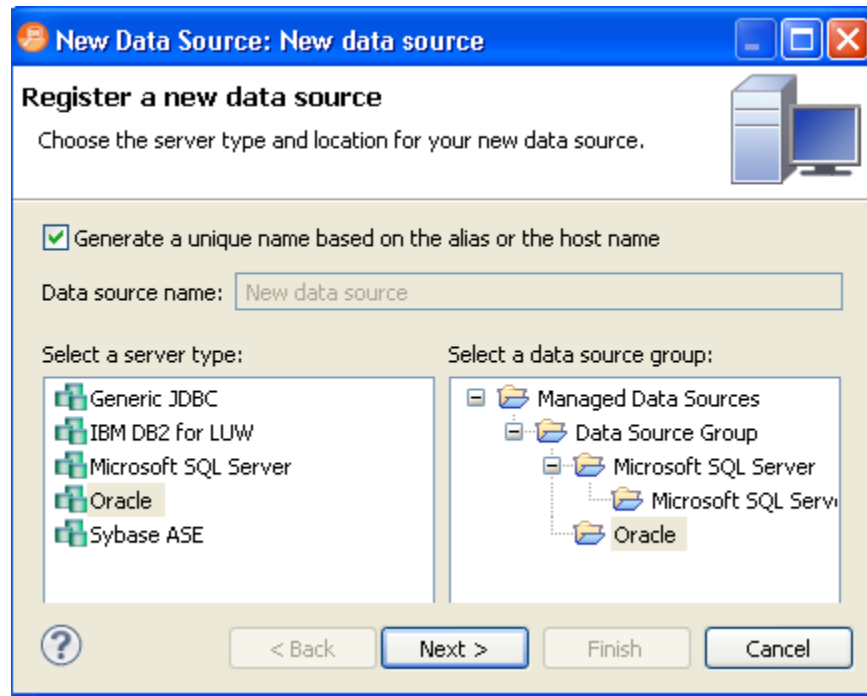
To profile and tune statements, you must register the data sources to be analyzed by providing connection information and other details to DB Optimizer from the Data Source Explorer. Once a data source has been registered, it remains stored in a catalog and does not need to be registered again each time you open DB Optimizer. Furthermore, it can be used in multiple jobs, archived, or otherwise “shared” with regards to the general functionality of the application.



The Data Source Explorer view provides an organizational tree of registered data sources and the parameters associated with them.

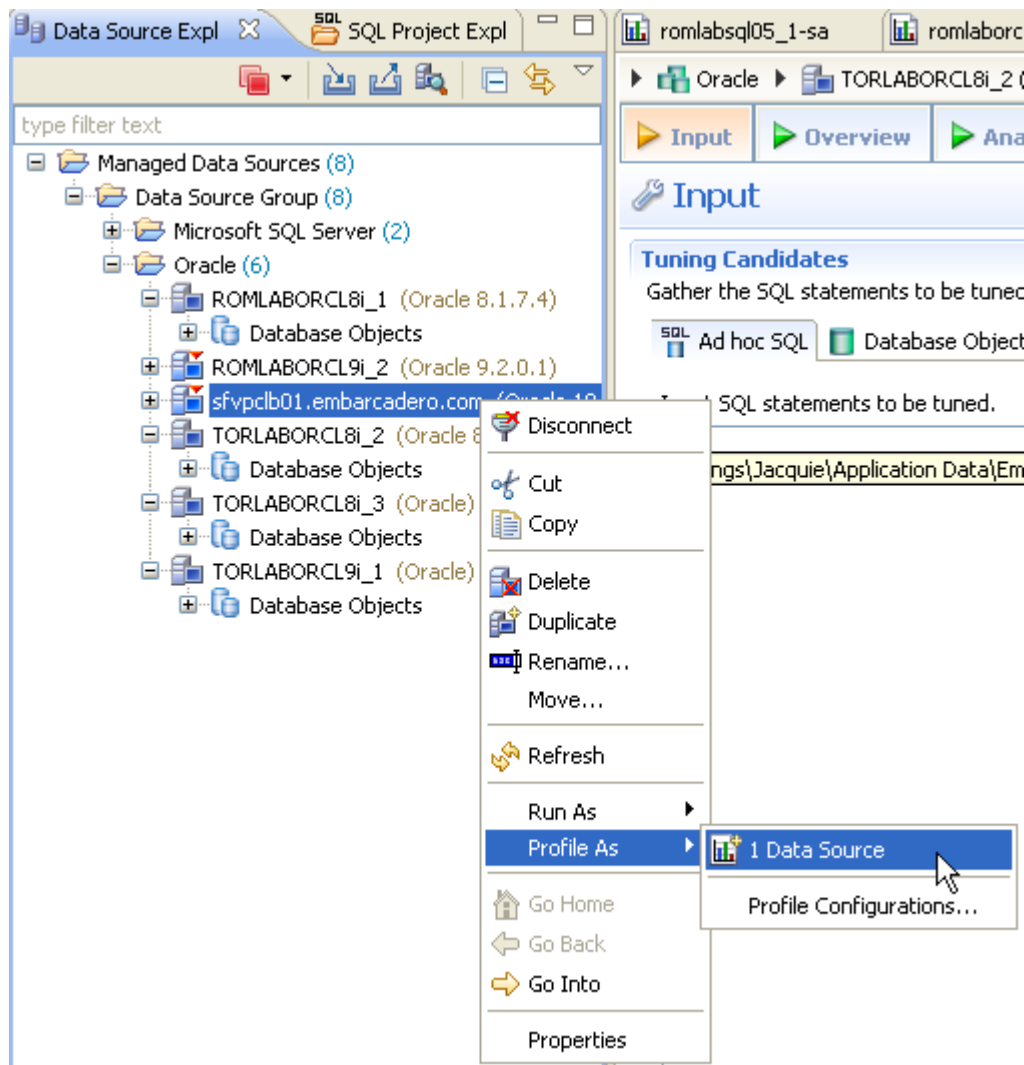
To add a data source:

Select **Data Source Explorer** and choose **File > New > Data Source** from the **Menu** bar. The **Data Source Wizard** appears. Follow the prompts to register the data source.



BROWSING DATA SOURCES

To analyze and tune statements on data sources within your enterprise, you must access the data source objects within the application. It is important to view databases and underlying code in an organized manner, especially when maintaining a large system. Data Source Explorer's tree structure can be used to view databases, tables, and other information about data sources. Expand the nodes of each registered data source to view details about each data source. Data Source Explorer is also used to launch profiling sessions, by enabling you to select the data source that you want to execute, and then launching a profiling session from your selection.



The Data Source Explorer tree provides a list of databases, and enables you to launch Optimizer features, such as SQL Profiler, via the context menu.

All objects listed in Data Source Explorer are organized, in descending order, by data source. Additionally, if you double-click on an object, SQL Editor will open and display the underlying SQL code.

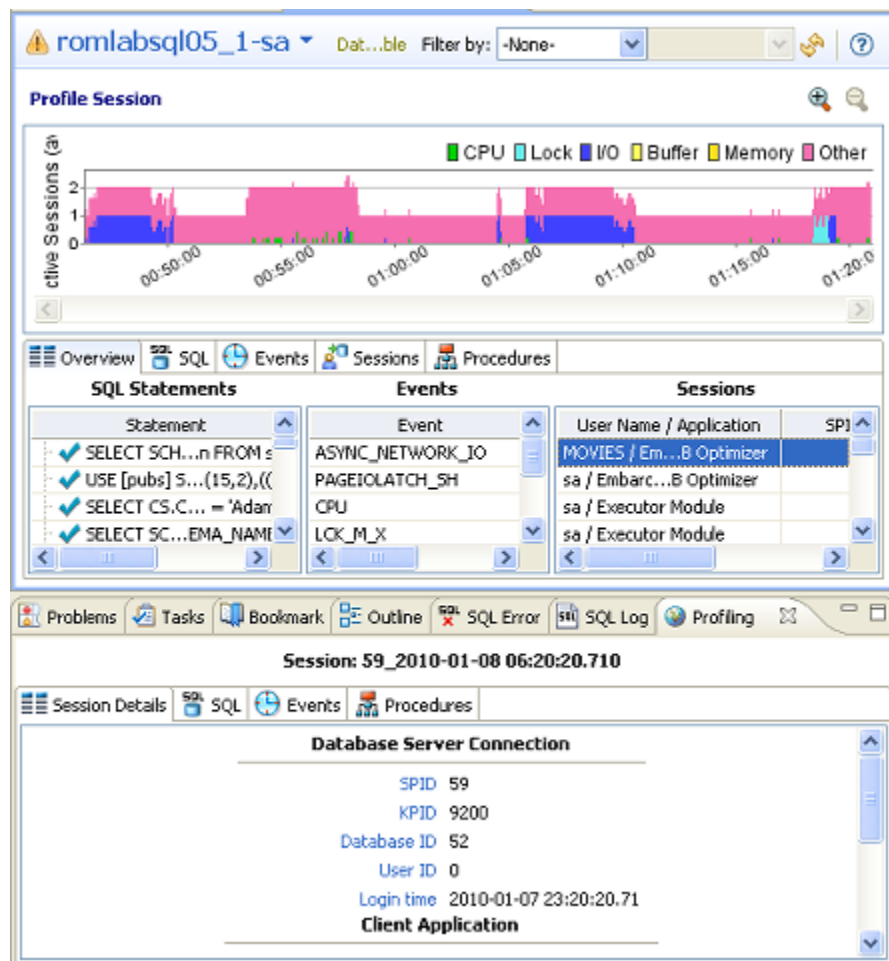
To browse a data source:

Expand the node of the data source that you want to examine. Double-click an object to view the code in SQL Editor.

PROFILING A DATA SOURCE

NOTE: The SQL Profiler is not available in the “Basic” version of DB Optimizer.

SQL Profiler samples a data source and builds a statistical model of the load on the database. Profiling is used to locate and diagnose problematic SQL code and event-based bottlenecks. Profiler also enables you to investigate execution and wait time event details for individual stored routines. Results are presented in the profiling editor, where you can identify problem areas and view individual SQL statements, as needed.



SQL Profiler analyzes and provides a statistical model of database load that identifies problematic code and event-based bottlenecks.

SQL Profiler is composed of the following diagnostic parts:

- The **Load Graph** provides a display of the overall load on the system. The bars represent individual aspects of the data source.
- The **Top Activity** section displays where load originates. Specifically, Top Activity displays the top SQL statements, the top events the database spends time in, and the top activity sessions that may be causing issues in terms of query time and response.
- The **Profiling Details View** displays detailed information for any item selected from Top Activity. For example, examining a SQL statement from Top Activity displays the identification parameters and the execution statistics of that specific statement selection.

STARTING A PROFILING SESSION

In order to access SQL Profiler, you need to run a profiling session on a data source registered in Data Source Explorer. Once a profiling session launches, it runs until you stop it or it has run for its specified length of time. You can stop a session by clicking the **Stop** icon on the upper left-hand side of the Profiler Editor.

To run a profiling session:

In Data Source Explorer, right-click on a data source and select **Profile As ...> Data Source**. The profiling session begins.

When the profiling session is complete, the first two sections (Load Chart and Top Activity) will be populated with information about the database load. You can then begin analyzing the data and identifying problem areas.

ANALYZING SESSION DATA

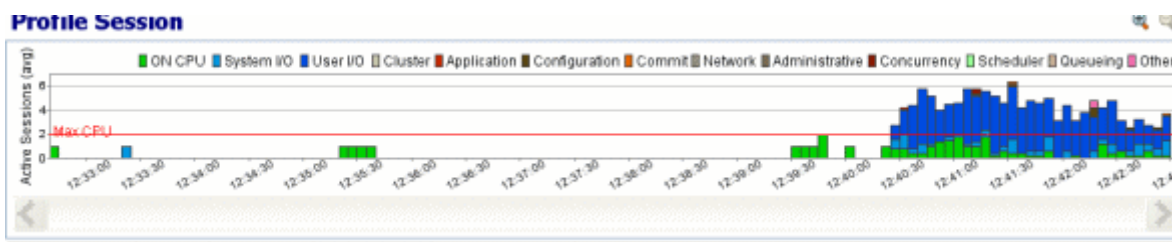
Profiler is composed of three essential diagnostic views that provide information about load on a particular database in the system. These views enable you to identify bottlenecks and view details about specific queries that execute and wait over the course of a profiling session.

The following describe the three components of SQL Profiler, in descending order of granularity:

- [Load Chart](#) on page 33
- [Top Activity](#) on page 33
- [Profiling Details](#) on page 34

LOAD CHART

The Load Chart provides an overview of the load on the system, and is designed as a high level entry point to reading session results. The colored bars represent individual aspects of the database, and the graph can be used to discover bottlenecks.



The Load Chart displays the overall load of the database that you analyzed with Profiler.

Time is displayed on the X axis, and the Y axis shows the average number of sessions waiting or executing. Each support platform type has a specific set of wait event times. For example, Sybase platforms will display CPU, Lock, Memory, I/O, Network, and Other. Use the chart legend to understand the graph. It displays a color and code scheme for executing and waiting session categories in the upper right-hand corner of the chart.

TOP ACTIVITY

Below the Load Graph, the Top Activity section displays where the load originates and outlines the top SQL statements, top events, and the top activity sessions on the database. It is composed of a series of tabs that provide detailed statistics on individual SQL statements and sessions that are waiting or executing over the length of the profiling session.

Overview SQL Events Sessions Object I/O								
U...e	Program	SID	S...#	Act...%)	Machine	Ses...pe	Cli...nfo	
SYSTEM		259	1865	91.02	rolcgrigore01	USER	10...52	
SYSTEM		285	2193	82.04	rolcgrigore01	USER	10...52	
SYSTEM		251	1809	57.49	rolcgrigore01	USER	10...52	
SYSTEM		286	566		rolcgrigore01	USER	10...52	

The Top Activity section displays a more detailed view of the Load Graph. It identifies top statements, events, and sessions that are waiting or executing over the length of the profiling session.

- The **Overview** tab provides summary information about SQL statements and events and their activity levels, and sessions, their system process IDs and their activity level. You can reorder the rows in any of the three sections of this tab. For example, clicking the Event column in the Events section changes the alphabetical order to ascending or descending.
- The **SQL** tab provides information about SQL statements and procedures. This includes all INSERT, SELECT, DELETE, and UPDATE statements that are executing or waiting to execute over the length of the profiling session.
- The **Events** tab displays information about wait events, and should be used to tune at the application or database configuration level. For example, if the top events are locks, then application logic needs to be examined. If the top events are related to database configuration, then the database setup should be investigated.

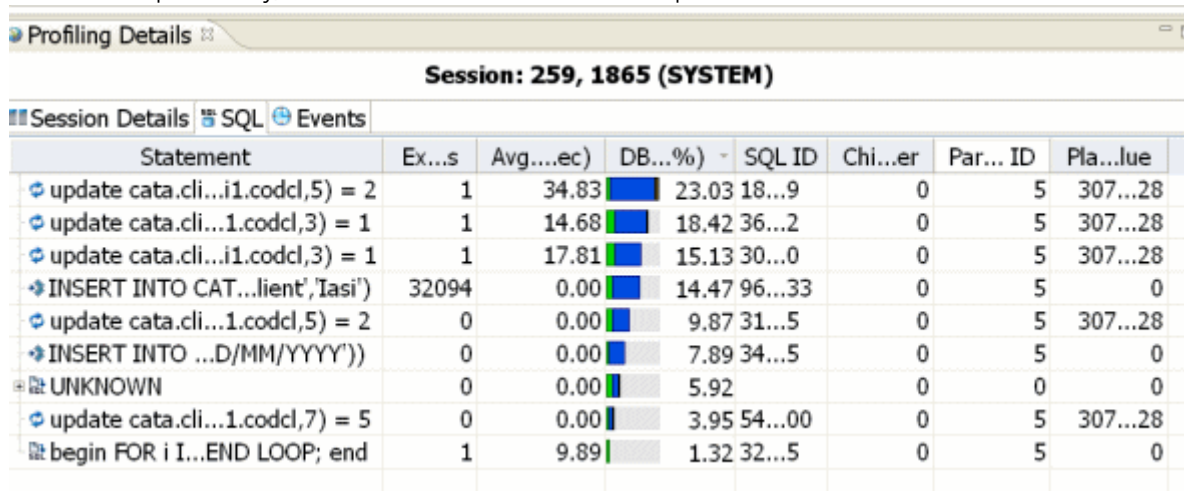
- The **Sessions** tab displays information about sessions, and can be used to discover sessions that are very active or bottlenecked.
- The **Object I/O** tab provides information about I/O when profiling an Oracle data source.
- The **Procedures** tab provides information about the procedures running in the session when profiling a SQL Server or Sybase data source.

PROFILING DETAILS

When you select any item from Top Activity, details are displayed on the Profiling Details view.

TIP: You may have to select **Windows > Show View > Profiling Details** to display the Profiling Details view.

The tabs that compose the Profiling Details view are dependent on the nature of the object selected in Top Activity, in order to reflect that item's specific information.



Statement	Ex...s	Avg....ec)	DB...%)	SQL ID	Chi...er	Par... ID	Pla...lue
update cata.cli...i1.codcl,5) = 2	1	34.83	23.03	18...9	0	5	307...28
update cata.cli...1.codcl,3) = 1	1	14.68	18.42	36...2	0	5	307...28
update cata.cli...i1.codcl,3) = 1	1	17.81	15.13	30...0	0	5	307...28
INSERT INTO CAT...lient',Iasi')	32094	0.00	14.47	96...33	0	5	0
update cata.cli...1.codcl,5) = 2	0	0.00	9.87	31...5	0	5	307...28
INSERT INTO ...D/MM/YYYY'))	0	0.00	7.89	34...5	0	5	0
UNKNOWN	0	0.00	5.92		0	0	0
update cata.cli...1.codcl,7) = 5	0	0.00	3.95	54...00	0	5	307...28
begin FOR i I...END LOOP; end	1	9.89	1.32	32...5	0	5	0

Profiling Details displays detailed parameter information on a statement, event, or session that was selected in Top Activity. The data displayed also varies by database platform.

Depending on the data source platform, the tabs that appear in the view will be different, in order to accommodate the parameter specifics of the statement you have selected. Depending on the top activity selected and the profiled platform types, some Profiling Details tabs may not be available. Click the tabs to view the profiling details for the selected object in the Top Activity section.

SAVING A PROFILING SESSION

A profiling session can be saved to a file with a .oar suffix that contains the name of the data source. When profiling Oracle datasources you can alternatively save the session data to the Profiling Repository. This enables you to open the file at a later time for analysis.

To save a profiling session as a .oar file:

Select the profiling session and then choose **File > Save As**. Specify the project location you want to save the file in and modify the file name, as needed. Click **OK**. The project is added to SQL Project Explorer.

Once you have saved a profiling session, it appears in the SQL Project Explorer under the name you saved it as. It can be opened again by double-clicking the project name.

To save a profiling session in the Profiling Repository:

You can also have the session saved automatically as a Profiling Repository file in an Oracle datasource. This allows you to profile your data source for a longer period of time, for days or even weeks.

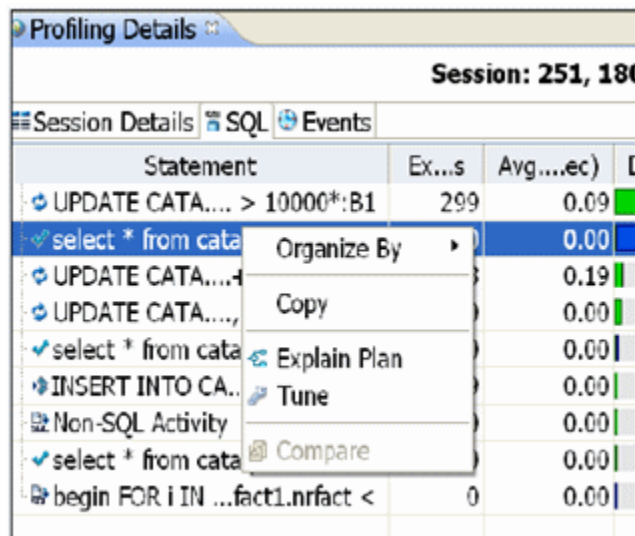
Right-click the datasource you want to profile and select **Profile As > Profile Configurations**.

In the **Profile Configurations** dialog, select **Save to data source** and then select the data source where you want to save the profile sessions.

The saved session file, named using the date and time when the profiling session was stopped, can be viewed from the Data Source Explorer and thus available to share with other DB Optimizer users.

IMPORTING STATEMENTS TO SQL TUNER

SQL Profiler enables you to submit one or more statements into SQL Tuner. This enables you to take advantage of Tuner's hint-based and transformation-based suggestions if you want to tune a problem statement that you detected over the course of a profiling session.



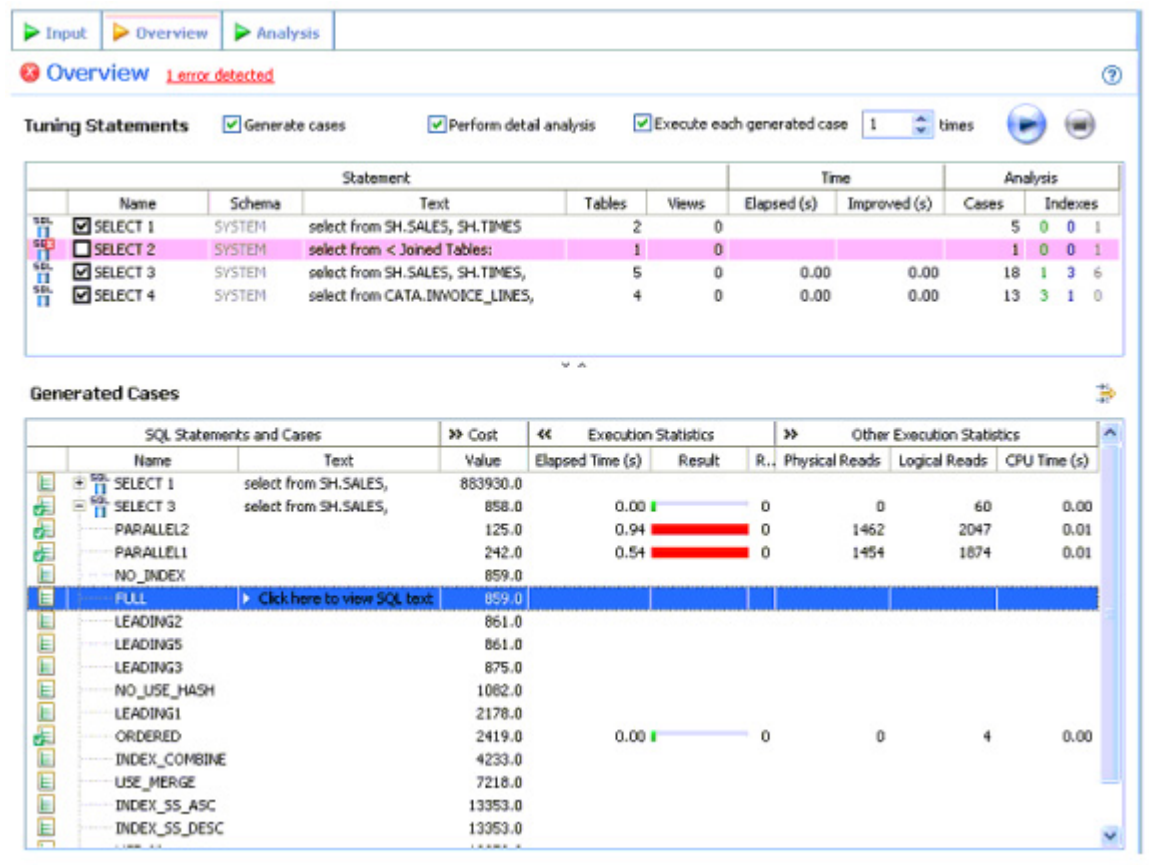
Context menu commands in SQL Profiler enable you to import statements to a tuning job directly from the Profiler interface.

To Import a statement from Profiler into Tuner:

Select one or more statements in Profiler, right-click and select **Tune** from the context menu. SQL Tuner opens and contains the selected statements in a new tuning job. You can now proceed to tune the problematic statements.

TUNING SQL STATEMENTS

SQL Tuner provides an easy and optimal way to discover efficient paths for queries that may not be performing as quickly or as efficiently as they could be. Tuner enables the optimization of poorly-performing SQL code through the detection and modification of execution paths used in data retrieval. This is primarily performed through hint injection, and index and statistics analysis.



Tuner analyzes specified SQL statements and then supplies execution path directives. This enables you to select alternate paths for queries, thus optimizing system performance based on analysis.

For example, if tuning is selecting from two tables (A and B), it will enable the joining of A to B, or B to A as well as the join form. Additionally, different joining methods such as nested loops or hash joins can be used and will be tested, as appropriate. Tuning will select alternate paths, and enable you to change the original path to one of the alternates. Execution paths slower than the original are eliminated, which enables you to select the quickest of the returned selections and improve query times, overall.

This enables the DBA to correctly optimize queries in the cases where the native optimizer failed.

CREATING A NEW TUNING JOB

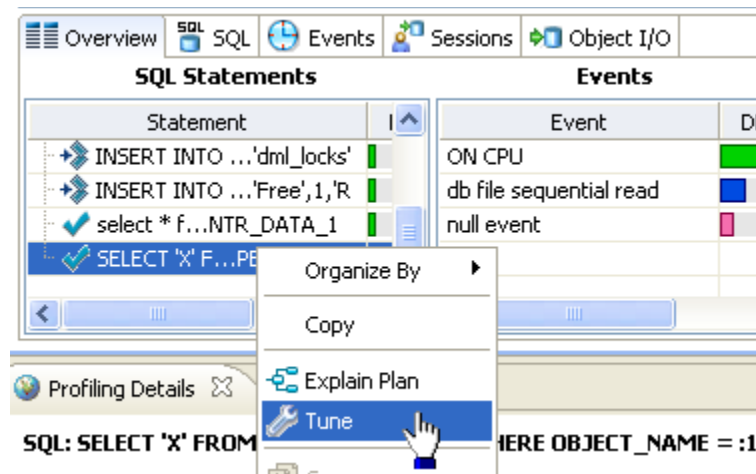
New tuning jobs are created from scratch where you can specify the statements to be tuned from a variety of sources, or statements can be directly imported from existing profiling sessions on data sources currently registered in the environment.



Tuning jobs are defined in SQL Tuner by specifying the data source and corresponding statements to be tuned and then executing the tuning job process.

To create a new tuning job:

To tune a specific SQL statement, in the Profile Session right-click the statement and select **Tune** as follows.



Once you have defined the input of the tuning job, you can save the file with a .tun suffix via the Save As command. The job is added to the SQL Project Explorer and can be re-opened and re-run at any time once it is saved to the system.

ADDING SQL STATEMENTS TO A TUNING JOB

Once you have created a name for the tuning job and indicated its source, you can add SQL statements that you want the job to tune. Statements are added to a job via the Ad hoc SQL box. All standard DML statements (SELECT, INSERT, DELETE, UPDATE) are viable for the tuning procedure.



Statements are added to a tuning job via the Input tab.

There are three or four different ways to add SQL statements to a job, as reflected by the tabs in the Tuning Candidates box:

Use the **Input** tab to specify which SQL statements to tune.

- **Ad hoc SQL:** Copy/paste SQL statements to the Ad hoc SQL tab or write queries by hand.
- **Database Objects:** Drag and drop database objects from the Data Source Explorer to the Database Objects tab.
- **SQL Files:** Browse the workspace or file system and select SQL files.
- **Active SQL in SGA:** For the Oracle platform only, you can also scan the System Global Area (SGA) for statements to tune.

To add an ad hoc statement, select the **Ad Hoc SQL** tab and manually type an SQL statement in the window or copy/paste the statement from another source.

To add a database object, select the **Database Objects** tab and click **Add**. The **Data Source Objects Selection** dialog appears. Type an object name prefix or pattern in the field provided and choose a statement from the window as it populates to match what you typed.

To add a saved SQL file, select the **SQL Files** tab and click **Workspace** or **File System**, depending on where the file you want to add is stored. Select a file from the dialog that appears and it will be added to the job.

To add Active SQL in SGA: Select the **Active SQL in SGA** tab and click **Scan**. Specify any filters as required and then click **Next**. From the list of SQL statements retrieved from the SGA, select those you want to optimize, and then click **Finish**. The selected statements are copied to the text area of the **Active SQL in SGA** tab.

RUNNING A TUNING JOB

After you add SQL statements to the job, click the **Overview** tab. Once you choose your tuning options and click the **Run Job** icon, the DML is parsed from the statements and added to the **Generated Cases** area. The Generated Cases are alternative execution paths or explain paths that could be better or worse than the default path the database uses. When these cases are executed you can use the execution statistics to determine which case would optimize performance.

Each extracted statement is listed by Name and Text. Additionally, each statement has a Cost, Elapsed Time, and Other Execution Statistics value that provide information on how effectively each case executes on the specified data source. These parameters let you compare the efficiency of the original statements to the cases generated by the tuning process when it is executed. You can click the Text field of a generated case to view the SQL source of the statement.

The screenshot shows the DB Optimizer interface with the Overview tab selected. The interface includes a top navigation bar with 'Input', 'Overview', and 'Analysis' tabs. Below the navigation bar, there's a 'Tuning Source Status' section with checkboxes for 'Generate Cases', 'Perform detail analysis', and 'Execute each generated case'. A 'Run/Cancel Job Controls' button is located to the right. The main area displays a table of statements with columns for Name, Schema, Text, Tables, Views, Elapsed (s), Improved (s), Cases, and Index. Below this is the 'Generated Cases' section, which includes a table of SQL Statements and Cases with columns for Name, Text, Cost, Elapsed Time (s), Physical Reads, and Other. Annotations with arrows point to various UI elements: 'Tuning Status Indicator' points to the status icon in the top left; 'Enable Execution Check Box' points to the 'Execute each generated case' checkbox; 'Increase/Decrease Pane Size Control' points to the vertical resize handle between the statements and generated cases tables; 'Column Set Expand/Collapse Control' points to the expand/collapse icons in the 'Generated Cases' table header; 'Run/Cancel Job Controls' points to the blue play button; 'Generated Case Expand/Collapse Control' points to the expand/collapse icons in the 'Generated Cases' table; 'Filter Control' points to the filter icon in the 'Generated Cases' table; 'Transformation Case' points to the 'USE_HASH' case; 'Hint-Based Cases' points to the 'LEADING1' case; and 'Extracted SQL Statements' points to the 'SELECT 1' statement.

Statement						Time		Cases	
Name	Schema	Text	Tables	Views	Elapsed (s)	Improved (s)	Cases	Index	
SELECT 2	SYSTEM	select from BROKER,			6.32	6.32	0		
SELECT 1	SYSTEM	select from			0.00	0.00	10		

SQL Statements and Cases			Cost	Execution Statistics	Other
Name	Text	Value	Elapsed Time (s)	Physical Reads	Logical Reads
SELECT 2	select from BROKER, CLIENT_TRANSACTION,	34014.0	6.32	2	
[Missing a ...sformation]		274.0	0.03	0	
SELECT 1	select from client_transaction, client,	4.0	0.00	0	
USE_HASH		14.0	0.00	0	
ORDERED		8.0	0.00	0	
NO_USE_NL		16.0	0.00	0	
LEADING4		8.0	0.00	0	
LEADING3		10.0	0.01	0	
LEADING2		7.0	0.01	0	
LEADING1		4.0	0.00	0	
INDEX_FFS		9.0	0.00	0	

The Generated Cases tab enables you to measure the various load costs of the original tuning statements and generated cases for each, which suggest alternative query paths to optimizing your data source.

The Tuning Status Indicator provides the status of each statement or case, and indicates if they are ready for execution. In some cases, SQL code may need to be corrected or bind variables may need to be set prior to executing statements.

Use the check boxes to select which statements and cases you want to run and then click the Run icon in the upper right-hand corner of the screen. The **Execute each generated case** field enables you to execute each selected statement or case.

Once you have executed a tuning job, the **Generated Cases** tab will reflect SQL Tuner's analysis of the specified statements. Once these have been analyzed, you can proceed to modifying the Tuner results and applying specified cases on the data source to optimize its performance.

To execute a tuning job:

- 1 Once you have a SQL statement that is a tuning candidate, navigate to the **Overview** tab.
- 2 In the **Tuning Statements** area, select the checkbox next to the Statement name that you want to analyze and then
- 3 *To analyze the SQL statements, click **Generate cases**.*

*To perform the analysis that populates the Analysis tab now, click **Perform detail analysis**. Otherwise, this analysis is performed when you click the Analysis tab.*

*To have the system generate execution statistics, click **Execute each generated case** and then select the number of times the system should execute each generated case. Multiple executions can verify that the case results are not skewed by caching. For example, the first time a query is run, data might be read off of disk, which is slow, and the second time the data might be in cache and run faster. Thus, one case might seem faster than another but it could be just benefiting from the effects of caching. Generally, you only need to execute the cases once, but it may be beneficial to execute the cases multiple times to see if the response times and statistics stay the same.*

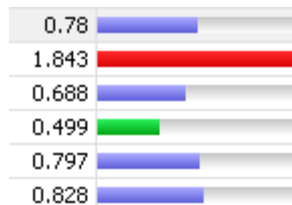
- 4 Then click the **Run Job** icon at the top right-hand side of the window. The tuning job runs, analyzing each statement and case, and providing values in the appropriate columns.

ANALYZING TUNER RESULTS ON THE OVERVIEW TAB

When you have executed a tuning job, the **Generated Cases** area of the **Overview** tab reflects Tuner's analysis of the specified statements and cases. The **Generated cases** create alternative execution or explain paths that could be more or less efficient than the default path the databases uses. Executing these cases provides the statistics necessary to optimize performance.

Once a tuning job has executed, use the **Cost** and **Execution Statistics** value columns to determine the fastest execution path for each statement. The **Cost** column shows the performance cost of an execution path as determined by the database. The **Execution Statistics** are the actual results of running the SQL statement using the generated case. This is where DB Optimizer can help you find where the database default path is actually not the optimal path. The **Elapsed Time(s)** and **Results** columns can more accurately show the most efficient execution path.

In the **Cost and Execution Statistics** columns, the values of the original statement are considered to be the baseline values. A **Cost** column can be expanded to provide a graphical representation of the values for statements and cases. Similarly, the **Execution Statistics** column can also be expanded to display a graphical representation of values as well. The bar length and colors are intended as an aid in comparing values, particularly among cases.



Case query times based on the original statement can be represented as colored bars on the Generated Cases area of the Overview tab to help you determine the fastest execution path for the given selections.

The baseline value of the original statement spans half the width of the column, in terms of bar length. For cases of the original statement, if one or more cases show a degradation value, the largest value will span the width of the column. Bar lengths for all other cases will then be displayed in comparison length to the highest degradation value.

The cost and execution results are color-coded as follows:

- **Light blue:** These cases are within the degradation and improvement threshold. Applying these changes may marginally improve or degrade the efficiency of the SQL statement.
- **Green:** These cases have values less than the improvement threshold. There is a high probability that changing the SQL statement with this alternative execution path will improve efficiency.
- **Red:** These cases have values higher than the improvement threshold. Implementing these changes will degrade the efficiency of the SQL statement.

To determine the best cases for statement execution path time:

Once the tuning job has executed, view the **Generated Cases** area of the **Overview** tab and determine the best possible case in terms of the **Execution Statistics** column values. This will indicate the most optimized query path for a given statement. Once you have determined the best case, you can execute that case on the specified data source and alter the database code to run the statement as that case on the native environment.

If you don't find an acceptably fast path, go to the **Analysis** tab. The **Analysis** tab can identify missing indexes and by examining the diagram you may be able to determine if there is something wrong with the SQL or schema.

FINDING MISSING INDEXES AND SQL PROBLEMS

The tuner performs the index and SQL analysis as part of the tuner run job performed on the **Overview** tab if **Perform Detail Analysis** is selected. Otherwise, the analysis is performed when you click the **Analysis** tab.

The screenshot shows the Oracle SQL Analysis tool interface. The top navigation bar includes 'Input', 'Overview', and 'Analysis' tabs. The 'SQL Analysis' section displays a 'Select statement of interest: UPDATE 1'. The main area shows a 'Visual SQL Tuning Diagram' (VST) with a tree structure of tables: RENTALITEM (ITM) at the top, connected to MOVIERENTAL (XXX) and MOVIECOPY (COP). MOVIECOPY (COP) is further connected to MOVIECATEGORY (CAT) and MOVIEID (MOVIEID). The diagram includes join types and cardinalities. Below the diagram, there is a 'Create Index button' with an arrow pointing to it. At the bottom, the 'Index Analysis' tab is active, showing a table of indexes:

Index Name	Table Owner	Table Name	Index Type
MOVIECATEGORY_PK	MOVIES	MOVIECATEGORY	CATEGOR
MOVIECOPY_PK	MOVIES	MOVIECOPY	MOVIECC
MOVIEID_PK	MOVIES	MOVIEID	MOVIEID
RENTALITEM_PK	MOVIES	RENTALITEM	RENTALIE

DB Optimizer can parse an SQL query and analyze the indexes and constraints on the tables in the query and display the query in graphical format on the Visual SQL Tuning (VST) diagram. The VST diagram can be displayed in either Summary Mode or Detail Mode. The VST diagram helps developers, designers, and DBAs see flaws in the schema design such as Cartesian joins, implied Cartesian joins and many-to-many relationships. The VST diagram also helps the user to more quickly understand the components of an SQL query, thus accelerating troubleshooting and analysis.

This section is comprised of the following topics:

- [Finding Missing Indexes](#) on page 44
- [Exploring the VST Diagram](#) on page 44
- [Finding Problematic SQL or Schema](#) on page 47
- [Applying Tuner Results to the Data Source](#) on page 48

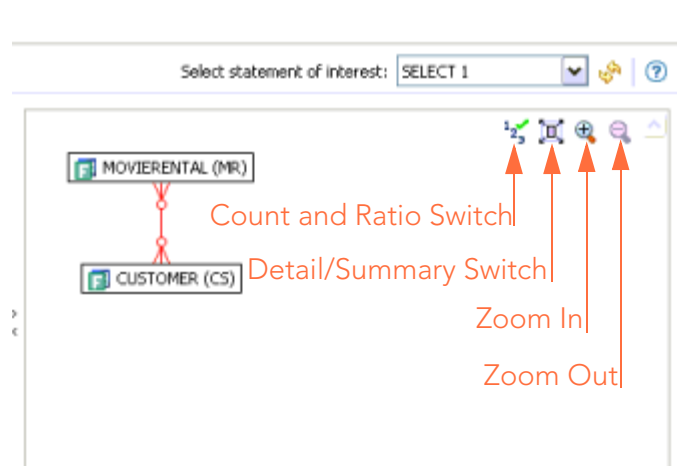
FINDING MISSING INDEXES

Missing indexes are color-coded **orange** in the **Collect and create indexes** area of the **Overview** tab. Creating a missing index can improve the execution path of the SQL statement being analyzed.

TIP: Indexes that are used are color-coded **green**. Indexes that are present but not used in this execution path are color-coded **grey**. Indexes that are usable but not used by the current execution path are color-coded **blue**.

Index Name	Table Owner	Table Name	Column Name
IDX_MOVIERENTAL_0	MOVIES	MOVIERENTAL	CUSTOMERID
IDX_MOVIERENTAL_1	MOVIES	MOVIERENTAL	TOTALCHARGE
IDX_CUSTOMER_ID	MOVIES	CUSTOMER	CUSTOMERID, LASTNAME
MOVIERENTAL_PK	MOVIES	MOVIERENTAL	RENTALID






EXPLORING THE VST DIAGRAM



DISPLAYING OR HIDING TABLE COLUMNS AND INDEXES

Click the **Detail/Summary** switch to display or hide details of the table display, including table columns and indexes. You can also switch between Summary Mode and Detail Mode for a specific table or view by double-clicking the object name. Only fields that are used in the WHERE clause are displayed in detail mode.

The following describes the icons used in tables displayed in Detail Mode.

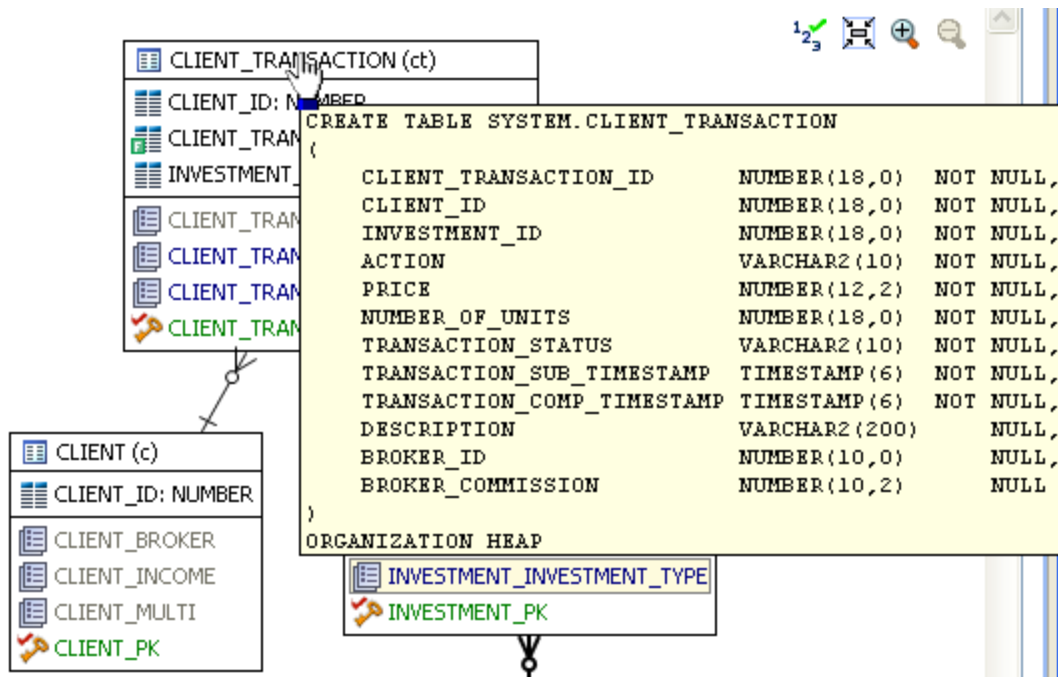
Table Icon	Description
	Table Name
	Field
	Field with a filter, used in the WHERE clause
	Index
	Primary Key

NOTE: You can move tables in the diagram by clicking and dragging them to the desired location. The position of the connecting lines is automatically adjusted.

VIEWING ALL FIELDS IN A TABLE

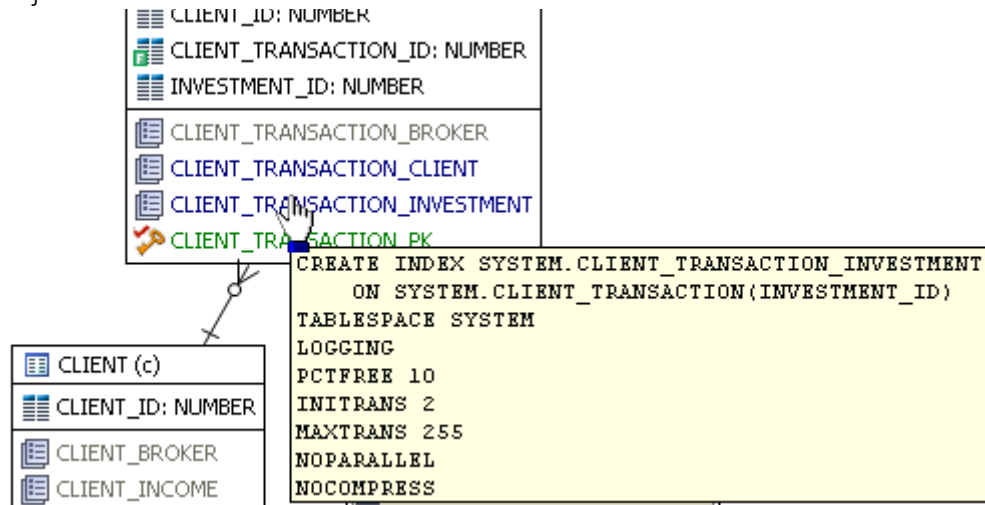
Hover the mouse over the table name to view all fields in the table in a pop-up window.

The illustration below shows the pop-up window that appears when hovering over the Viewing Object SQL.



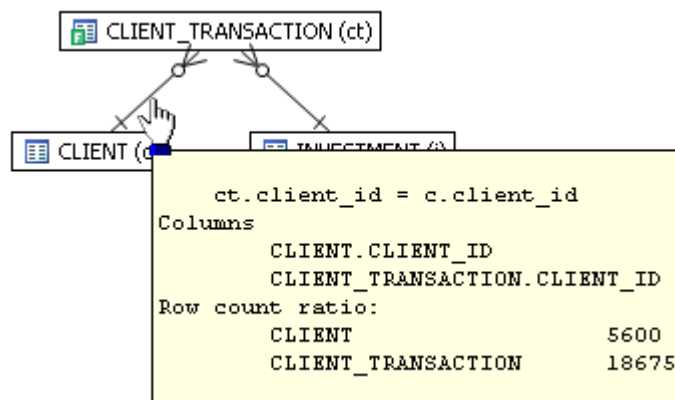
VIEWING OBJECT SQL

Hover the mouse over the table name, field, or index displays the SQL required to create that object.



VIEWING RELATIONSHIPS

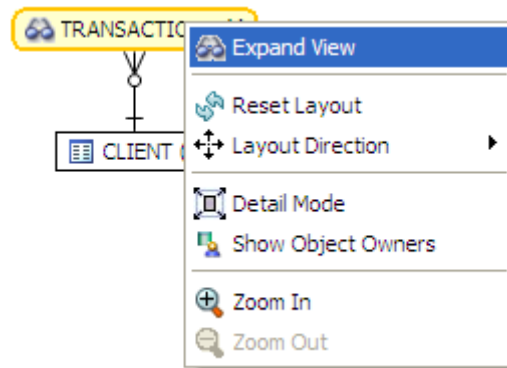
Hover over the join between two tables displays the relationship between the two tables.



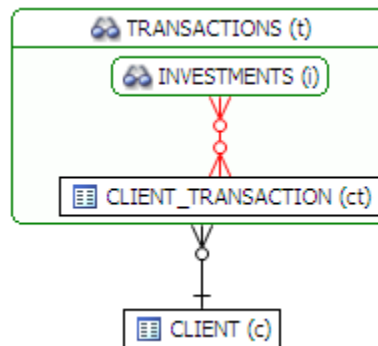
EXPANDING VIEWS IN THE VST DIAGRAM

For example, the following is the default layout from query join table CLIENT (c) to view TRANSACTIONS (t):

Right-click the view, TRANSACTION (t) and choose **Expand View**.

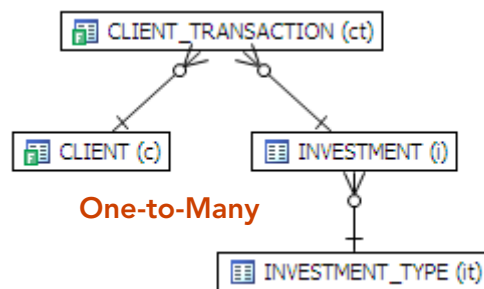


Now we can see the objects in the view:

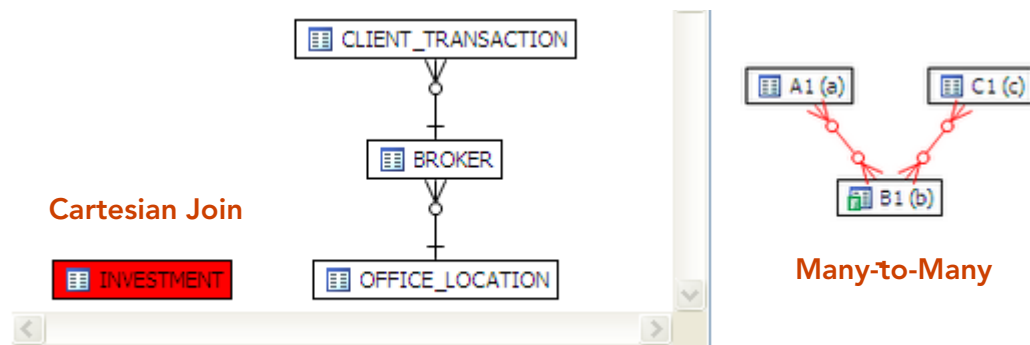


FINDING PROBLEMATIC SQL OR SCHEMA

In the Visual SQL Tuning (VST) diagram, a well-formed query would resemble the following:



Problems such as Cartesian joins, implied Cartesian joins, and many-to-many relationships are clearly represented in the VST.



Joins are represented with connecting lines between nodes. The following describes when a particular type of connecting line is used and the default positioning of the line.

Connecting Lines	When used
	One-to-One Join relationships are graphed horizontally using blue lines. One-to-one joins exist when two tables are joined on their primary key
	One-to-Many Join relationships are graphed with the many table above the one table.
	Cartesian Join shows the table highlighted in red with no connectors to indicate that it is joined in via a Cartesian join. The query is missing join criteria. A Cartesian join problem could be resolved by implementing a rewrite suggestion shown in the Generated Cases area of the Overview tab.
	Many-to-Many Join relationships are connected by a red line and the relative location is not restricted. If master detail information is missing then the VST diagram will have many-to-many connectors: The optimizer can more consistently optimize a well-formed query, so the query will run faster.

APPLYING TUNER RESULTS TO THE DATA SOURCE

Once Tuner has generated cases and statement results and analyzed the indexes, you can apply the suggested changes to the data source from the **Generated Cases** area of the **Overview** tab. You can create any recommended indexes from the **Index Analysis** area of the **Analysis** tab.

IMPLEMENTING RECOMMENDATIONS ON THE OVERVIEW TAB

To change an SQL statement based on a transformation or hint-based case:

- 1 In the **Generated Cases** area, right-click the **Name** field of the case you want to modify the original statement with and choose **Apply Change**.

The **Apply Change** dialog appears.

- 2 Choose **Execute** to apply the change to the statement automatically.

SQL Statements and Cases >>		Cost >>	Elapsed Time >>	Other Execution Statistics		
	Name	Value	Value (s)	Physical Reads	Logical Reads	Consistent Gets
<input checked="" type="checkbox"/>	SQL Statement 1					
<input checked="" type="checkbox"/>	[Null column comparison]					
<input checked="" type="checkbox"/>	SQL Statement 2					
<input checked="" type="checkbox"/>	SQL Statement 3					
<input checked="" type="checkbox"/>	SQL Statement 4					
<input checked="" type="checkbox"/>	SQL Statement 5					
<input checked="" type="checkbox"/>	SQL Statement 11					
<input checked="" type="checkbox"/>	SQL Statement 13					
<input checked="" type="checkbox"/>	SQL Statement 14	2.0	0	0	3	3
<input checked="" type="checkbox"/>	ALL_ROWS	2.0	0.829	1	3	3
<input checked="" type="checkbox"/>	FIRST_ROWS	2.0				
<input checked="" type="checkbox"/>	NO PARALLEL	2.0				

Cases can be selected and applied on the data source where the statement originated. This process improves the former statement's execution path and therefore lessens overall data source load.

IMPLEMENTING RECOMMENDATIONS ON THE INDEX ANALYSIS TAB

Once you have added tuning candidates to a tuning job, DB Optimizer can analyze the effectiveness of the indexes in the database and recommend the creation of new indexes where the new indexes can increase performance.

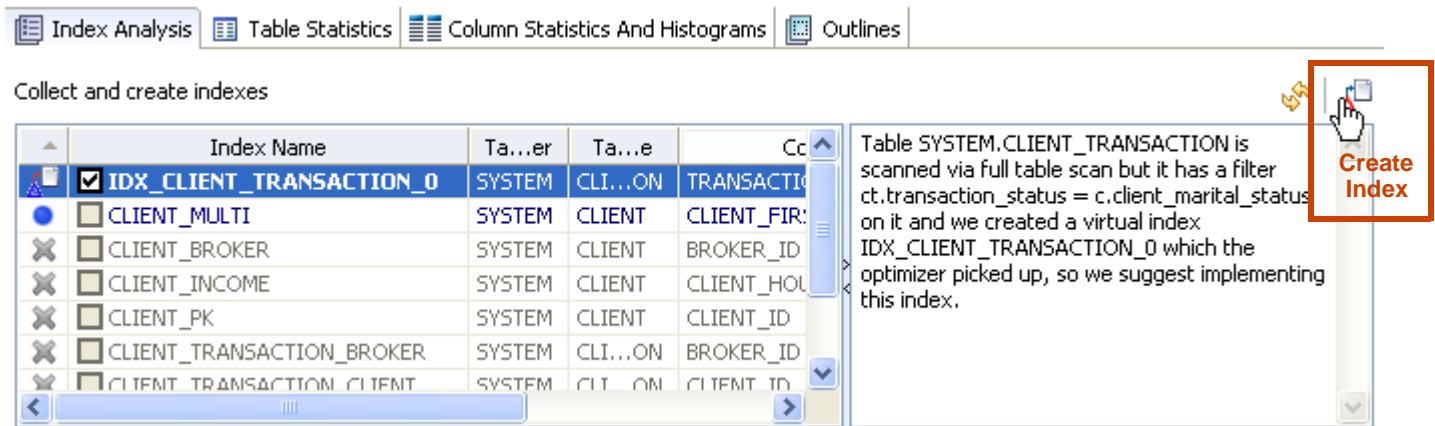
In the **Collect and create indexes** table, any indexes DB Optimizer recommends you create are marked in orange and have a create index icon next to its checkbox.

Index Analysis Table Statistics Column Statistics And Histograms Outlines					
Collect and create indexes					
	Index Name	Table Owner	Table Name	Column Name	Index
	<input checked="" type="checkbox"/> IDX_CLIENT_TRANSACTION_0	SYSTEM	CLIENT_TRANSACTION	TRANSACTION_STATUS	Normal
	<input type="checkbox"/> CLIENT_MULTI	SYSTEM	CLIENT	CLIENT_FIRST...NT_LAST_NAME	Normal
	<input type="checkbox"/> CLIENT_BROKER	SYSTEM	CLIENT	BROKER_ID	Normal
	<input type="checkbox"/> CLIENT_INCOME	SYSTEM	CLIENT	CLIENT_HOUSEHOLD_INCOME	Normal
	<input type="checkbox"/> CLIENT_PK	SYSTEM	CLIENT	CLIENT_ID	Unique
	<input type="checkbox"/> CLIENT_TRANSACTION_BROKER	SYSTEM	CLIENT_TRANSACTION	BROKER_ID	Normal
	<input type="checkbox"/> CLIENT_TRANSACTION_CLIENT	SYSTEM	CLIENT_TRANSACTION	CLIENT_ID	Normal

To accept the suggestion and have tuning automatically generate an index:

- 1 For any recommended index, click the checkbox to the left of the index you want to create.

For a selected index you can modify the Index type by clicking in the **Index Type** column and then selecting a type from the list: **Normal**, **Bitmap**, **Reverse Key**, **Reverse Key Unique**, or **Unique**.

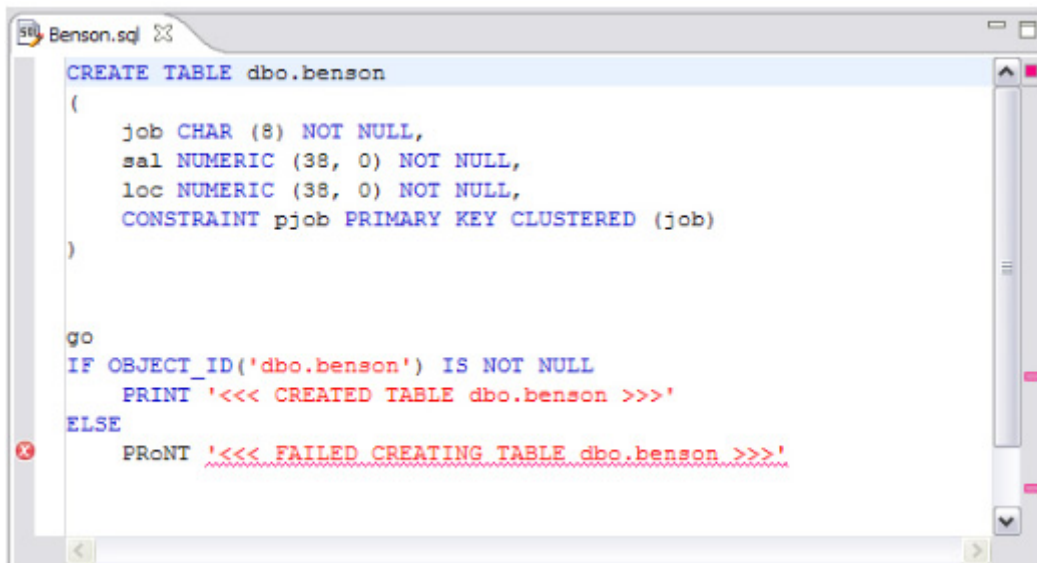


- 2 Click the **Create Index** button.
- The **Index Analysis** dialog appears.
- 3 To view the index SQL in an editor for later implementation, click the statement and then click **Open in a SQL editor**.
- 4 To run the index SQL and create the index on the selected database, click **Execute**.

SQL CODE ASSIST AND EXECUTION

SQL Code Assist is an interface component that enables the development and formatting of SQL code for the purposes of creating and modifying database objects. It provides a front-end application for the delivery of code through its object code extraction capabilities.

Code Assist provides a number of key features that assist in the coding process, ensuring greater accuracy in coding, faster development cycles, and a general increase in efficiency overall.



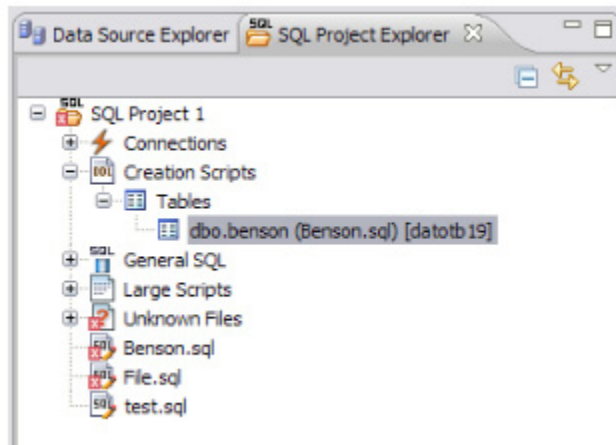
SQL Editor provides key features to ensure an increase in code accuracy and overall development efficiency.

The following key features are provided with SQL Code Assist:

- Code Extraction
- Code Highlighting
- Automatic Error Detection
- Code Complete
- Hyperlinks
- Code Formatting
- Code Folding
- Code Quality Checks

In order to access SQL Editor, you need to create a new file or edit existing code from Data Source Explorer.

Additionally, SQL Project Explorer provides a tree structure for all files created in DB Optimizer. You can access files from here by double-clicking the file name you want to open as well.



SQL Project Explorer provides a tree of all files developed and saved in DB Optimizer.

To create a new file:

Choose **File > New > SQL File**.

A blank instance of SQL Editor appears in the Workbench. If you save this file, it is automatically added to the SQL Project Explorer.

To edit an existing file:

Use Data Source Explorer or Project Explorer to navigate to the code you want to modify and double-click it.

An instance of SQL Editor appears in the Workbench, populated with the extracted code of the specified object or SQL project file.

CODE EXTRACTION

SQL Editor provides the ability to extract the underlying SQL code of database objects registered in DB Optimizer to provide a front-end application for the development and modification of data sources in your enterprise.

To extract underlying SQL code:

- Navigate to a database object in Data Source Explorer and select Extract via the right-click menu.

The object's underlying SQL code appears in SQL Editor and is ready for editing and further modification.

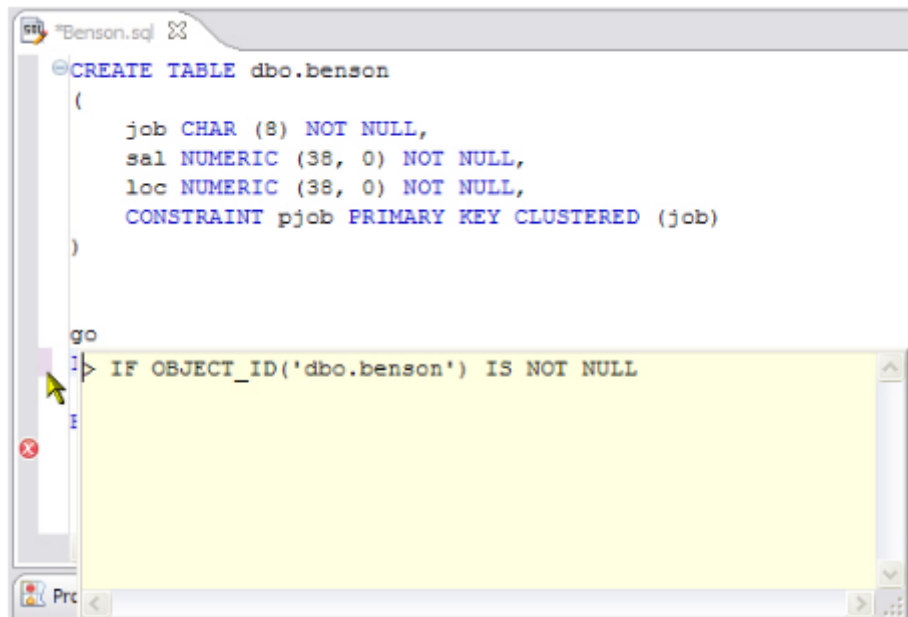
CODE HIGHLIGHTING

SQL Editor identifies commands and provides syntax highlighting changes that are automatically added to the code as you add lines, which enables you to clearly and quickly understand code when you read it in the interface.

The following syntax highlighting is automatically added to lines of code in SQL Editor:

Code	Formatting
Comments	<i>green italics</i>
SQL Commands	dark blue
Syntax Errors	<u>red underline</u>
Coding Errors	red

Additionally, SQL Editor provides a purple change bar in the left-hand column that indicates if a line of code has been modified from the original text. You can hover over this change bar to view the original code line. A red square icon in the right-hand column indicates that there are errors in the code line. You can hover over the icon to view the error count.



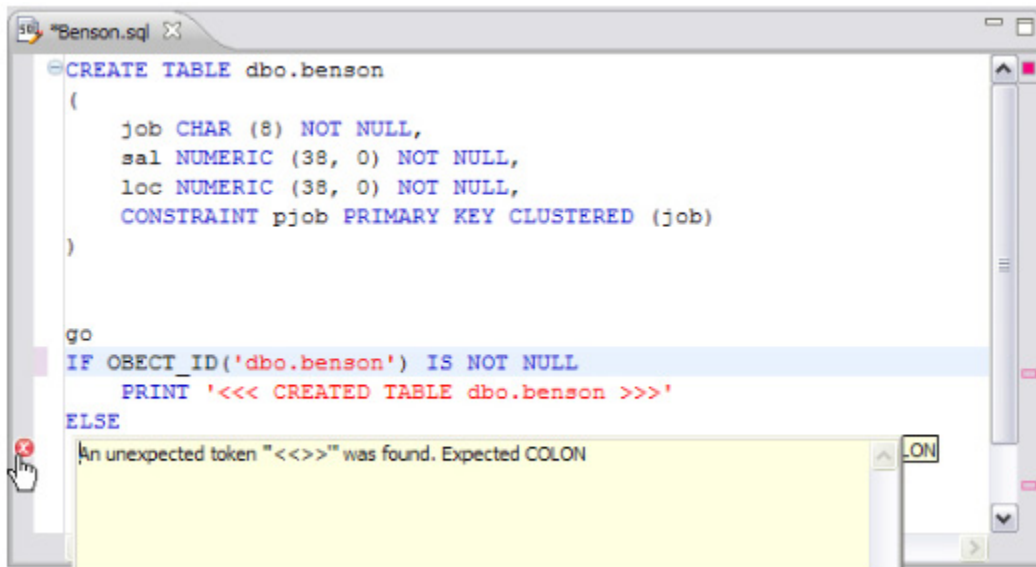
The purple change bar indicates if a line of code has been changed from its original text. Hover your mouse over the change bar to view the original text.

AUTOMATIC ERROR DETECTION

The automatic error detection functionality of the editor highlights errors and typos in the code as you work in “real time”.

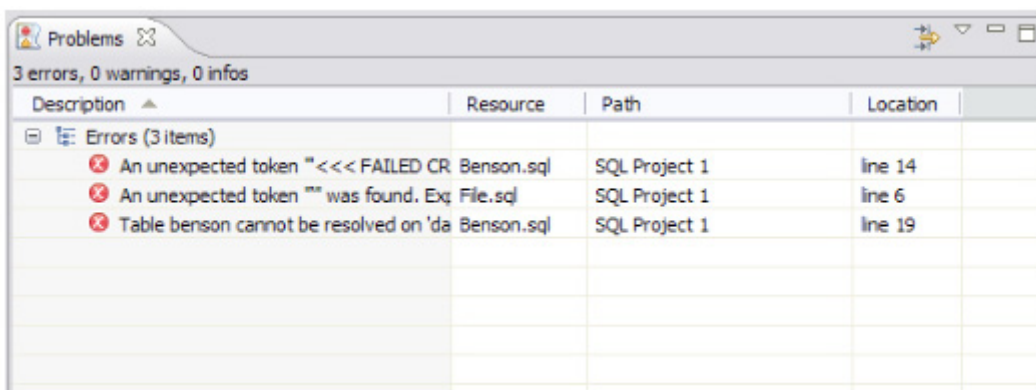
Getting Started with DB Optimizer > Automatic Error Detection

Automatic error detection automatically identifies and analyzes SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY statements. If it detects any syntax errors while you type these statements, the line is automatically flagged by the error icon in the left-hand column of SQL Editor. You can hover your mouse over the icon to view any errors.



Syntax errors are automatically flagged by line as you work with code in SQL Editor. Hover the mouse over the error icon to view the specific error message.

Additionally, all semantic errors are recorded in the Problems view, an interface component that automatically logs errors and warnings as you work with files.



The Problems view logs errors and warnings as you work with files in SQL Editor.

You can double-click on a line in the Problems view and DB Optimizer will automatically navigate you to that issue in SQL Editor.

CODE COMPLETE

SQL Editor provides suggestion capabilities for both DML statements and objects. It has the ability to look up object names in order to avoid errors when defining table names, columns, etc. in the development process. This feature provides lists of object and code suggestions that will make the development process more efficient as you will not be forced to manually look up object names and other statement values. SQL Editor provides code assist for SELECT, UPDATE, INSERT, and DELETE statements and object suggestion support for tables, alias tables, columns, alias columns, schemas, and catalogs.

To activate code assist:

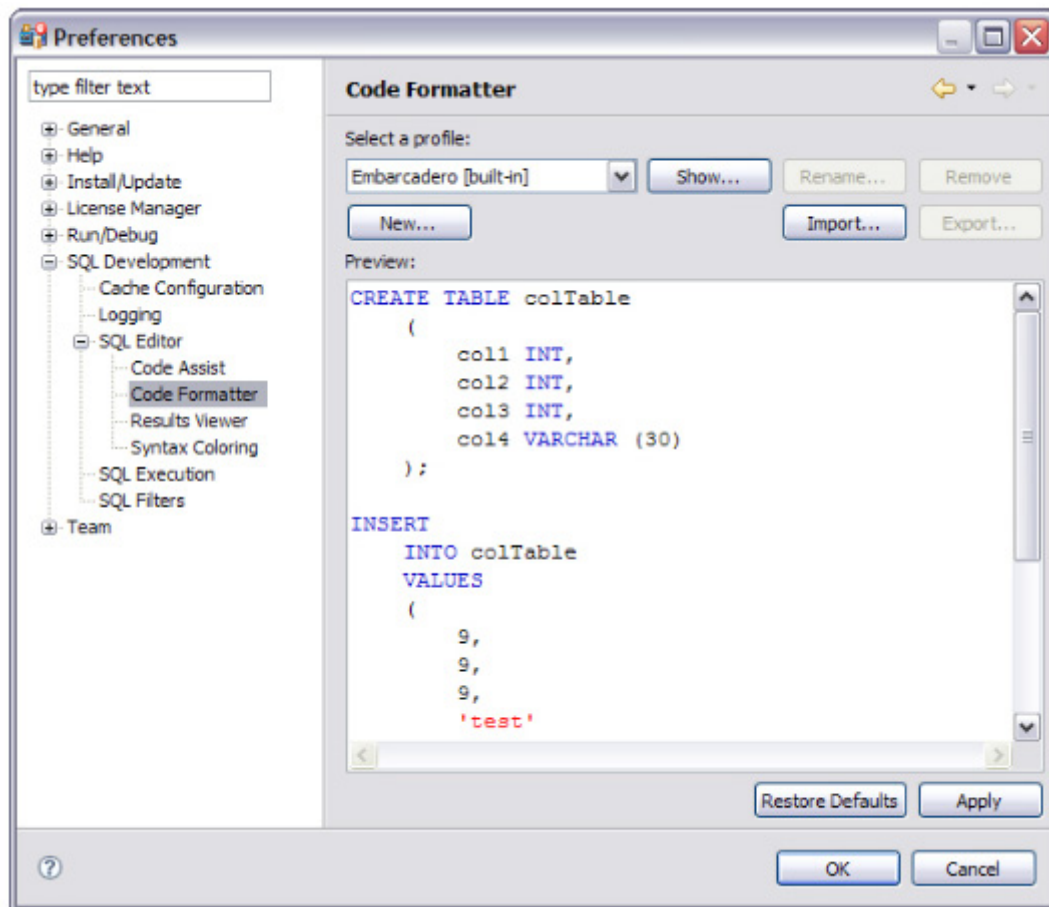
- 1 Click the line on which you want to activate the code assist feature.
- 2 Press **CTRL + Spacebar** on your keyboard. Code assist analyzes the line and presents a list of suggestions as appropriate based on the elements of the statement.

HYPERLINKS

Hyperlinks are used in SQL Editor to provide links to tables, columns, packages and other reference objects. When you select a hyperlinked object from a piece of code, a new editor opens and displays the source. Additionally, hyperlinks can be used to link procedures or the function of a call statement, as well as function calls in DML statements. To enable a hyperlink, hover your mouse over the object name and hold the CTRL key. It becomes underlined and changes color.

CODE FORMATTING

Code formatting is automatically applied to a file as you develop the file in SQL code. This enables you to set global formatting preferences one time, and then apply them to all code development, saving time and allowing for a more efficient code development process. The code formatter can be accessed by selecting **CTRL + Shift + F** in the editor. All code is automatically formatted based on parameters specified on the Code Formatter node of the Preferences panel. (Select Window > Preferences in the Main Menu to access this panel.)



Code formatting parameters can be globally set and then applied to all development work in SQL Editor.

In addition to formatting code per individual file, you can also format an entire group of files from Project Explorer. Select the directory of files that you want to apply formatting to and execute the **Format** command from the right-click menu. The files will be automatically formatted based on the global preferences.

CODE FOLDING

The code folding feature automatically sorts code into a tree-like outline structure in SQL Editor. This increases navigation and clarification capacities during the code development process. It ensures that the file will be easily understood should any future work be required on the code. As you work in SQL Editor, collapsible nodes are automatically inserted into the appropriate lines of code. Statements can then be expanded or collapsed as needed, and this feature is especially useful when working on parts of particularly large or complicated files.

CODE QUALITY CHECKS

On Oracle-based code, the code quality checking feature provides suggestions regarding improved code on a statement-by-statement basis. As you work in SQL Editor, markers provide annotations that prevent and fix common mistakes in the code.

Notes regarding code quality suggestions appear in a window on any line of code where the editor detects an error, or otherwise detects that the code may not be as efficient as it might be. Code quality check annotations are activated by clicking the light bulb icon in the margin, or by selecting **Ctrl + I** on your keyboard.

The following common errors are detected by the code quality check function in the editor:

- Statement is missing valid JOIN criteria
- Invalid or missing outer join operator
- Transitivity issues
- Nested query in WHERE clause
- Wrong place for conditions in a HAVING clause
- Index suppressed by a function or an arithmetic operator
- Mismatched or incompatible column types
- Null column comparison

To activate code quality checks:

- Click the light bulb icon in the margin of the editor or select **Ctrl + I** on your keyboard.

The editor suggestions appear in a window beneath the selected statement. When you click a suggested amendment, the affected code is automatically updated.

SQL EXECUTION

When you have finished developing or modifying code, you can then execute the file from within the DB Optimizer environment, on the database of your choosing. This enables you to immediately execute code upon completion of its development. Alternatively, you can save files for execution at a later point in time.

In order to execute a file, you must first associate it with a target database. This is performed by using the drop-down menus located in the Toolbar. When a SQL file is open in the Workbench, the menus are enabled. Select a data source and a corresponding database to associate the file with and then click the green arrow icon to execute the file.



The pair of drop-down menus indicate that the SQL file is associated with the dataotb19 data source and EMBCM database. When the green arrow icon on the right-hand side of the menus is selected, the file is executed on the specified data source and database.

Additionally, if you have turned off auto-committal in the **Preferences** panel (**Window > Preferences**) you can commit and execute transactions via the **Commit Transaction** and **Start Transaction** icons located beside the **Execute** icon.

To execute a file:

Open the file you want to run and ensure it is associated with the correct database, then click the **Execute** icon. DB Optimizer executes the code on the database you specified.

To execute a transaction:

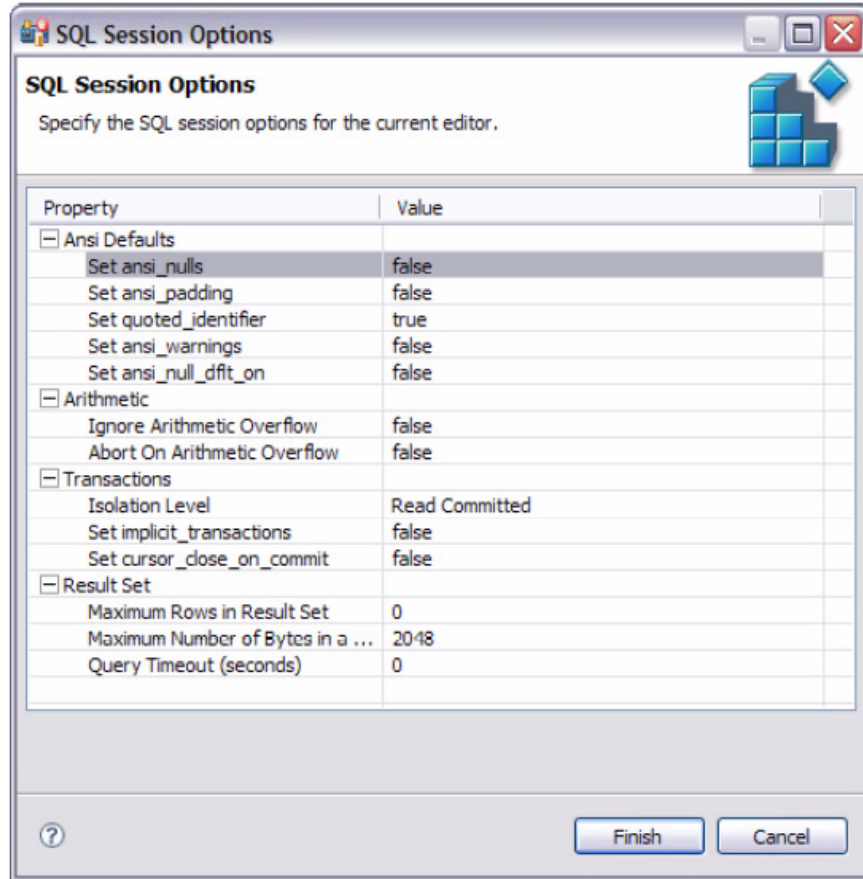
Open the transaction file you want to run and ensure it is associated with the correct database, then click the **Start Transaction** icon. DB Optimizer executes the transaction on the database you specified.

To commit a transaction:

Open the transaction file you want to commit, ensure it is associated with the correct database, and click the **Commit Transaction** icon. DB Optimizer commits the transaction on the database you specified.

CONFIGURING SQL EXECUTION PARAMETERS

Use the **SQL Session Options** dialog to modify the configuration parameters that determine how DB Optimizer executes code. These options ensure that code is executed the way you want on an execution basis, ensuring accuracy and flexibility when running new or modified code.



SQL Sessions Options provide you with the flexibility to adjust execution parameters on a session-by-session basis.

To modify SQL session options:

- 1 Click the SQL Session Options icon on the toolbar. The **SQL Sessions Options** dialog appears.
- 2 Click on the individual parameters in the **Value** column to change the configuration of each property, as specified.
- 3 Click **Finish**. The session options are changed and DB Optimizer executes the code as specified by your options.

NOTE: SQL Session Options are only applied to the currently-selected code, and are not retained across different files with regards to execution.