



## **Embarcadero® DB Optimizer™ 1.5.1: Using SQL Profiling**

Copyright © 1994-2009 Embarcadero Technologies, Inc.

Embarcadero Technologies, Inc.  
100 California Street, 12th Floor  
San Francisco, CA 94111 U.S.A.  
All rights reserved.

All brands and product names are trademarks or registered trademarks of their respective owners.

This software/documentation contains proprietary information of Embarcadero Technologies, Inc.; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with Restricted Rights, as defined in FAR 552.227-14, Rights in Data-General, including Alternate III (June 1987).

Information in this document is subject to change without notice. Revisions may be issued to advise of such changes and additions. Embarcadero Technologies, Inc. does not warrant that this documentation is error-free.

# Contents

Using Profiling .....	2
Understanding the Interface .....	3
Running a Profiling Session .....	4
Execute a Profiling Session .....	5
Working with Session Results .....	6
Save Profiling Sessions .....	21
Import Statements to Tuning .....	21
Using Other Profiling Commands .....	22
Configuring Profiling .....	24
Configuring DBMS Properties and Permissions .....	24
Building Launch Configurations .....	28
Using Load Editor .....	31

# Using Profiling

Profiling continuously samples the data source to build a statistical model of the load on the database. Profiling can be used to locate and diagnose problematic SQL code and event-based bottlenecks. Additionally, profiling enables you to investigate execution and wait time event details for individual stored routines. Results are presented in the profiling editor, which enables users to identify problem areas and subsequently drill down to individual, problematic SQL statements.

For details on working with profiling, see the following topics:

[Understanding the Interface](#)

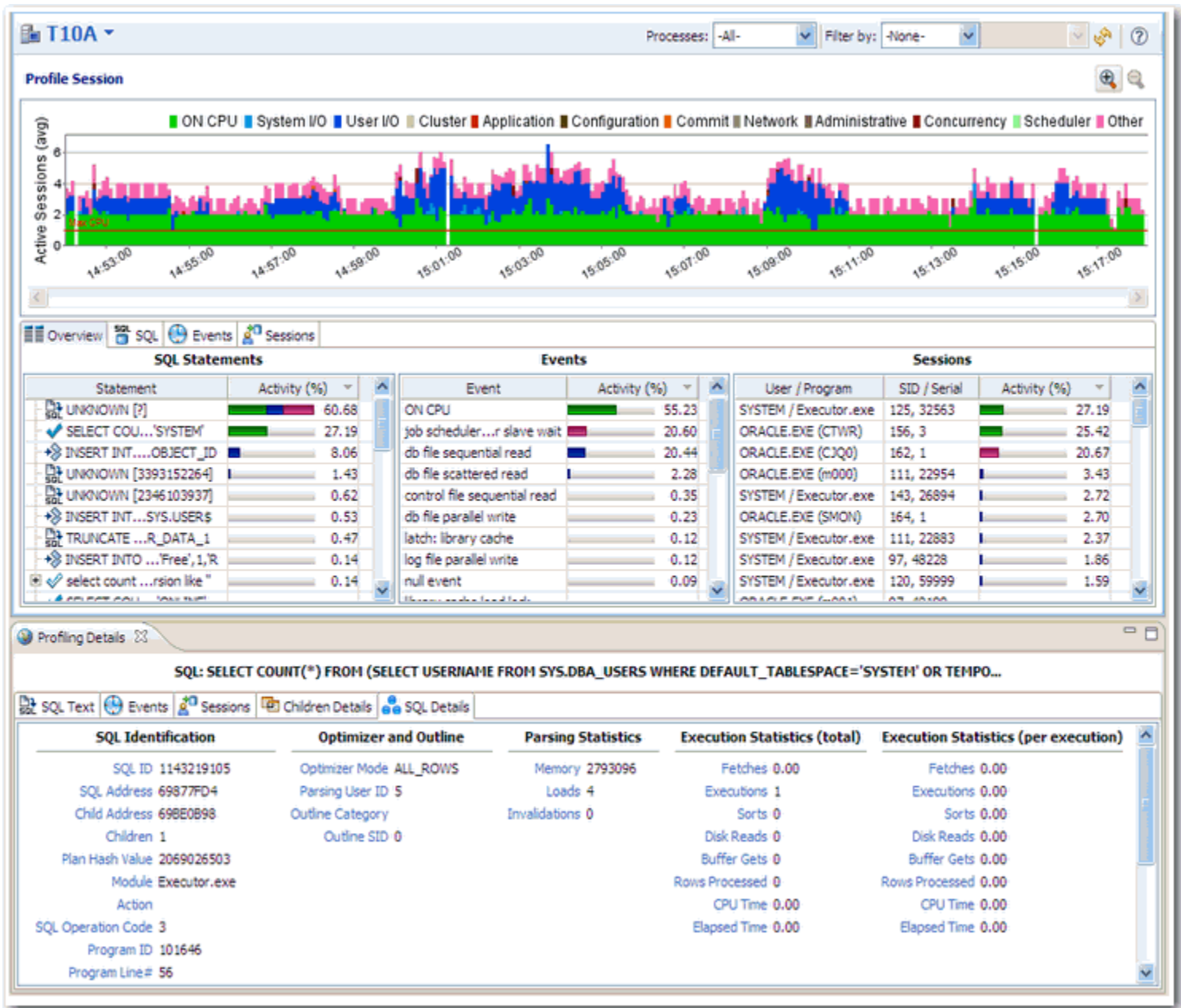
[Running a Profiling Session](#)

[Configuring Profiling](#)

[Using Load Editor](#)

## Understanding the Interface

The profiling interface is divided into three major parts:



- The **Load Graph** is located on the top section of the editor and provides a display of the overall load on the system. The bars represent individual aspects of the enterprise, and the view can be used to find bottlenecks.
- **Top Activity** is located on the middle section of the editor and displays where the load originates. Specifically, the top SQL statements, top events that the database spends time in, as well as the top activity sessions.
- The **Profiling Details View** is located on the bottom section of the editor and displays detailed information on any item selected in the middle section. For example, a SQL statement, an Event, or a Session.

The graphical portion of the profiling editor presents the distribution of sessions executed over the length of the profiling process, and those that were waiting in DBMS-specific events. It provides a first and most important step in identifying problem areas. Results can be viewed in real-time.

The **Load Graph** and **Top Activity Section** compose one view in the editor, while the **Profiling Details** view is a separate interface component that only activates when an item in the **Top Activity Section** is specified.

**Note:** It is recommended you use a 1280 x 1024 monitor resolution when viewing profiling information. Smaller resolution sizes can obscure details in the view.

## Running a Profiling Session

Profiling provides the continuous monitoring of a data source and builds a statistical model based of database load based on the users state every second. The created profile can then be saved to file, and the data can be saved, analyzed, and optimized by importing statements into the tuning component and running a tuning job.

The following list provides the general workflow and overhead tasks, when attempting to monitor data sources and store query information.

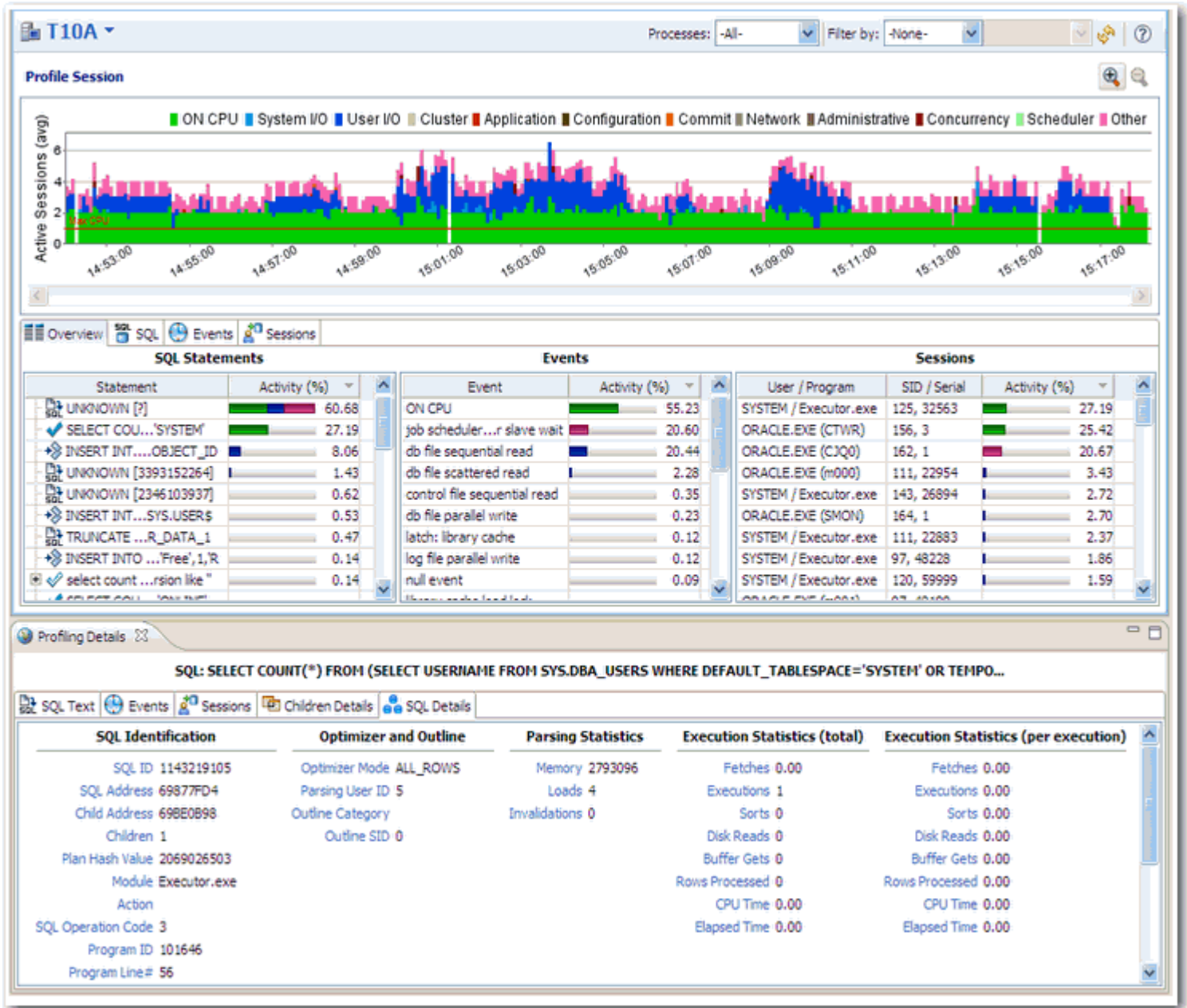
- 1 [Execute a Profiling Session](#)
- 2 [Working with Session Results](#)
- 3 [Save Profiling Sessions](#)
- 4 [Import Statements to Tuning](#)

In addition to the workflow tasks outlined above, the profiling interface also enables a number of important functions to help in statement analysis and diagnosis. This additional, or extra, functionality can be found in [Using Other Profiling Commands](#).

Furthermore, in some cases you will need to configure system variables and parameters in order to get the results you need from the application. See [Configuring DBMS Properties and Permissions](#) for more information on how to configure profiling and your registered data sources prior to running a session.

## Execute a Profiling Session

Profiling is monitored and managed via profiling's three major interface components: the **Load Chart**, **Top Activity Section**, and **Profiling Details** view.



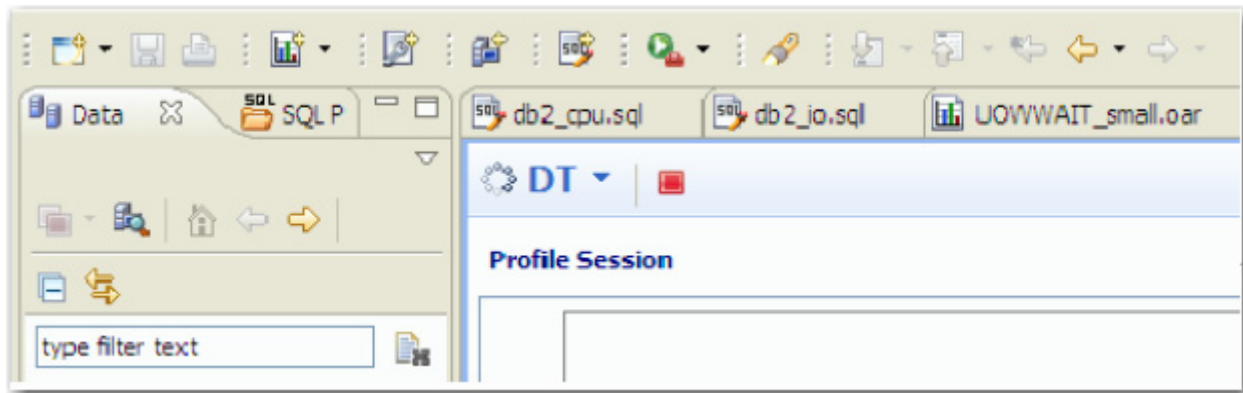
### To execute a profiling session:

- 1 In Data Source Explorer, right click on the data source you want to profile and select **Profile As** from the menu, and then choose **Data Source**. The profiling session begins. Additionally, clicking the **Profiling** icon on the Toolbar automatically runs a profiling session for the last data source you selected.

Once a profiling session launches, it runs until you stop it. When a session has run for a length of time, you can then interpret and analyze the results. See [Working with Session Results](#).

**To stop a profiling session:**

You can stop a profiling session at any time by clicking the **Stop** icon in the upper left-hand side of the screen.

**Executing a Session from the Command Line**

You can launch a profiling session from the command line using the following syntax:

```
dboptimizer.exe profile ds:ROM*L*ABORCL10G_1 duration:20 tofile:c:\testprofile.oar
```

In the above command, the user has specified ROM\*L\*ABORCL10G\_1 as the **data** source, and indicates a profiling session of 20 minutes. The **tofile** variable specifies the directory and name of the file to which the profiling session will be saved.

**Working with Session Results**

Results are displayed in the editor whenever a profiling session is executed. Results can appear in real time (if real time profiling is enabled) or once a session as finished its execution.

Results are displayed in the three major interface components of the editor, which you can use to analyze the overall efficiency and capacity of queries running on the data source, to various levels of detail:

[Analyze the Load Chart](#)

[Analyze the Top Activity Section](#)

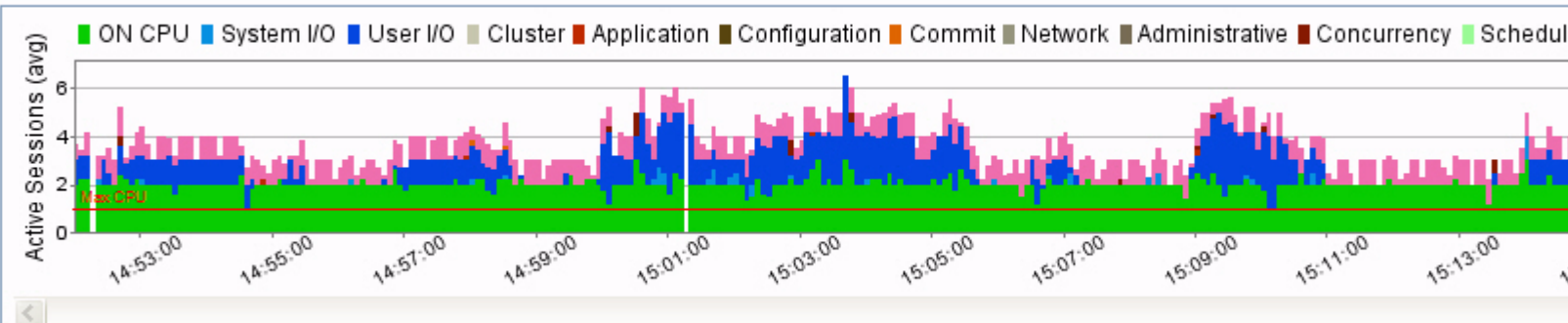
[Analyzing Profiling Details](#)



## Analyze the Load Chart

The **Load Chart** is located on the top section of the editor and provides a display of the overall load on the system. The bars represent individual aspects of the enterprise, and the view is used to discover bottlenecks.

Profile Session



The graph is designed as a high level entry point to profile session results. Subsequently, you can use the Top Activity and Details views to examine more detailed information on waiting and executing sessions over the length of the session. Alternatively, you can select one or more bars on the graph to populate the Top Activity section (and subsequently, the Details View) with information on a specific subset of the graph.

The **Load Chart** displays the distribution of waiting and executing sessions over the length of a profiling session.

- Time is displayed on the X axis. You can zoom in and zoom out on the graph via the icons in the upper right hand corner of the graph, once a profiling session is stopped.
- The Y axis shows the average number of sessions waiting or executing. Each supported platform has a specific set of wait event times.

DBMS	Wait Event Category
IBM DB2	Fetch, Cursor, Execution, Operation, Transaction, Connectivity, Lock, Other
Oracle	On CPU, System I/O, User I/O, Cluster, Application, Configuration, Commit, Other
SQL Server	CPU, Lock, Memory, Buffer, I/O, Other
Sybase	CPU, Lock, Memory, I/O, Network, Other

- A chart legend displays a color and code scheme for executing and waiting session categories, in the upper right-hand corner of the view.

## Analyze the Top Activity Section

The **Top Activity Section** is located in the middle section of the editor and displays where the load originates. Specifically, the top SQL statements, top events that the database spends time in, as well as the top activity sessions.

The Top Activity Section is composed of a series of tabs that provide detailed statistics on individual SQL statements and sessions that were waiting or executing over the length of a profiling session.

Overview				SQL		Events		Sessions	
				SQL Statements		Events		Sessions	
				Statement	Activity (%)	Event	Activity (%)	User / Program	Activity (%)
				UNKNOWN [?]	60.68	ON CPU	55.23	SYSTEM / Executor.exe	
				SELECT COU... 'SYSTEM'	27.19	job scheduler...r slave wait	20.60	ORACLE.EXE (CTWR)	
				INSERT INT... OBJECT_ID	8.06	db file sequential read	20.44	ORACLE.EXE (CJQ0)	
				UNKNOWN [3393152264]	1.43	db file scattered read	2.28	ORACLE.EXE (m000)	
				UNKNOWN [2346103937]	0.62	control file sequential read	0.35	SYSTEM / Executor.exe	
				INSERT INT... SYS.USER\$	0.53	db file parallel write	0.23	ORACLE.EXE (SMON)	
				TRUNCATE ...R_DATA_1	0.47	latch: library cache	0.12	SYSTEM / Executor.exe	
				INSERT INTO ... 'Free', 1, 'R	0.14	log file parallel write	0.12	SYSTEM / Executor.exe	
				select count ...rsion like "	0.14	null event	0.09	SYSTEM / Executor.exe	

- The **SQL Tab** provides more detailed information than provided on the Overview tab, in terms of executing SQL statements and procedures.
- The **Events Tab** displays information about wait events profiled by the execution process.
- The **Sessions Tab** displays information about sessions profiled by the execution process.
- The **I/O Tab** is Oracle-specific and will not appear in the **Top Activity Section** unless the data source being profiled is an Oracle platform. This tab displays information about the I/O profiled by the execution process.

When you select any item from the **Top Activity** tabs, details are displayed in the **Profiling Details** view. Depending on the nature of the **Top Activity** item that you selected, the tabs that compose the **Profiling Details** view will change to reflect that item's specific session information.

**Note:** When right-clicking on a SQL statement in the **Top Activity Section** in Profiling, if the SQL statement is run by a different user than the user who is running DBO, than the **User Mismatch** dialog appears, with an example of the following message: "This query was executed by [SOE] and you are currently connected as [system]. We recommend you reconnect as [SOE] to tune the SQL. Would you like to continue anyway?" This message indicates that the statement is being tuned by a user other than the user who originally ran the query, and tables may be missing based on the different schemas. Click **OK** to run the query, or click **Cancel** and run tuning under the original user.

[Viewing Details on the SQL Tab](#)

[Viewing Details on the Events Tab](#)

[Viewing Details on the Sessions Tab](#)

[Viewing Details on the I/O Tab \(Oracle-Specific\)](#)

### Viewing Details on the SQL Tab

The Profile editor's SQL tab shows a representation of all SQL statements that are executing or waiting to execute over the length of the profiling session or within the currently selected graph bars.

Overview SQL Events Sessions						
Statement	Executions	Activity (%)	SQL ID	Child Number	Parsing User ID	
UNKNOWN [?]	n.a.	60.68	0	0	n.a.	
SELECT COUNT(*) FROM (S...LESPACE_NAME='SYSTEM'	1	27.19	1143219105	0	5	
INSERT INTO PERFCNTR_D... AND L.ID1=O.OBJECT_ID	4	8.06	1135785965	0	5	
UNKNOWN [3393152264]	n.a.	1.43	3393152264	0	n.a.	
UNKNOWN [2346103937]	n.a.	0.62	2346103937	0	n.a.	
INSERT INTO PERFCNTR_D... USER# FROM SYS.USER\$	2	0.53	530002544	0	5	
TRUNCATE TABLE PERFCNTR_DATA_1	1	0.47	3440472121	0	5	
INSERT INTO PERFCNTR_DAT...ODE(STATE,0,'Free',1,'R	1	0.14	1765240657	0	5	
select count (*) from product_...t) like " and version like "	2	0.14		0	5	
SELECT COUNT(*) FROM SY...HERE STATUS = 'ONLINE'	1	0.12	1545848752	0	5	
INSERT INTO PERFCNTR_DAT...DE R NAME=ldh_black_...	1		2815406587	0	5	

Statements can be grouped by type by right-clicking the view and selecting **Organize > By Type**. The following statement types are organized:

#### INSERT, SELECT, DELETE, and UPDATE

**TIP:** Statements are grouped when they differ only by their clause values. This enables the roll-up of SQL statements that only differ by a variable value. For example: select \* from emp where empno=1; and select \* from emp where empno=2. A '+' symbol appears beside rollup statements. You can click the symbol to expand and view the different statement predicates.

Additionally, the SQL tab displays two other groupings:

Group	Description
OTHER	Includes all recognized statements other than INSERT, SELECT, UPDATE, and DELETE statements.
UNKNOWN	Statements that are not recognized by the application.

All statements are displayed in a tree structure with the following statement components:

Statement Component	Description
Subject	The DML statement type (and FROM clause, as appropriate).
Predicate	The WHERE clause.
Remainder	Any statement component following the WHERE clause.

For example, all statements with common subjects are shown as a single entry with multiple children; one child for each unique predicate. Predicates are similarly broken down by remainders.

**Note:** Right-clicking in the **SQL** tab and selecting **Organize By** lets you choose between **Statement Type** grouping and **None**. The **None** option disables grouping by statement.

Statistics are provided at each level of the tree structure. This lets you evaluate costs and spot wait event problems not just at the level of entire SQL statements, but also at the level of statement components. For each subject, predicate or remainder entry, the following statistics are provided:

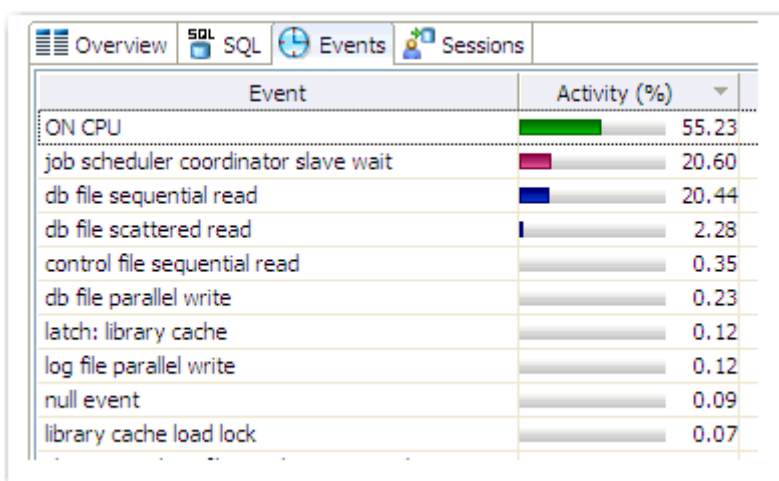
Statistic	Notes
Executions	The number of active executions for the statement or statement component over the length of the profiling session or the selected graph bars.
Activity (%)	A graphical representation of the distribution of execution and wait time for the statement or statement component.
SQL ID	The ID value of the SQL statement. This statistic only appears on Oracle data sources.
Child Number	The child number in the database. This statistic only appears on Oracle data sources.
Parsing User ID	The ID of the user who parsed the statement. This statistic only appears on Oracle data sources.
Plan Hash Value	The execution value of the statement. This statistic only appears on Oracle data sources.

Other values displayed on this tab differ depending on the data source platform. For example, the tab displays the values **SQL ID**, **Child Number**, **Parsing User ID**, and **Plan Hash Value** when analyzing a data source.

Additionally, DB2 platforms have an additional tab named **Execution Activity**, which contains the following statistical rows: **Rows Read**, **Rows Written**, **Fetch Count**, **Statement Sorts**, **Sort Time**, and **Sort Overflows**.






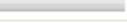
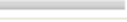




### Viewing Details on the Events Tab

The **Events** tab displays information about wait events on the resources involved in the profiling process. This display is 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 top events are related to database configuration, then database setup should be investigated.




### Viewing Details on the Sessions Tab










The **Sessions Tab** provides information about individual sessions. This tab provides information about sessions that are very active or bottlenecked.

User Name	Program	SID	Serial#	Activity (%)	Machine	Session Type
SYSTEM	Executor.exe	125	32563	 27.19	EMBARCADERO\ROWEBITA01	USER
	ORACLE.EXE (CTWR)	156	3	 25.42	TORLABORCL10G_1	BACKGROUND
	ORACLE.EXE (CJQ0)	162	1	 20.67	TORLABORCL10G_1	BACKGROUND
	ORACLE.EXE (m000)	111	22954	 3.43	TORLABORCL10G_1	BACKGROUND
SYSTEM	Executor.exe	143	26894	 2.72	EMBARCADERO\ROWEANITUCA01	USER
	ORACLE.EXE (SMON)	164	1	 2.70	TORLABORCL10G_1	BACKGROUND
SYSTEM	Executor.exe	111	22883	 2.37	EMBARCADERO\ROWSNOVAC01	USER
SYSTEM	Executor.exe	97	48228	 1.86	EMBARCADERO\ROWSNOVAC01	USER
SYSTEM	Executor.exe	120	59999	 1.59	EMBARCADERO\ROWSNOVAC01	USER
	ORACLE.EXE (m001)	97	48199	 1.54	TORLABORCL10G_1	BACKGROUND
	ORACLE.EXE (SMON)	164	1	 2.70	TORLABORCL10G_1	BACKGROUND

### Viewing Details on the I/O Tab (Oracle-Specific)

The **I/O Tab** is specific to the Oracle data source platform, and displays information about Oracle I/O loads on the profiled data source.

Object	Type	DB Activity (%)	Tablespace	File ID
EMP	TABLE	 100.00	SYSTEM	1

local v  
 read l  
 db file  
 db file  
 direct  
 direct  
 direct  
 direct  
 Undo  
 Other

The following parameters are displayed on the **I/O** tab:

Value	Description
Object	The name of the data source object affecting the Oracle I/O.
Type	The object type. For example, table, partition, or index.
DB Activity (%)	Use the color chart on the right-hand side of the I/O tab to view the I/O load on the data source during the profiling session.
Tablespace	The name of the tablespace where the object resides.
File ID	The unique ID value of the file from where specified object resides.

## Analyzing Profiling Details

The **Profiling Details** view displays detailed information on any item selected in the **Top Section View**. For example, a SQL statement, an Event, or a Session.

The screenshot shows the 'Profiling Details' window with the SQL tab selected. The SQL statement is: `SQL: INSERT INTO PERFCNTR_DATA_1(QUERYID, VALUE1) SELECT 951, COUNT(*) FROM SYS.V_$LOCK L, SYS.DBA_O`. The window displays three main sections: SQL Identification, Optimizer and Outline, and Execution Statistics.

SQL Identification	Optimizer and Outline	Execution Statistics (total)	per execution	per row
SQL ID 1135785965	Optimizer Mode ALL_ROWS	Fetches 0.00	0.00	0.00
SQL Address 6998CE60	Parsing User ID 5	Executions 1	1.00	1.00
Child Address 698AA520	Outline Category	Sorts 0	0.00	0.00
Children 1	Outline SID 0	Disk Reads 1004	1,004.00	1,004.00
Plan Hash Value 3592252508	<b>Parsing Statistics</b>	Buffer Gets 13261	13,261.00	13,261.00
Module Executor.exe	Memory 162433	Rows Processed 1	1.00	1.00
Action	Loads 134	CPU Time 93,750.00	93,750.00	93,750.00
SQL Operation Code 2	Invalidations 132	Elapsed Time 70,279,820.00	70,279,820.00	70,279,820.00
Program ID 101644				
Program Line# 195				

Depending on the data source platform you have specified, the tabs that appear in the view will be different, in order to accommodate the parameter specifics of the statement you have selected.

[Viewing Details on the SQL Tab](#)

[Viewing Details on the Sessions Tab](#)

[Viewing Details on the Events Tab](#)

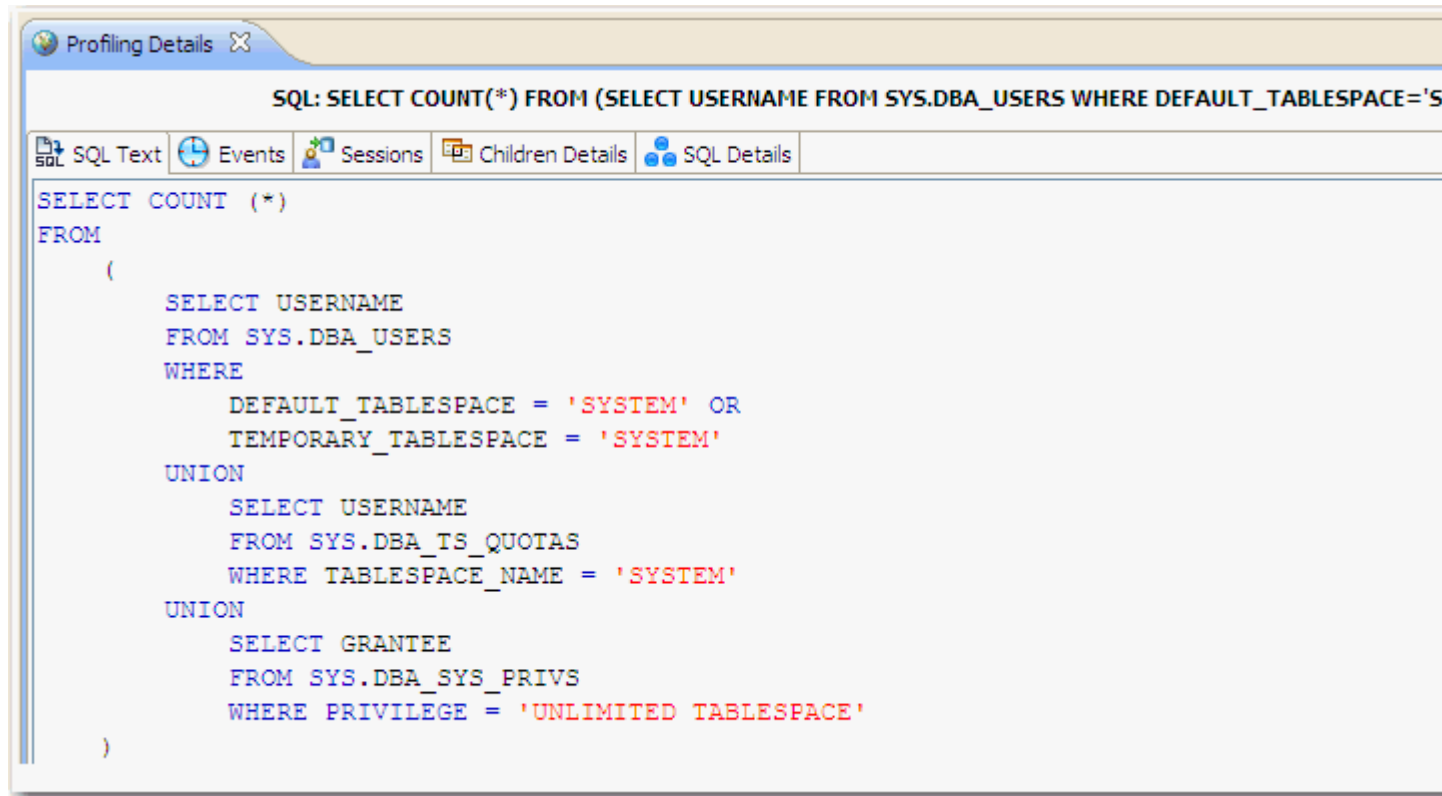
### Viewing Details on the SQL Tab

In the **Top Activity Session**, selecting a statement entry on the **SQL** tab displays information in the **Details** view. The graph portion and details on the event category tabs on the new editor pertain only to the selected statement. Additionally, new tabs become available:

- The [SQL Text](#) tab shows the full code of the SQL statement.
- The [Events](#) tab displays information about the events the selected statement is associated with.
- The [Sessions](#) tab displays information about the sessions that the selected statement is associated with. This tab is only displayed for Oracle data sources.
- The [SQL Details](#) tab displays execution details. This tab is only displayed for Oracle data sources.

**SQL Text**

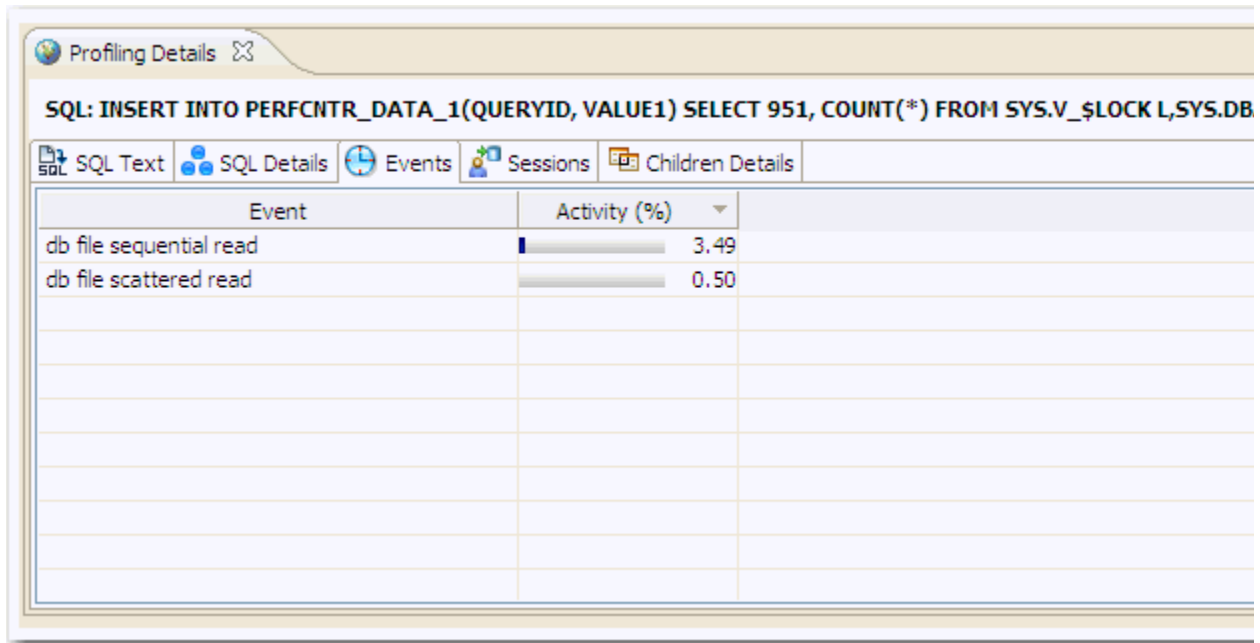
The **SQL Text** tab displays the full code of the SQL statement.





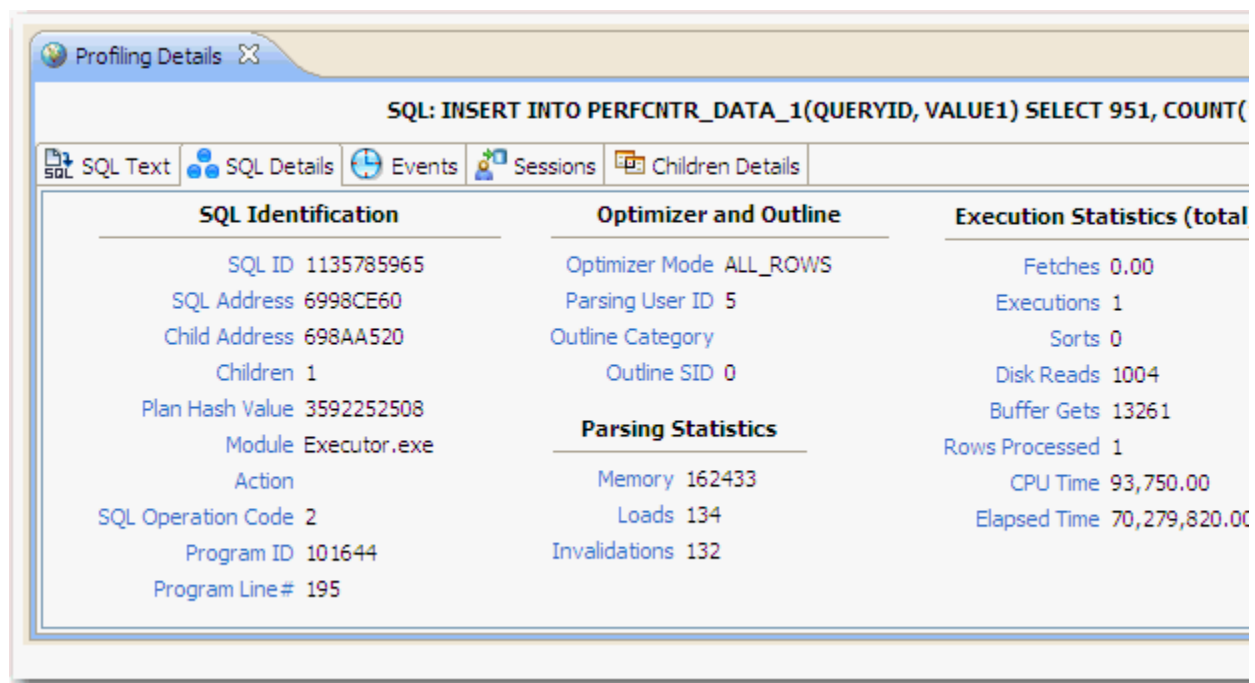
## Events

The **Events** tab provides details about the events that the statement is associated with.



## SQL Details

The **SQL Details** tab provides information and the execution of the statement and other information related to how it is running. It is only applicable to Oracle data sources:





**SQL Details** include:

Parameters	Description
SQL Identification Values	The SQL ID value of the statement.
Optimizer and Outline Values	Optimizer-specific values pertaining to the parsing user ID value and outline SID.
Parsing Statistics	Information regarding memory, loads, and invalidation values.
Execution Statistics	The execution statistics of the statement. This category includes disk reads, buffer gets, rows, and values that represent CPU and elapsed time.

## Sessions

The **Sessions** tab provides information about any sessions the statement is associated with:

User Name	Program	SID	Serial#	Activity (%)	Machine
SYSTEM	Executor.exe	145	9180	3.14	EMBARCADERO\ROWSNOVAC01
SYSTEM	Executor.exe	145	9242	0.85	EMBARCADERO\ROWSNOVAC01

Session details include information on different parameters, depending on the platform. For example, on Oracle platforms, the following parameters are displayed: **User Name**, **Program**, **SID**, **Serial #**, **Activity (%)**, **Network Machine Name**, and **Session Type**.

### To select a SQL tab statement entry:

- On the **SQL** tab, click on a statement with no child nodes or on a leaf node in the statement structure.

The new profiling editor page opens, as reflected by the breadcrumb trail at the top left of the editor. You can continue to drill down into the statement, as needed.

### Viewing Details on the Sessions Tab

In the **Top Activities Section**, selecting a statement entry on the **Sessions** tab displays information in the **Details** view. The graph portion and details on the event category tabs on the new editor pertain only to the selected statement. Additionally, new tabs become available.

Selecting an event type entry on an event category tab opens a new profiling editor page. The graph portion and details on the **Sessions** tab and event category tabs on the new editor page pertain only to the selected wait event and to SQL statements that waited in that event.

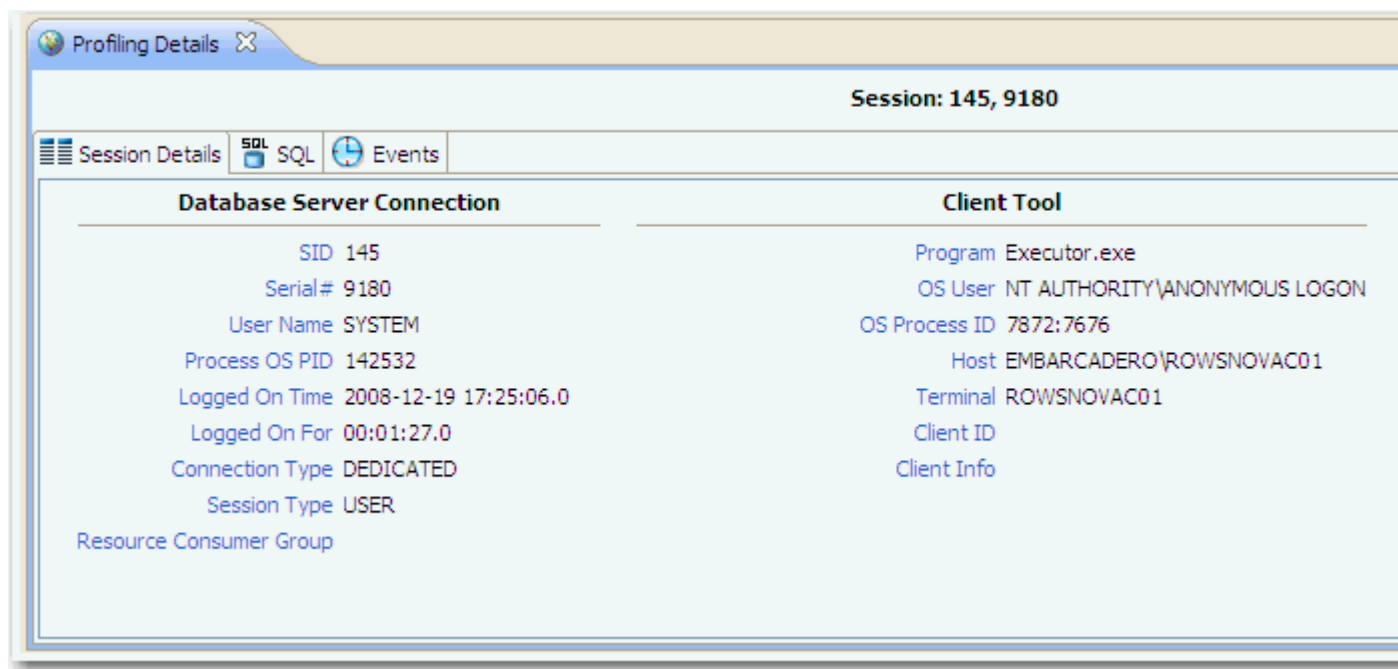
The **Sessions** tab shows system details about the selected session.

The **SQL** tab displays information about the SQL files that the selected session is associated with. This tab only appears on Oracle platforms.

The **Events** tab displays the time and parameter information about the selected session.

### Oracle Session Details

The **Session Details** tab provides further information about the selected session. This tab is only available when profiling Oracle data sources.

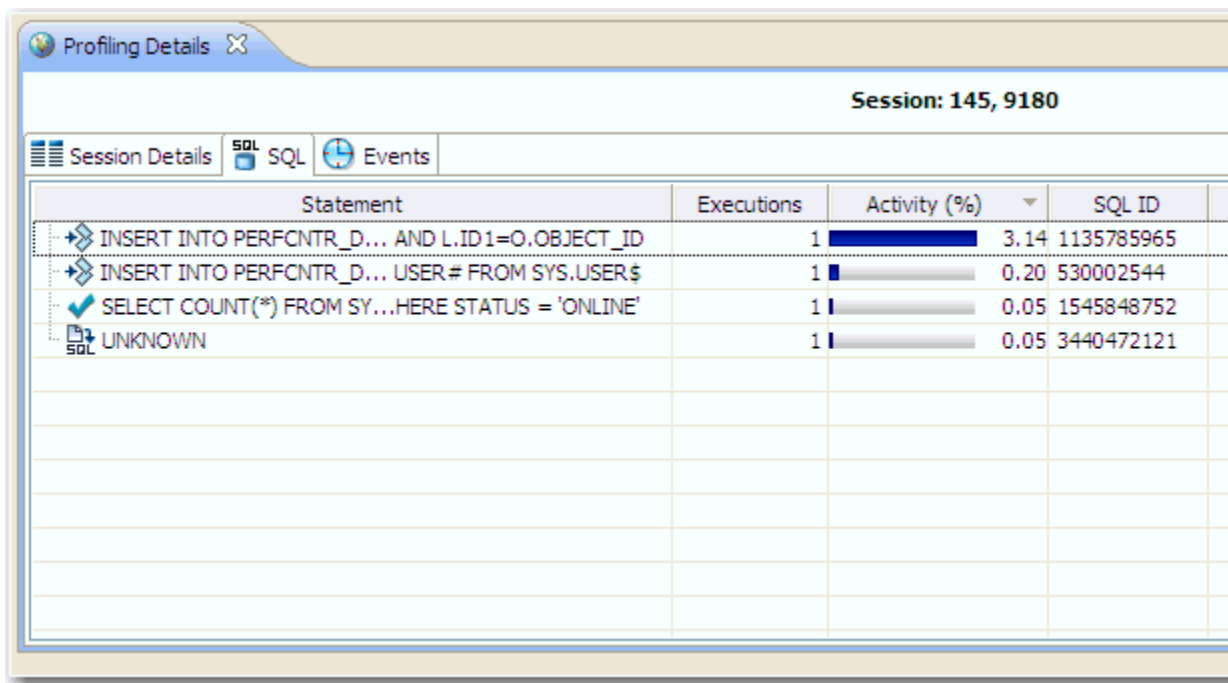


The **Session Details** tab provides information about the session, as follows:

Value	Notes
<b>Database Server Connection Information</b>	SID, Serial Number, User Name, Process OS PID, Logged On Time, Logged On For, Connection Type, Session Type, Resource Consumer Group
<b>Client Tool</b>	Program, OS User, OS Process ID, Host, Terminal, Client ID, Client Info
<b>Application</b>	SQL ID, SQL Operation Code, Last Call Elapsed Time, Module, Action, SQL Trace

**SQL**

The **SQL** tab displays information about the statements associated with the session.

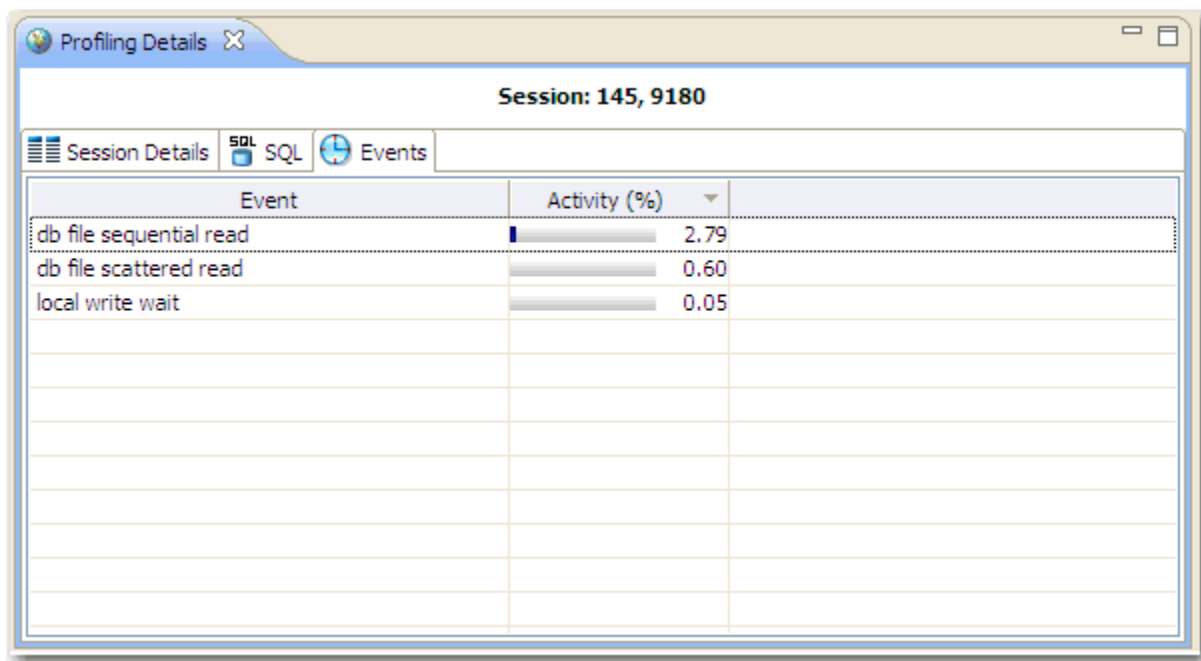


SQL statements are listed by the following parameters:

Value	Notes
Statement Name	The name of the statement.
Executions	The number of times the statement was executed during the session.
Activity (%)	A graphical representation of the distribution of execution and wait time for the statement or statement component.
SQL ID	The SQL ID value of the statement.
Child Number	The child number in the database.
Parsing User ID	The ID of the user who parsed the statement.
Plan Hash Value	The execution value of the statement.

## Events

The **Events** tab provides details about the events that the session is associated with.



Events are listed by the following values:

Value	Notes
Event Name	The name of the event.
Activity (%)	A graphical representation of the distribution of execution and wait time for the statement or statement component.

### Viewing Details on the Events Tab

In the **Top Activities Section**, selecting a statement entry on the **Event** tab displays information in the **Details** view. The graph portion and details on the event category tabs on the new editor pertain only to the selected statement. Additionally, new tabs become available.

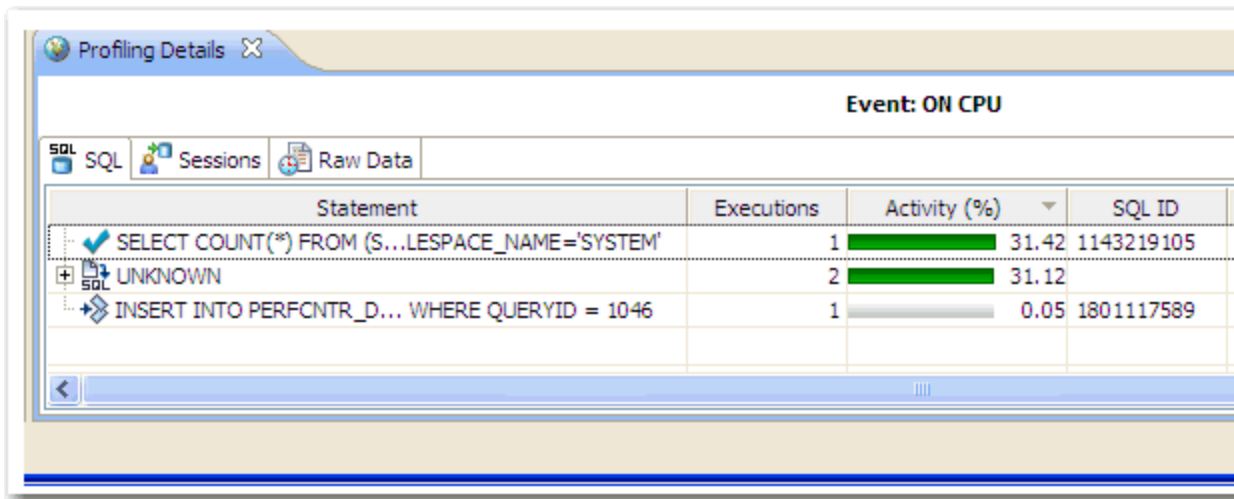
Selecting an event type entry on an event category tab opens a new profiling editor page. The graph portion and details on the **Event** tab and event category tabs on the new editor page pertain only to the selected wait event and to SQL statements that waited in that event.

The **SQL** tab shows the statements involved in the selected event.

The **Sessions** tab displays information about the sessions that the selected event is associated with.

**SQL**

The **SQL** tab displays information about the SQL statements involved in the selected event.



The **SQL** tab displays the following parameters:

Value	Notes
Statement	The name of the statement.
Executions	The number of times the statement was executed.
Activity (%)	A graphical representation of the distribution of execution and wait time for the statement or statement component.
SQL ID	The ID value of the SQL statement.
Child Number	The child number in the database.
Parsing User ID	The ID of the user who parsed the statement.
Plan Hash Value	The execution value of the statement.

## Sessions

The **Sessions** tab displays the sessions and related information regarding those that were associated with the selected event.

User Name	Program	SID	Serial#	Activity (%)	Machine
SYSTEM	Executor.exe	125	32563	31.42	EMBARCADERO\ROWEBITA01
	ORACLE.EXE (CTWR)	156	3	29.48	TORLABORCL10G_1
	ORACLE.EXE (DBW0)	167	1	0.75	TORLABORCL10G_1
	ORACLE.EXE (MMON)	161	1	0.45	TORLABORCL10G_1
IGNITE	ORACLE.EXE (J002)	149	26903	0.10	TORLABORCL10G_1
	ORACLE.EXE (SMON)	164	1	0.10	TORLABORCL10G_1
	ORACLE.EXE (PSP0)	169	1	0.10	TORLABORCL10G_1
IGNITE	ORACLE.EXE (J015)	118	1	0.05	TORLABORCL10G_1
SYS	ORACLE.EXE (J007)	124	3890	0.05	TORLABORCL10G_1
SYSTEM	Executor.exe	145	9242	0.05	EMBARCADERO\ROWSNOVAC01
IGNITE	ORACLE.EXE (J000)	159	5	0.05	TORLABORCL10G_1

The following parameters are displayed on the **Sessions** tab:

Value	Notes
User Name	The user name under which the session was run.
Program	The name of the executable under which the session was run.
SID	The SID value of the session.
Serial Number	The serial number of the machine from which the session executed.
Activity (%)	A graphical representation of the distribution of execution and wait time for the statement or statement component.
Machine	The machine name and network location of the machine from which the session executed.
Session Type	The type of session.

## Save Profiling Sessions

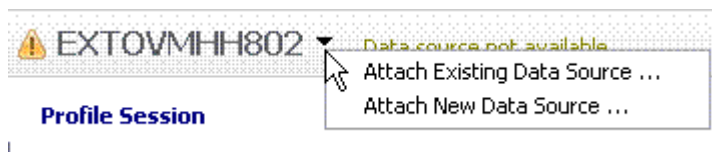
A profiling session can be saved in the current workspace in an archive file with a **.oar** suffix and with a default file name of:

- The name of the data source if the session was not initiated from a named launch configuration
- The name of the launch configuration if the session was initiated from a named launch configuration

The time period of the saved session is the amount of data on the chart. The maximum amount of data on the chart is determined when profiling is started (1 hour default).

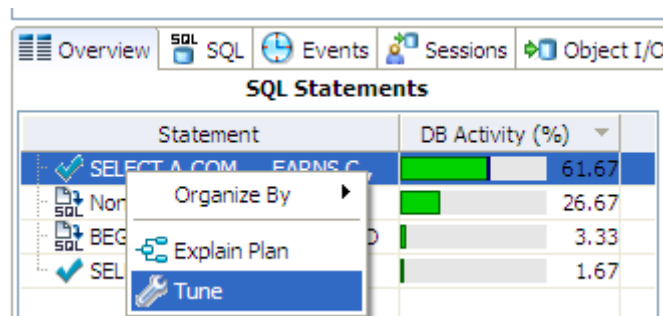
This lets you open the archive at a later time for subsequent analysis. Use standard DB Optimizer file techniques to save, open, or close SQL Profiling archives.

If you open a profiling archive on a machine on which the associated data source is not registered, a **Data source not available** warning appears in the profiling editor header. Use the associated control to specify a data source already defined on the machine or to register a new data source.



## Import Statements to Tuning

The profiling feature lets you submit one or more SQL tab statements for tuning by the tuning feature. This lets you take advantage of tuning's hint-based and transformation-based suggestions, detailed execution statistics, and explain plan costing, in tuning a statement.



**To open a tuning job on a statement appearing on the SQL tab of the profiling editor:**

- Select one or more statements, right-click and select **Tune** from the context menu. Tuning opens on the selected statement.

**Note:** The SQL will be tuned as the user/schema that profiling was running under. If the query being tuned was run by another user/schema, it is recommended to connect to the database as that user/schema and copy/paste the query into tuning, rather than import the statement directly from profiling.

## Using Other Profiling Commands

In addition to the default viewing options provided by the views, profiling also provides the following features and functionality:

- [Zooming In and Out](#)
- [Filtering Results](#)
- [Building Launch Configurations](#)

### Zooming In and Out

**To zoom in or out on the Load Graph:**

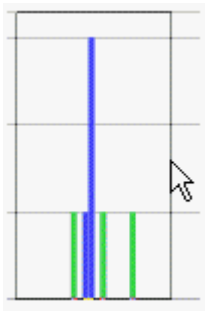
- In the upper right-hand corner of the Load graph, click the **Zoom In** or **Zoom Out** icons, respectively.

**Note:** The **Zoom In** and **Zoom Out** commands are only available when a session has been stopped.

By default, the information contained on the Load Chart spans the entire length of the profiling session. You can select one or more bars of the graph to have the tabbed view populated with statistics for only the selected subset of the graph.

**To display statistics for one or more bars on the graph, use one of the following methods:**

- Click-drag across one or more bars.



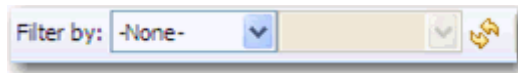


## Filtering Results

You can display filtered subsets of the original profiling results set for each section of profiling based on DBMS platform type:

- IBM DB/2 for Windows, Unix, and Linux - **Creator ID**, **Cursor Name**, **Package Name**, and **Statement Type**
- Microsoft SQL Server - **Application Name**, **Command**, **Database Name**, and **Hostname**
- Oracle - **Action Hash**, **Module Hash**, and **Program**
- Sybase - **Application**, **Database ID**, **Host**, **IP Address**, and **Process priority**

You filter results using the filter controls in the upper, right-hand part of the profiling editor.



Additionally, on Oracle platforms, you can filter results by user, or foreground, or background activity. Select **All**, **User (Foreground)**, or **Background** to filter out the specified process activity, respectively.

### To filter profile editor results:

- 1 Use the **Filter By** menu to select a filter type. The second menu becomes active based on your selection in the first menu.
- 2 Use the second menu to specify a value.

The profiling editor is updated to show only results associated with your choice.

**TIP:** Select **-None-** from the **Filter by** dropdown to restore the unfiltered results.

## Configuring Profiling

[Configuring DBMS Properties and Permissions](#)

[Building Launch Configurations](#)

### Configuring DBMS Properties and Permissions

Profiling supports the following DBMS platforms:

- IBM DB/2 for Windows, Unix, and Linux
- Microsoft SQL Server
- Oracle
- Sybase

Select a platform below for more information regarding how to set up a platform to utilize Optimizer:

- [Configuring IBM DB/2 for Windows, Unix, and Linux](#)
- [Configuring Microsoft SQL Server](#)
- [Configuring Oracle](#)
- [Configuring Sybase](#)

### Configuring IBM DB/2 for Windows, Unix, and Linux

By default, DB2 Monitor flags are set to OFF. As a result, when attempting to launch a Profile job on a DB2 data source, users may encounter the following message: “One or more errors have occurred that prevent session profiling against this data source. Examine the details below and consult your data source administrator and/or the data source documentation to resolve the problem(s).”

You can resolve this error using one of two methods:

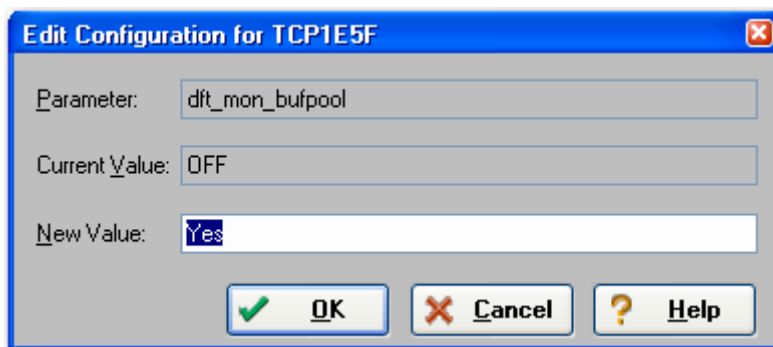
- Enabling DB2 Monitor Flags via Embarcadero DBArtisan
- Command Line Option

**To resolve the error through DBArtisan:**

1 Ensure the following DB2 Monitor Flags are turned on in DB2:

- dft\_mon\_uow
- dft\_mon\_uow
- dft\_mon\_stmt
- dft\_mon\_timestamp
- dft\_mon\_lock
- dft\_mon\_bufpool
- dft\_mon\_table

You can set view and set Monitor Flags via DBArtisan. Ensure that the **New Value** field for each variable is set to **Yes**, as shown below.



2 Restart the DB2 data source to enable the changes, then launch DB Optimizer and begin profiling.

**To resolve the error through the command line:**

This solution must be performed from DB2 CLP, on the DB2 server. If you attempt to perform these tasks through a client, an error message will result.

- 1 Navigate to START/PROGRAMS/IBM DB2/CMDLINE/COMMAND LINE PROCESSOR
- 2 Turn the monitor switches on using the following commands:

```
db2 update dbm cfg using dft_mon_lock on dft_mon_bufpool on dft_mon_sort on
dft_mon_stmt on dft_mon_table on dft_mon_uow on
db2stop
db2start
```

- 3 Ensure that the switches are turned on by connecting to the server with the following command:

```
Db2 connect to database username password password
```

The following screen provides an example of the input and output from the server:

```
db2 => connect to gim user db2admin
Enter current password for db2admin
db2 => get monitor switches
```

#### Monitor Recording Switches

```
Switch list for db partition number 0
Buffer Pool Activity Information (BUFFERPOOL) = ON 03/05/2009 19:14:06.6125
Lock Information (LOCK) = ON 03/05/2009 19:14:06.6125
Sorting Information (SORT) = ON 03/05/2009 19:14:06.6125
SQL Statement Information (STATEMENT) = ON 03/05/2009 19:14:06.6125
Table Activity Information (TABLE) = ON 03/05/2009 19:14:06.6125
Take Timestamp Information (TIMESTAMP) = ON 03/05/2009 18:50:44.0003
Unit of Work Information (UOW) = ON 03/05/2009 19:14:06.6125
```

## Configuring Microsoft SQL Server

Perform the following tasks to ensure that SQL Server is compatible with Optimizer:

- If you are setting up SQL Server 2000 or 2005, ensure the current user is a member of **sysadmin**.
- If you are setting up later versions of SQL Server, the current user must meet one of the following requirements:
  - Be a member of **sysadmin**, or have the **VIEW SERVER STATE** permission enabled.
  - Be a member of **sysadmin**, or have the **SELECT** permission enabled.

### On SQL Server 2000 only:

You can enable profiling to capture more SQL by adding the following flag:

```
DBCC TRACEON(2861)
```

Trace flag 2861 instructs SQL Server to keep zero cost plans in cache, which SQL Server would typically not cache (such as simple ad-hoc queries, set statements, commit transaction, and others). In other words, the number of objects in the procedure cache increases when trace flag 2861 is turned on because the additional objects are so small, there is a slight increase in memory that is taken up by the procedure cache.

Ensure you restart the server for your changes to take affect.

## Configuring Oracle

Oracle users need access to V\$ views. In order to configure Oracle to provide users with these privileges:

- If you are setting up Oracle 10 or later, ensure you are logged in as **sys** or **system** with the **sysdba** role, or the **SELECT\_CATALOG\_ROLE** has been granted to *user\_name*.
- If you are setting up an earlier version of Oracle, ensure you are logged in as **sys** or **system** with the **sysdba** role.

## Configuring Sybase

Perform the following tasks to ensure that Sybase is compatible with Optimizer:

- Ensure the following system configuration properties are activated:
  - **Enabling Monitoring** (`sp_configure "enable monitoring", 1`)
  - **Wait Event Timing** (`sp_configure "wait event timing", 1`)
  - **Max SQL Text Monitored** (`sp_configure "SQL batch capture", 1`)
  - **SQL Batch Capture** (`sp_configure "max SQL text monitored", 4096`)

Additionally, perform the following tasks, as necessary:

- If a user does not have **mon\_role** enabled, the user will not be able to access Adaptive Server's monitoring tables.
- If the **monProcess** table is missing, the user will not be able to view currently connected sessions.
- If the **sysprocesses** table is missing, the user will not be able to view information about Adaptive Server processes.
- If the **monWaitEventInfo** table is missing, the user will not be able to view information about wait events.
- If the **monProcessSQLText** table is missing, the user will not be able to view currently executing SQL statements.

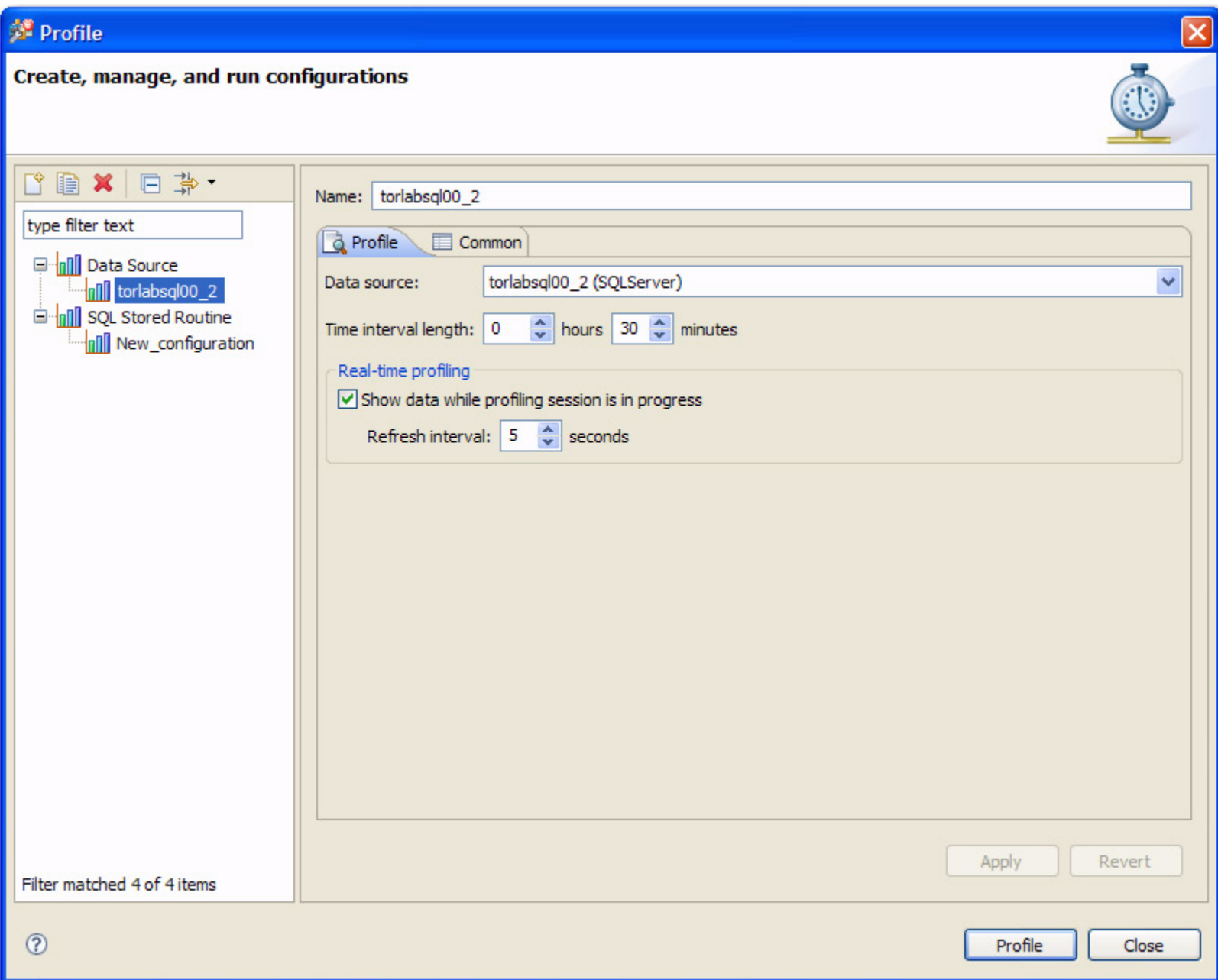
**Note:** These packages should only be installed by the DBA.

Profiling enables you to create a set of launch configurations to store the basic properties for each profiling session that you run on a regular basis. A launch configuration enables you to start profiling sessions from a single menu command, rather than re-define configuration parameters each time you want to run one.

## Building Launch Configurations

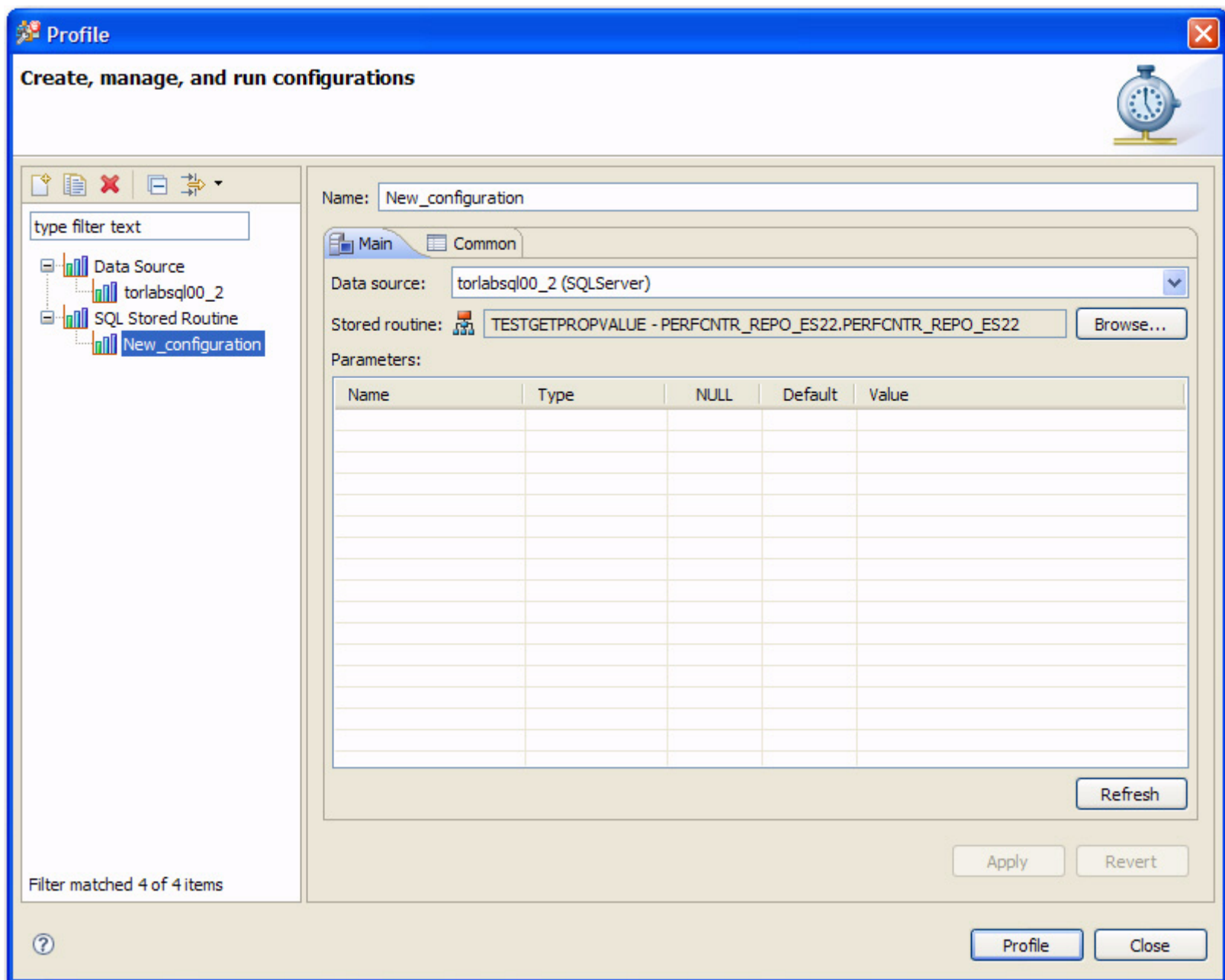
Profiling enables you to store parameters related to specific profiling sessions, in a launch configuration for stored routines. Multiple configurations can be created for each data source in your enterprise and saved with unique names that identify them in the application.

**Note:** On all supported platforms, support for stored routines includes functions and procedures. On Oracle, stored routine support also includes package functions and package procedures.



- **Data Source** indicates the name of the data source to which the profile applies.

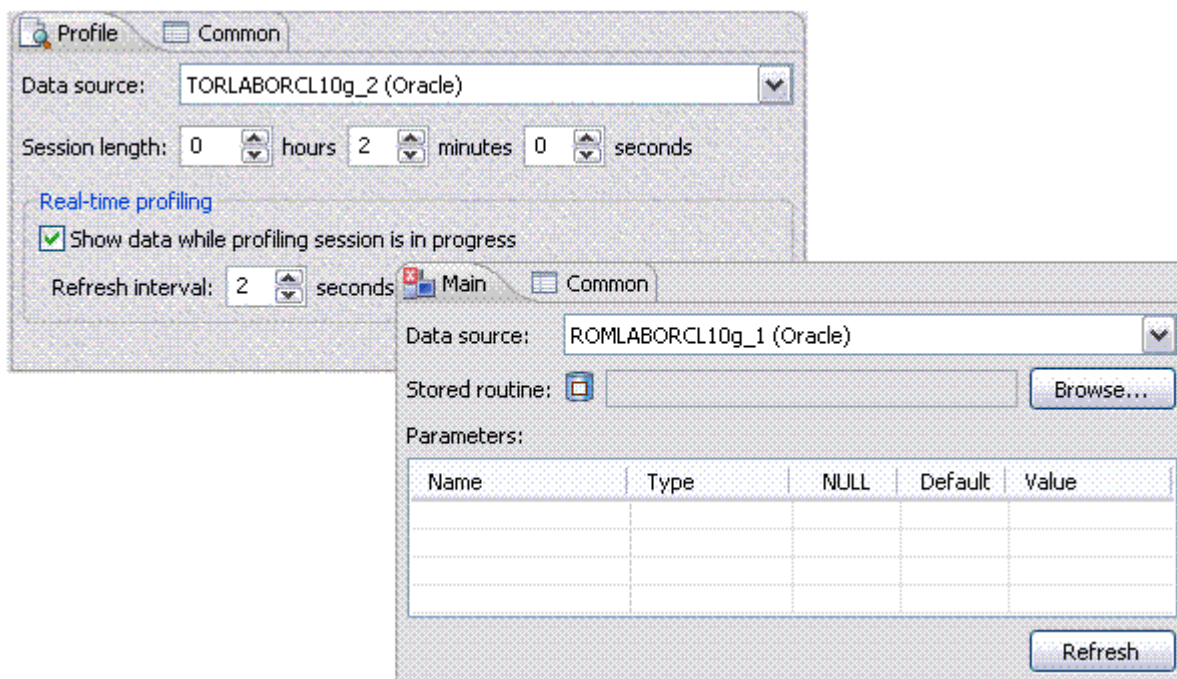
- **Time Interval Length** indicates how long the session profiles the data source term, in the format of hours and minutes. This parameter also indicates the total width of the time load graph. The longer a profile is, the larger the saved file will be. It is not recommended to make the time interval length value beyond eight hours.
- The **Show Data While Profile Session is in Progress** check box enables “real time” profiling, which refreshes the data of the session as profiling runs. The **Refresh Interval** specifies how often profiling updates this data in minutes.



**Note:** Profiling can run sessions based on ad hoc parameters you designate before executing the profiling process. However, by building profile launch profiles, it is a much more efficient method of managing standard, frequent, or common profiling sessions.

**To create a launch configuration:**

- 1 Right-click on the data source you want to build a configuration for and select **Profile As** from the menu, then choose **Open Profile Dialog**. The **Profile** dialog appears.
- 2 Select the name of the data source and modify the parameters on the **Profile** tab, as needed.
- 3 In the left pane, select the name of the data source, and click the **New Launch Configuration** icon. The right pane of the dialog populates with the parameters needed to define the launch configuration of the type that you specified.



- 4 In the Name field, provide a name for the launch configuration. You should select a name that will make the launch configuration unique and easily identified once it is saved in the application.
- 5 Ensure that the selected value in the **Data Source** is accurate. If it is not, then use the drop down menu to select the proper data source.
- 6 Use the **Browse** button to specify the stored routine. Depending on the specified stored routine, profiling will populate the fields of the **Parameters** table with the specified input parameters. These parameters will be used when executing the stored routine when running a session from the launch configuration.
- 7 Click **Apply**. The launch configuration is stored in the application.

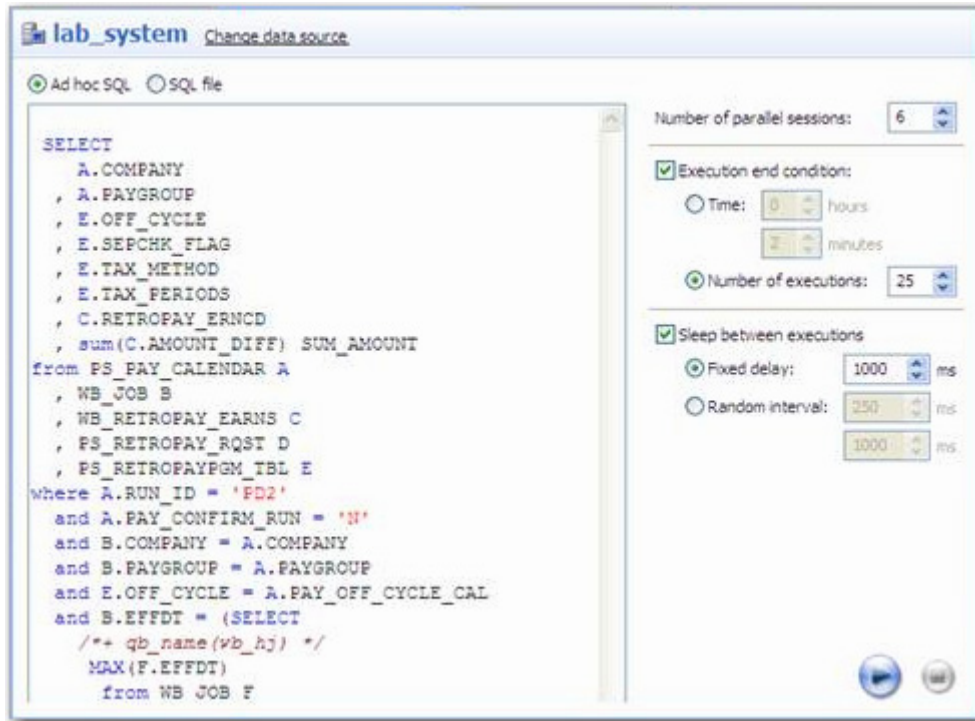
Once a launch configuration is defined, you can execute it in profiling. For more information, see [Running a Profiling Session](#).

**Note:** The parameters provided when you select the data source name in the left pane control session parameters for the specified data source. To set these controls, see [Configuring DBMS Properties and Permissions](#).



# Using Load Editor

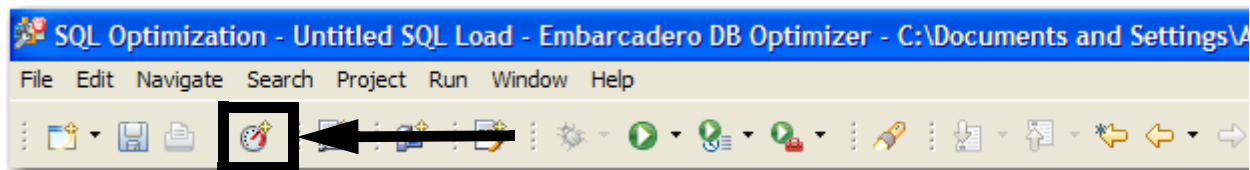
SQL Load Editor enables you to configure and execute SQL code against a data source.



This feature enables you to specify a data source against which the code will be executed, and then provides options that enable you to choose a period of time that you want the script to execute for, and at what intervals the execution “loop” occurs.

On execution, SQL Load Editor runs in the background. It can therefore be run in conjunction with a profiling session in order to analyze the effects of the executing load against the specified data source. Once you run a SQL script via Load Editor, you can start the SQL Profiling function and analyze the results of the load.

The Load Editor is accessed via the **Load Editor** icon on the Toolbar:



When you open Load Editor, click **Select Data Source** to specify the data source against which you want the SQL script to run.

Choose **Ad hoc SQL** and manually type (or copy/paste) the SQL code into the window provided, or select **SQL file** and navigate to the SQL file you want to run. The window populates with the code from the selected file.

The following configuration parameters are set with Load Editor prior to executing the SQL script:

Configuration Parameter	Description
<b>Number of Parallel Sessions</b>	Specifies the number of jobs that the execution script will operate on.
<b>Execution End Condition</b>	Specifies if the script execution process runs for a set amount of time or script executions.  Choose <b>Time</b> if you want the script to execute over a specific period of time, or <b>Number of Executions</b> if you want the script to execute a specific number of times.
<b>Sleep Between Executions</b>	Specifies if Load Editor will wait before running the execution script again.  Select the check box and choose <b>Fixed Delay</b> or <b>Random Interval</b> , depending on whether you want the script to execute at a specific time, or at random intervals within a specified range of time.

#### To run Load Editor:

- 1 Access Load Editor by selecting the icon on the Toolbar. Load Editor opens.
- 2 Click **Select Data Source** and choose a data source you want to run the SQL code against.
- 3 Choose **Ad hoc SQL** or **SQL file**, and then copy/paste or manually type the code you want to execute in the window provided, or navigate to the location of the file, respectively.
- 4 In the right-hand panel, choose the execution configuration parameters to specify how you want Load Editor to handle the script.
- 5 Click the **Execute** icon in the lower right-hand corner of the screen. The script starts to execute against the specified data source, using the configuration parameters you selected.
- 6 If you are profiling a data source, start and run a new profiling session on the data source you specified in Load Editor. The session will reflect how your SQL script executes against the specified data source.

>

## **A**

archives, profiling  
    opening/saving 21

## **F**

filtering profile results 23

## **P**

permissions, SQL Profiler 24  
profiling sessions  
    configuring DBMS for 24  
    filtering results 23  
    opening/saving 21  
    submitting tuning sessions from 21

## **T**

tuning sessions  
    opening from profiling session 21