

Users Manual

SQLShot™

*For Oracle Database
From 8i to 11g*





DATABASE Tuning Release 4.2 User's Guide

| | |
|--|------------------|
| <u>ABOUT DATABASE TUNING</u> | <u>4</u> |
| <u>DATABASE TUNING'S TECHNICAL SUPPORT</u> | <u>5</u> |
| <u>ENHANCEMENTS FEATURED IN THIS 4.2 RELEASE.....</u> | <u>6</u> |
| <u>FROM INSTALLATION TO MONITORING</u> | <u>7</u> |
| <u>ARCHITECTURE AND PREREQUISITES OF DATABASE TUNING.....</u> | <u>8</u> |
| MONITORED ORACLE INSTANCE | 8 |
| CLIENT APPLICATIONS..... | 8 |
| DATABASE TUNING RELEASE 4.2 | 8 |
| APPLICATION DATA | 8 |
| %APPDATA% | 9 |
| «.MNT» FILES | 9 |
| «.XLS» FILES | 9 |
| «.TXT» FILES | 9 |
| «.FLT» FILES..... | 9 |
| «.GPH» FILES..... | 9 |
| «DATABASE TUNING.LOG» FILE | 9 |
| «.TMP» FILES | 9 |
| «.SQL» FILES..... | 9 |
| «.MON» FILES | 9 |
| <u>INSTALLATION.....</u> | <u>10</u> |
| <u>LICENSE KEY REGISTRATION</u> | <u>14</u> |
| <u>HOW TO START AND STOP MONITORING.....</u> | <u>16</u> |

THE PROPERTIES OF THE CONNECTION.....17
THE PERFORMANCE ITEMS18
SNAPSHOT INTERVALS18
LICENSE18
THE ACTIVATION OF THE SESSION18
VERIFICATION THAT DATABASE TUNING IS PROPERLY SET UP.....19
MONITORING SESSION’S SHUTDOWN19
GRANT MONITOR PRIVILEGES TO A USER.....20

MONITORING SQL STATEMENTS.....22

CONTENTS OF THE SESSION’S REPORT.....22
DETAILED SQL STATEMENT23
 DETAIL, GENERAL25
 DETAIL, SQL STATEMENT26
 DETAIL, BATCH TEXT27
 DETAIL, EXECUTION PLAN28
 DETAIL, PROCEDURE, TRIGGER AND PACKAGE29
 DETAIL, INSTANCE’S PERFORMANCES.....30
 DETAIL, PRINT32
SQL STATEMENTS’ GRAPHS.....33
FILTERS.....35
 FILTER, SID35
 FILTER, USER36
 FILTER, ORACLE INSTANCE <-> CLIENT HOST.....37
 FILTER, APPLICATION38
 FILTER, DATE-TIME39
 FILTER, SQL STATEMENT39
 FILTER, BATCH TEXT40
 FILTER, PROCEDURE, TRIGGER AND PACKAGE40
 FILTER, EXECUTION PLAN41
 FILTER, PERFORMANCES.....41
 FILTER - HOW TO MANAGE FILTER FILES42
DISPLAY OPTIONS42
SORTS43
NEW SESSION.....44

MONITORING THE ORACLE INSTANCE’S PERFORMANCES.....46

ORACLE INSTANCE’S PERFORMANCE INDICATORS48
HOW TO CONFIGURE THE INDICATORS.....48
HOW TO RESTORE A CONFIGURATION49
HOW TO ADD AN INDICATOR.....50
HOW TO MODIFY AN INDICATOR51
HOW TO REMOVE AN INDICATOR52
HOW TO SAVE THE CONFIGURATION53
HOW TO REFRESH THE GRAPH54
SQL STATEMENTS RELATED TO A TIME PERIOD54
GRAPHS’ DISPLAY OPTIONS55
NEW GRAPH56

HOW TO SAVE RESULTS57

HOW TO SAVE TO A DATABASE57
 ACCESS MANAGEMENT57
 CONNECTING TO THE ARCHIVE DATABASE.....58

HOW TO SAVE TO A FILE 59

HOW TO SAVE AUTOMATICALLY 60

EXPORTS 60

HOW TO EXPORT TO EXCEL 60

HOW TO EXPORT TO A TEXT FILE 61

HOW TO RETRIEVE RESULTS 63

HOW TO RETRIEVE FROM A DATABASE 63

HOW TO RETRIEVE FROM A FILE..... 64

AUTOMATIC REFRESH..... 65

RESULTS DELETIONS..... 67

EXECUTING DATABASE TUNING IN A BACKGROUND TASK 68

CREATING A SCHEDULED TASK 68

PARAMETERS OF THE BACKGROUND TASK EXECUTION 72

HOW TO SHUTDOWN A SESSION IN BACKGROUND MODE 73

DATABASE TUNING'S ARCHIVE DATABASE MODEL 74

TABLES..... 75

SSV_STMT 76

SSV_BATCH..... 77

SSV_OBJECT 77

SSV_TEXTOBJECT 77

SSV_EXEPLAN..... 78

SSV_TEXTPLAN 78

SSV_MONITOR..... 78

SSV_MONITOR_SERVERNAME..... 79

SSV_MONITOR_TYPE_INFO 79

SSV_VERSION 79

About DATABASE Tuning

Copyright © 2004-2010 by DATABASE Softwares

For any additional information about this handbook and/or other products and services thereof, please contact:

DATABASE Softwares

2, rue Troyon
92310 Sèvres
France

Phone: +33 (1) 41 14 60 70

Fax: +33 (1) 41 14 60 71

Email: <mailto:support@dbarc.com>

Web Site: <http://www.dbarc.com>

Oracle is a trademark by Oracle Corporation. Powerbuilder is a trademark by Sybase Corporation. UNIX is a trademark of UNIX Systems Laboratories. Windows, Windows 95, Windows NT, Windows 2000, Windows XP, Windows Vista and Windows 7 are trademarks of Microsoft Corporation.

This document is an unpublished work containing confidential information owned by DATABASE Softwares. This document and all attached information thereof are the property of DATABASE Softwares and may not be copied, used, or transferred to anyone, either in part or in full, without the express consent of DATABASE Softwares.

The information collected by DATABASE Tuning has been obtained from the values collected in the shared memory and more particularly the SGA (SQL Global Area). DATABASE Tuning directly collects the indicators of performances from the ORACLE instances.

The SQL Queries, as well as its parameters and all performance information related to the execution context, (performance indicators' values at the Adaptive Server level), are automatically displayed by DATABASE Tuning and can be saved in an ASCII file or in an ORACLE database (whose model is provided in the appendix). Thanks to DATABASE Tuning, it is possible to consult the performance incidents in real time, as soon as they are detected, including the SQL Statements that are in a pending state.

DATABASE Tuning is an application that has been developed using Sybase PowerBuilder and Microsoft Visual C++. It can be deployed on any Windows platform.

For performance monitoring purposes, DATABASE Tuning captures the SQL Statements in a pending state exceeding a preset performance threshold and it does so in real time. DATABASE Tuning makes it possible to analyze each SQL statement captured. Impact depends on preset performance thresholds. Thanks to DATABASE Tuning, it is possible to detect the database design's mistakes and the applications' coding errors before they create major problems during the production phase. DATABASE Tuning makes it possible to pinpoint the SQL Queries that have a real impact on the ORACLE instance's performance. Thanks to DATABASE Tuning's off-set design, the impact on the performance of monitored systems is linked to a preset performance threshold.

For optimization purposes, DATABASE Tuning allows database administrators and project managers to instantly locate ORACLE instance's performance problems that are caused by heavy load in the execution of SQL statements.

For quality control and load tests purposes, DATABASE Tuning analyzes the performance of all your applications, whether they are being developed or in a pre-production stage. In the production context, DATABASE Tuning captures the SQL Statements exceeding a preset performance threshold. Along with load-test tools, DATABASE Tuning makes it possible to test your applications in an environment mimicking production conditions and also to identify the causes of saturations related to SQL statements.

For migration purposes, DATABASE Tuning is essential to performing migrations from one ORACLE release to another. DATABASE Tuning captures critical SQL Queries sent by your applications in the current ORACLE release. The results can then be compared with the results of another monitoring session in the new ORACLE release.

DATABASE Tuning's Technical Support

Database Architects
"Architecting for the Future"

Services offered | Products | Resources | Company | Partners | Events | Contact

Database Architects global consulting team has delivered hundreds of large scale, high impact database data architecture, high availability and performance and tuning engagements.

Our strengths are:

- Database Design and Data Analysis/Modeling
- Distributed Data Architectures/Data Replication
- Database Encryption/Security Frameworks
- High Availability
- Performance & Tuning
- Data Standards and Governance
- Delivery on time and within budget
- Extreme Quality of deliverables
- Knowledge Transfer and Education

Publications

Products

[SQL SuperVision](#)
[DATABASE Tuning](#)

Technical Support Documents

Contact Database Architects: you can contact us here or email us

[USER MANUAL](#) | [Installation Instructions](#)

Technical Support Request

Contact Database Architects team

Firstname:

Lastname:

Company:

Email:

Request:

Message:

The Database Architects Team

News

- Nov. 2-5, 2009 PA 8 8 Summit 2009 Seattle
- Aug. 20, 2009 MS SQL User Group San Diego
- Jul. 27, 2009 Database Tuning version 4.0 available for 30-days Free Trial Website
- Jul. 27, 2009 SQL Supervision version 4.0 available for 30-days Free Trial Website
- Jul. 27, 2009 SQL Shot version 4.0 available for 30-days Free Trial Website
- Jul. 17, 2009 UFE West USA Job/Career Solidarity Initiative SF - Bay Area
- Jul. 1, 2009 Interested in reducing your IT costs? Services
- Jun. 12, 2009 PA 8 8 Summit 2009 - Paul Bertuool Speaker! Seattle
- Jun. 6, 2009 SQL Saturday Portland
- Jun. 6, 2009 SQL Shot Leunoh at SQL Saturday Portland

[>more...](#)

Related

- P&T Tools
- Events
- Performance & Tuning
- Services Offerings
- Books
- White Papers
- Presentations
- Partners
- Clients
- Customer Testimonials
- Case Studies
- Management Team

Sign up for our newsletter

Be the first to hear about new products, events and Database Architects news.

Going to <http://www.dbarchitects.com/?page=support> you will have access to the Database Softwares' support web-page. You can then, send us any question about DATABASE Tuning licensing, Technical Support or ORACLE performance issue. You can also send us directly an email to support@dbarchitects.com.

Enhancements featured in this 4.2 release

This new 4.2 release makes it possible to define the frequency of performance information's snapshots related to queries and metrics of ORACLE instance. These frequencies are configured in milliseconds, meaning that it is now possible to define a frequency of less than one second. Although the access method has been optimized, we would like you to note that the impact on the monitored instance is proportional to the frequency selected. We therefore strongly recommend that you avoid to set up the frequency below 500 ms on a production instance. For testing and for developments, it is preferable to use a high speed snapshot. In such a situation you need to keep an eye on the activity of the monitored instance so as not to pollute the events being observed.

From installation to monitoring

Please follow the steps below to set up and configure DATABASE Tuning.

Architecture and prerequisites of DATABASE Tuning

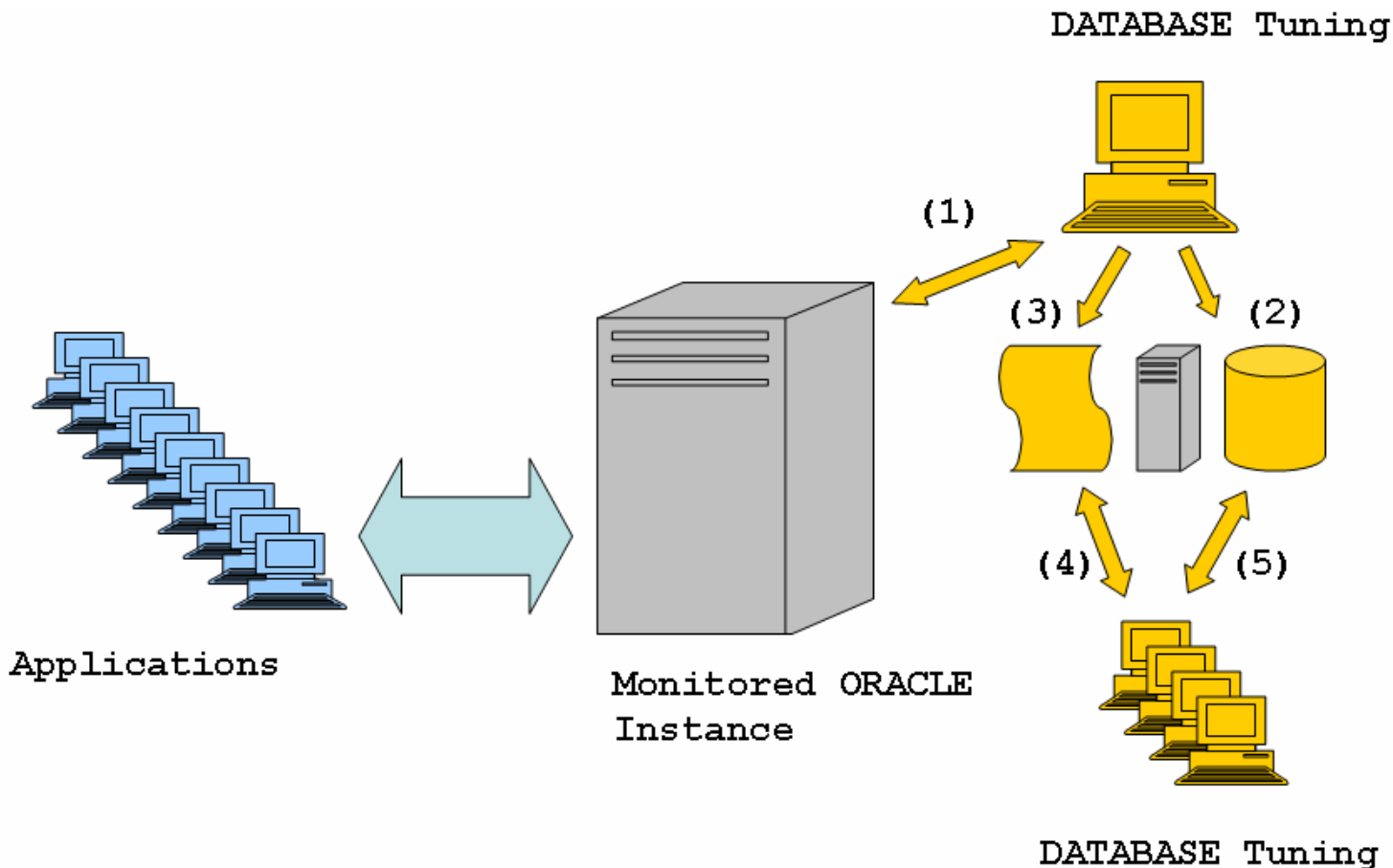
Installation

License key registration

How to start and stop monitoring

The setting of the network connections do not come with the DATABASE Tuning package. Please have your Database Administrator perform these operations beforehand. Although these settings are platform-specific, please find below the information needed to set up DATABASE Tuning.

Architecture and prerequisites of DATABASE Tuning



- (1) DATABASE Tuning connects, using an ODBC datasource, to the ORACLE instance you want to monitor. Thus, you need to create an ODBC datasource for the targeted instance. You must use ORACLE ODBC driver of ORACLE Client 10g and later.
- DATABASE Tuning connects to another ORACLE instance. This connection is used to write (2) or to read (5) monitoring results into an archive database. These both connections use ODBC as well.
- (3) (4) DATABASE Tuning makes it possible to write and read monitoring results in a flat file.

Monitored ORACLE instance

All ORACLE 8i+ releases installed on any platform are supported by DATABASE Tuning. However, one parameter needs to be set, when you create a session of monitoring, DATABASE Tuning checks that the parameter «TIMED_STATISTICS» is set to 'TRUE'. If it is not, DATABASE Tuning works normally, except for performance indicator 'CPU time'.

Client applications

There are no prerequisites for the latter.

DATABASE Tuning release 4.2

To install DATABASE Tuning you will need to have a PC workstation running with Microsoft Windows 9x/NT/2000/XP/VISTA/7 and ORACLE Client release 10g+. DATABASE Tuning will require a minimum of 25 MB free disk space. This can be located on the PC's hard drive or on a network's shared disk drive.

Application data

The files used by DATABASE Tuning are stored in the directory %APPDATA% (see below), as follows.

See also:

%APPDATA%

«.MNT» files

«.XLS» files

«.TXT» files

«.FLT» files

«.GPH» files

«DATABASE Tuning.log» file

«.tmp» files

«.sql» files

«.mon» files

%APPDATA%

To display the name of that directory, run the DOS command:

Echo %APPDATA%

«.MNT» files

These files contain performance data collected by DATABASE Tuning. These files are stored by default in the «%APPDATA%\DATABASE Tuning\Data Files» directory.

«.XLS» files

By default, the data exported to Microsoft Excel spreadsheets are stored in the «%APPDATA%\DATABASE Tuning\Data Files» directory.

«.TXT» files

By default, the data exported to text format files are stored in the «%APPDATA%\DATABASE Tuning\Data Files» directory.

«.FLT» files

By default, the filters' configuration files are created in the «%APPDATA%\DATABASE Tuning\cfg\flt» directory.

«.GPH» files

By default, the graphs' configuration files are stored in the «%APPDATA%\DATABASE Tuning\cfg\gph» directory.

«DATABASE Tuning.log» file

The DATABASE Tuning's alertlog file is in the «%APPDATA%\DATABASE Tuning\log» directory.

«.tmp» files

The files are used to control background mode executions. They are created in the «%APPDATA%\DATABASE Tuning\log» directory.

«.sql» files

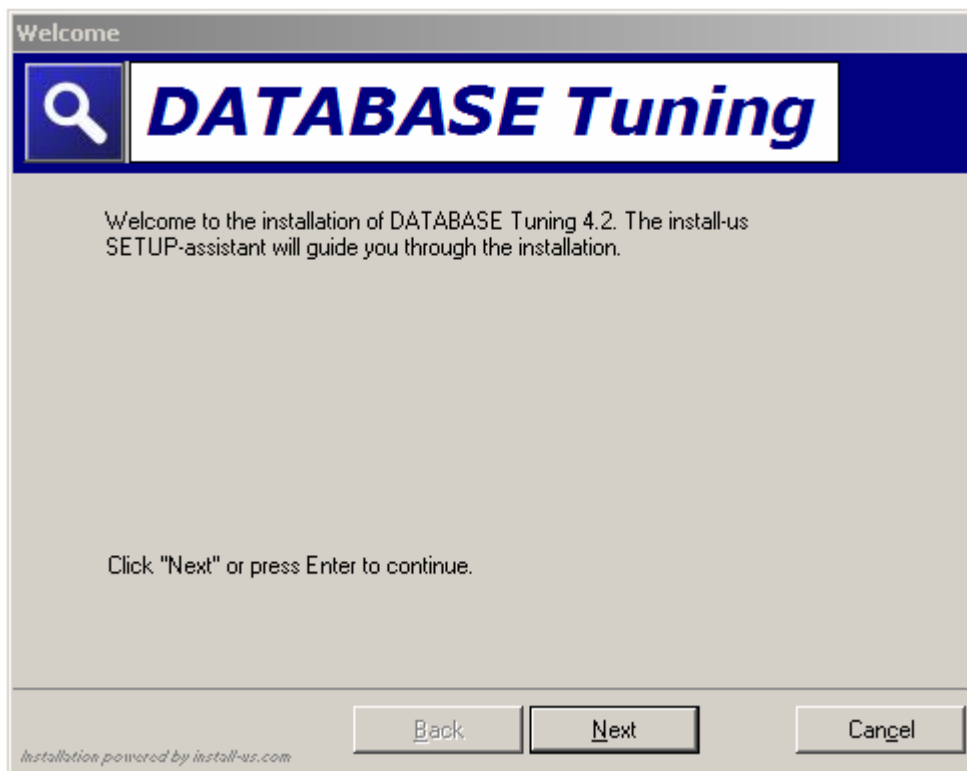
These files are used to grant a user to monitoring. The «grant_monuser_9i.sql» file is used for an ORACLE 9i instance monitoring. The «grant_monuser_10g+.sql» file is used for the ORACLE 10g and later instances. These files are located in the «%APPDATA%\DATABASE Tuning\sql» directory.

«.mon» files

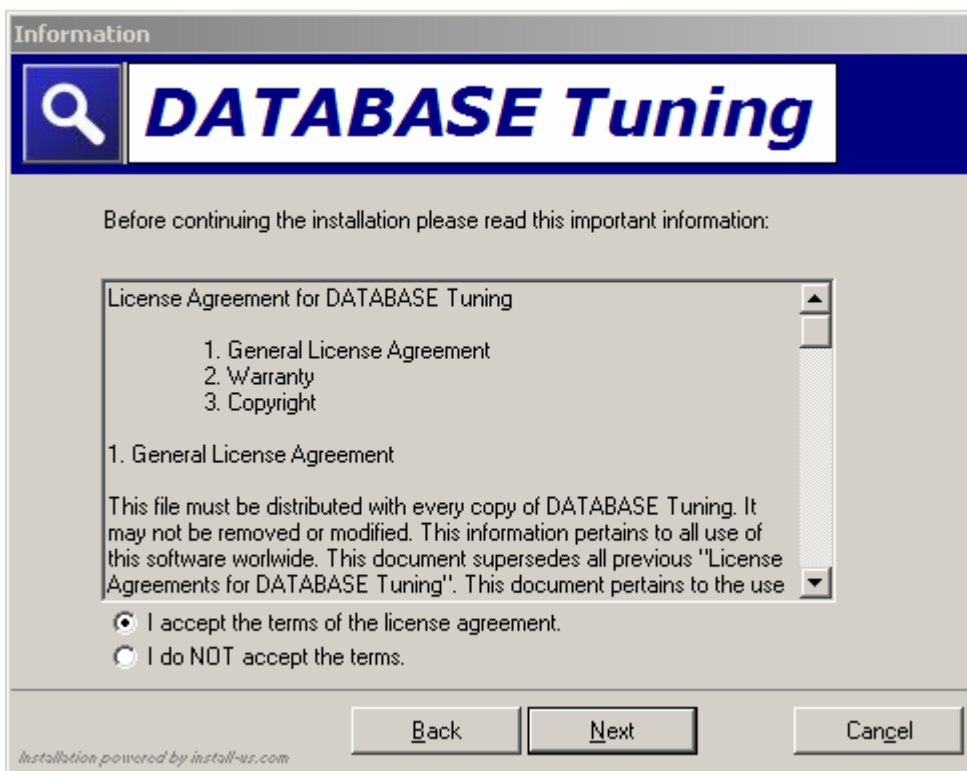
These files are used to collect the ORACLE instance's performance indicators. The «instance_8i+.mon» file is used for an ORACLE 8i and later instances. These files are located in the «%APPDATA%\DATABASE Tuning\cfg\mon» directory.

Installation

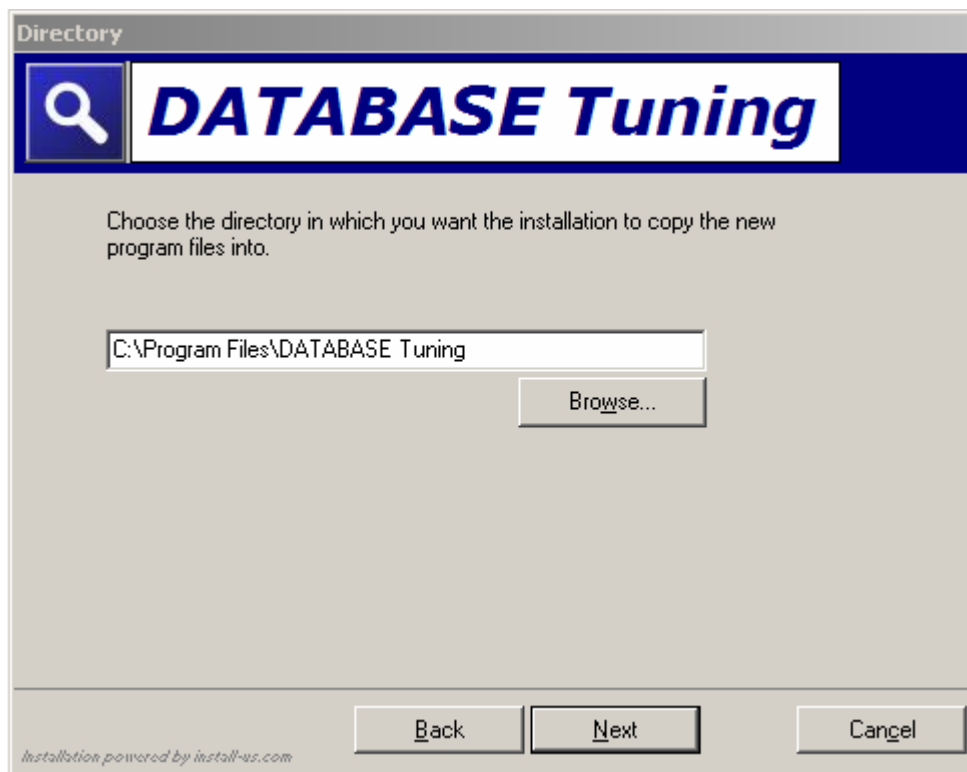
Insert your CDROM into the drive. The installation dialog box appears. Alternatively, you can run the following program «<CDROM drive>:\setup.exe».



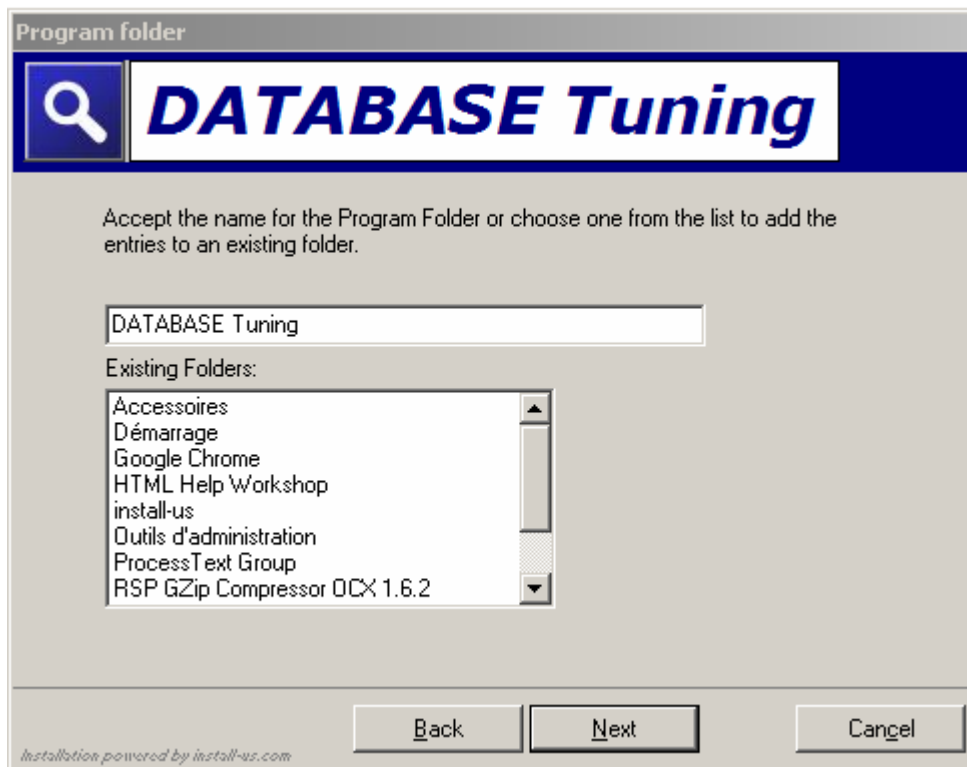
Set up starts.



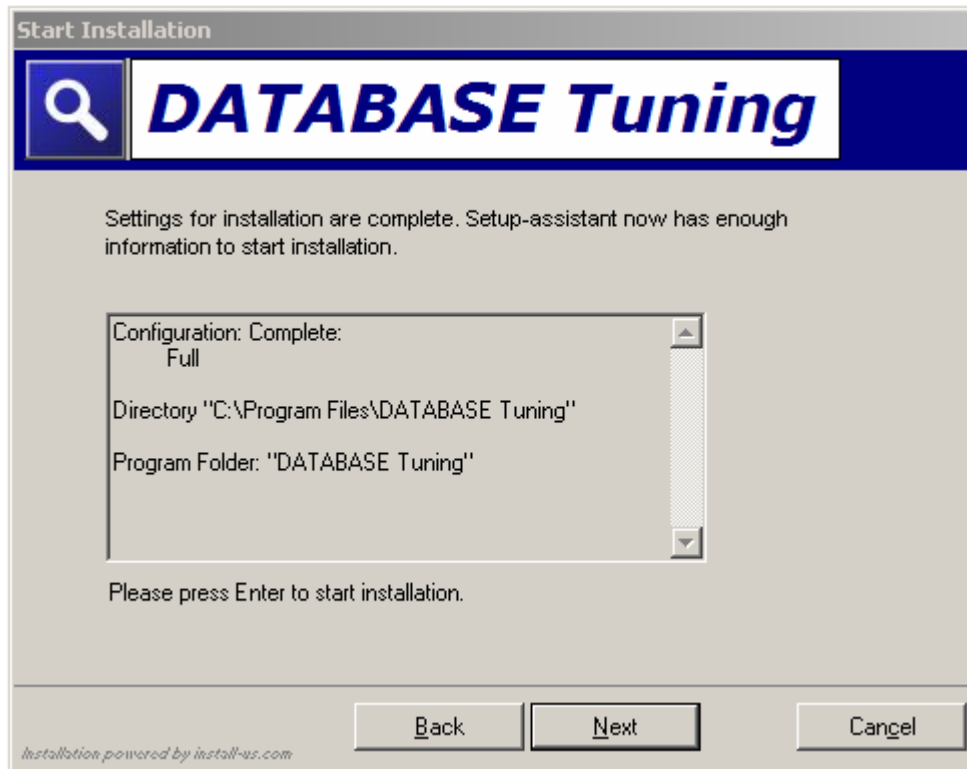
Enter DATABASE Tuning's installation directory. DATABASE Tuning needs to be installed on a computer that is able to connect to the Oracle instances (see paragraph on «Architecture and prerequisites of DATABASE Tuning»), or that can access DATABASE Tuning's files or archive database.



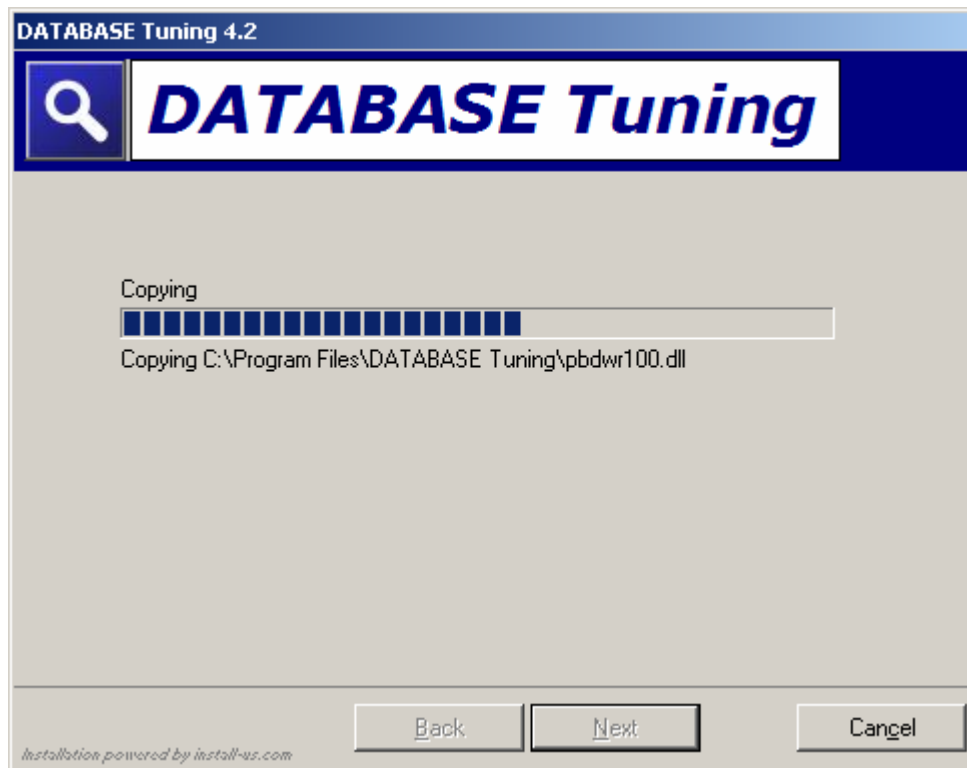
Next, enter the folder where you want to store DATABASE Tuning.



The installation program displays a validation window.



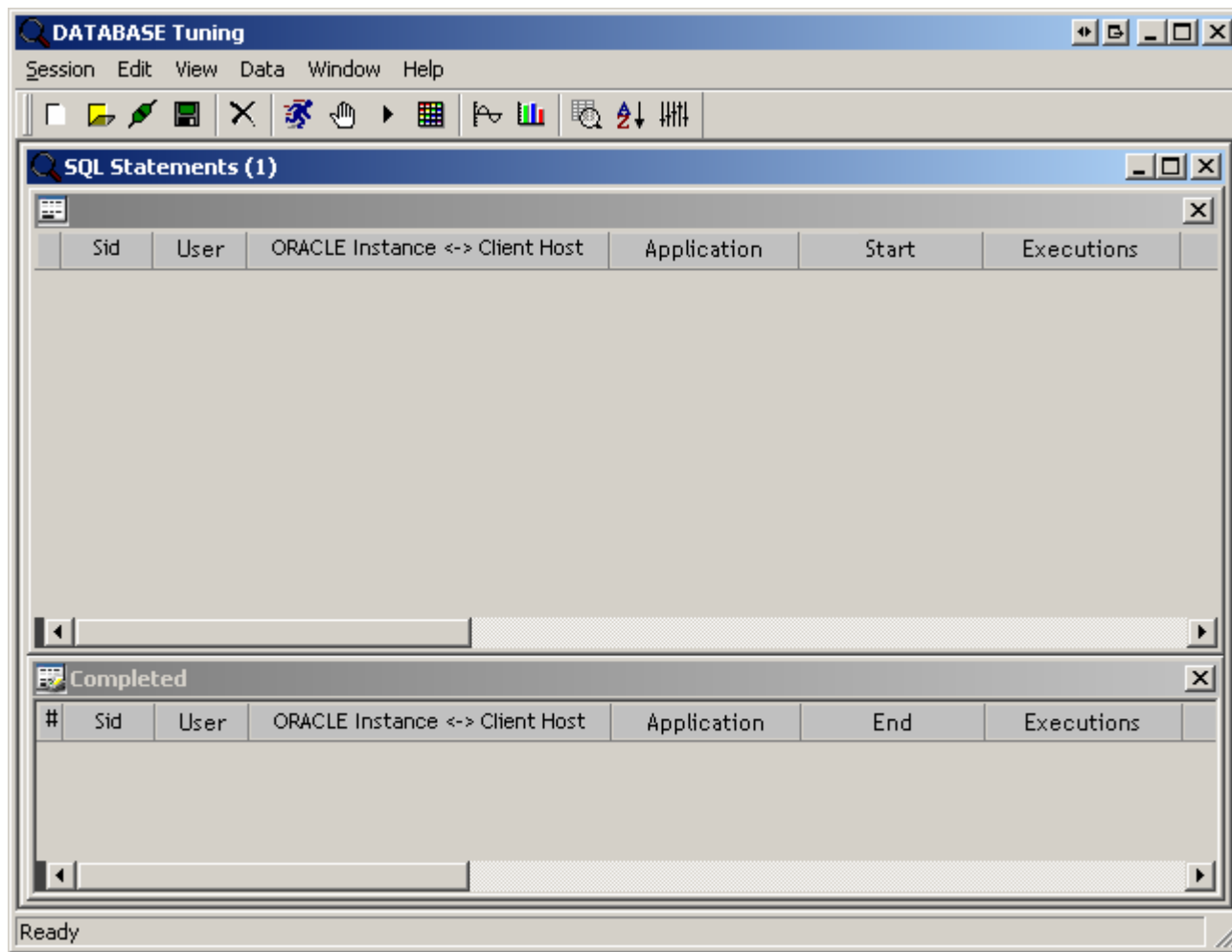
The installation proceeds automatically.



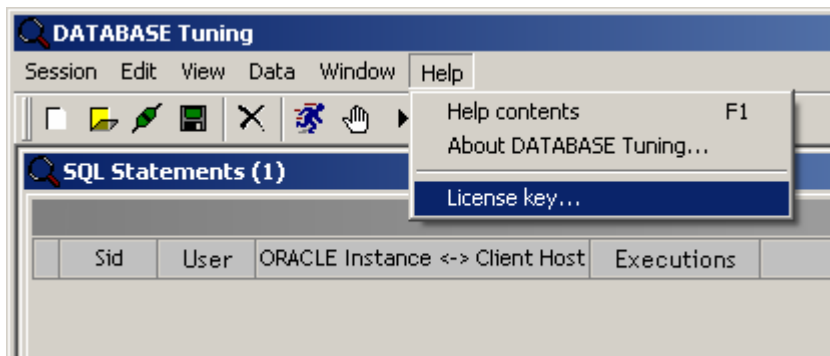


License key registration

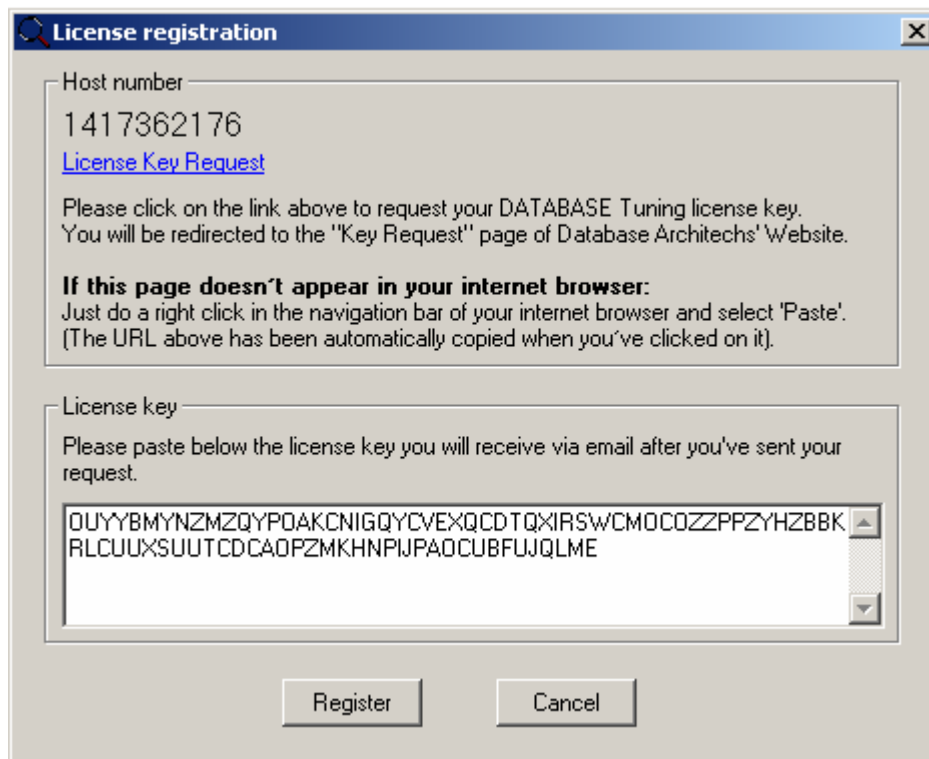
Proceed as follows: Go to the «Start - Programs - DATABASE Tuning» menu and, run the «DATABASE Tuning» program. The following window appears. Once started, DATABASE Tuning accesses its initialization file to use the display configuration. This configuration is saved each time you leave the program.



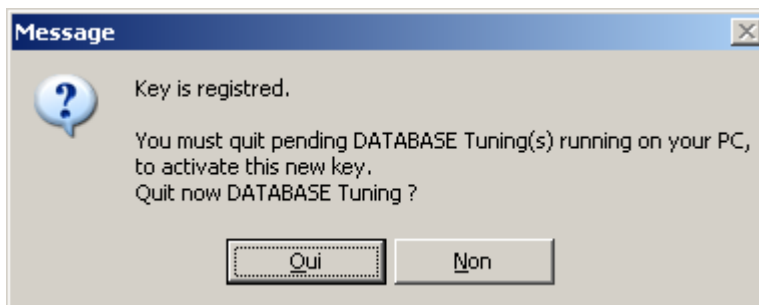
Following the installation, you are asked to register your license key. The latter manages monitoring connections to the ORACLE instances and authorizes a defined number of simultaneous sessions for a specific number of connected users, based on the type of license purchased. To obtain your license key, choose the «Help - License key...» menu item.



The following dialog box appears. The program displays the PC's identification (host ID). You will need to forward it to [DATABASE Tuning's Technical Support](#), so they can give you your personal DATABASE Tuning's license key. Next, type your key in the «License key» edit box and click the «Register» button.



A validation dialog box appears. Quit the program to allow DATABASE Tuning to recognize the key.

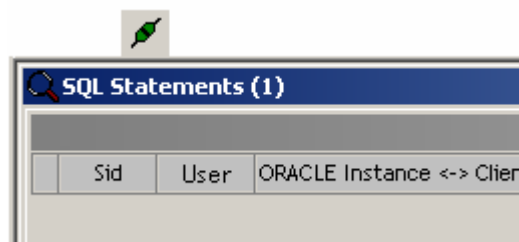
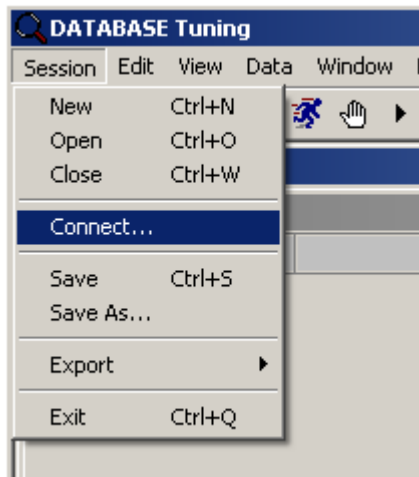


How to start and stop monitoring

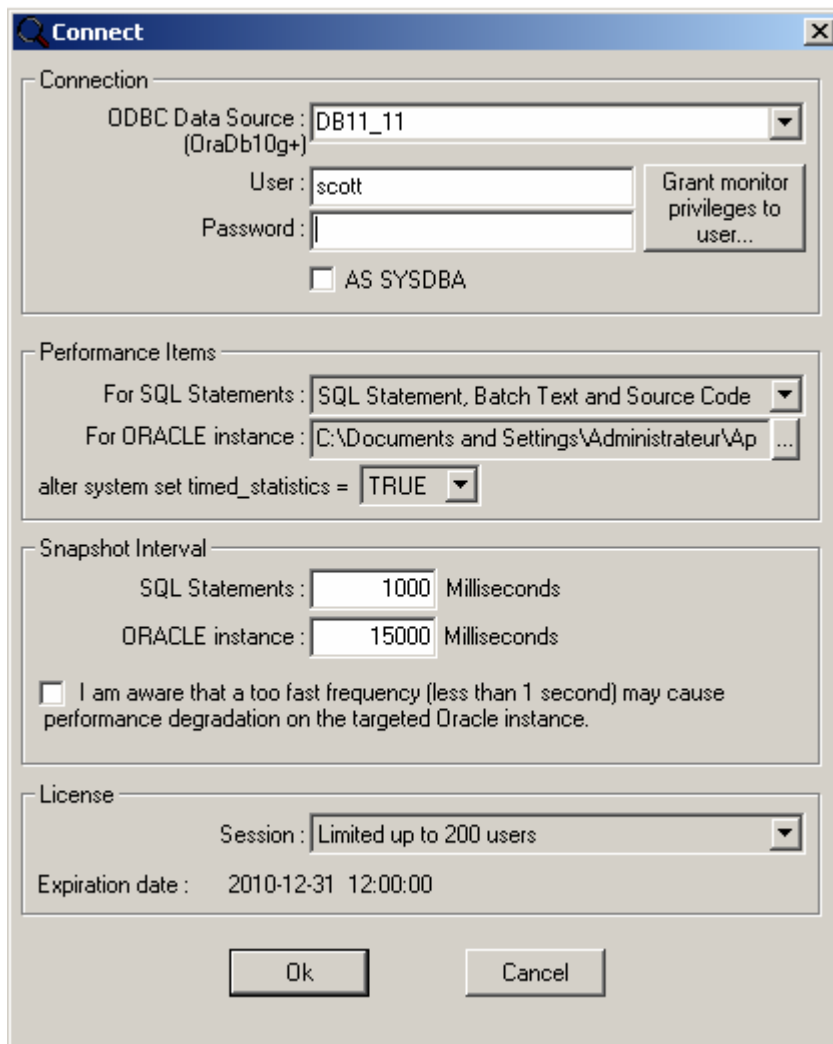
See also:

- The properties of the connection
- The performance items
- Snapshot intervals
- License
- The activation of the session
- Verification that DATABASE Tuning is properly set up
- Monitoring session's shutdown
- Grant monitor privileges to a user

To start a monitoring session on the targeted ORACLE instance, select the «Session - Connect...» menu item. Alternatively, you can click on the «Connect/Disconnect» menu icon.

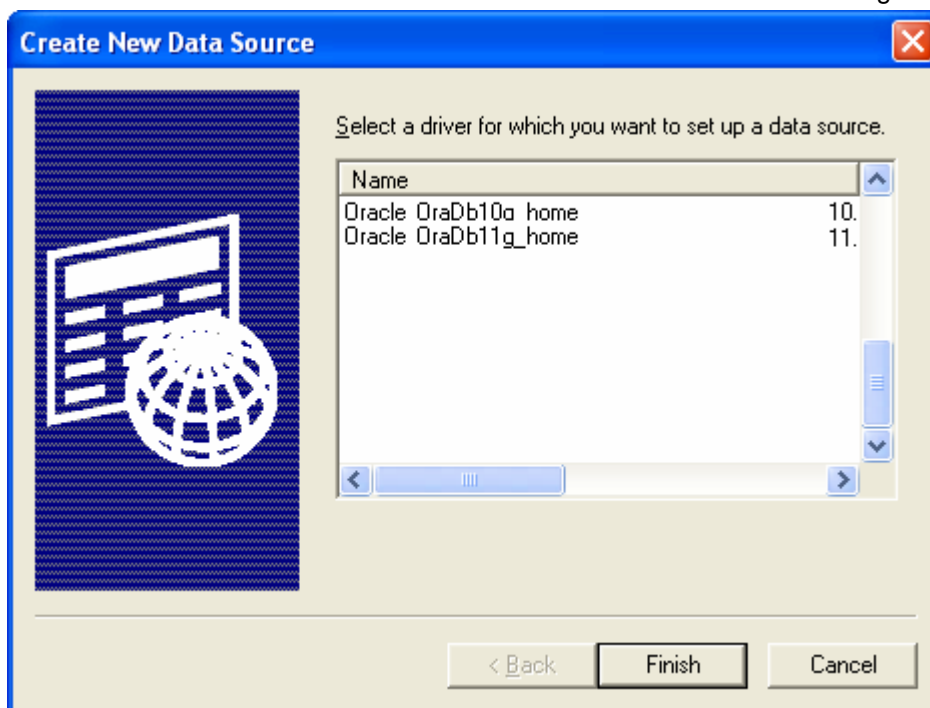


The connection dialog box appears.



The properties of the connection

ODBC datasource is the ORACLE database that you want to connect to for monitoring. The DSN configuration must be the following. The driver must be ORACLE release 10g or later. You can only use ORACLE 10g driver to connect to an ORACLE instance release 8i.



You must connect using a granted user, or sysdba user. The monitoring needs a connection with a user granted SYDBA for releases older than ORACLE 9i. Starting from ORACLE 9i, you can grant a user for monitoring using the «Grant monitor privileges to a user...» function, to do that, you will click on the button in question from the connect dialog box.

The performance items

DATABASE Tuning uses three monitoring levels (see paragraph on «Contents of the session's report»). The least CPU load mode is «SQL Statement» and the most is the full mode : «SQL Statement, Batch Text Batch and Source Code». According to your needs for real time monitoring and characteristics of your environment (production, pre-production, development) you will select the corresponding accuracy level. Contact DATABASE Softwares to obtain all needed information regarding this subject.

You select a pre-formatted file containing SQL queries built to collect ORACLE instance's performance values. For further information, see paragraph on «How to configure the indicators».

The setting to 'TRUE' is not mandatory. If this parameter is set to 'FALSE', DATABASE Tuning will not be able to evaluate CPU load used by the SQL Statements and by instance's processes.

Snapshot Intervals

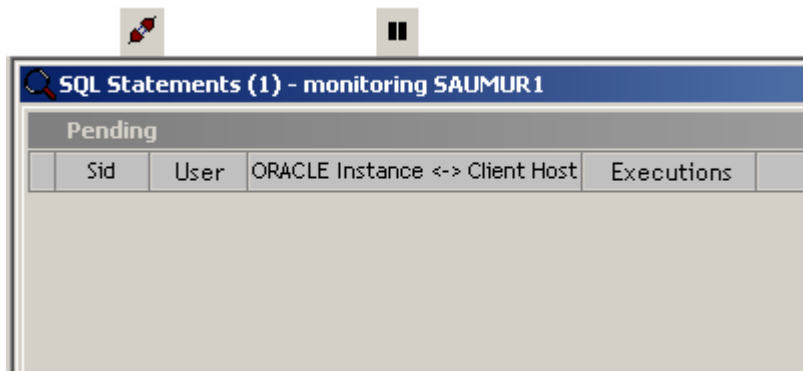
Enter the snapshot frequencies for both SQL statements and ORACLE instance, caution: too high frequency will cause performance degradation on monitored instance.

License

Choose the license to be used for this monitoring session. The contents of this list depend on the acquired license and change based on the currently active monitoring sessions. License (expiration date): temporary key for trial or lease. The purchased keys are updated on simple call or sending email approximately every 12 months.

The activation of the session

You click on the «ok» button. As soon as the session is connected, the window's title changes: SQL Statements (1) monitoring < name of the targeted SQL Server > <last snapshot datetime> <scan interval>, the «Connect/Disconnect» menu icon changes to red to indicate that the session is in progress and the «Refresh» menu shows that the graphical user interface is displaying information in real time. The child window's upper part (See the title of this window) changes to a «Pending» state.

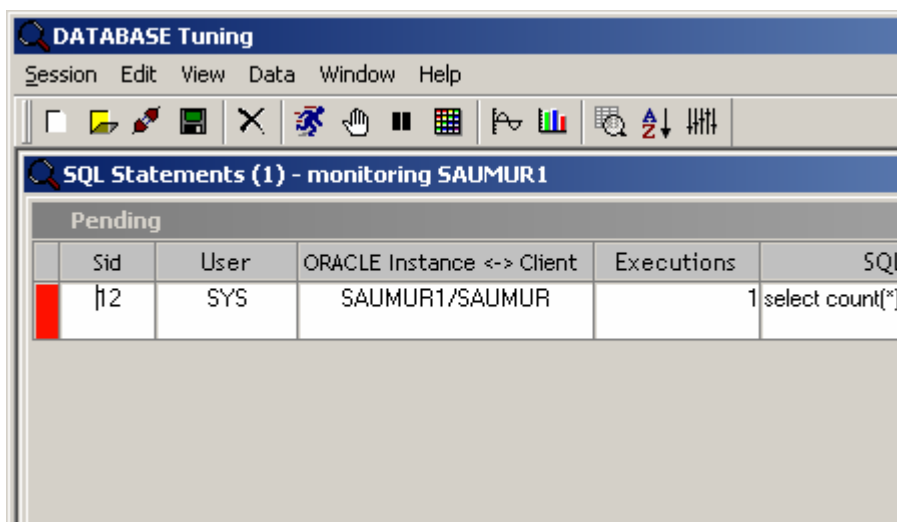


Verification that DATABASE Tuning is properly set up

The session is now connected and the Automatic Refresh is activated, SQL Statements appear in the session's report (based on the SQL activity of the targeted instance). To make sure that the monitoring is properly set up, simply connect to the targeted instance and run a SQL Statement having a longer execution than the selected snapshot for SQL Satatements, for example :

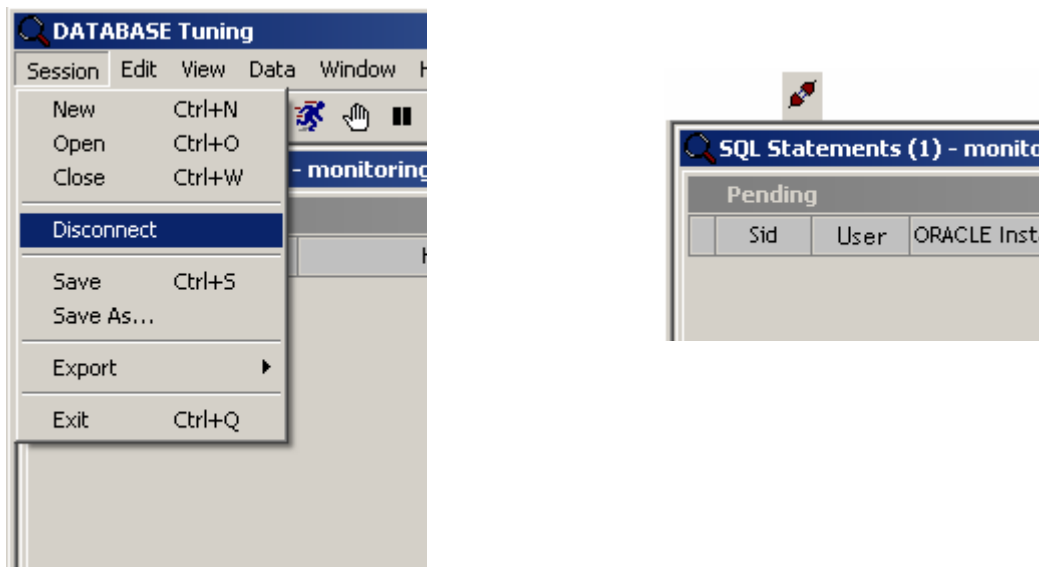
```
select count(*) from dba_source t1, dba_source t2;
```

After a few seconds, you see appearing this SQL Statement in the session's report. This indicates that the monitoring is operational.

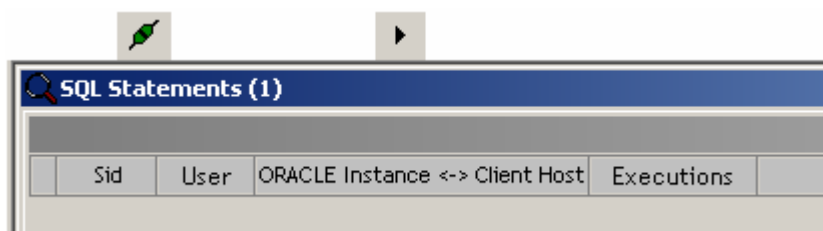


Monitoring session's shutdown

To stop a monitoring session, click on the «Connect/Disconnect» menu icon, or select the «Session - Disconnect» menu item.

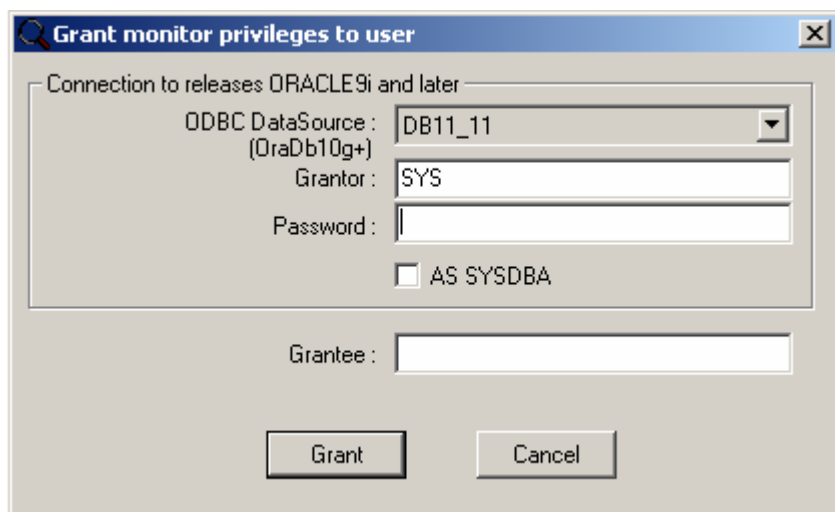


The window's title changes to display the «SQL Statements (1)», which means that the session is no longer active. The «Connection» and «Automatic Refresh» menu icons are deactivated state and the title located in the upper part of the screen is no longer visible.

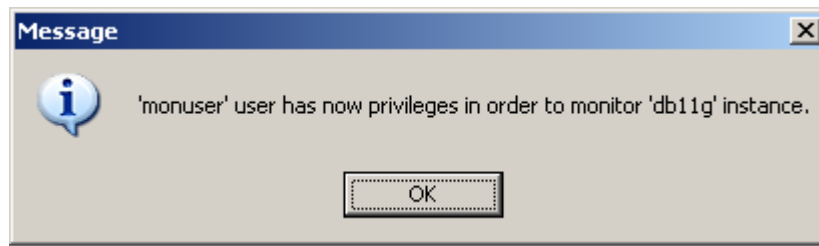


Grant monitor privileges to a user

For versions prior to Oracle 9i, only the «SYS» user can audit an instance. Therefore, this feature will be used only for versions Oracle 9i and later. From the connection window, you click the button "Grant monitor privileges to user...", the dialog box below asks for the user (grantor) and for the user receiving the authorization (grantee). The latter must have been created before. Monitoring SQL objects (tables and views) will be created in the schema of the user. You can view the privileges in question and edit the "%APPDATA%\DATABASE Tuning\sql\grant_monuser_9i.sql" or "...grant_monuser_10g.sql" file according to the version of the monitored instance.



Once the privileges are granted, the tool displays the confirmation message that you can use this user to create a monitoring session on the targeted ORACLE instance.



The best thing to do is to create a user, and to assign a dedicated tablespace resource. It is not necessary because there are no data generated on the target instance by the monitoring activity. Alternatively, you can assign the «SYSDBA» role to the monitoring user.

Monitoring SQL Statements

The captured SQL Statements appear in the session's report in real time. In the upper part of the window, quantitative information change with each passing interval until the end of the SQL Statement's execution. At this moment, the SQL Statement moves from the top part («pending») to the bottom part of the window («completed»).

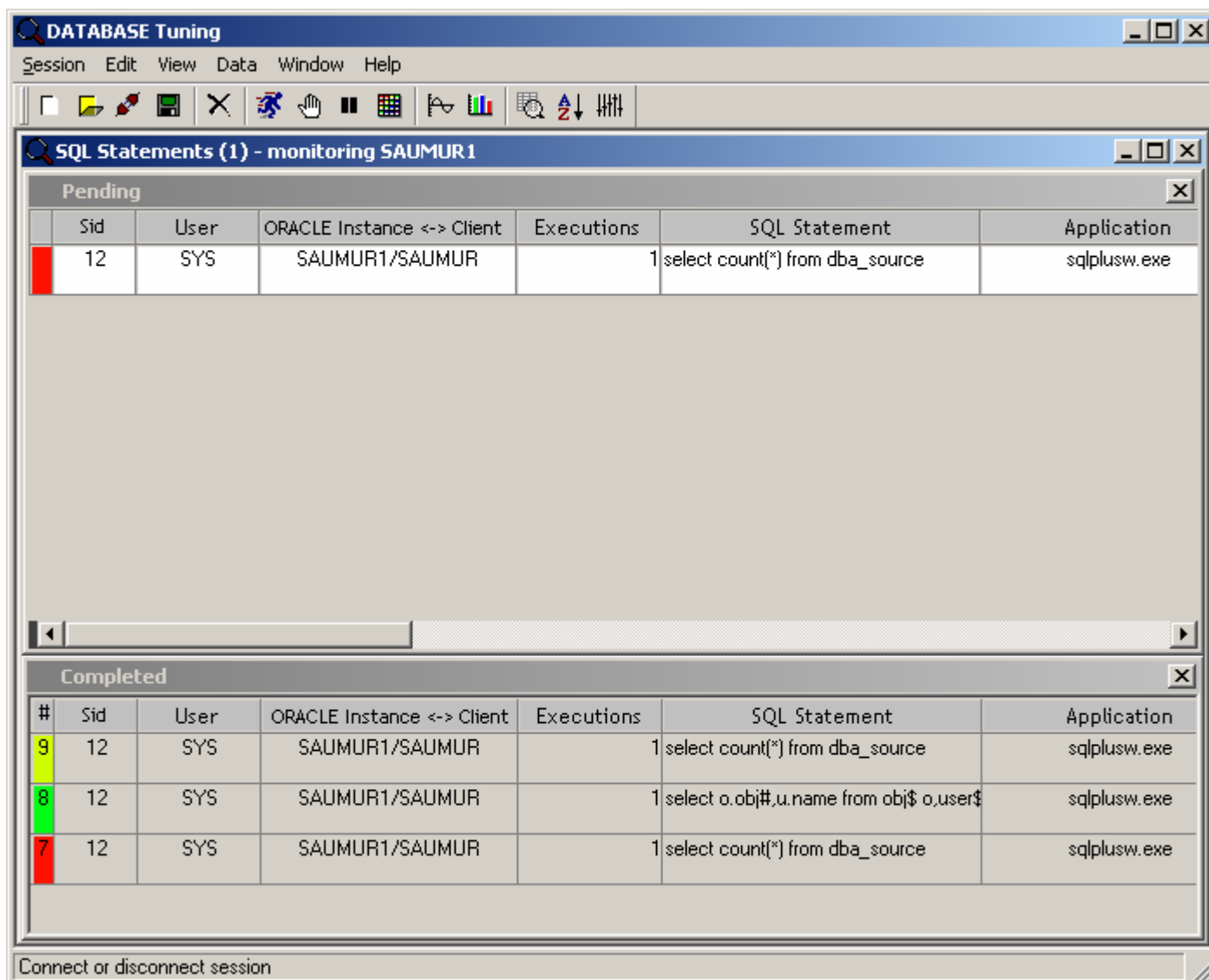
See also:

- How to start and stop monitoring
- Contents of the session's report
- SQL Statements' graphs
- Filters
- Display options
- Sorts
- New session


Contents of the session's report

See also:

- Detailed SQL Statement







The session's report contains two parts. The displayed SQL Statements located in the window's upper part are in a pending state. Once completed, these statements are moved to the lower part of the screen. When the option «Automatic

Refresh» is active (icon: ), the upper part of the window (SQL Statements that are in a pending state) is the only one that is enabled and the bottom part of the window (completed) cannot be selected. The same goes for the SQL Statements' graphs. To click on or scroll through the completed SQL statements' reports, you need to switch off the «Automatic Refresh» option. The monitoring remains active, but the captured pending SQL Statements are no longer refreshed, but are still saved in the background. Once you activate the «Automatic Refresh» option, the results saved in the background are immediately inserted into the session's report and all information regarding the ongoing SQL Statements becomes available again, in real time.

For each captured SQL Statement, the program displays the following information:

| | |
|----------------------------------|---|
| Sid: | The process identifier. |
| User: | User's name. |
| ORACLE instance <-> Client Host: | Instance's name, Client Host's name. |
| Application: | Application name. |
| Start: | The date and time of the pending SQL Statements. (These data vary according to the time zone of the Adaptive Server and they correspond to the beginning of the SQL Statement's execution). |
| End: | The date and time at which the execution of the SQL Statement ended. (These data vary according to the time zone of the Adaptive Server and they correspond to the completion of the SQL Statement). |
| Elapsed Time: | Time needed to execute the SQL Statement. |
| Batch Text: | The query sent by the client application. |
| SQL Statement: | The SQL Statement captured by DATABASE Tuning. This indicates the performance issue that has been identified. |
| Execution Plan: | The SQL Statement's execution plan. |
| Procedure - Trigger - Package: | The Database containing the stored procedure, the trigger or the function, followed by the procedure, the trigger or the function's owner name, followed by the procedure, the trigger or the function's name. When a SQL Statement is not part of a procedure, a trigger or a function (dynamic SQL), the character string «NoOwner/NoObject/NoType» is displayed. |
| Executions: | The number of executions of the SQL Statement (for example within a while loop). |
| Logical reads: | The quantity of data page reads (in data cache) consumed by the SQL Statement. |
| Physical reads: | The quantity of data page reads (on physical device) consumed by the SQL Statement. |
| CPU Time: | The computation time used by the SQL Statement (this value is equal to 0 during the execution of the SQL Statement. If the SQL Statement is part of a PL/SQL block composed of several SQL Statements, CPU usage is not known when it ends). |
| Locks : | The quantity of demands for locks requested by the SQL Statement that were not satisfied immediately. |

Monitoring (See Paragraph on «How to start and stop monitoring») and SQL Statements' Automatic Refresh displays (See Paragraph on «Contents of the session's report») work independently from each other. This means that it is impossible to view the details on the SQL Statements contained in the completed statements' report when the option «Automatic

Refresh» is active (icon: ) and when the «Completed» report is greyed out. However, it is possible to interrupt the refreshing task ( icon) without losing the results of the ongoing monitoring session. During the interruption, the captured SQL Statements are saved in the background and you can view the details of the SQL Statements contained in the reports are viewable. When the option «Automatic Refresh» is active, you can view the pending SQL Statements' details at any time, in real time, as was the case for the former releases. The  icon, located in the «View - SQL Statements - Pending» menu, opens up the session's report containing the pending SQL Statements. The  icon, located in the «View - SQL Statements - Completed» menu, opens up the session's reports containing the executed SQL Statements.

Each session's report can be split in two (at the left of the horizontal scrollbar). This enables you to view the pending SQL Statements' reports as well as the completed SQL Statements' reports. Each report comes with its own horizontal scrollbar. However, it is impossible to view synchronized columns in the window while scrolling horizontally.

Detailed SQL Statement

See also:
Detail, General

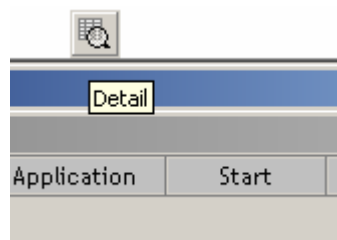
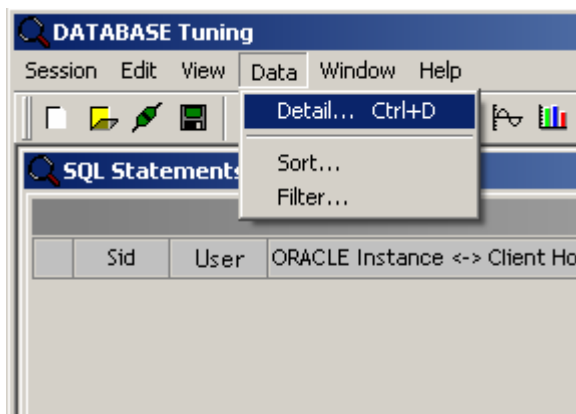
- Detail, SQL Statement
- Detail, Batch Text
- Detail, Execution Plan
- Detail, Procedure, Trigger and Package
- Detail, Instance's Performances
- Detail, Print

To see a detailed SQL Statement, simply click on a specific report's line, in the «#» column. To select the line, make sure it is highlighted. To unselect lines, simply click on any of the report's columns.

| # | Sid | User | ORACLE Instance <-> Client | Executions | SQL Statement | Application |
|----|-----|------|----------------------------|------------|--|-------------|
| 11 | 12 | SYS | SAUMUR1/SAUMUR | 1 | select o.obj#,u.name from obj\$ o,user\$ | sqlplusw.e |
| 10 | 12 | SYS | SAUMUR1/SAUMUR | 1 | select count(*) from dba_source | sqlplusw.e |
| 9 | 12 | SYS | SAUMUR1/SAUMUR | 1 | select count(*) from dba_source | sqlplusw.e |

Start/Stop read historical file

Once selected, you choose «Data - Detail...» menu item, or click on the «Detail» menu icon (cf. figures below), or use right click, the contextual menu appears, then choose the «Detail» menu item. You have also the possibility to double-click on any column of the chosen line.

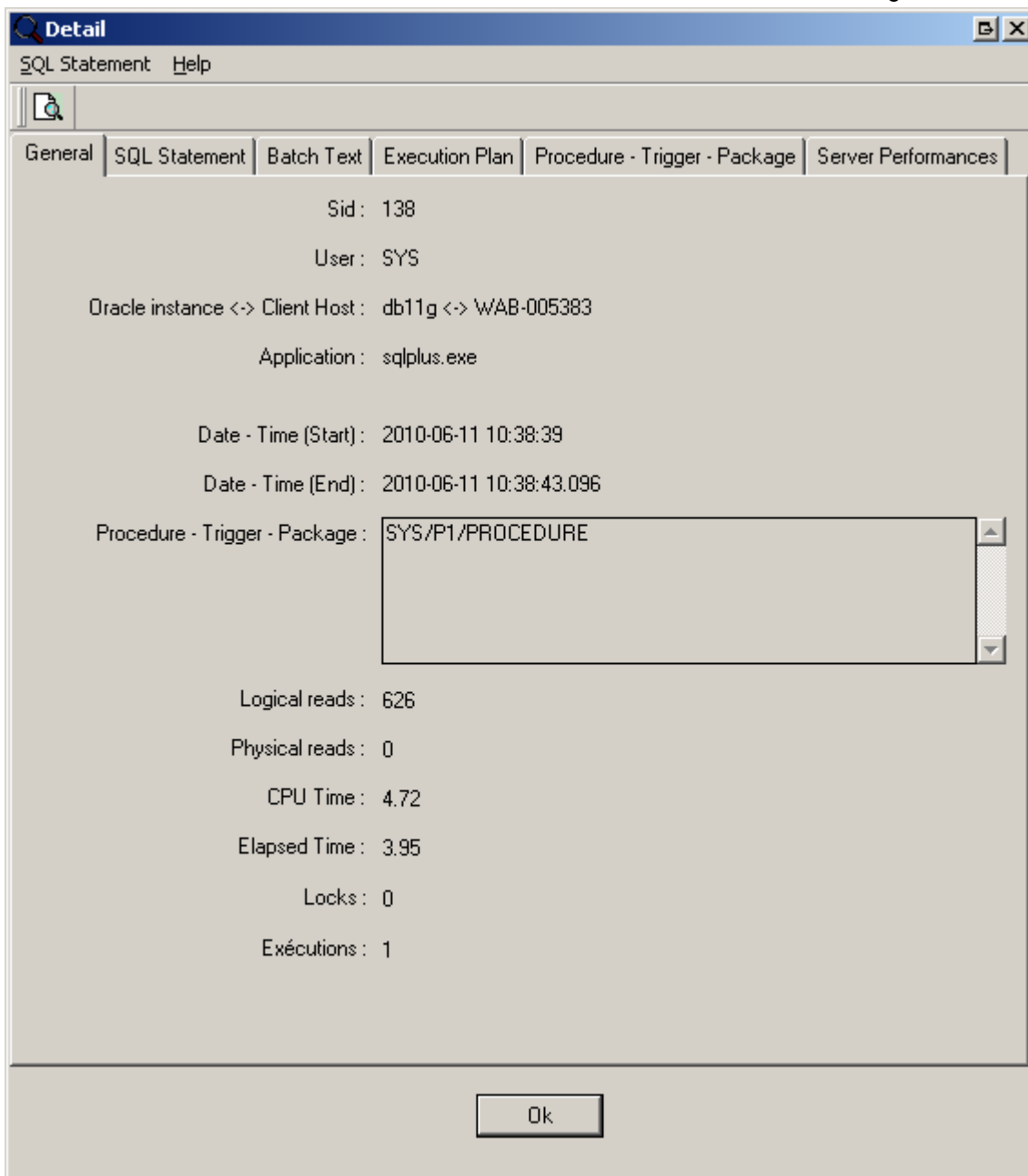


| | Executions | SQL Statement |
|-----|------------|--|
| 4UR | 1 | select o.obj#,u.name from obj\$ o,user\$ |
| 4UR | 1 | select count(*) from dba_source |
| 4UR | 1 | select co |

A window with six tabs is displayed.

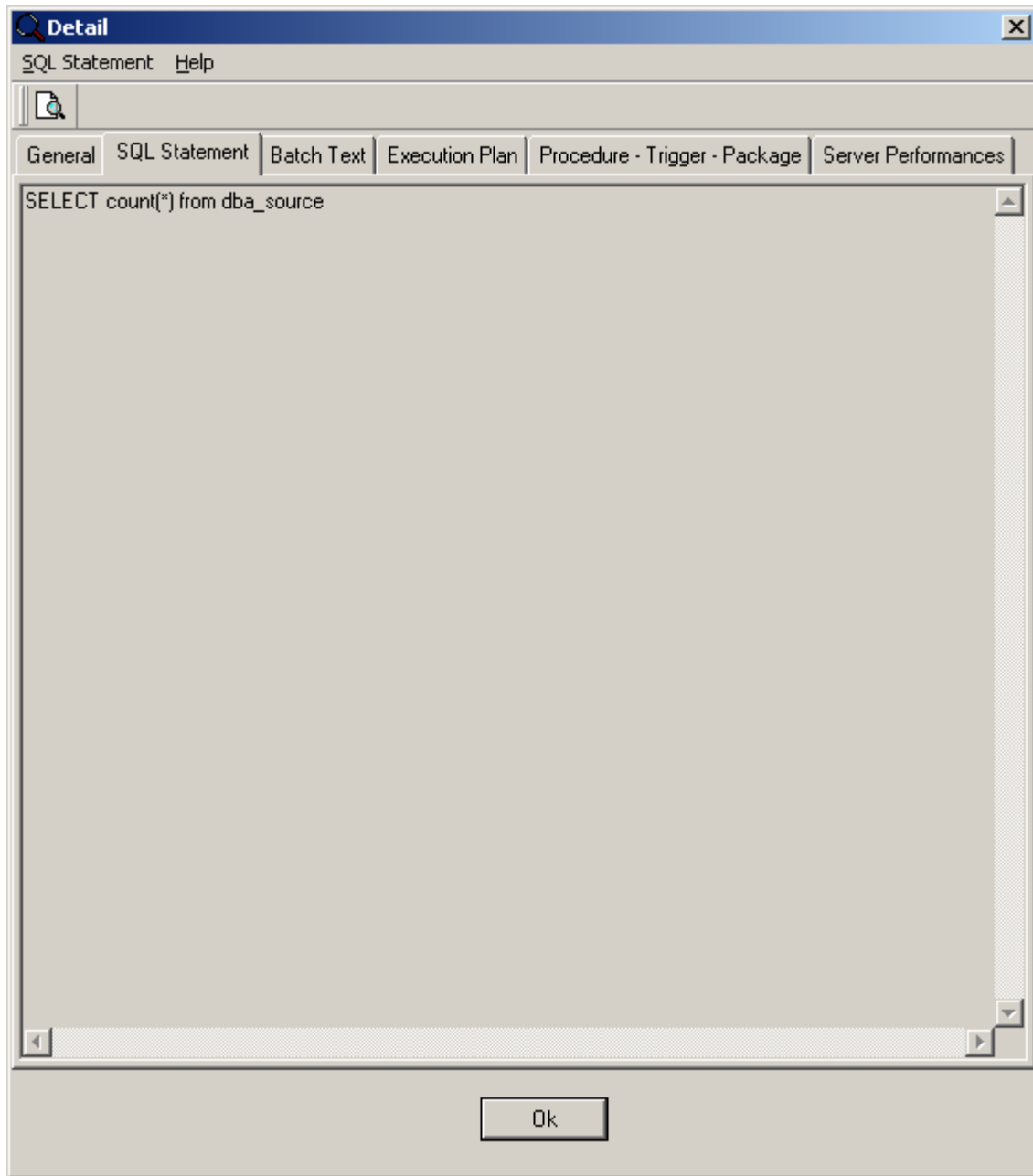
Detail, General

This window displays the following data: sid, user name, host name, application name, date and time of the execution's start, date and time of the execution's end, and statistics of execution. Moreover, it shows the procedure's name or the trigger's name (if the SQL Statement is part of a PL/SQL object). Whenever there is a dynamic SQL statement, the detail tab reads: NoOwner/NoObject/NoType.



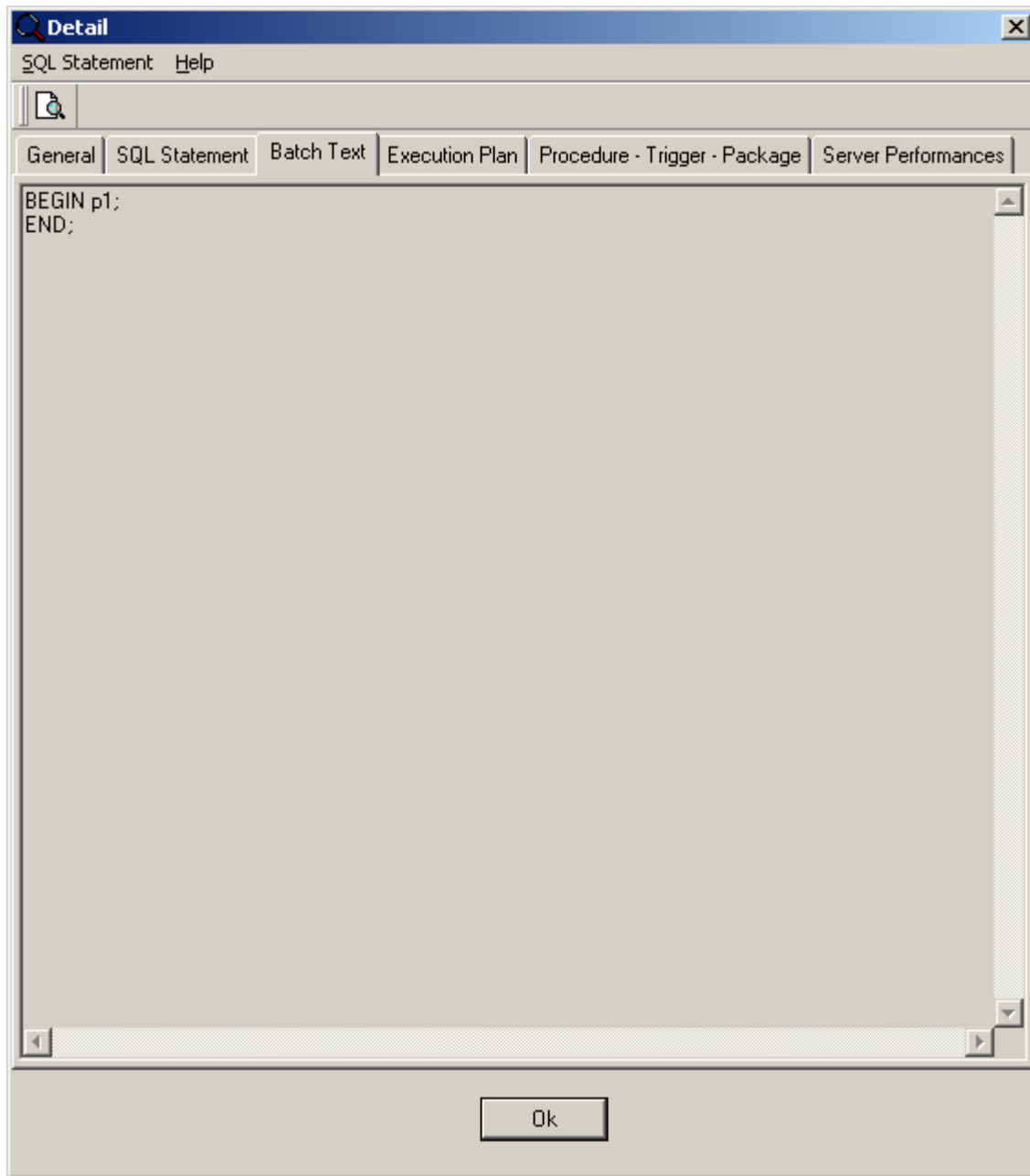
Detail, SQL Statement

The SQL Statement can either be part of a SQL batch sent by the client application, or part of a stored procedure or a trigger or a package. The execution duration of this SQL Statement exceeded the scan interval. This elapsed time threshold is triggered at the SQL Statement's level, not at the transaction's level. It is not set in motion by the duration for a stored procedure or of a trigger or of a package.



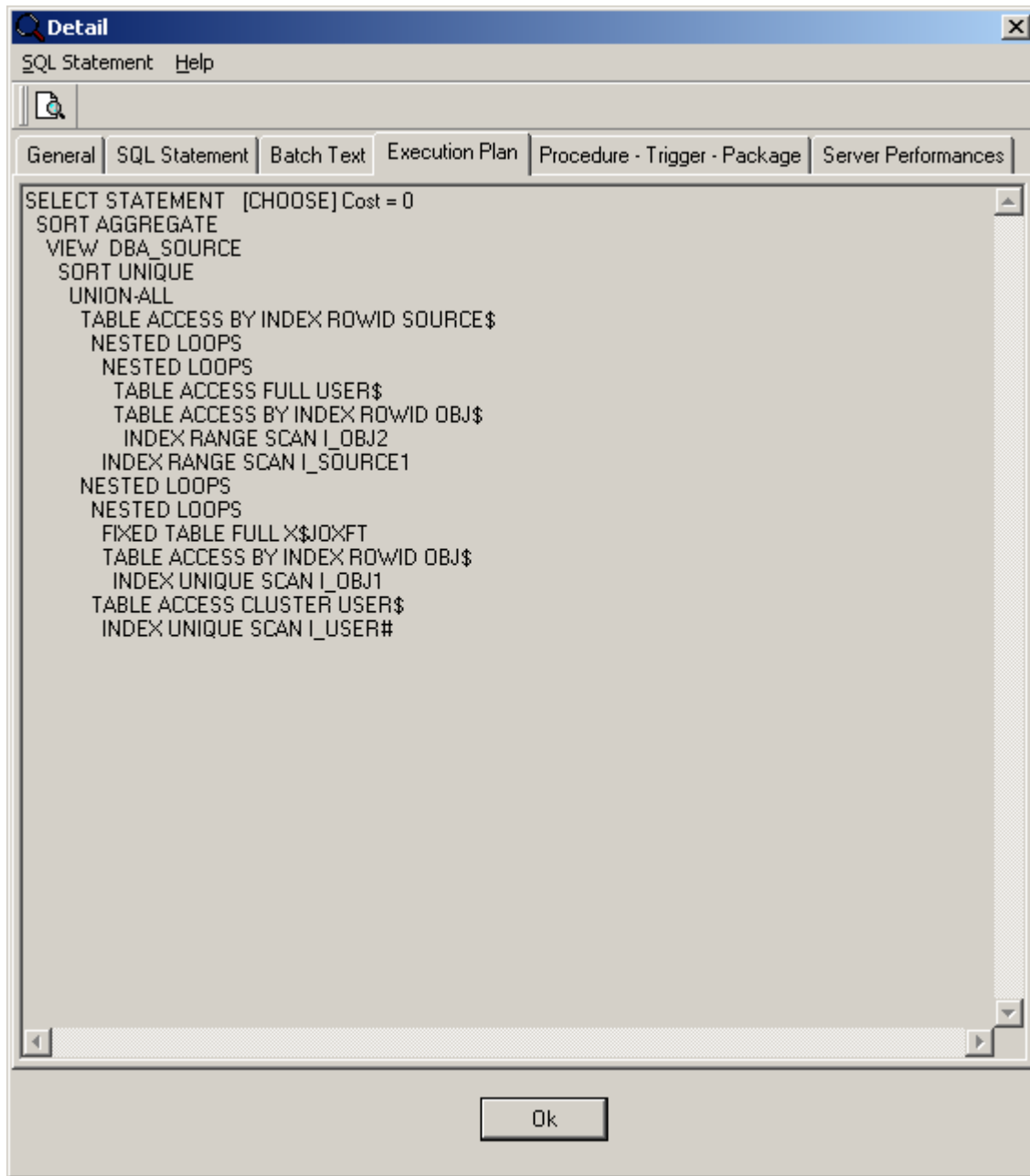
Detail, Batch Text

This is the SQL batch sent by the Client Application to the ORACLE instance. (It contains a stored procedure call as well as its parameters, SQL BatchText, RPC, cursors...).



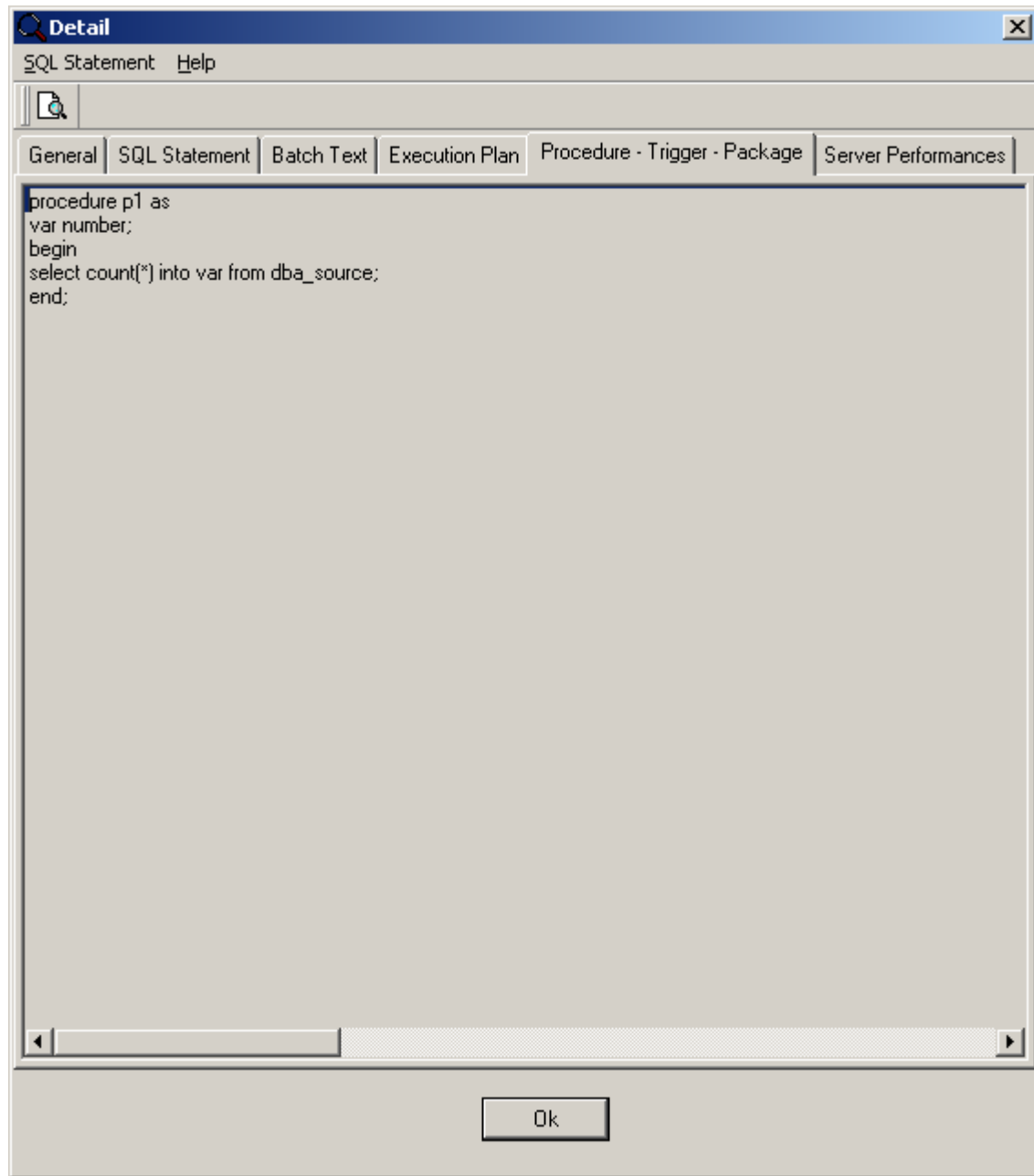
Detail, Execution Plan

This is the SQL Statement's execution plan as computed by the ORACLE instance's query optimizer.



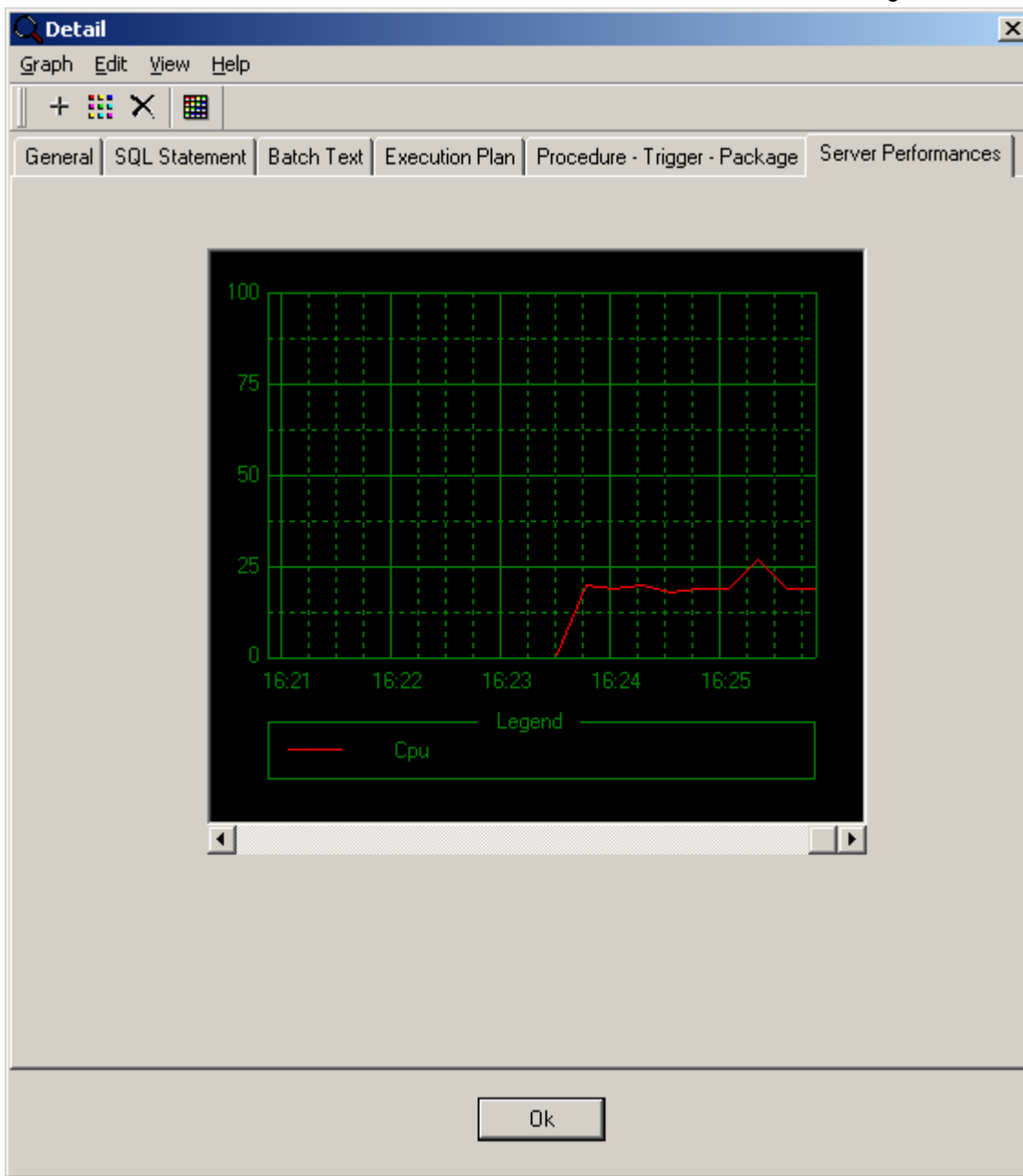
Detail, Procedure, Trigger and Package

This is the stored procedure, trigger or package source code containing the captured SQL Statement (provided the latter is part of a procedure or trigger or package). In the source code text, the captured SQL Statement is selected. Whenever the SQL Statement is not a part of an ORACLE PL/SQL procedure or a trigger or package, it is a part of the Batch Text, namely a dynamic SQL. When it happens, the tab displays the following character string: «NoOwner\NoObject\NoType».

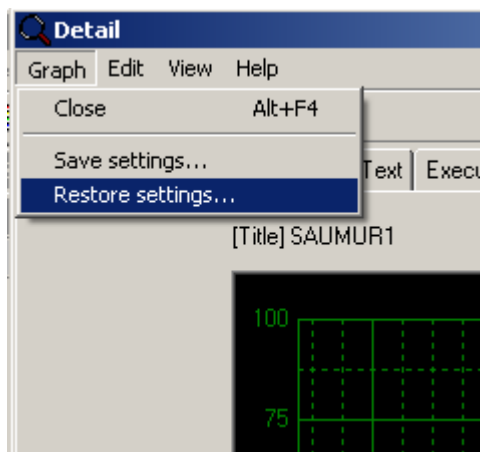


Detail, Instance's Performances

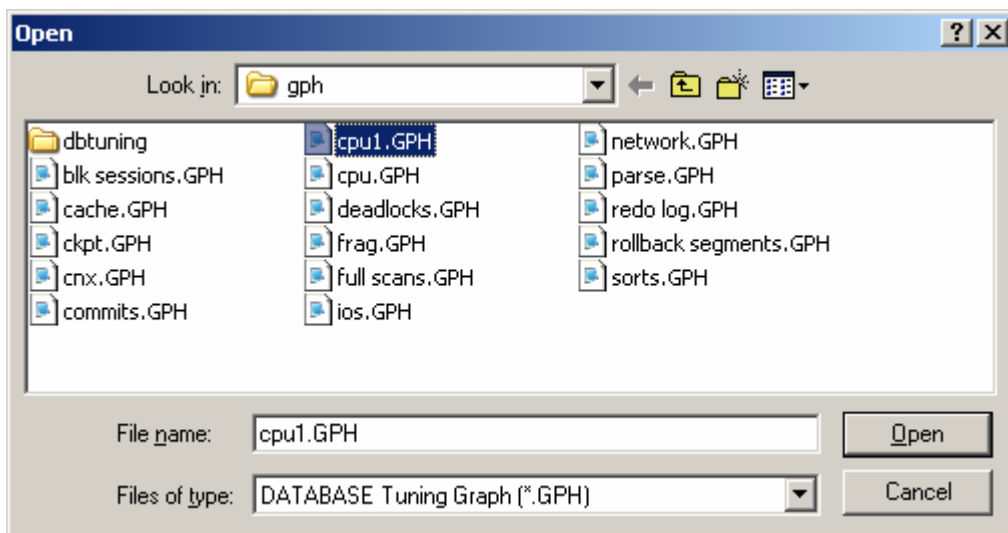
This tab displays the graph related to the chosen ORACLE instance's performance indicator. These are the values that have been collected during the execution of the current SQL Statement.



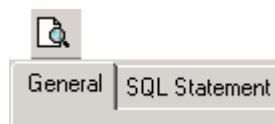
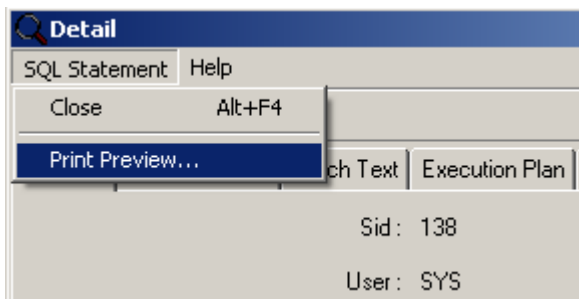
When selecting the «Graph - Restore settings...» menu item, you can view the different performance indicators that are available.



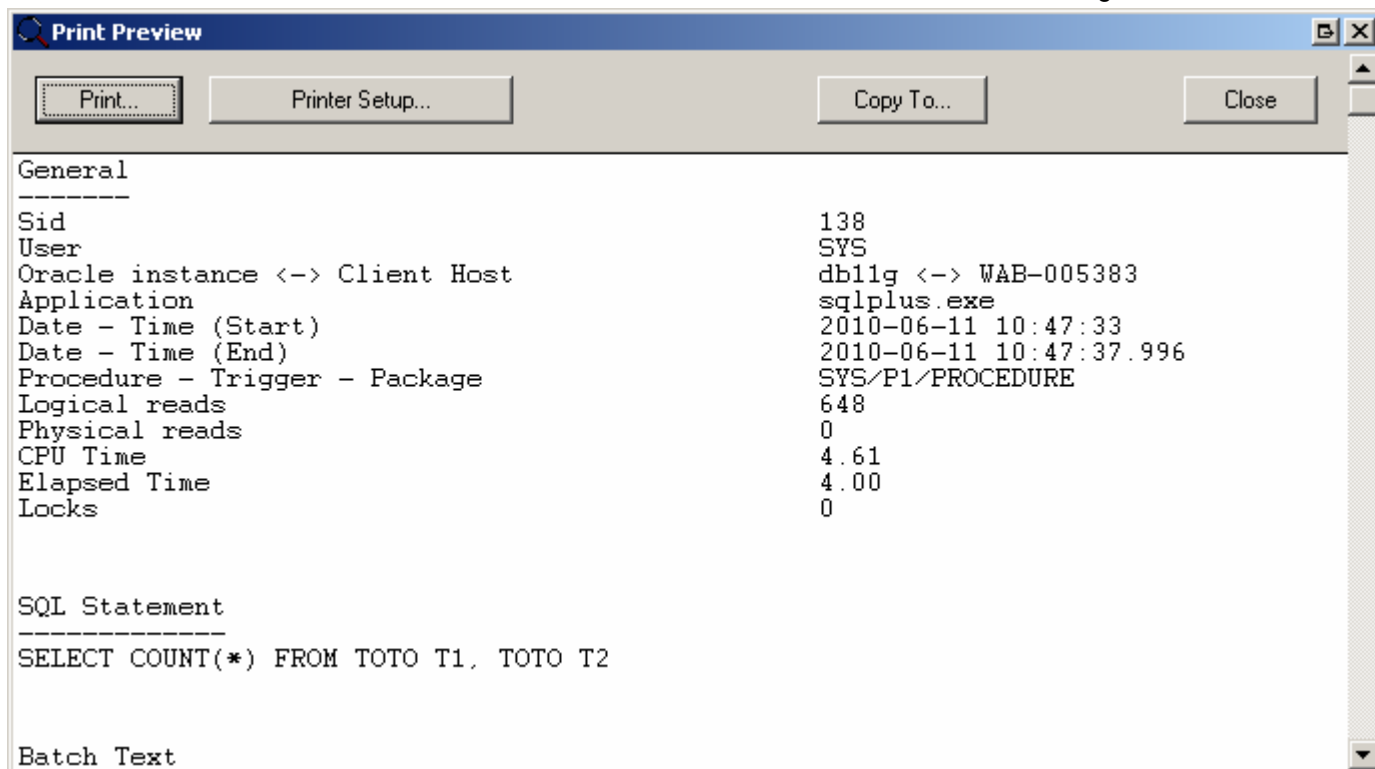
To select, open a configuration file previously configured with the following DATABASE Tuning's feature «ORACLE instance's performances' monitoring». To learn how to save and change the graph configurations, see paragraph on «Monitoring the ORACLE instance's performances».



Detail, Print

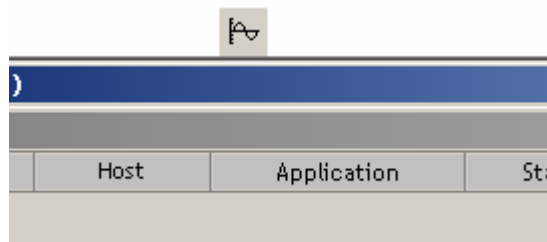
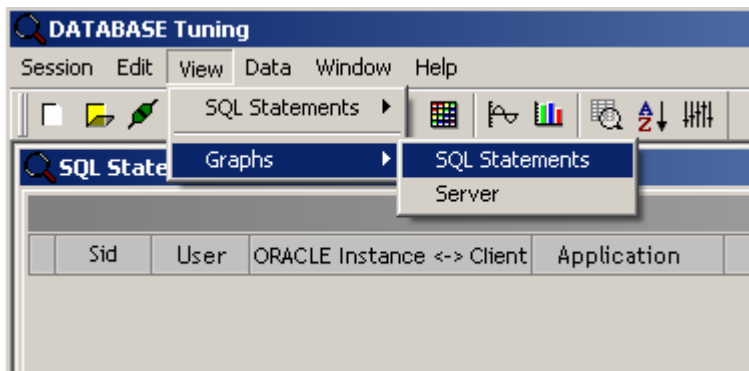


This feature allows you to print the contents of the «General», «SQL Statements», «Batch Text» and «Execution Plan» tabs. You can then select to print, configure the printing settings or copy the preview to a file.



SQL Statements' graphs

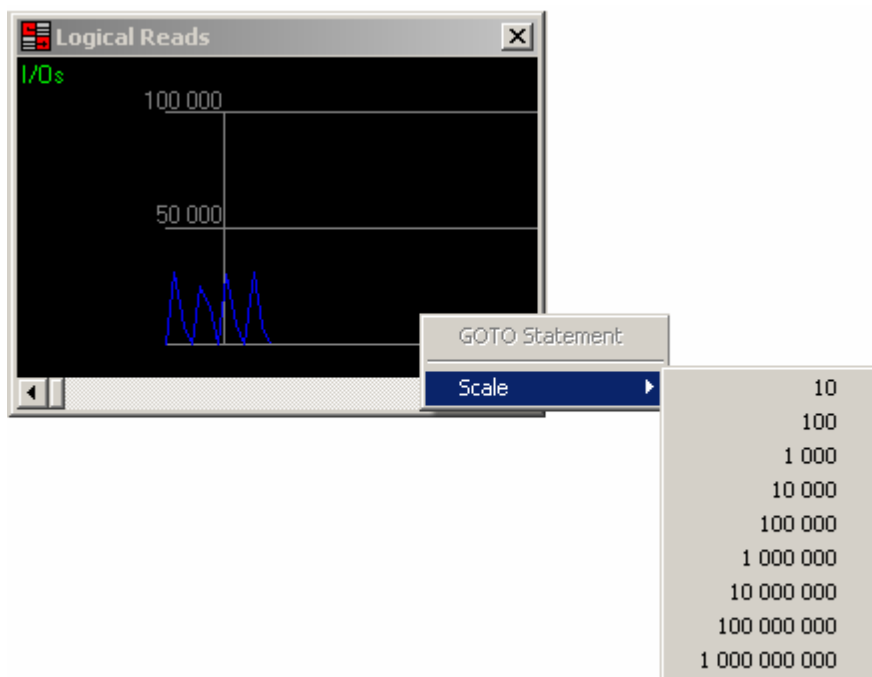
To display the SQL Statements' graphs, please select the «View - Graphs - SQL Statements» menu item. Alternatively, you can click on the «SQL Stmt graphs» menu icon (See Figures below). These graphs are only available for the completed SQL Statements.



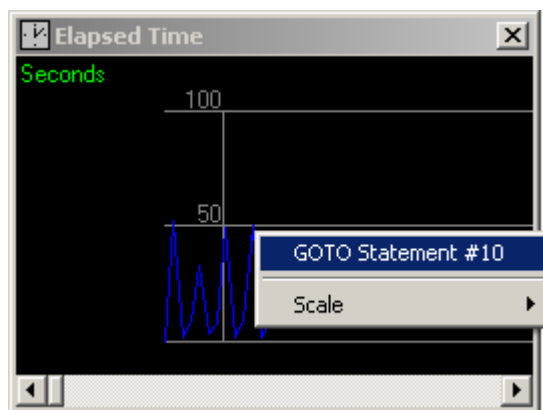
Two graphs are displayed:



These graphs display the execution duration (elapsed time) in seconds as well as the logical inputs/outputs used by the SQL Statements. They vary according to the sort order selected for the session's report. Each SQL Statement corresponds to a point in the curve.



Depending on the values obtained, you can adjust the scale by clicking on the right button. A contextual menu appears. Select the most suitable scale. Thanks to these graphs, you can identify the peaks showing which SQL Statements use most logical I/Os or Elapsed times. To point to the SQL Statement generating a peak, right-click on the peak in question. Next, select the «GOTO Statement #(i)», from the contextual menu. (i) is the SQL Statement's line number (column «#» in the session's report). Once clicked, the selected SQL Statement is displayed at the top of session's report (in the completed SQL Statements' part).

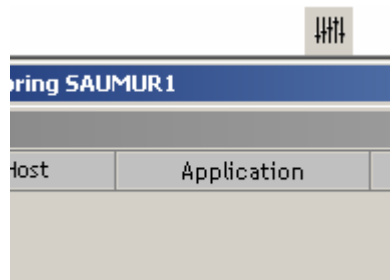
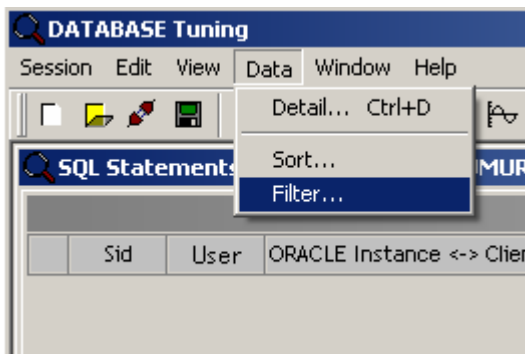


Filters

See also:

- Filter, Sid
- Filter, User
- Filter, Host
- Filter, Application
- Filter, Date-Time
- Filter, SQL Statement
- Filter, Batch Text
- Filter, Procedure, Trigger and Package
- Filter, Execution Plan
- Filter, Performances
- Filter, How to manage filter files

Filters are used to reduce the quantity of SQL Statements displayed in the session's report or captured by DATABASE Tuning. These filters can be applied to a session in progress or to an archive containing the history of previously recorded results. The filters affect completed SQL Statements' reports (bottom part of the window) and pending SQL Statements' reports (upper part of the window). To apply these filters, you need to select the «Data - Filter...» menu item. Alternatively, click on the «Filter» menu icon (See Figures below).



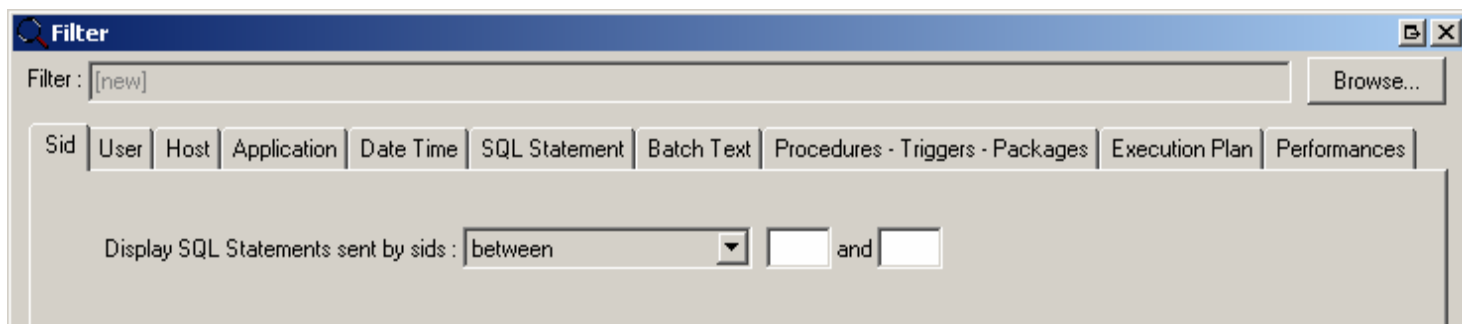
A window with ten tabs is displayed. Each tab represents a particular type of the session report's information. If several filter types are set, the filtered SQL Statements will need to fulfill all the defined filters (with an «AND» operator).

Example: A filter defines the user and logical I/Os higher than 5000. The filtered SQL Statements are linked to the SQL Queries sent by the user and consuming more than 5000 logical I/Os.

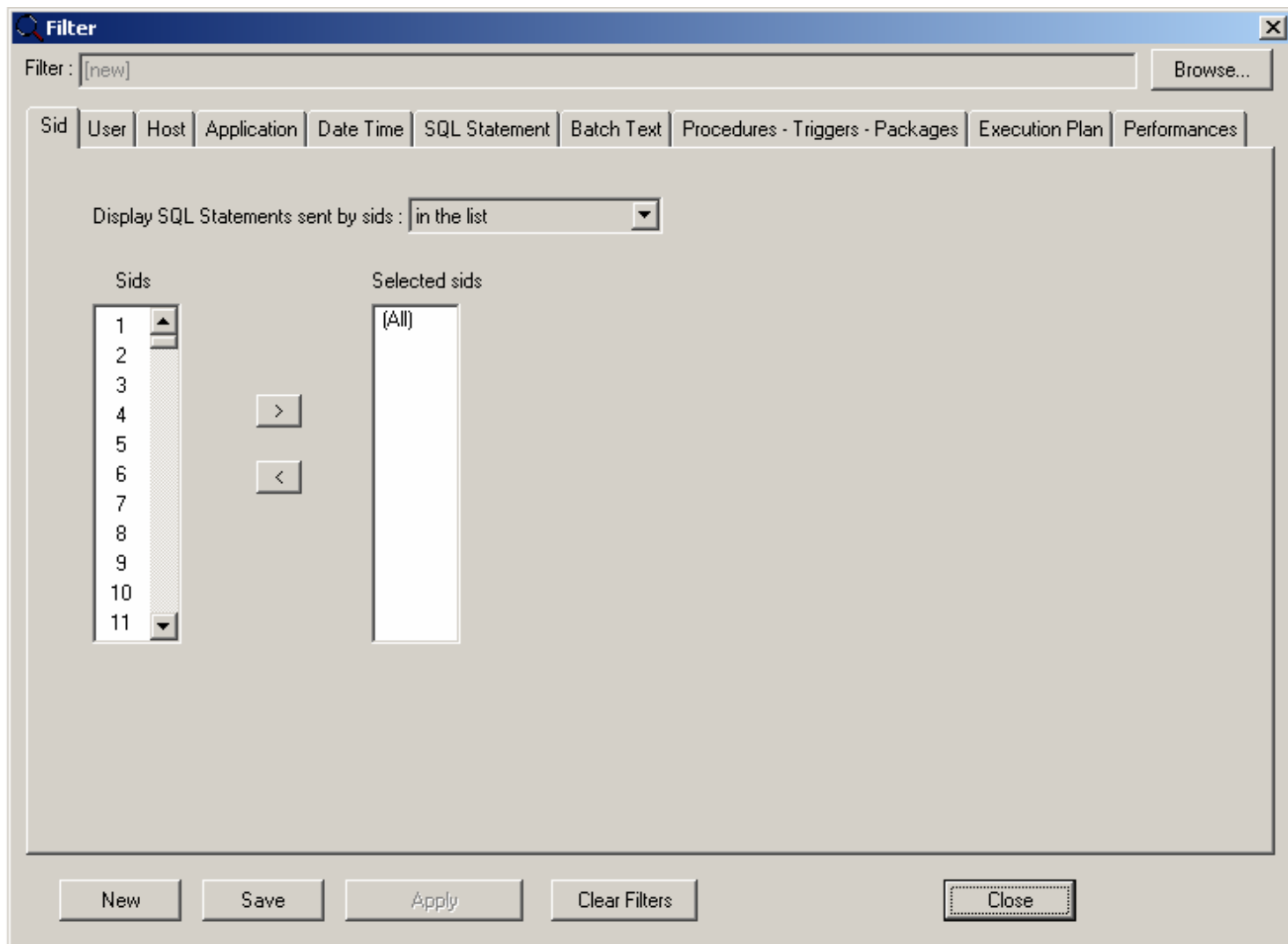
Filter, Sid

Only the SQL Statements executed by the defined Sid(s) will be displayed. The list of sids can be set within a list or within

a values interval.

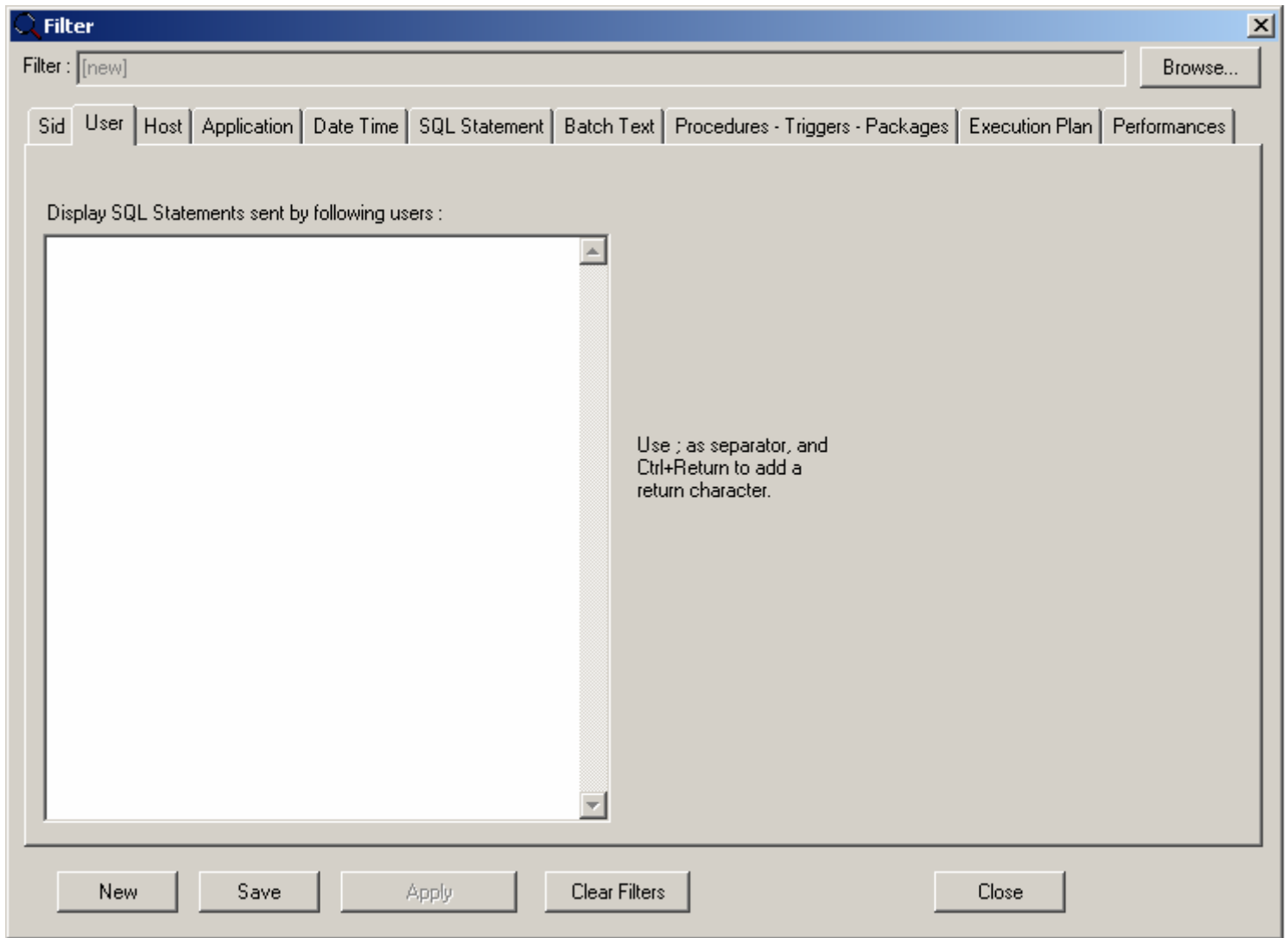


or within a list.



Filter, User

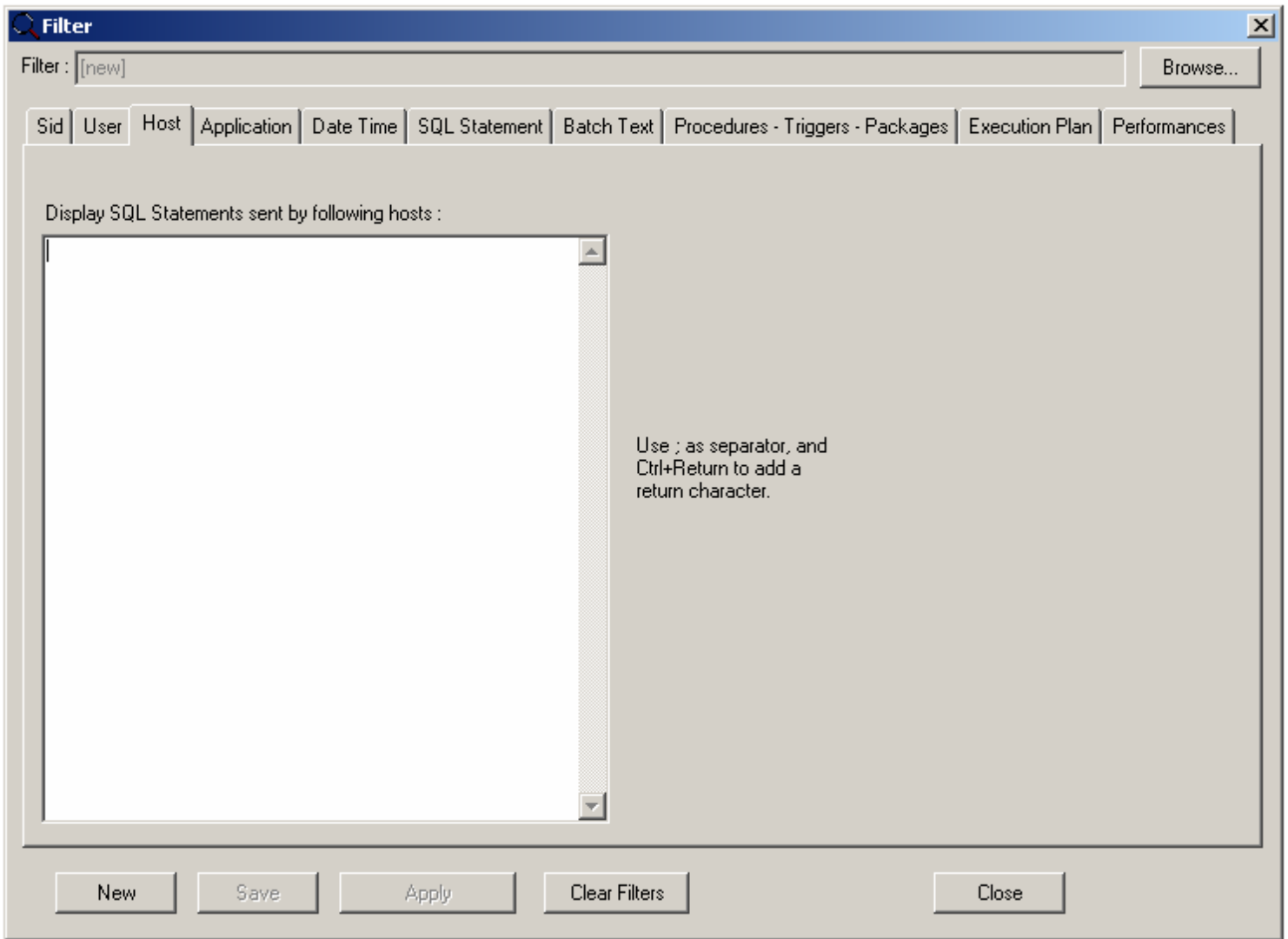
Only the SQL Statements executed by the defined users will be displayed. The users are entered in a list and are separated by a semicolon.



Filter, ORACLE instance <-> Client Host

Only the SQL Statements sent by the defined workstations, or sent to the defined ORACLE instance, will be displayed. The hosts and ORACLE instances are entered in a list and they are separated by a semicolon.

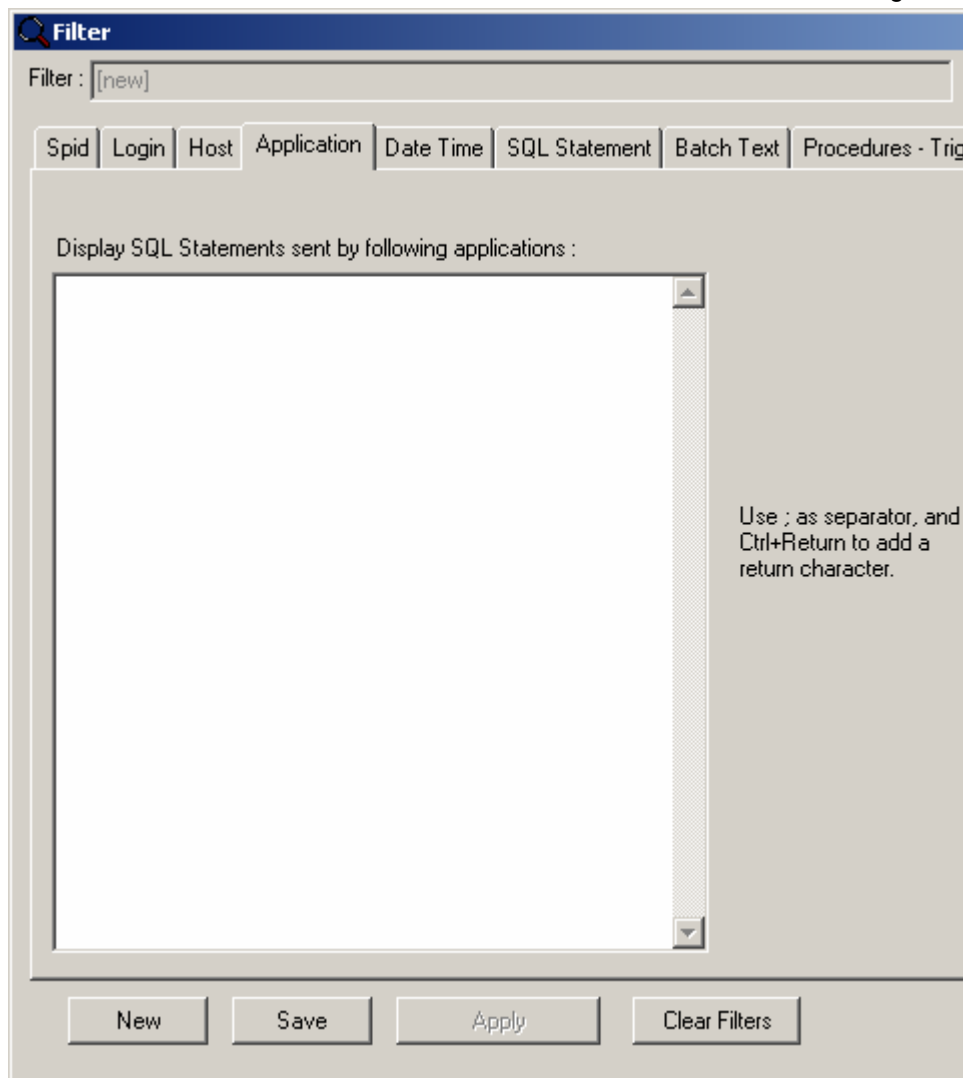
The «Host» information is the property of a given client application's connection. Since this property is optional, it may not always be mentioned. If you select this value as a filter, you need to make sure that the applications being monitored mention this property.



Filter, Application

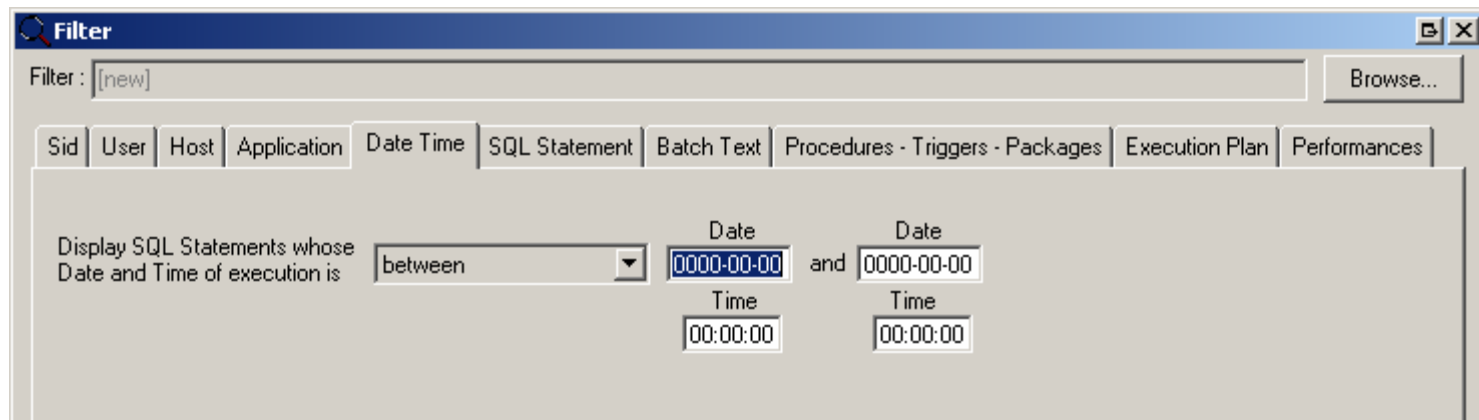
Only the SQL Statements executed in the defined applications will be displayed. The application names are entered in a list and are separated by a semicolon.

The «Application» information is a property of a given client application's connection. Since this property is optional, it may not always be mentioned. If you select this value as a filter, you need to make sure that the applications being monitored mention this property.



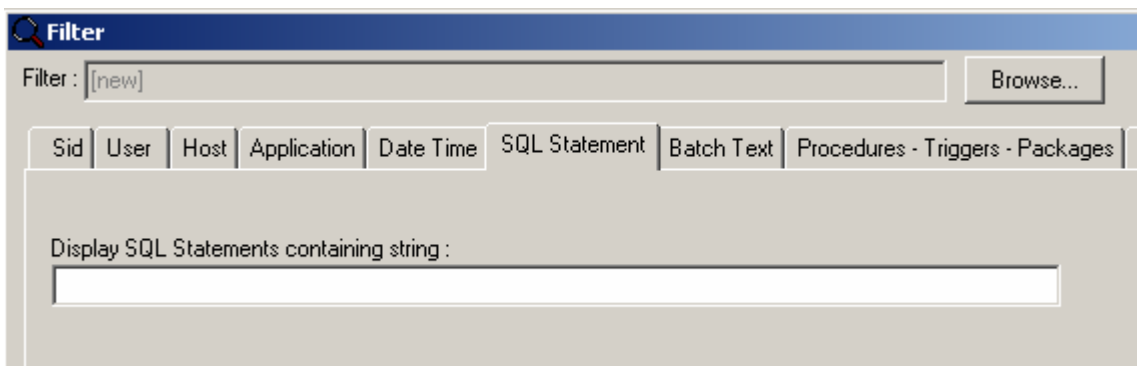
Filter, Date-Time

This filter allows you to only display the SQL Statements that are being executed within a specific range of time. You can select to display the SQL Statements starting from a particular date and time or you can select a particular segment between two time ranges.



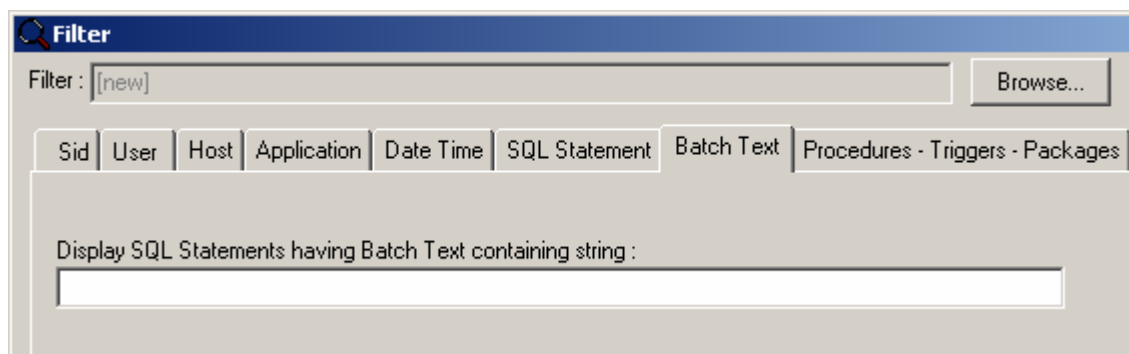
Filter, SQL Statement

This filter allows you to only display the SQL Statements containing a defined character string (e.g., the name of a given table). Please note that the character string entered needs to match the SQL Statement's text exactly.



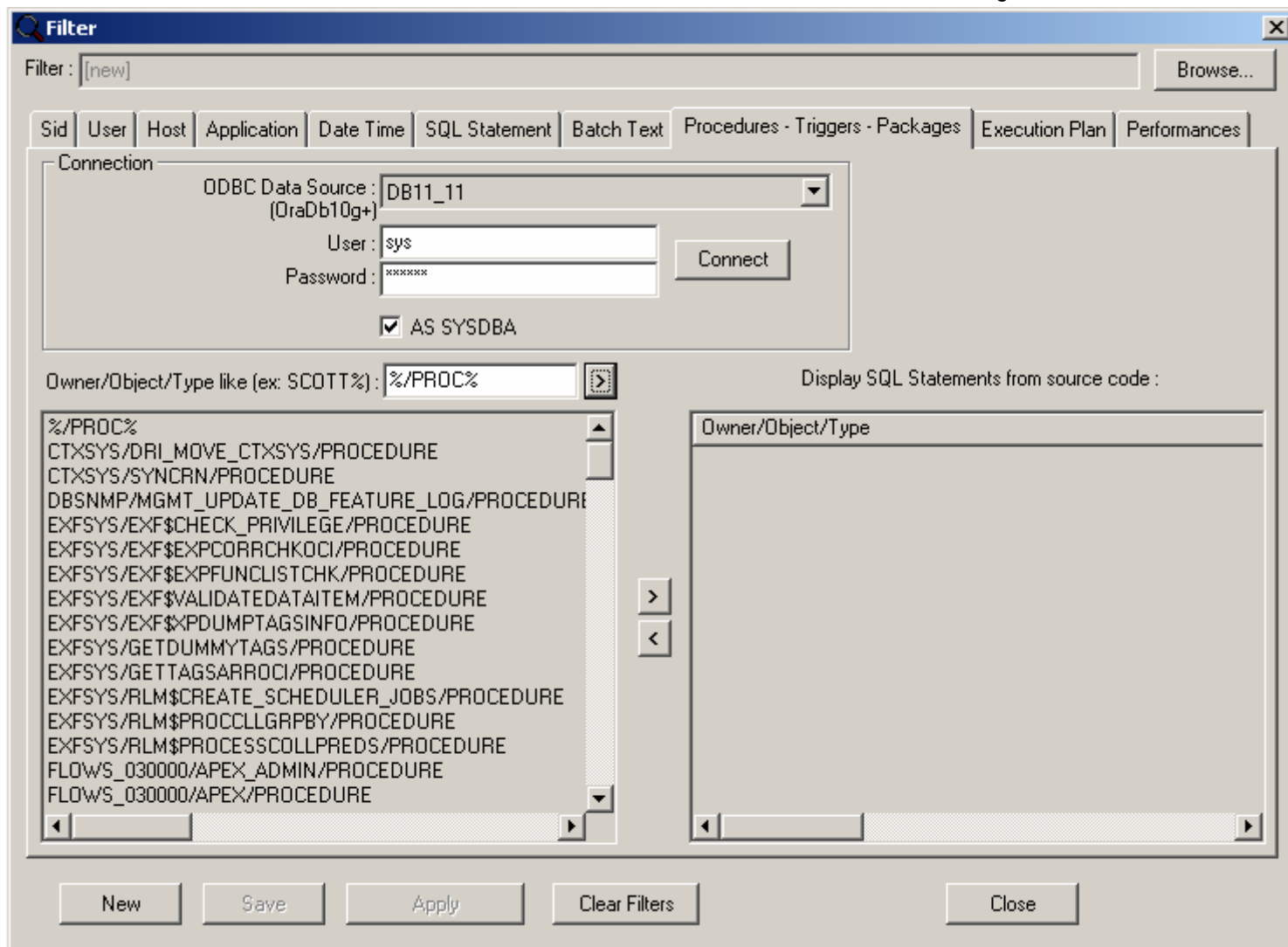
Filter, Batch Text

This filter allows you to only display the SQL Statements whose Batch Text contains a defined character string (e.g., the name of a given stored procedure). Please note that the character string entered needs to match the Batch Text exactly.



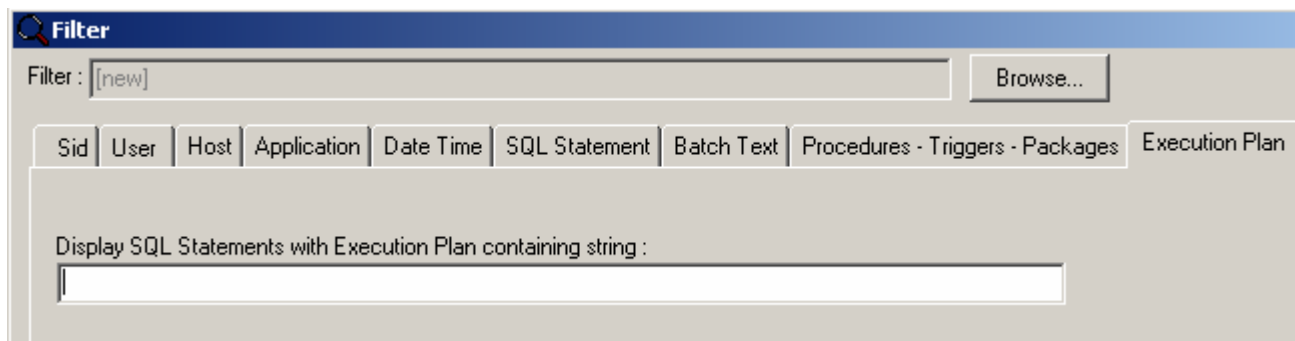
Filter, Procedure, Trigger and Package

This filter allows you to only display the SQL Statements which are part of a defined list of stored procedures or triggers or packages. To select the list, you first need to connect to ORACLE instance. Next, select the list of procedures or triggers or packages required.



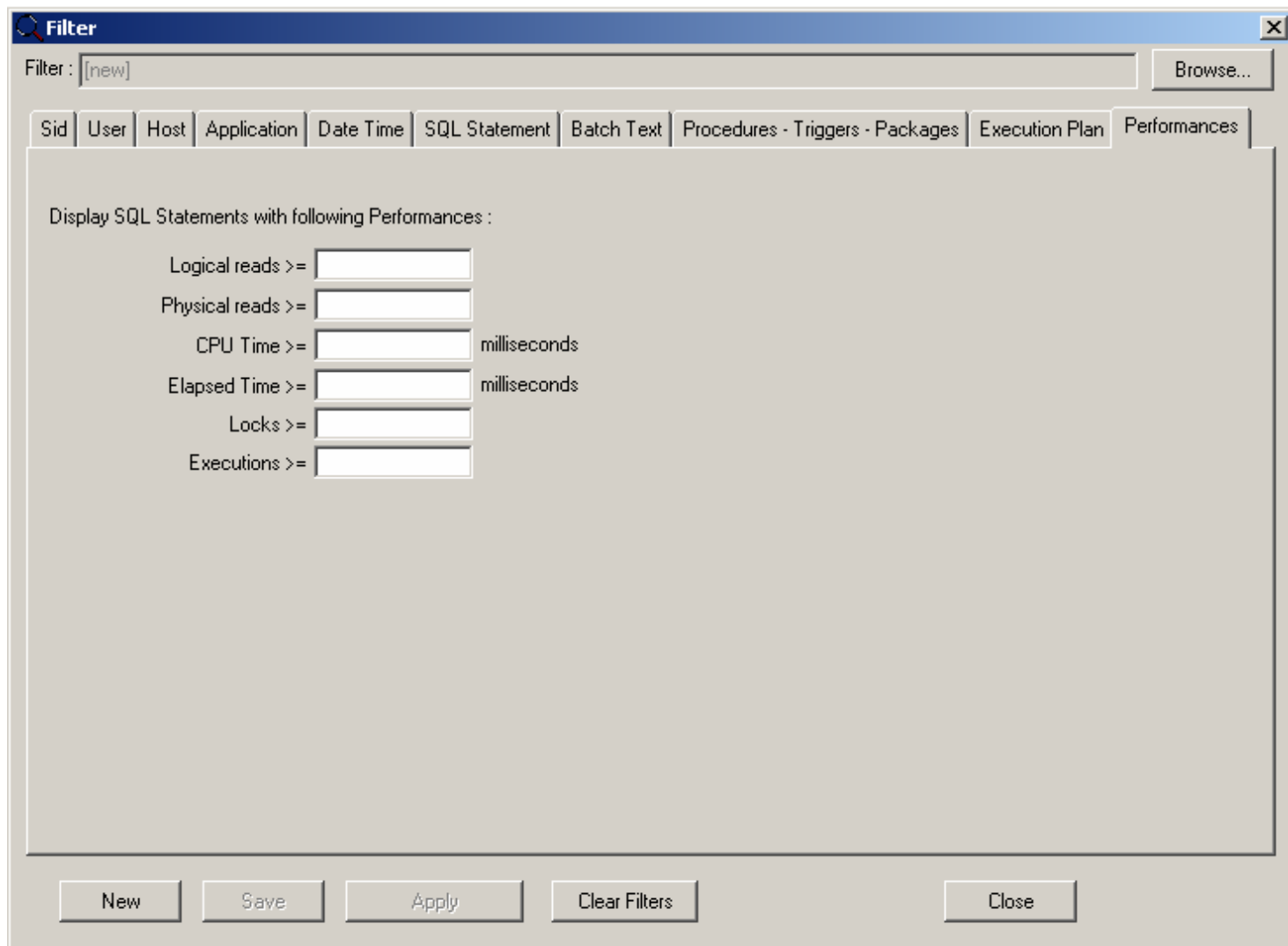
Filter, Execution Plan

This filter allows you to only display the SQL Statements whose execution plan contains a defined character string (e.g., «FULL SCAN»). Please note that the character string entered needs to match the execution plan text exactly.



Filter, Performances

These various filters allow you to only display the SQL Statements whose performance indicators exceed the preset thresholds. By default, these thresholds are set to 0, except for the Elapsed Time threshold, which cannot be less than one second.

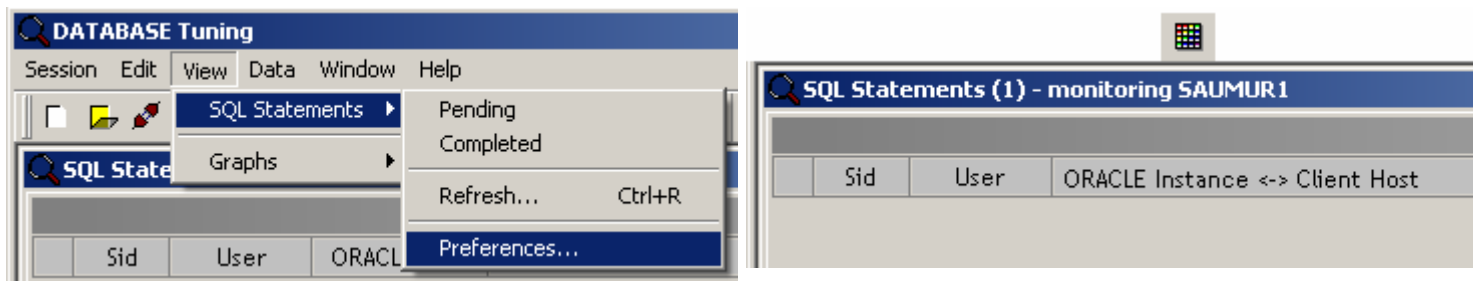


Filter - How to manage filter files

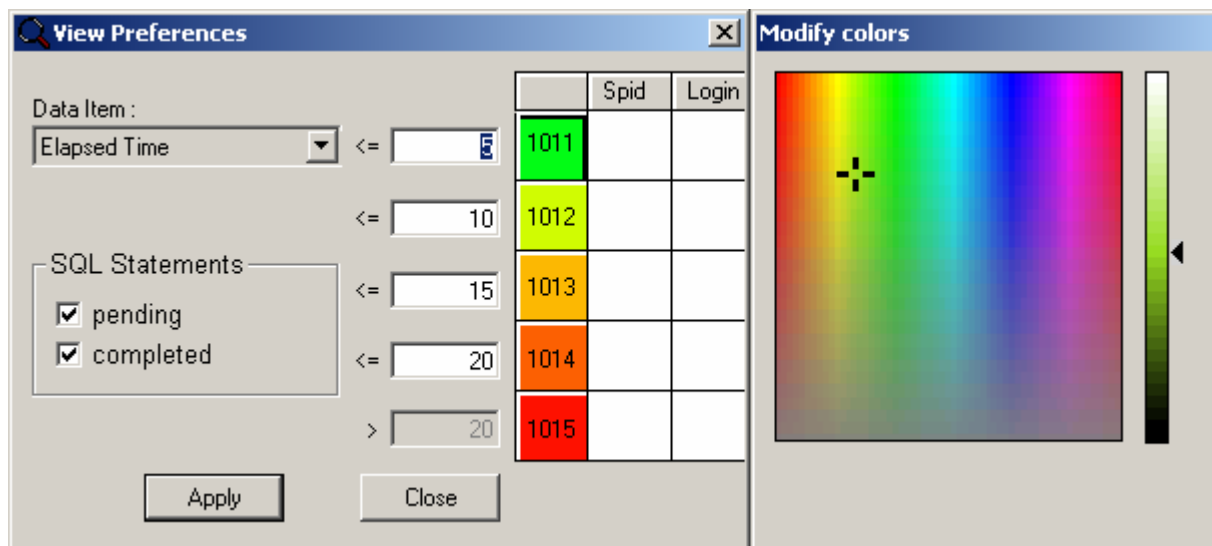
Once you have defined the elements you need to filter, save these settings in a file. To do this, click on the «Save» button and select a filename with the FLT suffix. After clicking on the «Apply» button, this file is used to apply the filters. To retrieve previous settings, click on the «Browse» button to select a filter file. The «New» button is used to clear all settings from the various tabs. The «No Filter» setting unselects the filtering and displays the monitoring windows without applying any filter. By default, «.FLT» files are stored in «%APPDATA%\DATABASE Tuning\cfg\flt» directory.

Display options

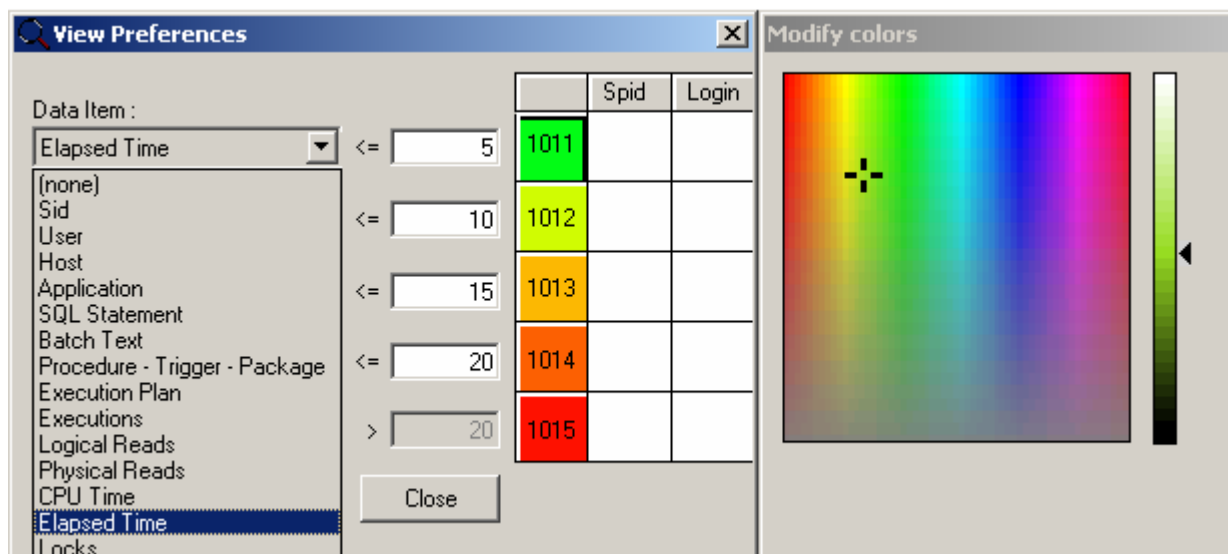
For an easy identification of the most consuming SQL Statements, DATABASE Tuning allows you to associate colors, defined by thresholds, with a specific performance indicator. By default, these thresholds are set to the «Elapsed Time» indicator and the colors range from green to red, based on the SQL Statement's execution duration. The display options make it possible to choose the performance indicator to which the thresholds will apply and to configure the values and colors associated with each threshold. To modify the display options, select the «View - SQL Statements - Preferences» menu item. Alternatively, click on the «Preferences» menu icon (See Figures below).



The following window is displayed:



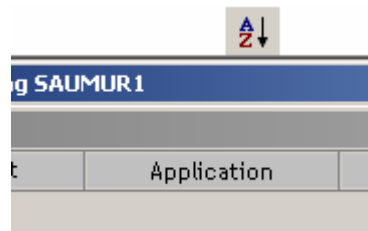
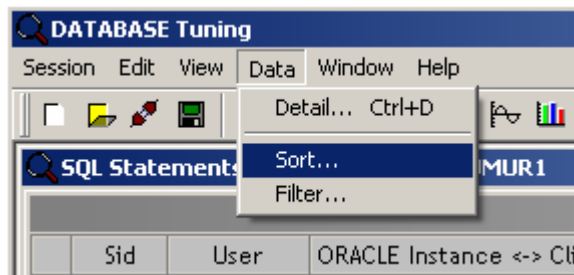
Select the performance indicator to which the thresholds will apply.



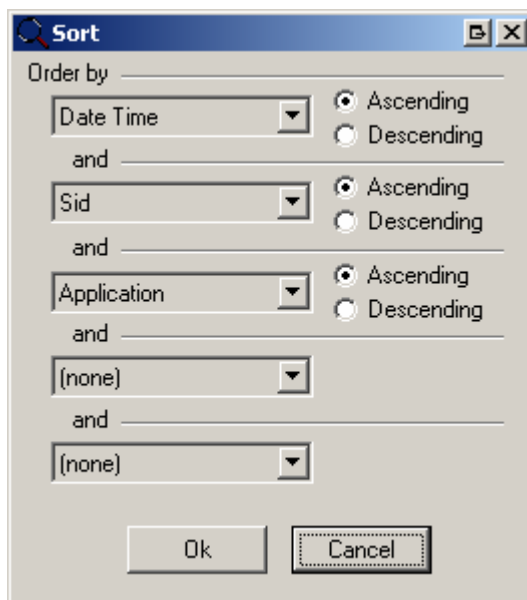
Choose colors as well as values for the various thresholds. Next, apply these display options to the pending SQL Statements, to the completed SQL Statements, or to both.

Sorts

To sort the lines of a given report, select the «Data - Sort...» menu item. Alternatively, click on the «Sort» menu icon (See Figures below).



The following window is displayed:

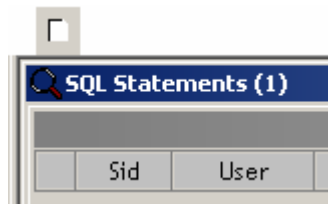
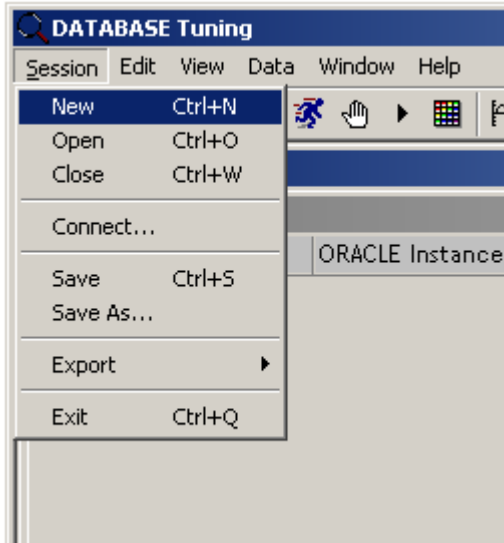


You will need to select the sort keys (up to five levels) as well as the direction (ascending or descending). Each sort key corresponds to a specific column of a given session's report.

Caution: For performance's sake, sorting is not allowed while viewing results in real time (Automatic Refresh).

New Session

To create a new session, select the «Session - New...» menu item or click on the «New» menu icon (See Figures below). For performance's sake, we recommend that you not create several monitoring sessions simultaneously, but rather that you launch as many DATABASE Tuning versions as needed for each monitoring session. The monitoring frequency of each session is evenly distributed by the Windows load-balancing task on the various processors of your computer. Because you can create new sessions, you can view several archive files simultaneously.



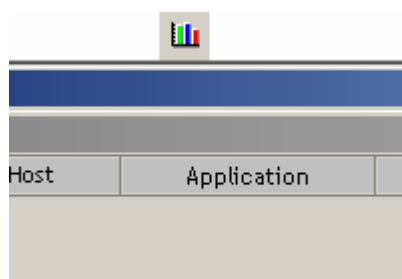
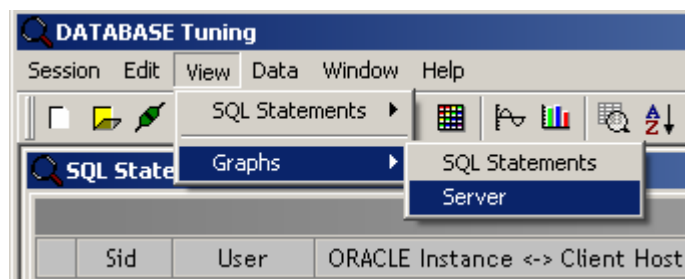
Within a new session, you can create a new connection to the Adaptive Server. To do so, use the («Session - Connect...» menu. (See Recommendation above). You can also open a historical archive by using the («Session - Open...» menu).

Monitoring the ORACLE instance's performances

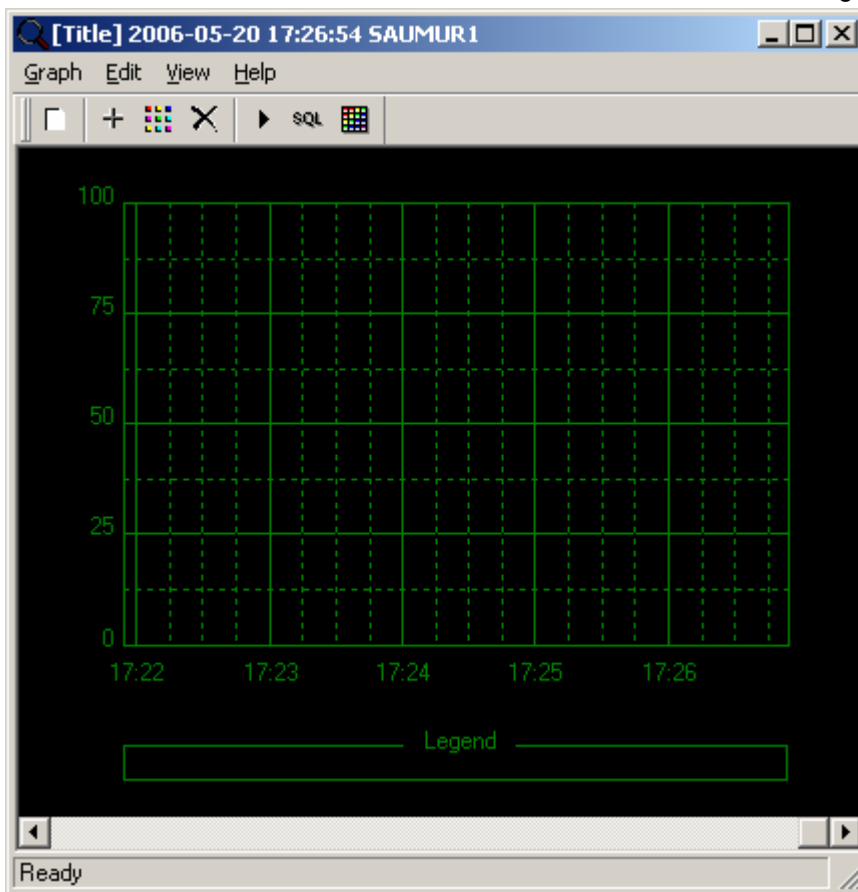
See also:

- How to start and stop a monitoring
- ORACLE instance's performance indicators
- How to configure the indicators
- How to restore a configuration
- How to add an indicator
- How to modify an indicator
- How to remove an indicator
- How to save the configuration
- How to refresh the graph
- SQL Statements related to a time period
- Graphs' display options
- New graph

To view the monitoring results relating to particular performance indicators of the Adaptive Server, select the «View - Graphs - Server» menu item from the session's report window. Alternatively, click on the «Server graphs» menu icon (See Figures below).



The following window is displayed:



From this window, you can view the ORACLE instance's performance curves. However, this window is not initialized. You first need to select the indicators that you want to see displayed. You also need to define the appropriate graph's scale.

ORACLE instance's performance indicators

| Metric | Description |
|-----------------|---|
| CNX | Number of open connections in the time interval. |
| CPU | Amount in milliseconds of CPU time spent by the user sessions in the time interval. In order to make this metric available, it is necessary to activate the system parameter named "timed_statistics" (cf. paragraph "How to start a monitoring session"). A high rate means that queries are consuming too many buffer_get, full scans in memory. You must consult then what was recorded in the queries audit report, and fix those which are consuming the most i/o in memory (logical reads). |
| BUF SPACE REQ | A number of allocations of free buffers in the time interval. A high rate means that the cache is too small. |
| CKPT | Number of Checkpoints. Each checkpoint accesses to disc (Datafile). The checkpoints must be specific and not be triggered more often than every 30s even during a very strong load. The recommended value is a minimum of 3mn, optimum is 30mn. It is necessary to increase the cache size (db_block buffers, db_cache_size) or to increase the size of the redo logs and to check the following parameters: LOG_CHECKPOINT_INTERVAL and LOG_CHECKPOINT_TIMEOUT. |
| PHYS IO | Number of physical i/o (reads and writes) in the time interval. |
| REDO LOG | During the time interval, numbers of issues, for allocations of the redo log buffers. This value must be close to 0. If it is not the case, it is necessary to increase the log cache size (log_buffer) and/or the size of redo logs. |
| COMMITTS | Ratio of the number of commits compared to the number of executions. This value is calculated for the time interval. Too many commits slows down the activity of the database by forcing the writes to disc. Oracle empties its log cache (log_buffer) and waits for the acknowledgement of the writes to disc. LGWR avoids the use of system cache, but not the hardware cache. If the ratio is higher than 5%, then the running applications will execute too many commits. |
| SORTS | During the time interval, SORTS is the ratio of the number of sorts executed using disc i/o compared to those executed in memory. It is necessary to increase both SORT_ARE_SIZE and PGA_AGGREGATE_TARGET parameters if the ratio exceeds 5%. |
| FULL SCANS | This is the number of full scans executed on huge tables. |
| FRAG | This is the number of fragmented block use. |
| PARSE | Ratio = number of Parse / number of Executions. |
| NET RECEIVED | Number of Mega Bytes received by database. |
| NET SENT | Number of Mega Bytes sent by database. |
| CACHE HIT RATIO | Percentage of hits in cache. |
| DEADLOCKS | Number of deadlocks. |
| BLK SESSIONS | Number of sessions in a wait status. |
| ROLLBACK | Ratio of wait for the rollback segments. This value must be close to 0. In case not, it is necessary to increase the number of rollback segments, the number of extents of each rollback segment, and the size of the extents as well. |

There are two ways to display an indicator. You can either use the preset graphs' configuration files or you can manually define the graph's aspect and its contents (See «How to save the configuration»).

How to configure the indicators

Monitoring performance metrics designed for the ORACLE instance are defined by a skilled database administrator. The %APPDATA%\cfg\mon*.mon files contain queries executed on the targeted instance to get performance metrics. These queries must be tested to avoid to add CPU load on the monitored instance. This is the responsibility of the database administrator. In addition, you must grant the monitoring user to access targeted views or tables.

It is necessary to respect the following syntax:

[QUERIES]

query_<query identifier>=select <metric identifier>, <metric name>, <metric value (numeric)>, 'S or C'

Where 'S': means that indicator's value is the result of the difference with the previous sample
 'C': means that indicator's value is taken as it is received.

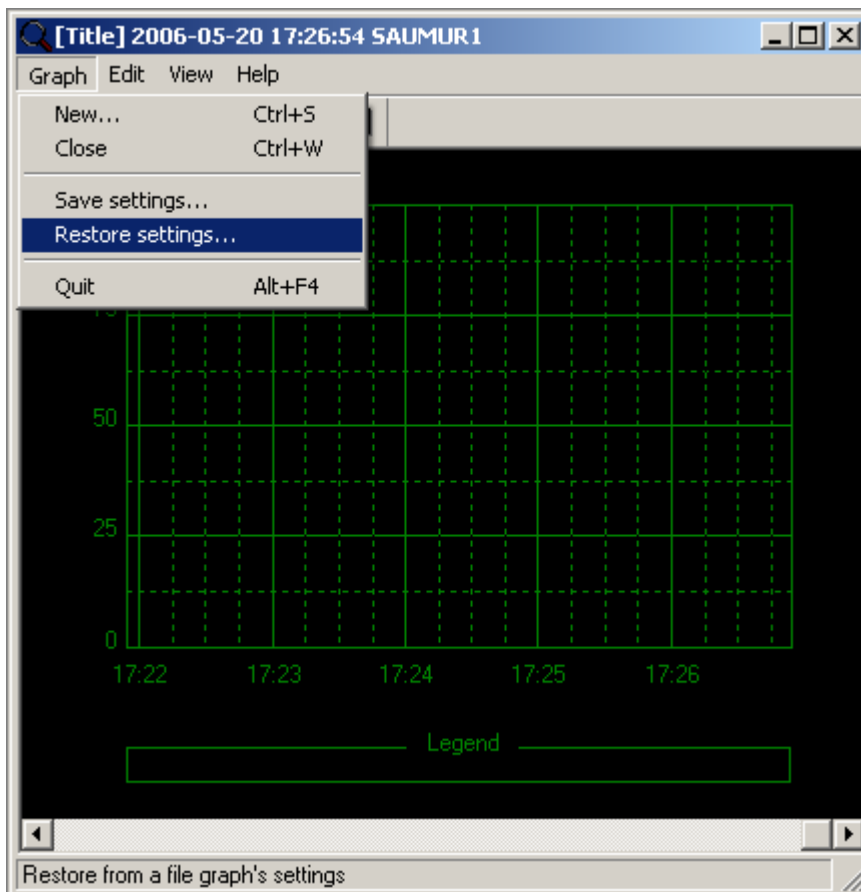
Example:

[QUERIES]

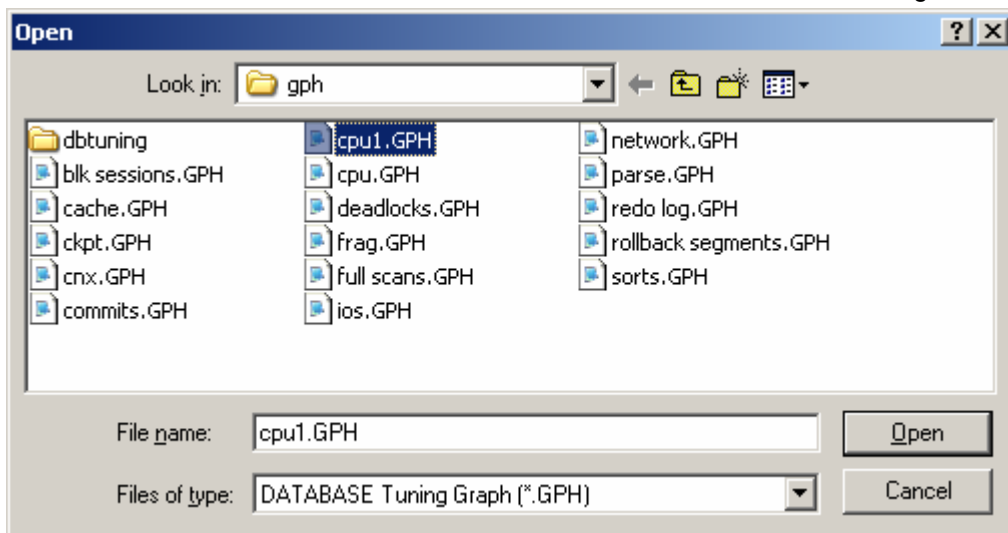
query_5=select 2002, 'gets', sum (gets), 'S' from v\$rowcache

How to restore a configuration

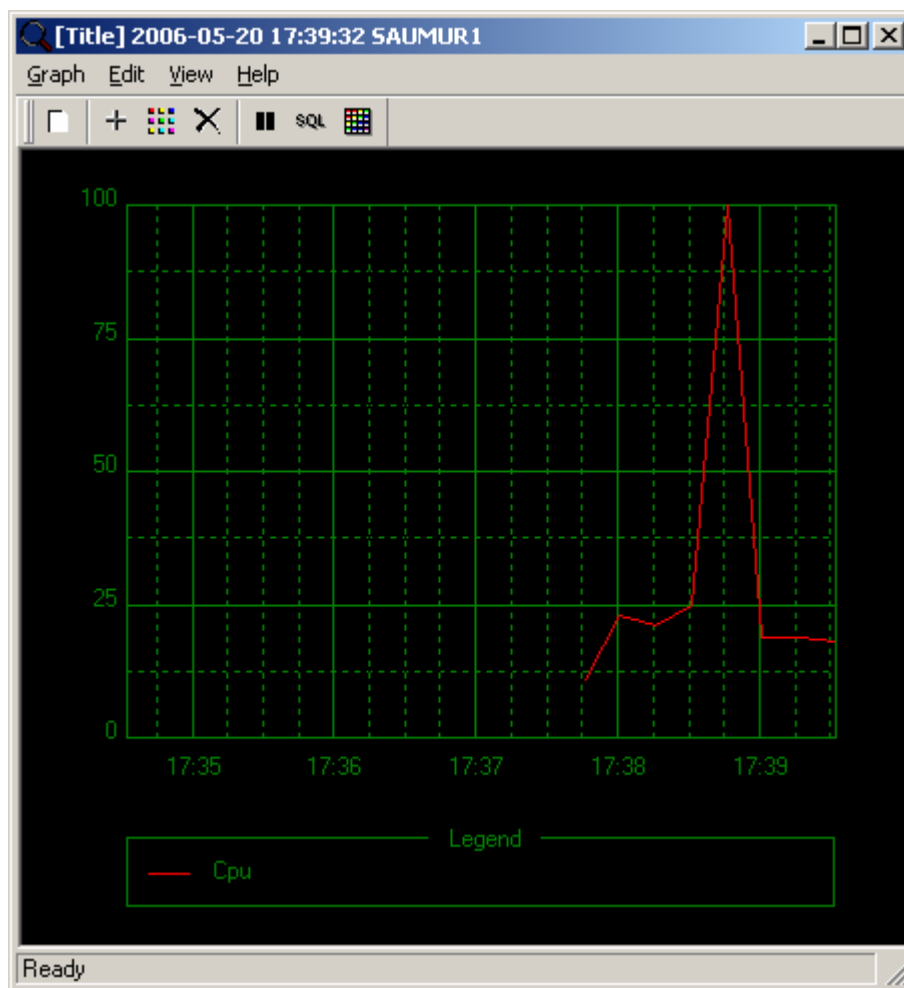
Select the «Graph - Restore settings...» menu item.



Next, determine the location and the name of the file containing the requested graph configuration. The default location is «%APPDATA%\DATABASE Tuning\cfg\gph» directory.

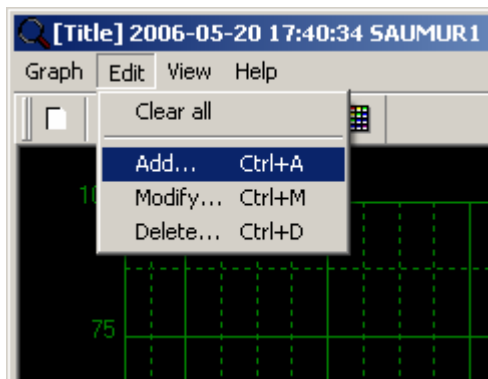


When you have selected the indicator you want to view, the graph adapts automatically to the configuration saved in the file (See below). The configuration files mentioned above are provided by default with the DATABASE Tuning's installation. However, you can create your own configuration files or modify those provided to suit your needs.

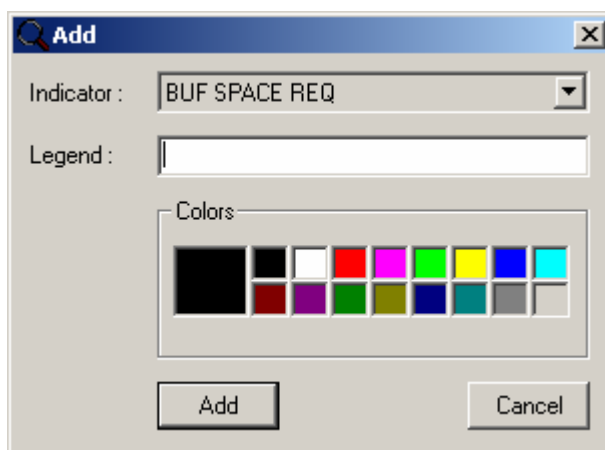


How to add an indicator

To add a new indicator to your graph, select the «Edit - Add...» menu item. Alternatively, click on the «Add» menu icon, as illustrated below.



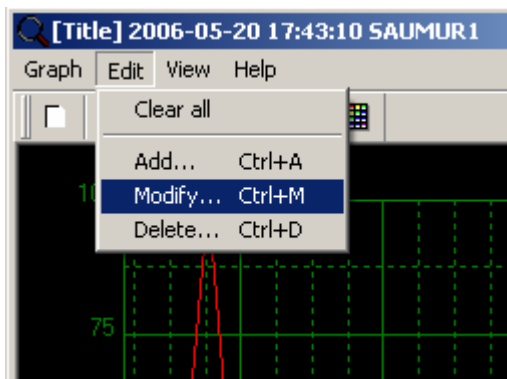
The following dialog box appears:



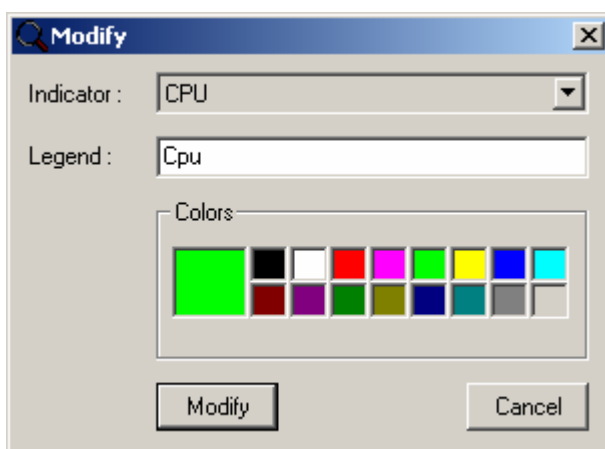
From this window, you can define the indicator you want to view. First, select the «Indicator» dropdown list box containing the various types described above (See paragraph on «ORACLE instance's performance indicators»). Next, enter the text you want to see displayed in the lower part of the graph. You can then select the color of the values' curve. When you click on the «Add» button, the graph changes immediately to reflect the new values recorded by the selected indicator. If the dropdown list box does not contain any value, it means that no monitoring session was created or that no values were read in the archive database or file. If the graph doesn't change after you have added an indicator, you need to change the graph's scale (See Paragraph below on «Graphs' display options»). The absence of change in the graph can also mean that there is no value for this indicator in the time range displayed in a given session's report. (If this happens, please contact [DATABASE Tuning Technical Support](#) for a better assessment of the case.)

How to modify an indicator

To modify the display of a specific indicator, select the «Edit - Modify...» menu item. Alternatively, click on the «Modify» menu icon, as illustrated below.



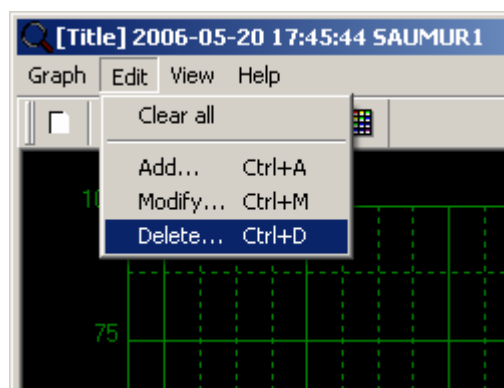
The following dialog box appears :



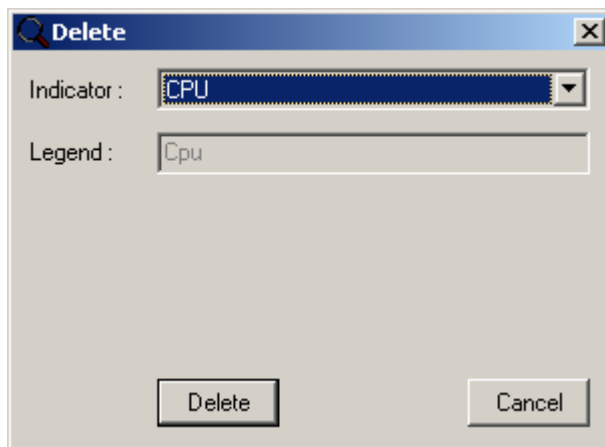
At this stage, you can modify the display characteristics that were previously added to the graph. To modify the texts and/or colors, select the indicators you need to change from the «Indicator» dropdown list box. Next, click on the «Modify» button. The graph changes automatically to reflect the new display settings.

How to remove an indicator

To remove an indicator from your graph, select the «Edit - Delete...» menu item. Alternatively, click on the «Delete» menu icon, as illustrated below.



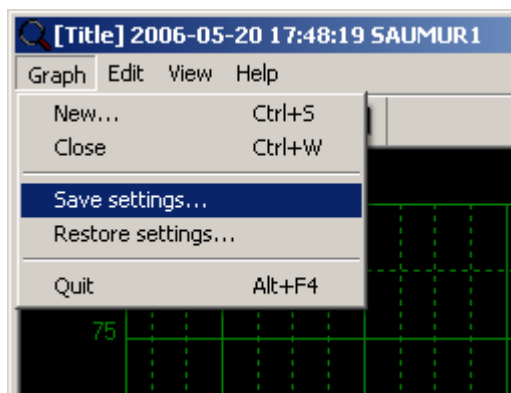
The following dialog box appears:



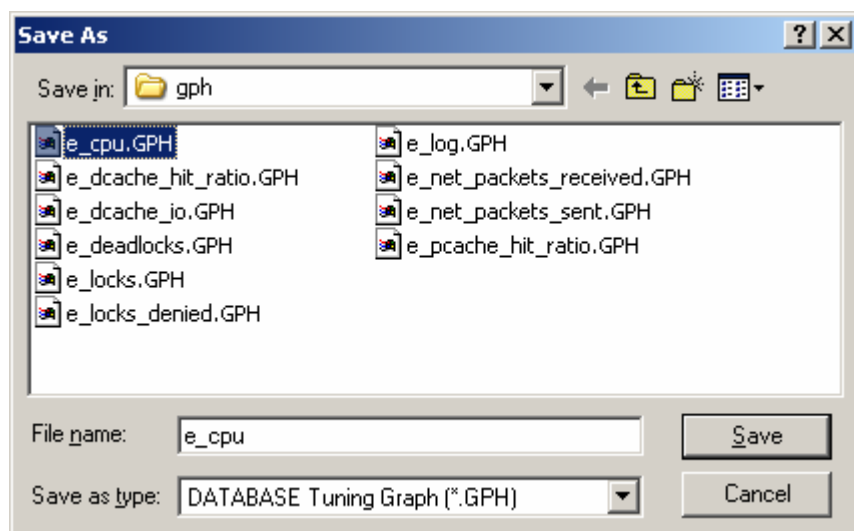
You can now remove indicators that were previously added to the graph. To do so, click on the «Delete» button. The selected indicator's values are immediately removed from the graph.

How to save the configuration

To save a graph's configuration, you choose «Graph - Save settings...» menu item.

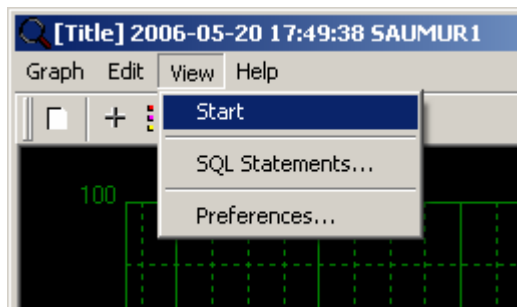


You now need to determine the location and the filename containing the graph configuration you have just saved. By default, «.GPH» files are stored in «%APPDATA%\DATABASE Tuning\cfg\gph» directory.



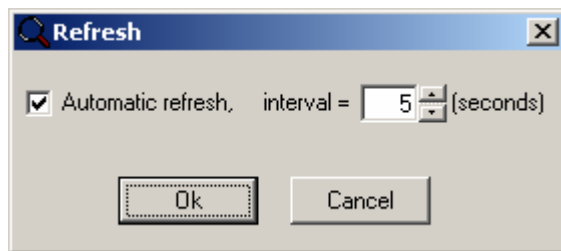
How to refresh the graph



To perform the automatic refresh of the graph, select the «View - Start...» menu item. Alternatively, click on the «Refresh» menu icon (See Figures below).



This option makes it possible to refresh the display of a given report or graph, whether they are being produced locally or by a monitoring session in progress that is executed by another DATABASE Tuning (See paragraph on “How to save automatically”). This option allows you to view the events captured by a workstation dedicated to monitoring from a distant client PC workstation (see paragraph on “Executing DATABASE Tuning in a background task”).

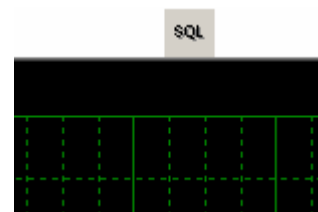
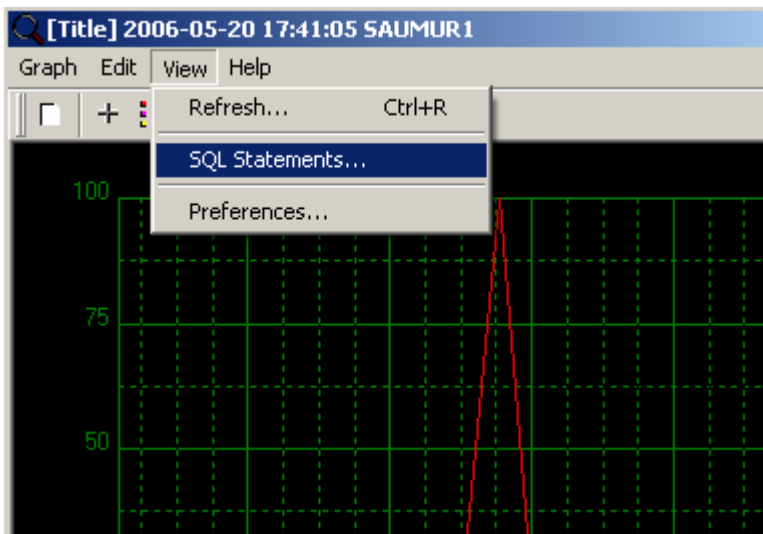
When you activate the automatic refresh function while reading a file or an archive database extract, a dialog box asks you to define the refresh frequency. In direct monitoring mode (DATABASE Tuning is connected to an ORACLE instance), the automatic refresh function is performed with the same frequency as the monitoring (i.e. the refreshing frequency cannot be modified). However, you can interrupt the refreshing, if you wish to.



As soon as you activate the automatic function, the refresh icon changes from the «read» mode  to the «pause» mode .

SQL Statements related to a time period

To view the SQL Statements captured during the time range displayed in the graph's window, select the «View - SQL Statements...» menu item. Alternatively, you can click on the «SQL Statements» menu icon (See Figures below).



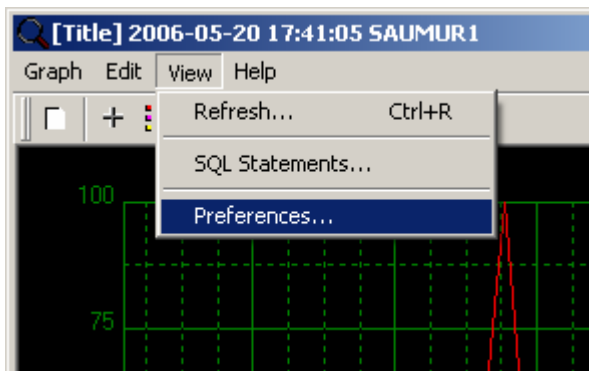
The following window appears. It only contains the SQL Statements that have been captured from the beginning date-time of the graph (left) to the ending date-time of the graph (right).

| End | Exécutions | Elapsed Time | CPU Time | SQL Statement | Batch Text |
|---------------------|------------|--------------|----------|---------------------------------|-------------------|
| 2006-05-20 17:38:34 | 1 | 10,00 | 9,12 | SELECT count(*) from dba_source | BEGIN p1; END; |
| 2006-05-20 17:38:23 | 1 | 35,00 | 0,00 | SELECT count(*) from dba_source | BEGIN p1; END; |

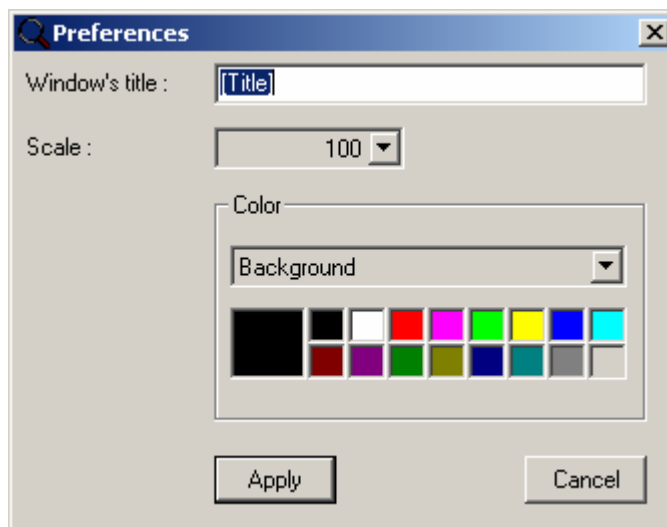
From this window, you can export the data to Excel or to a .txt format file. You can also display the SQL statements' graphs and view the details of a given SQL Statement. These actions obey the same principles as those concerning a monitoring session's report.

Graphs' display options

You can modify the graphs display options (the graph's title, the graph's scale, the window's fill color, the graph grid's color). To do that, you select «View - Preferences...» menu item, or click on the «Preferences» menu icon (cf. figures below).



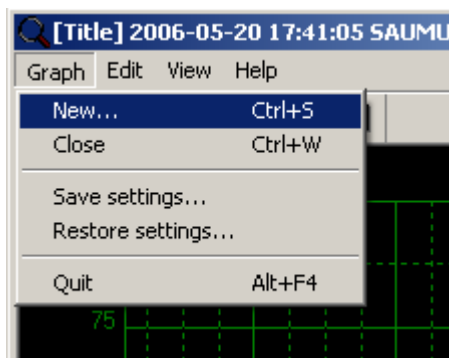
The following dialog box appears:



The preferences defined from this dialog box can be saved in a configuration file (See paragraph on «How to save the configuration»).

New graph

To create a new graph, select the «Graph - New...» menu item. Alternatively, click on the «New» menu icon (See Figures below). You can display as many graphs as you wish, depending on your PC workstation's capacity.



If at any time you need to re-display a graph that was left in the «close» menu box, simply activate the «Graphs - Server» function from the session's report window.

How to save results

See also:

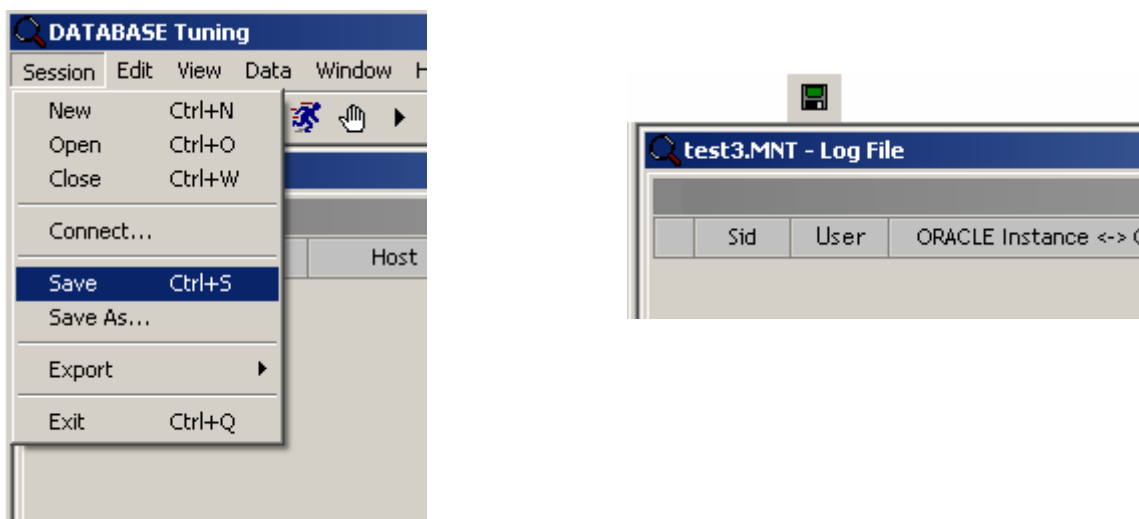
How to save to a database

How to save to a file

How to save automatically

Exports

To save the contents of a session's report, select the «Session - Save/Save As...» menu item. Alternatively, click on the «Save» menu icon (See Figures below).



Next, you have two ways of saving your information: you can save it to a file or to a database.

How to save to a database

You can save the contents of a session's report to an ORACLE database. There is no technical prerequisite attached to this database. The database's physical model is provided in the appendix (See paragraph on "DATABASE tuning's archive database model"). This feature is only available in the DATABASE Tuning's Enterprise Edition.

For performance's sake, the archive database should not be located on a WAN (from the DATABASE Tuning collecting the monitoring results). We recommend that you not install this database on a production server. You need to allow for 10 Kb per captured SQL Statement, on average, and for 1 Mb, for ORACLE instance's indicators values, per monitoring hour. As an example, it is typical to capture more than 5000 SQL Statements per day on a targeted production instance experiencing performance problems. This represents a total volume of 50 Mb for the SQL Statements and of 24 Mb for the instance's indicators values. Approximately 75 Mb are delivered, each monitoring day. For DATABASE Tuning's stored delete procedures, see paragraph on «Results deletions». It is necessary to manage the archive database's free space. (A good way to do this is to set alarms to manage tablespaces's free space. We recommend that you create a database with a minimum size of 5 Gb for data and indexes and of 500 Mb for redolog.

Access management

The user used to record the results in the archive database must be that of the schema owner. The users who view the information are simple «users» granted to select the tables. They are able to execute deletes on the results (see paragraph on «Results deletions»).

The viewing of information within a centralized database can be of critical importance to your security. If such is your intention, you can create views as follows:

You create a user, you allow him to use the archive database, you modify the view «ssv_stmt_user».

```
Drop view ssv_stmt_user;
```

```
create view ssv_stmt_user
```

```
as
```

```
select * from ssv_stmt
where < criteria of restriction >;
```

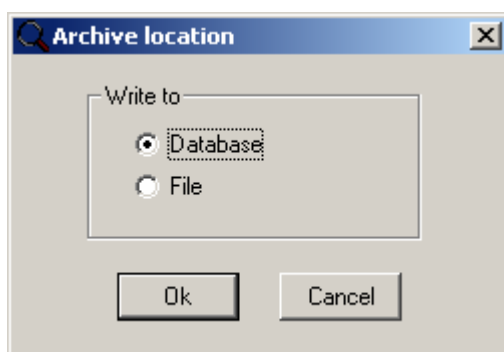
example:

```
drop view ssv_stmt_user;
create view ssv_stmt_user
as
select * ssv_stmt
where (user = «DEVELOP» and servename = «ORAPROD» and appname = «PORTFOLIO»)
```

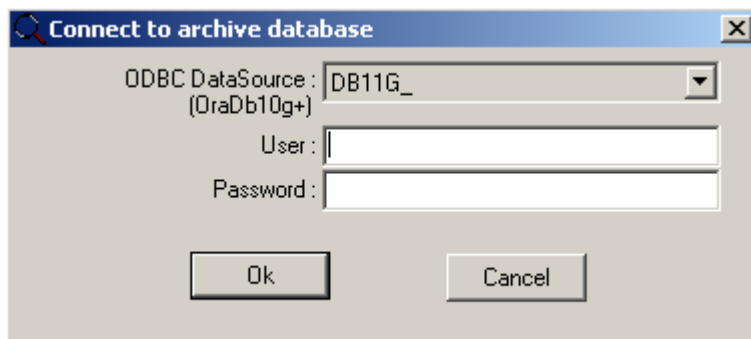
The «PORTFOLIO» application developers can only view the performance results of their own stored procedures. Do not hesitate to contact [DATABASE Tuning's technical support](#) to validate your criteria concerning the DATABASE Tuning's database model and performances using indexes.

Connecting to the archive database

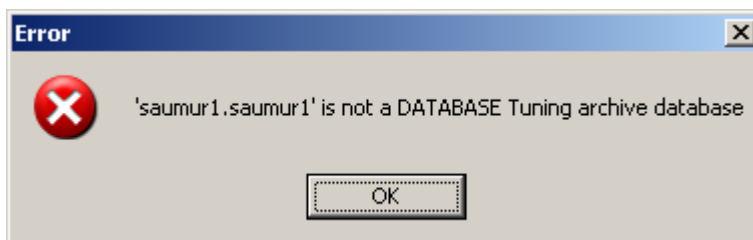
From the following dialog box, select the required location for the archive.



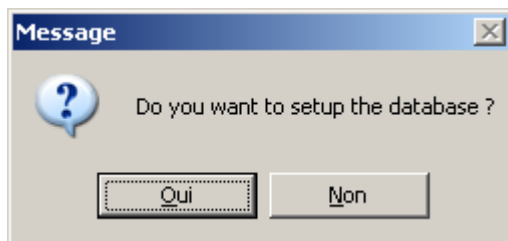
After selecting «Database», you connect to it.



If the database is not yet formatted or if its structure has been modified, DATABASE Tuning displays the following message.



Next, DATABASE Tuning suggests that you execute the full database setup. The connected user must be that of the schema owner.

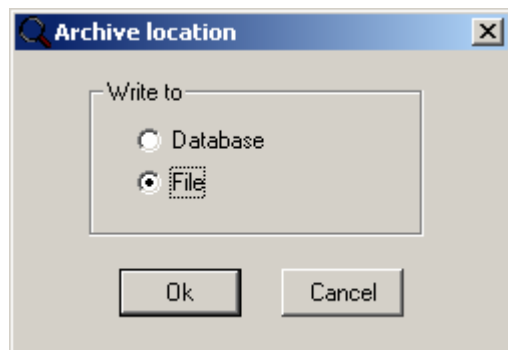


Caution: The DATABASE Tuning package does not come with a specific procedure to restore data. If you choose: «yes» (database setup), all previous data will be lost. While saving and restoring databases, use the tools provided and certified by ORACLE.

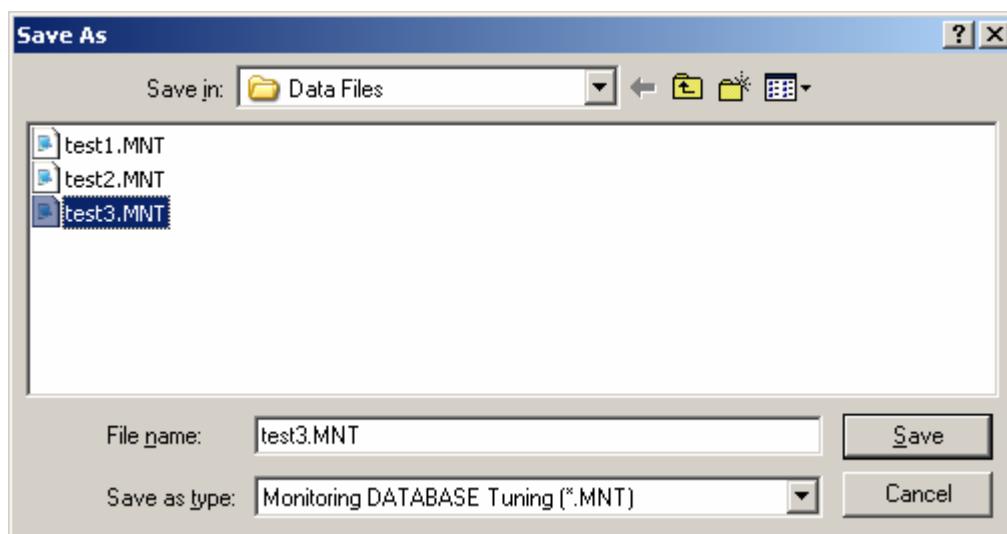
How to save to a file

For performance's sake, the archive file containing the monitoring results should not be located on a WAN (from the DATABASE Tuning version recording the results). We recommend that you not create this file on a production server. You need to allow for 10 Kb per captured SQL Statement, on average, and for 1 Mb, for Adaptive Servers' indicators values, per monitoring hour. As an example, it is typical to capture more than 5000 SQL Statements per day, on a targeted production instance experiencing performance problems. This represents a total volume of 50 Mb for the SQL Statements and of 25 Mb for the instance's indicators values. Approximately 75 Mb are delivered, each monitoring day. It is imperative that you manage the free space that is on the archive directories.

From the following dialog box, select the required location for the archive.

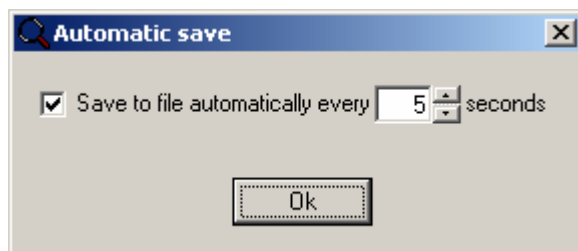


After selecting «File», you need to define the location and name of the file. The default location is «%APPDATA%\DATABASE Tuning\Data Files» directory.



How to save automatically

When a monitoring session is active, a window appears to setup the session's reports automatic-save frequency which is defined in seconds. The newly captured SQL Statements are then automatically saved to the file at the selected frequency. The automatic-save frequency operates independently from the monitoring frequency. (and the latter cannot be modified.).



Exports

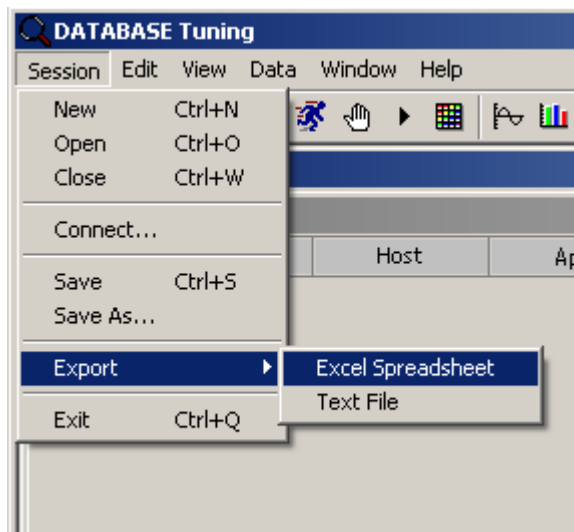
See also:

How to export to Excel

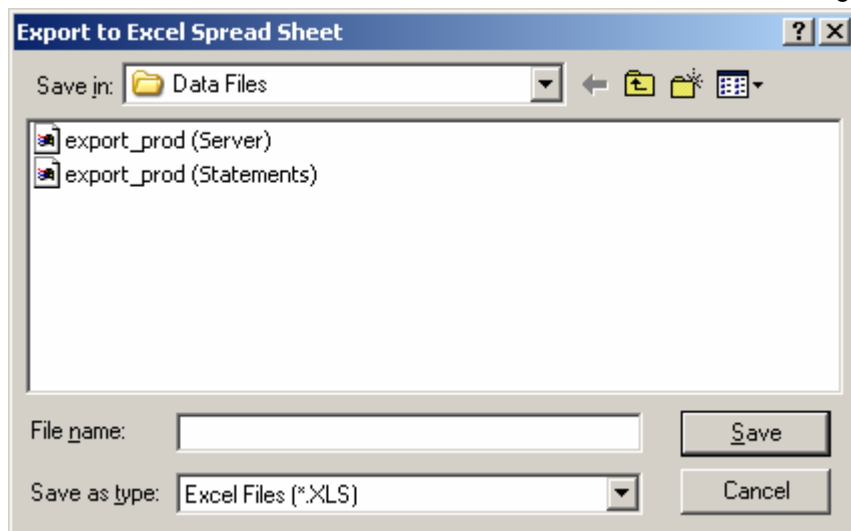
How to export to a text file

DATABASE Tuning enables you to export all the SQL Statements or part thereof to a file (in Excel or Text format). Instead of exporting results to a database, you can save them to a database (see paragraph on «How to save to a database»). To do this, select the SQL Statements you wish to export. Next, select the «Session - Export» menu item and determine the file format required to export the data.

How to export to Excel

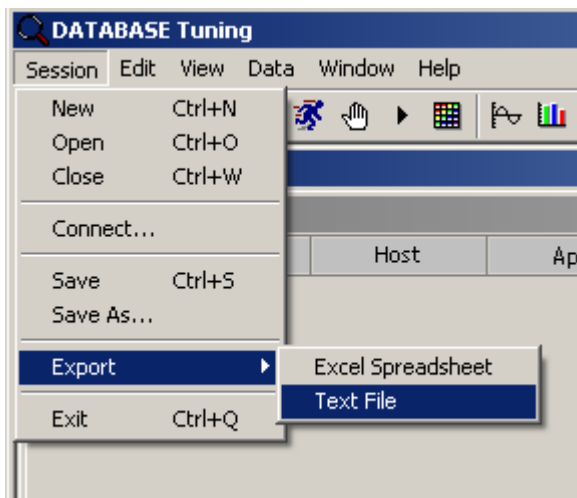


Enter a file name in the directory of your choice. The default location is «%APPDATA%\DATABASE Tuning\Data Files» directory.

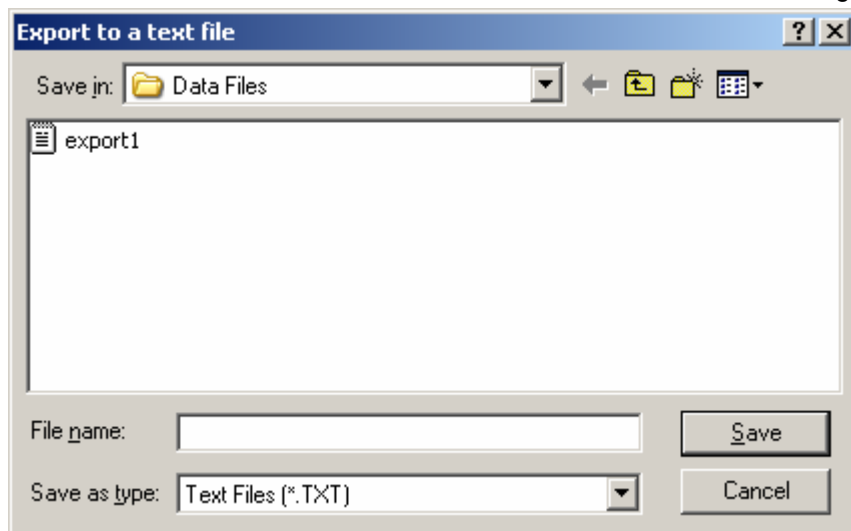


The export generates two files: < file name > (Statements).XLS, which contains the lines of the session's report and < file name > (Server).XLS, which contains the instance's performance indicators' values.

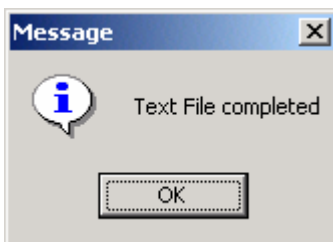
How to export to a text file



You enter a file name in the directory of your choice.



When the exporting is complete, the following message appears:



The export generates a file: < name of file>.TXT in ASCII format, which contains the lines of the session's report. The instance's performance indicators' values are not exported to the text file.

How to retrieve results

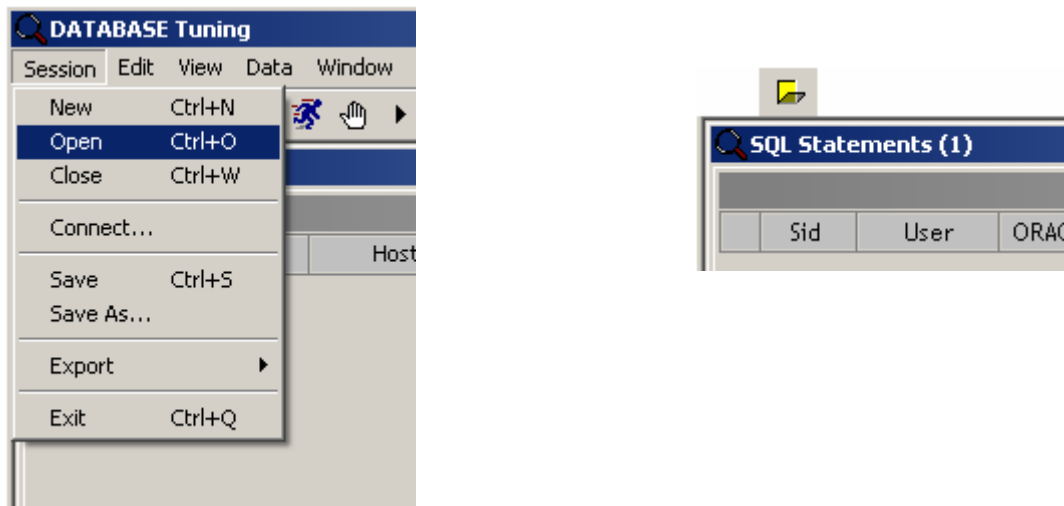
See also:

How to retrieve from a database

How to retrieve from a file

Automatic refresh

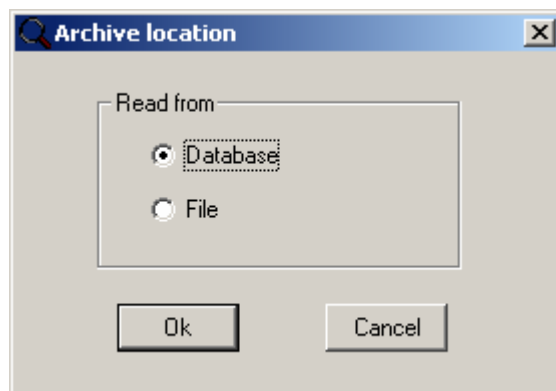
To open an archive, all you need to do is select the «Session - Open...» menu item, or click on the «Open» menu icon (see Figures below).



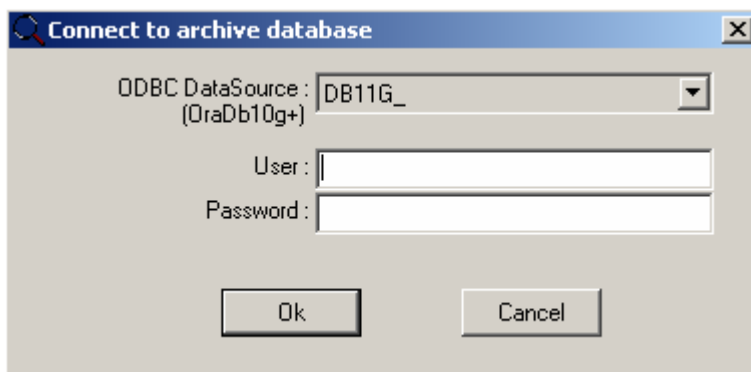
DATABASE Tuning allows the following two archive formats: ASCII files and ORACLE dedicated databases.

How to retrieve from a database

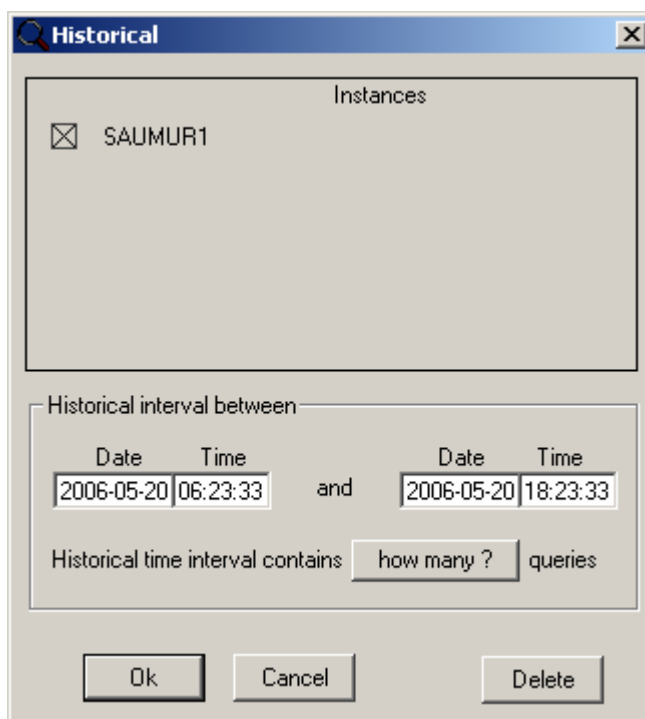
From the following dialog box, select the required location for the results.



After selecting «Database», you connect to the archive database containing the monitoring results recorded by DATABASE Tuning.



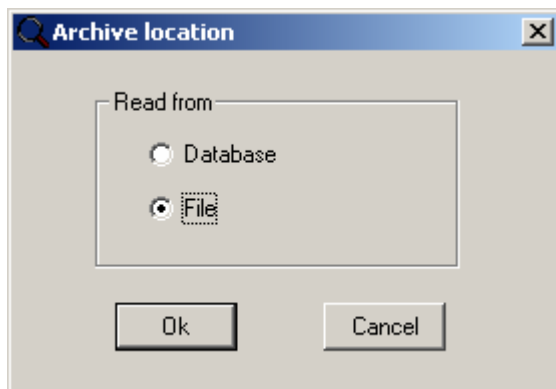
The dialog box below enables you to extract the required data.



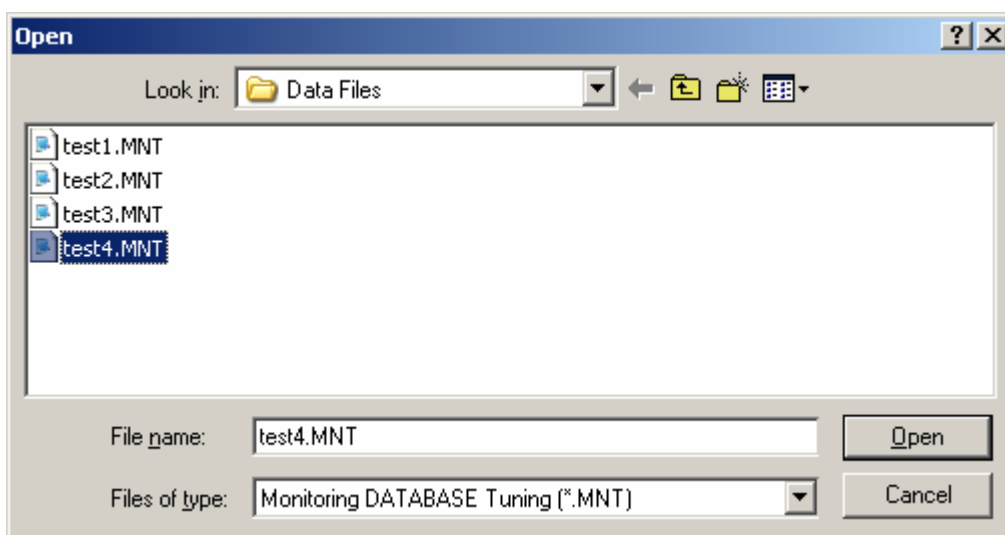
The «instances» list box contains the various ORACLE instances' names for the monitoring data that exists in the archive database. Define the time interval needed. Thanks to the «how many?» button, you can find out how many SQL Statements are present in the database according to these criteria. Check that your PC workstation's memory is not saturated with a bulky session's report. When you click on this button, the number of SQL Statements replaces the previous string («how many?»). The «Delete» feature is explained in the paragraph on «Results deletions».

How to retrieve from a file

From the following dialog box, select the location required.



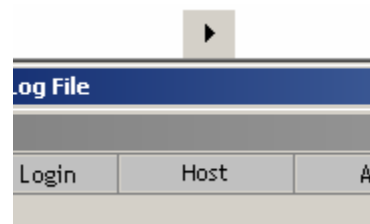
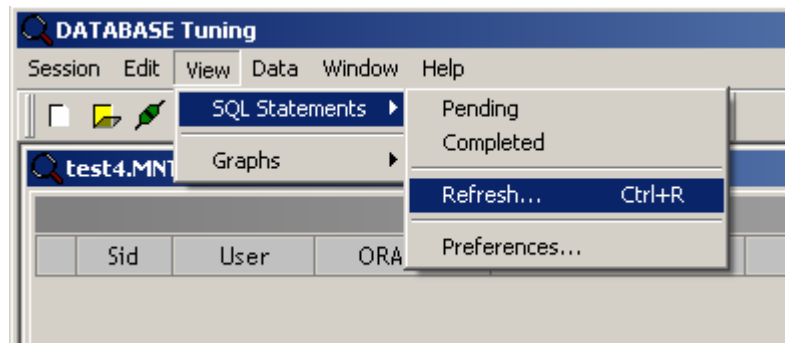
After selecting «File», define the directory and the file name containing the DATABASE Tuning's monitoring results. By default, the «.MNT» files containing performance data collected by DATABASE Tuning are stored in the «%APPDATA%\DATABASE Tuning\Data Files» directory.



All SQL Statements contained in this file are loaded in the session's report.

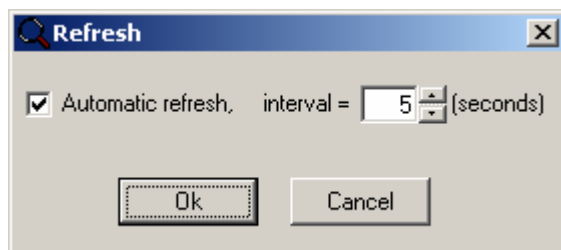
Automatic refresh





It is possible to read a file that is currently being processed by an active monitoring session running on a different PC workstation. To perform an automatic refresh of the display, select the «View - Refresh...» menu item. Alternatively, you can click on the «Refresh» menu icon (See Figures below).



When you activate the automatic refresh function of the display, a dialog box asks you to define the required

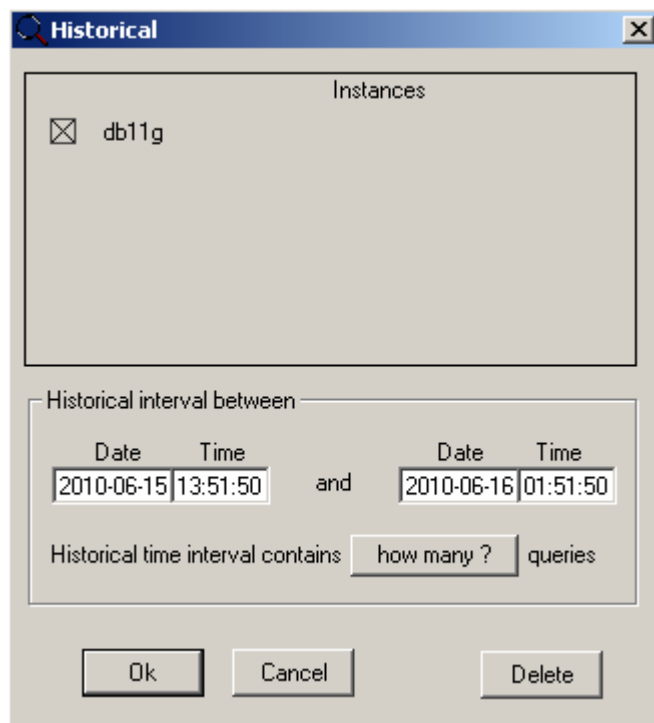
refreshing frequency. In the direct monitoring mode (DATABASE Tuning is connected to the ORACLE instance), the automatic refresh function is performed with the same frequency as the monitoring. In this case, the refreshing frequency cannot be modified. However, you can interrupt the refreshing at any time.



As soon as the automatic refresh function is activated, the refresh icon goes from the «read» mode  to the «pause»  mode. The session's connection (See Paragraph on «How to start and stop monitoring ») and the automatic refresh function of SQL Statements' display (See Paragraph on «Contents of the session's report») run independently from each other. What this means is that you cannot view the details of the SQL Statements of the «Completed» report when the Automatic Refresh function is active (icon , and when the «Completed» report is greyed out). However, you can interrupt the automatic refresh and resume it ( icon) without losing the results of the monitoring session in progress. During the interruption, the captured SQL Statements are saved in the background and you can view the details of the SQL Statements in the reports. Former releases allowed you to view the details of the SQL Statements (in a pending state) in real time, while the Automatic Refresh function was active.

Results deletions

From this window, you can activate the deletion of the monitoring results (SQL Statements and instance's indicators' values) of a given archive database (See below: «Delete» button). The SQL Statements and the ORACLE instance's performance indicators' values associated with the selected instances in question and the selected time period are removed.

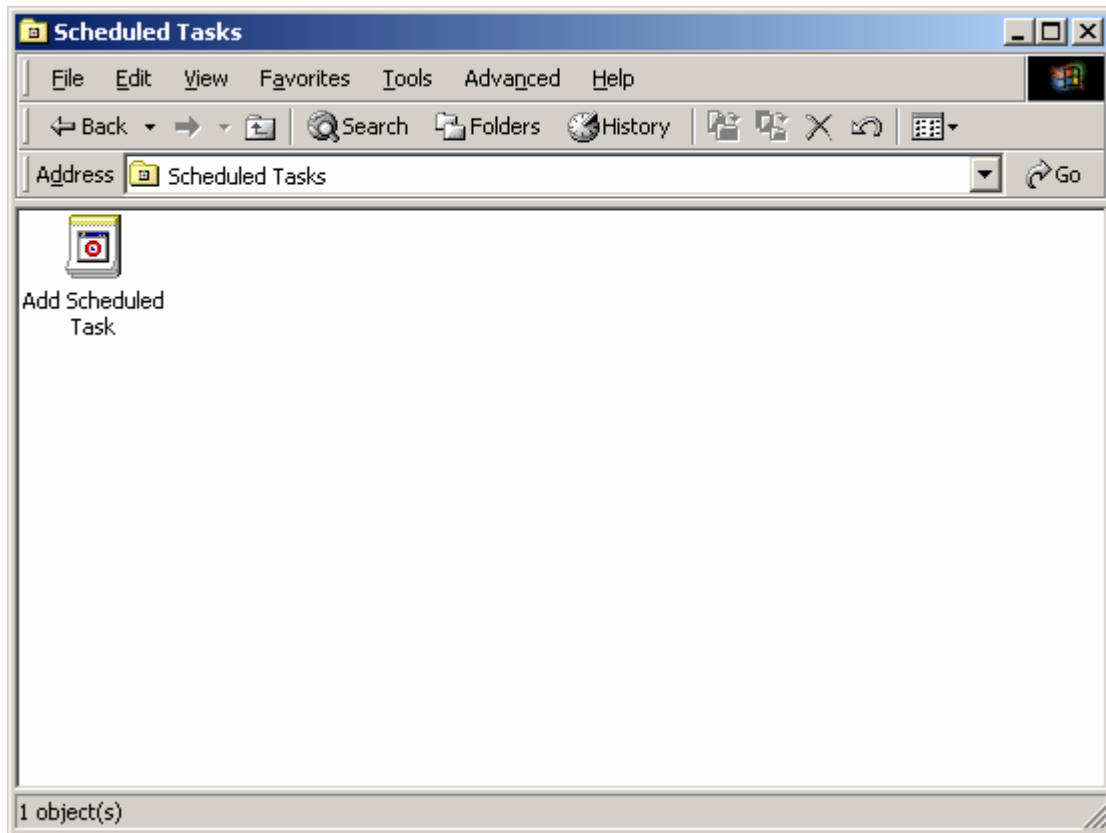


Executing DATABASE Tuning in a background task

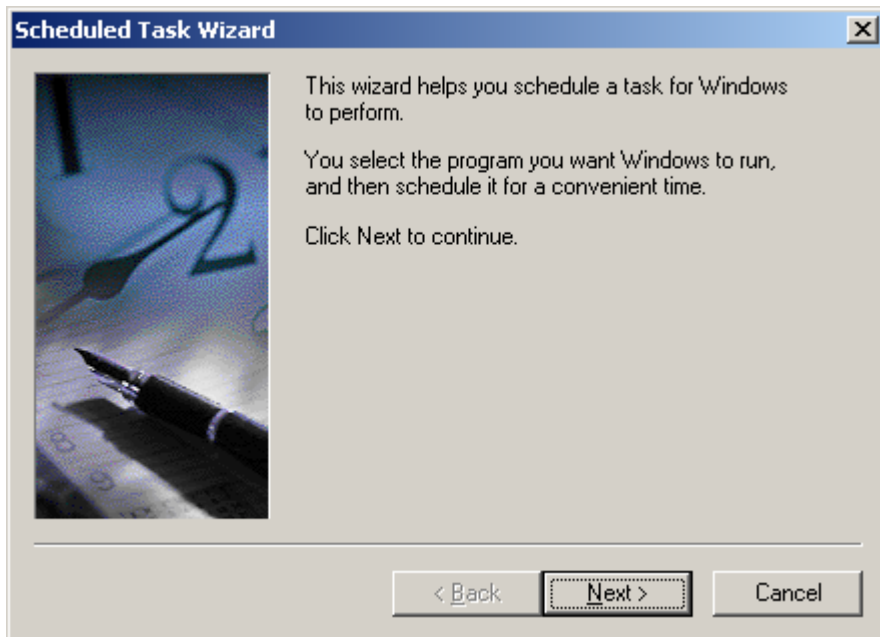
You can launch DATABASE Tuning in a background task. To do so, you need to create a scheduled task. Next, use the background task's execution parameters. This feature is only available in the DATABASE Tuning's Enterprise Edition.

Creating a scheduled task

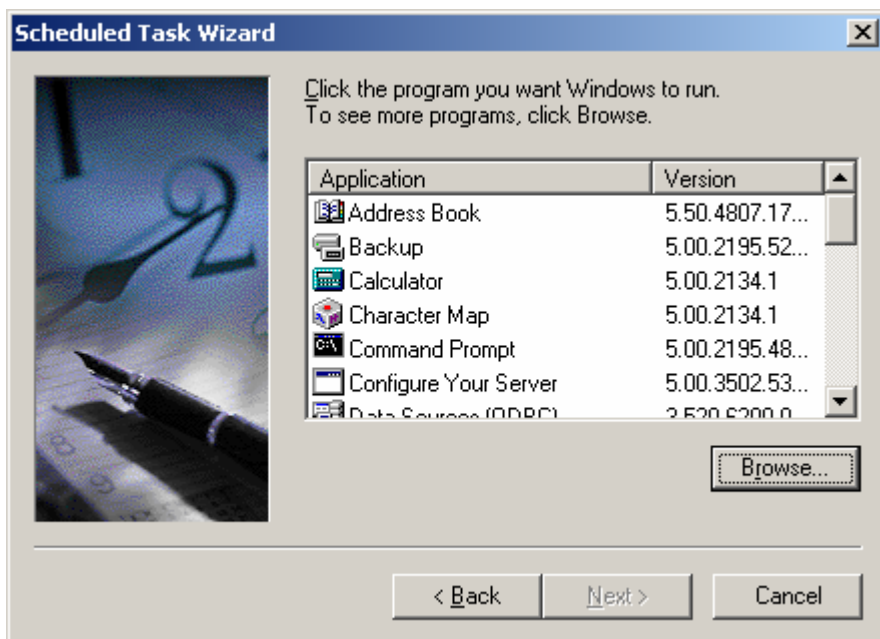
The following example details the steps to follow. From the «Start - Settings - Control panel» menu, select the «Scheduled Tasks» utility (to setup an automatic process execution in a background task of your computer). The window dedicated to the management of scheduled tasks helps you create a new task.



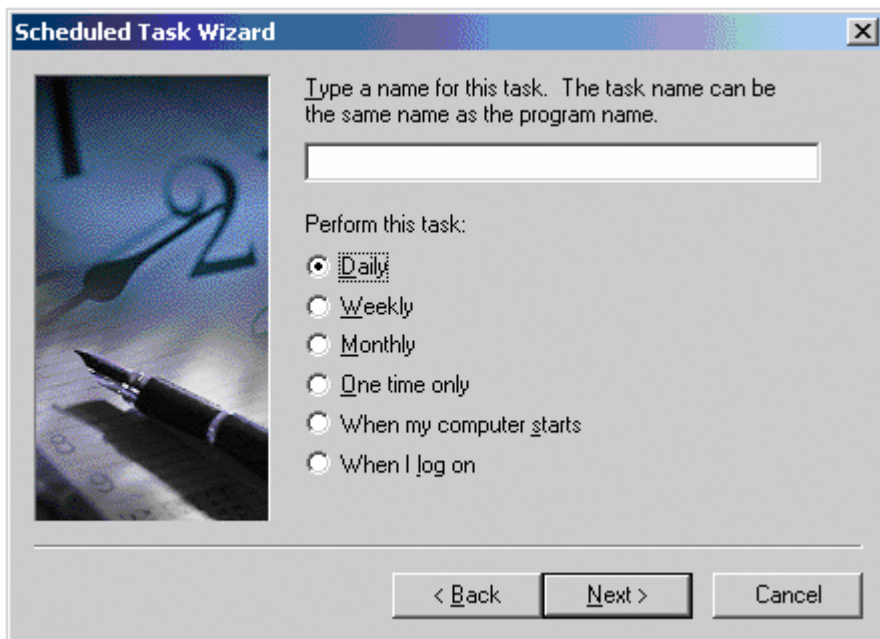
Next, click on «Add Scheduled Task».



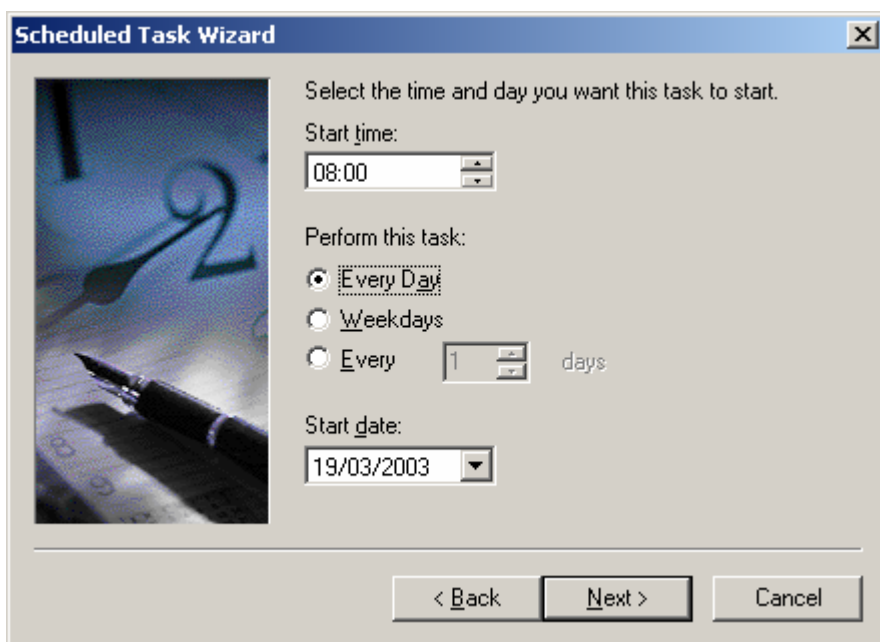
Now, click on the «Next>» button.



Click on the «Browse...» button to look for the DATABASE Tuning's EXE file (dbt42.exe), which is in the installation directory. Next, select the launch options. We recommend that you define a daily execution, especially if you intend to record the results to a file, so that it is not too bulky.



You can now define the time when you want the process to start.



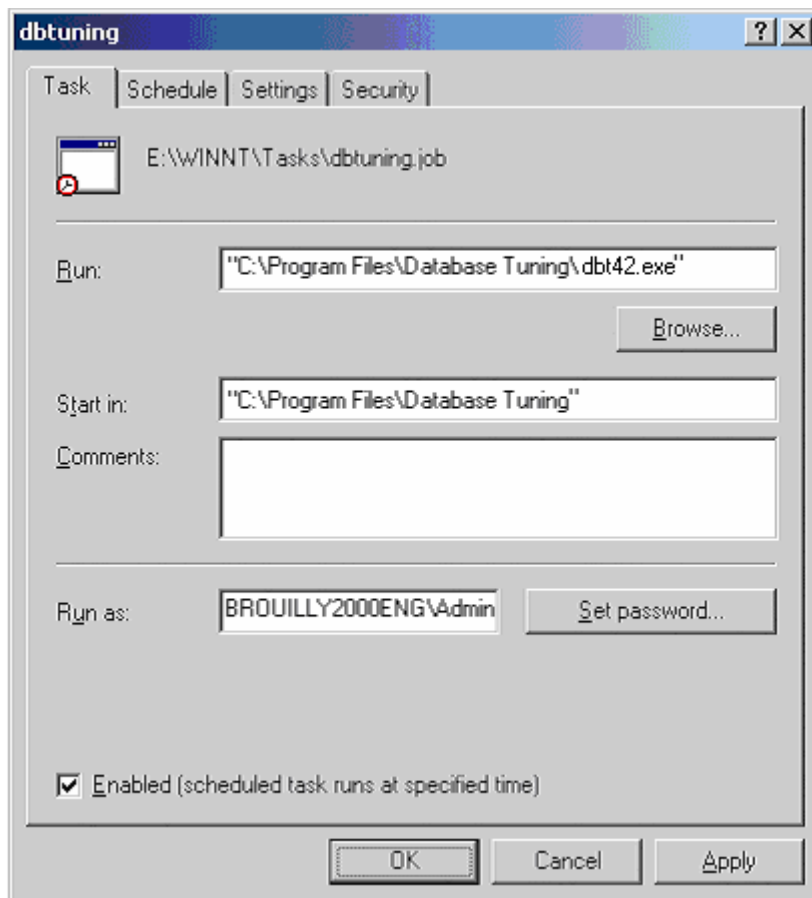
Enter the login of the background task's owner (the person who is launching the scheduled task). It is preferable to use a login that is not set to a specific person, in order to avoid any inappropriate password modification.



The scheduled task creation is almost complete.



You now need to complete the command line (in the following «Run:» window) using DATABASE Tuning's background task execution's options described below. The «Start in:» edit line defines the execution's directory. The DATABASE Tuning's alertlog file named «DATABASE Tuning.log» is stored in the «%APPDATA%\DATABASE Tuning\log» directory. It contains DATABASE Tuning's execution traces and all messages related to potential incidents.



Parameters of the background task execution

- [-D] session's duration (minutes)
- [-F] filters file (by default, this file is located in the «%APPDATA%\DATABASE Tuning\cfg\flt» directory
- [-L] license level
- [-O] archive file
 \$\$dd\$ day_month_year
 \$\$hh\$ hour-minute-second
 example:
 ORA_PRD_\$\$dd\$_\$\$hh\$.MNT generates the file:
 ORA_PRD_05-11-2009_15-01-03.MNT
- [-P] password of the user connected to the monitored ORACLE instance
- [-S] monitored ORACLE instance's name (ODBC datasource name)
- [-U] user that connects to the monitored ORACLE instance
- [-V] application is visible (YES/NO), NO is the default
- [-W] user connected to the ORACLE instance hosting the archive database
- [-X] password of the user connected to the ORACLE instance hosting the archive database
- [-Y] archive ORACLE instance's name (ODBC datasource)
- [-I] snapshot frequency for the SQL Statements (milliseconds), default value = 1000
- [-J] snapshot frequency for the ORACLE instance's performance indicators' monitoring (milliseconds), default value = 15000
- [-K] Option used to enforce snapshot frequency under 1 second
- [-H] timed_statistics (TRUE/FALSE), this system parameter to be activate or not.

Example 1:

The following command creates a session whose task is to monitor the «ORAPROD» instance for 10 hours. The results are recorded in a file located in the execution directory. The name of this file is built dynamically from the current date and hour at which the program is launched.

```
«C:\Program Files\DATABASE Tuning\dbt42.exe» -D600 -L5000 -OORAPROD_$$dd$_$$hh$.MNT -P***** -SORAPROD -USYS
```

Example 2:

The following command creates a session whose task is to monitor the «ORAPROD» instance for 10 hours. The results are recorded in a database named «ORAARCH». DATABASE Tuning connects to the latter with the user «scott». The snapshot's frequency for the queries is 500 milliseconds, the -K option is mandatory to enable DATABASE Tuning to launch a session with that snapshot frequency, i.e. faster than 1 second (see chapter on «How to start and stop monitoring»). Note that "as sysdba" is part of the password parameter.

```
«C:\Program Files\DATABASE Tuning\dbt42.exe» -D600 -L5000 -P***** as sysdba -SORAPROD -USYS -Wscott -X***** -YORAARCH
```

You now need to replace the «Run: « edit line with your own command line, based on the method illustrated in the above examples.

The «.MNT» files containing performance data collected by DATABASE Tuning are stored in the default directory named «%APPDATA%\DATABASE Tuning\Data Files».

How to shutdown a session in background mode

When launched in the background mode, DATABASE Tuning creates a file called <monitored instance>.tmp in the «%APPDATA%\DATABASE Tuning\log» directory. It is controlled by DATABASE Tuning at every cycle of the scan Interval. If this file is missing, DATABASE Tuning stops the session started in the background mode. To stop a DATABASE Tuning session launched in the background mode, it is imperative to carry out a manual removal of this file. Please do not cancel the process from the task manager.

Tables

See also:

ssv_stmt

ssv_batch

ssv_object

ssv_textobject

ssv_exeplan

ssv_textplan

ssv_monitor

ssv_monitor_servername

ssv_monitor_type_info

ssv_version

ssv_stmt

Each line of this table corresponds to a line in the session's report. The properties of the connection associated with SQL Statement as well as the attributes of performances are present in this table.

| Column of the table | Column of the queries audit report | Description |
|---------------------|--|---|
| stmtid | | Primary key, single identifier in this table. This number does not appear in the session's report in the graphical interface of DATABASE Tuning. The line number displayed in the session's report corresponds to a numbering method appearing in the tabular window, as in MS Excel, to establish a link with the graph of the SQL Statements |
| keymonitor | | DATABASE Tuning's license key from the PC workstation, which is connected to the ORACLE instance. This information can be used to detect if several stations are connected to the same ORACLE instance. |
| status | | "1" = SQL Statement is pending, "0" = SQL Statement is completed. |
| spid | Sid | The Process ID, which executes the SQL Statement. |
| kspid | | The Kernel Process ID, which executes the SQL Statement. It is an internal identifier of the ORACLE instance that is incremented and allotted to each new connection. |
| stmt_batchid | | The query's number, within a client connection. This batch's identifier is augmented each time a new query is sent by a client PC workstation within a connection. |
| login | User | User name of the connected user. |
| servername | ORACLE instance | The monitored ORACLE instance's name. |
| hostname | Client Host | Name of the client PC workstation that sent the query to the ORACLE instance. |
| appname | Application | Name of the application that sent the query to the ORACLE instance. |
| date_time | Date Time (Start) Date Time (End) | In case of pending SQL Statements, it is the date and time at which the execution started. For completed SQL statements, it is the date and time at which the execution ended. |
| batchid | Batch Text (Put the following tables together: «ssv_stmt » and «ssv_batch») | Foreign key used to establish a link with the "ssv_batch" table. |
| sql_stmtid | SQL Statement (Put the following tables together: «ssv_stmt» and «ssv_batch») | Foreign key used to establish a link with the "ssv_batch" table. |
| objectid | Procedure - Trigger - Package (Put the following tables together: «ssv_object» and «ssv_textobject») | Foreign key used to establish a link with the "ssv_object" and "ssv_textobject" tables. If "objectid" = "-1", the recorded SQL Statements belong neither to a stored procedure nor to a trigger nor a package and they correspond to a query sent as part of a dynamic SQL. Please note that the reverse is untrue: a recorded SQL Statement can belong to a stored procedure executed in a trigger that is activated by a query sent as part of a dynamic SQL. |
| textoff | | Not used with ORACLE. |
| exeplanid | Execution plan (Put the following tables together: «ssv_exeplan» and «ssv_textplan») | Foreign key used to establish a link with the "ssv_exeplan table". |
| logical_reads | Logical reads | Number of reads (I/Os in memory) consumed by the SQL Statement. |
| physical_reads | Physical reads | Number of physical reads consumed by the SQL Statement. |
| elapsed_time | Elapsed Time | Elapsed time spent by the SQL Statement. |
| cpu_time | CPU Time | Computing time consumed by the SQL Statement. |
| lock_g_not | Locks | Number of locks not allocated immediately with the execution of the SQL Statement. |
| commentid | | Not used with ORACLE. |

ssv_batch

The rows of this table correspond to SQL queries sent by the client PC workstations which are at the origin of the execution of the SQL Statements captured by DATABASE Tuning. There is thus a (1, N) association, N for the number of line of the SQL query, between the table «ssv_stmt» and the table «ssv_batch».

| Column of the table | Column of the queries audit report | Description |
|---------------------|--|--|
| batchid | | This column belongs to the primary key (batchid, linenum). This identifier is used to establish a link with the foreign key "batchid" of the "ssv_stmt" table. |
| linenum | | This column belongs to the primary key (batchid, linenum). The text of the query is cut out in several lines of 255 characters. |
| linelength | | Length of the text line. |
| linetext | SQL Statement (after the address inside the source code has been calculated, when a query was sent in dynamic SQL mode) | The query's text line. |

ssv_object

The rows of this table correspond to the stored procedures or triggers which contains SQL Statements captured by DATABASE Tuning. There is thus a (0, 1) association between the table «ssv_stmt» and the table «ssv_object». Indeed, there is no correspondence (cardinality 0) between the two tables when impacted SQL Statement is from dynamic SQL.

| Column of the table | Column of the queries audit report | Description |
|---------------------|---|---|
| objectid | | Primary key of the table. This identifier is used to establish a link with the foreign key "objectid" of the "ssv_stmt" and "ssv_textobject" tables. |
| servername | | Name of the ORACLE instance containing the object. |
| dbname | | Database name. |
| ownername | Procedure - Trigger - Package (join with the table « ssv_stmt ») | Name of the owner of the object. |
| procname | Procedure - Trigger - Package (join with the table « ssv_stmt ») | Name of the object (procedure, trigger, or package). |
| type | | Type of the object (procedure, trigger or package). |
| crdate | | Date on which the stored procedure or trigger or package was created. The versions are archived, which makes it possible to compare the performances recorded by DATABASE tuning on the various versions of a stored procedure or trigger or package. |

ssv_textobject

The rows of this table are the sources code of the stored procedures or triggers which contains SQL Statements captured by DATABASE Tuning. There is a (1, N) association, N being the number of lines of the source code, between the table «ssv_stmt» and the table «ssv_object».

| Column of the table | Column of the queries audit report | Description |
|---------------------|--|---|
| objectid | | This column belongs to the primary key (objectid, linenum). This identifier is used to establish a link with the foreign key "objectid" of the "ssv_stmt" table and the primary key "objectid" of the "ssv_object" table. |
| linenum | | This column belongs to the primary key (objectid, linenum). The text of the source code (stored procedure or trigger or package) is cut out in several lines of 255 characters. |
| linelength | | Length of the text line. |
| linetext | SQL Statement (after the address inside the source code has been calculated, when the SQL statement is part of a PL/SQL object) | Text line of the source code of the stored procedures, the triggers and the packages. |

ssv_exeplan

The rows of this table correspond to execution plans of the SQL Statements captured by DATABASE Tuning. There is a (1, N) association, N being the number of lines of the execution plan, between the table «ssv_stmt» and the table «ssv_exeplan».

| Column of the table | Column of the queries audit report | Description |
|---------------------|--|---|
| exeplanid | | This column belongs to the primary key of the table (exeplanid, linenum). This identifier is used to establish a link with the foreign key "exeplanid" of the "ssv_stmt" table. |
| linenum | | This column belongs to the primary key (exeplanid, linenum). The execution plan is composed of several lines of 255 characters. |
| lineid | Execution Plan (Put the following tables together: « ssv_stmt » and « ssv_textplan ») | Foreign key linked to the primary key "lineid" of the "ssv_textplan" table. |

ssv_textplan

The rows of this table correspond to the lines (example: to gain place, and because the literal ones of the execution plans are very repetitive, in the same way for the others, is recorded only once in the archive database) of the execution plans. There is a (1, 1) association between the table «ssv_exeplan» and the table «ssv_textplan». Indeed, the table «ssv_textplan» contains the character strings of the execution plans of table «ssv_exeplan».

| Column of the table | Column of the queries audit report | Description |
|---------------------|---|---|
| lineid | | Primary key of the table. This identifier is used to establish a link with the foreign key "lineid" of the "ssv_exeplan" table. |
| linetext | Execution Plan (Put the following tables together: « ssv_stmt » and « ssv_exeplan ») | "Standard" line of execution plan text. Example: "FULL SCAN" |

ssv_monitor

The rows of this table contain the values of performances indicators collected on the level of ORACLE instance. These values are consultable in interface DATABASE Tuning with graphs.

| Column of the table | Description |
|---------------------|---|
| date_time | Timestamp of the value for the indicator defined by "id_type_info". |
| id_servername | Foreign key linked with to the "id_servername" primary key of the "ssv_monitor_servername" table. |
| id_type_info | Foreign key linked with the "id_type_info" primary key of the "ssv_monitor_type_info" table. |
| info | value |
| status | 0 |

ssv_monitor_servername

The rows of this table contain the names of the monitored ORACLE instance.

| Column of the table | Description |
|---------------------|-------------------------|
| id_servername | Instance identifier. |
| servername | ORACLE instance's name. |

ssv_monitor_type_info

The rows of this table contain the names of the performance metrics.

| Column of the table | Description |
|---------------------|---|
| id_type_info | Indicator identifier. |
| type_info | Indicator's name. CNX CPU BUF SPACE REQ CKPT PHYS IO REDO LOG COMMITTS SORTS FULL SCANS FRAG PARSE NET RECEIVED NET SENT CACHE HIT RATIO DEADLOCKS BLK SESSIONS ROLLBACK (For further details, See Chapter on « ORACLE instance's indicators ») |

ssv_version

This table contains one row; this is the version of DATABASE Tuning archive model.

| Column of the table | Description |
|---------------------|--|
| Version_text | Character string characterizing the version. Run the following in the database: select * from ssv_version You obtain the following result: « DATABASE Tuning Database - 6.0.00 - Feb 15 2010 » or later. |