

Users Manual

SQLShotTM

*For Microsoft SQL Server
Including SQL Server 2008 R2*



<u>About SQL Shot</u>	<u>5</u>
<u>SQL Shot's Technical Support</u>	<u>6</u>
<u>From Installation to Monitoring</u>	<u>7</u>
<u>Prerequisites</u>	<u>8</u>
Architecture	8
Microsoft SQL Server	8
Client applications	8
SQL Shot 4.2	8
Application data	9
%APPDATA%	9
« .MNT » files	9
« .XLS » files	9
« .TXT » files	9
« .FLT » files	9
« .GPH » files	9
« SQL Shot.log » file	9
« .tmp » files	9
« .sqlplan » files	9
<u>Installation</u>	<u>10</u>
<u>License key registration</u>	<u>12</u>
<u>How to start and stop monitoring</u>	<u>14</u>
ODBC DataSource for SQL Server configuration	14
Verification that SQL Shot is properly set up	18
Monitoring session's shutdown	19
<u>SQL Statements' monitoring</u>	<u>20</u>
Session report's contents	21
SQL Statement's detail	22
Detail - General	24
Detail - SQL Statement	25
Detail - Batch Text	26
Detail - Execution Plan	27

Detail - Procedure - Trigger - Function	28
Detail - Server Performances	29
Detail - Print	30
SQL Statements' graphs	31
Filters	33
Filter - Spid	35
Filter - Login	36
Filter - Host	37
Filter - Application	38
Filter - Date - Time	39
Filter - SQL Statement	40
Filter - Batch Text	41
Filter - Procedure - Trigger - Function	42
Filter - Execution Plan	43
Filter - Performances	44
Filter - How to manage filter files	44
Display options	44
Sort	46
New Session	47
<u>SQL Server's performances' monitoring</u>	48
The SQL Server's indicators	49
How to restore a configuration	49
How to add an indicator	51
How to modify an indicator	51
How to remove an indicator	52
How to save the configuration	53
How to refresh the graph	53
SQL Statements related to a time period	54
Graphs' display options	55
New graph	55
<u>How to save results</u>	57
How to save to a database	57
Technical recommendations	57
Permissions management	57
Connect 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
<u>How to retrieve results</u>	63
How to retrieve from a database	63
How to retrieve from a file	64
Automatic refresh	65
<u>Results' suppressions</u>	67

SQL Shot's execution in a background task **68**

Create a scheduled task	68
Background task execution	72
How to shutdown the background mode session	73

SQL Shot's archive database model **74**

Tables	75
ssv_stmt	76
ssv_batch	77
ssv_object	78
ssv_textobject	79
ssv_exeplan	80
ssv_monitor	81
ssv_release	82
ssv_timestamp	83

About SQL Shot

Copyright © 2009-2010 by DATABASE Softwares.

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

DATABASE Softwares
2, rue Troyon
92310 Sèvres, France

SQL Server, SQL Server Management Studio, Windows, Windows 95, Windows NT, Windows 2000, Windows Vista, Windows 7 and Windows XP are trademarks of Microsoft Corporation. Sybase PowerBuilder, is a trademark of Sybase 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.

SQL Shot's architecture depends on Microsoft SQL Server. The information collected by SQL Shot are derived from the values sampled by Microsoft's SQL Server. SQL Shot directly collects performance indicators' values on Microsoft SQL Server.

SQL Queries as well as its parameters and all performance information related to the execution context (performance indicators' values at the SQL Server level) are automatically displayed by SQL Shot and can be saved in an ASCII file or in a Microsoft SQL Server database (whose model is provided in the appendix). Thanks to SQL Shot, 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.

SQL Shot is an application that has been developed using Sybase PowerBuilder and Microsoft Visual C++. It can be deployed on any Windows 32bits platform.

For performance monitoring purposes, SQL Shot captures the SQL Statements in pending state exceeding a preset performance threshold and it does so in real time. SQL Shot makes it possible to analyze each SQL statement captured. Impact depends on preset performance thresholds. Thanks to SQL Shot, it is possible to detect the database design's mistakes or the applications' coding errors before they create big production problems. SQL Shot makes it possible to pinpoint the SQL Queries having a real impact on the SQL server's performances. Due to SQL Shot's off-set design, the impact on monitored systems' performances depends exclusively on a preset performance threshold.

For optimization purposes, SQL Shot allows database administrators and project managers to instantly locate SQL Servers' performance problems caused by a heavy load in the execution of SQL statements.

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

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

SQL Shot'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 Launch 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 Architects' support web-page.

You can then, send us any question about SQL Shot licensing, Technical Support or Microsoft SQL Server performance issue.

You can also send us directly an email to support@dbarchitects.com.

From Installation to Monitoring

The following steps need to be completed to set up SQL Shot:

Prerequisites

Installation

License key registration

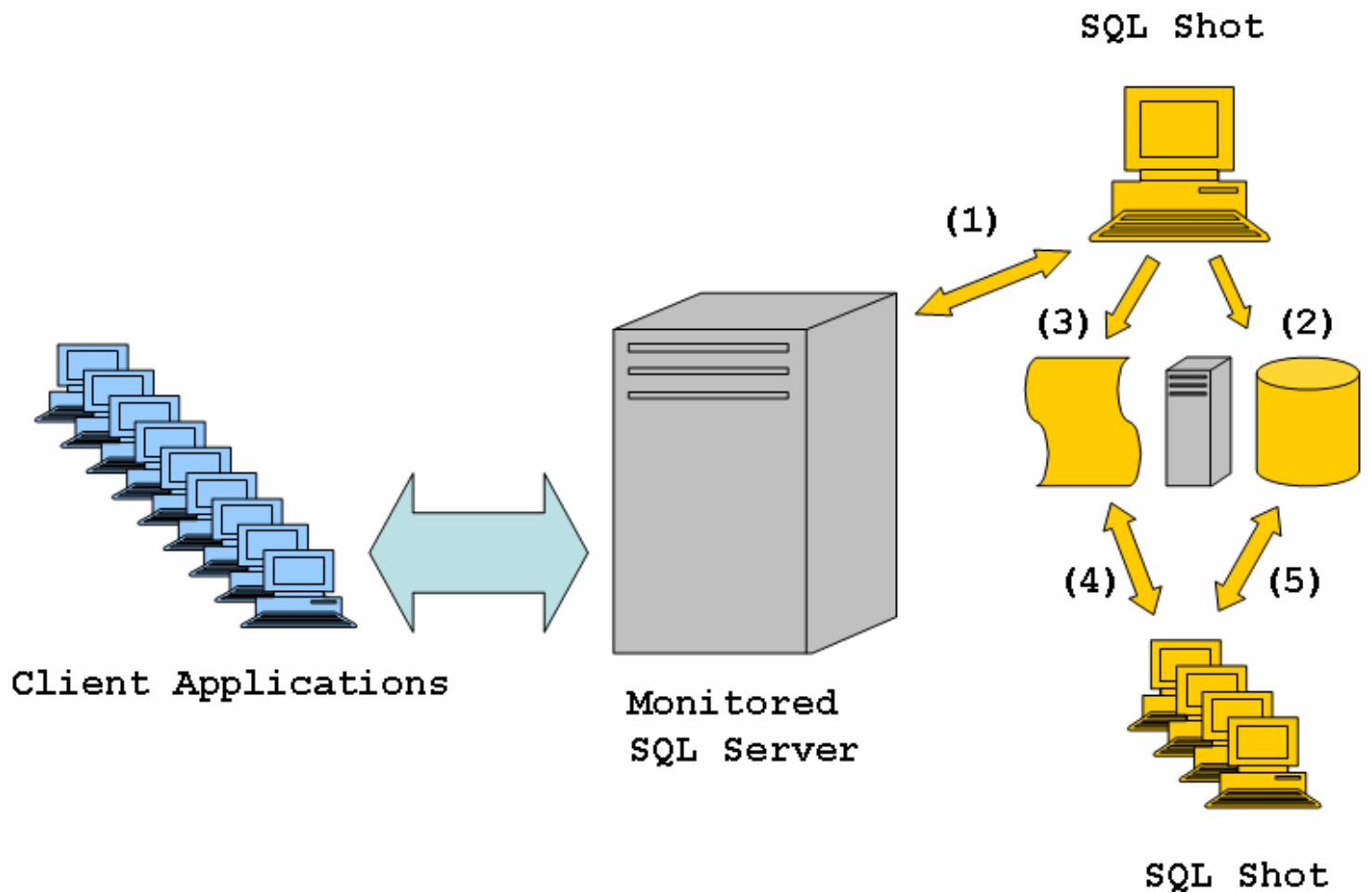
How to start and stop monitoring

Network connections' settings are not part of the SQL Shot package. These operations must have been previously prepared by the Database Administrator. Although these settings are platform specific, this chapter describes the information items that are necessary to set up SQL Shot.

Prerequisites

Architecture
 Microsoft SQL Server
 Client applications
 SQL Shot 4.2

Architecture



- (1) SQL Shot connects to the monitored SQL Server. To make it work, you must define an ODBC user datasource.
- (2) SQL Shot connects to the SQL Server in order to insert the monitoring results in a database and to retrieve (5) them as well. Similarly to monitored SQL Servers, you must define an ODBC user datasource.
- (3) (4) SQL Shot makes it possible to record the monitoring results in a flat file.

Microsoft SQL Server

2005 and 2008 MS SQL Server releases installed on any Windows platform are supported by SQL Shot.

Client applications

No prerequisites.

SQL Shot 4.2

To install SQL Shot, you will need to have a PC workstation running on Microsoft Windows 2003/XP/VISTA. SQL Shot will require a minimum of 25 MB of free disk space. This disk space can be located on the PC's hard disk or on a network's shared disk drive.

Application data

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

See also:

%APPDATA%

« .MNT » files

« .XLS » files

« .TXT » files

« .FLT » files

« .GPH » files

« SQL Shot.log » file

« .tmp » files

« .sqlplan » files

%APPDATA%

For the name of that directory, run the DOS command:

echo% APPDATA%

« .MNT » files

These files contain performance data collected by SQL Shot. These files are stored by default in the %APPDATA%\SQL Shot\Data Files directory.

« .XLS » files

By default, the data exported to Microsoft Excel are stored in the %APPDATA%\SQL Shot\Data Files directory.

« .TXT » files

By default, the data exported to text format are stored in the %APPDATA%\SQL Shot\Data Files directory.

« .FLT » files

By default, the filters' configuration files are created in the %APPDATA%\SQL Shot\cfg\flt directory.

« .GPH » files

By default, these configuration files are present in the %APPDATA%\SQL Shot\cfg\gph directory.

« SQL Shot.log » file

The SQL Shot's errorlog file is in the %APPDATA%\SQL Shot\log directory.

« .tmp » files

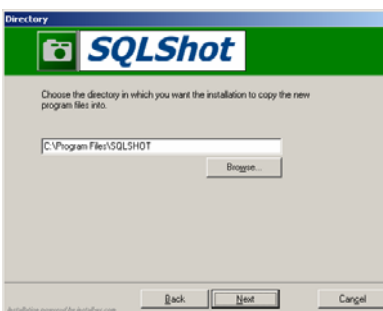
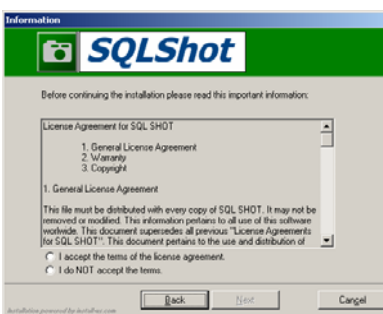
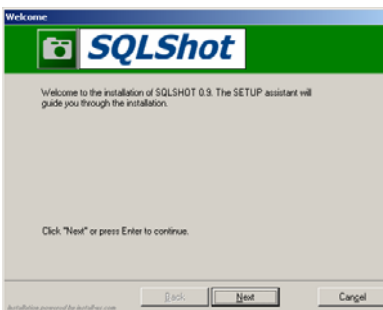
The files used to control background mode executions are created in the %APPDATA%\SQL Shot\log directory.

« .sqlplan » files

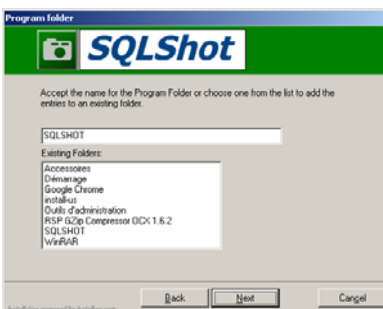
The XML files used to display the execution plans with SQL Server Management Studio are created in the %APPDATA%\SQL Shot\Query Plans directory. These are temporary files that are automatically deleted when older than a day. The execution plans are not lost, they are stored in the archive database or in the historical files.

Installation

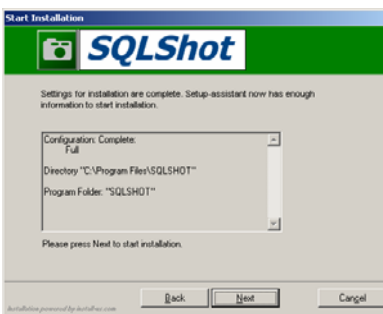
Insert your CDROM into the drive. The installation dialog box appears. Alternatively, run the following program: «<CDROM drive>:\setup.exe”.



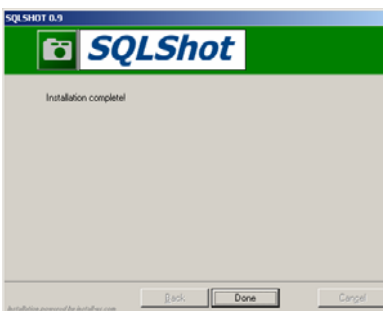
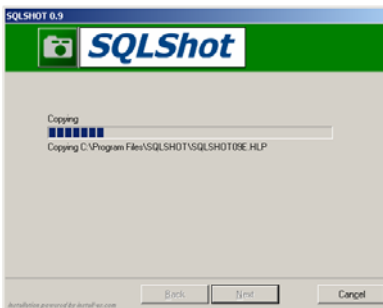
Enter SQL Shot's installation directory. SQL Shot must be installed on a computer able to connect to SQL Server (See Paragraph «About SQL Shot - prerequisites») or access SQL Shot's files or archive database. The SQL Shot's errorlog file, named "SQL Shot.log" is stored in the %APPDATA%\SQL Shot\log directory.



Next, enter the name of the menu where you want to access SQL Shot.



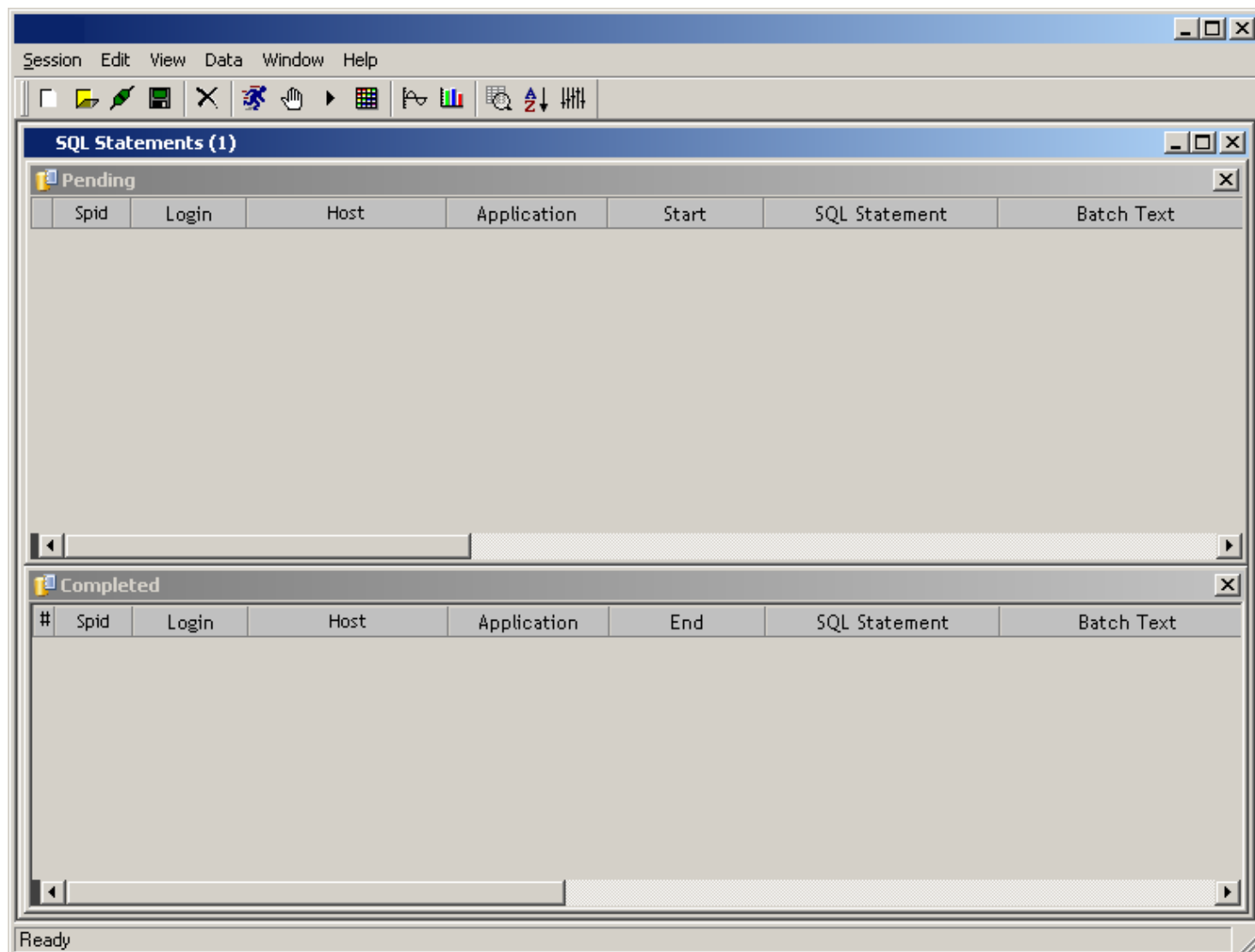
The installation program displays a validation window.



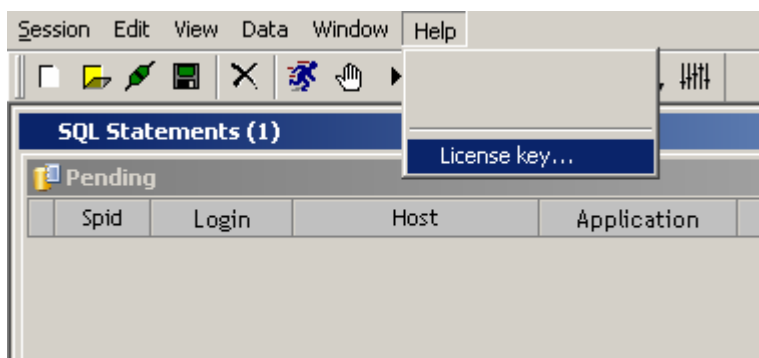
The installation proceeds automatically.

License key registration

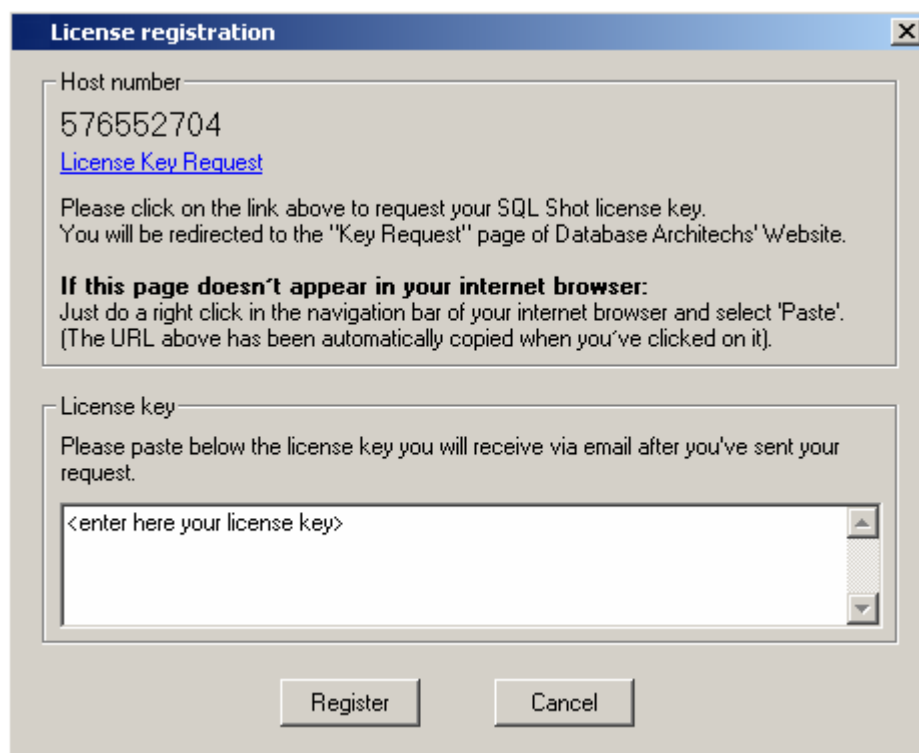
«Start - Programs - SQL Shot» menu, run «SQL Shot» program. The following window appears. Once started, SQL Shot first accesses its initialization file to use the display configuration. This configuration is saved each time you leave the program.



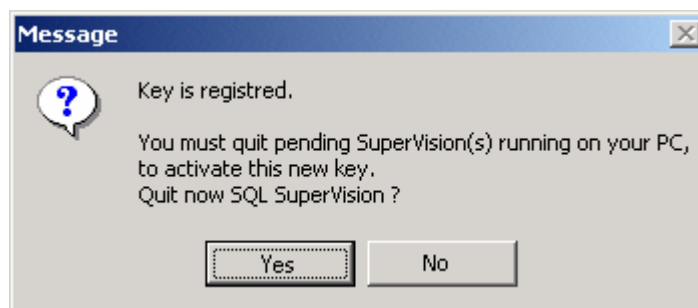
Following installation, you are invited to register your license key. The latter manages monitoring connections to the SQL Server and authorizes a defined number of simultaneous sessions for a specific number of connected users based on the type of license purchased. To obtain your own license key, choose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



The following dialog box appears. In this dialog box, the program displays the PC's identification (host ID). You will need to give this identification to [SQL Shot's Technical Support](#), which, in turn, will give you your SQL Shot's license key (See Chapter on "[SQL Shot's Technical Support](#)"). Next, type your key in the "License key" edit box and click on the «Register» button.



A validation dialog box appears. Quit the program to allow SQL Shot to acknowledge the key.



How to start and stop monitoring

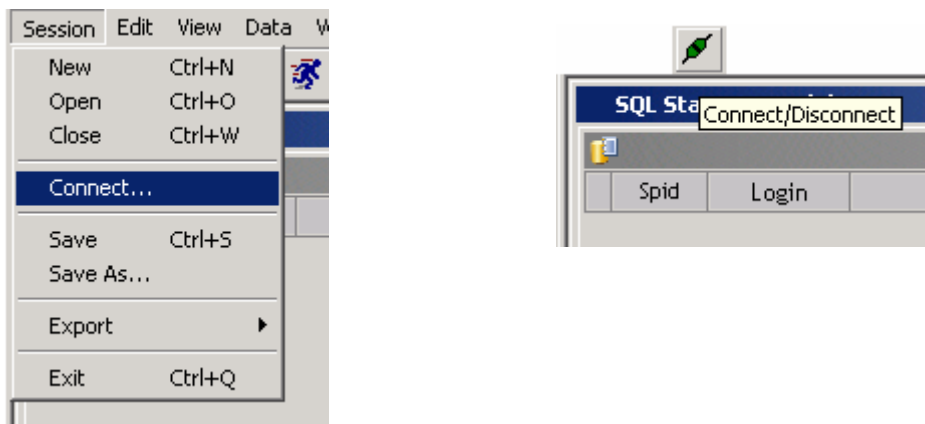
See also:

ODBC DataSource for SQL Server configuration

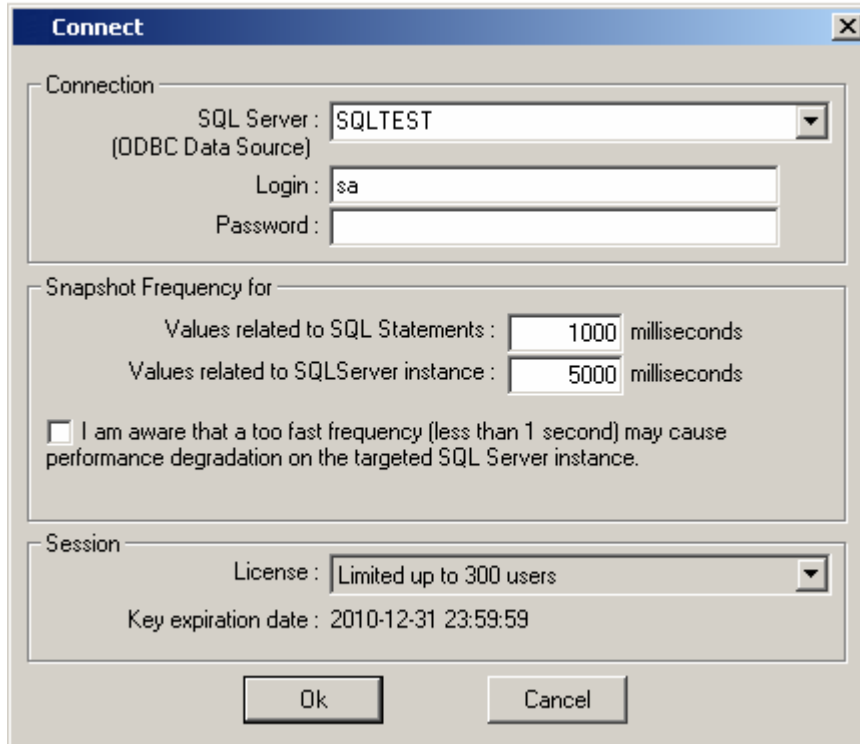
Verification that SQL Shot is properly set up

Monitoring session's shutdown

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



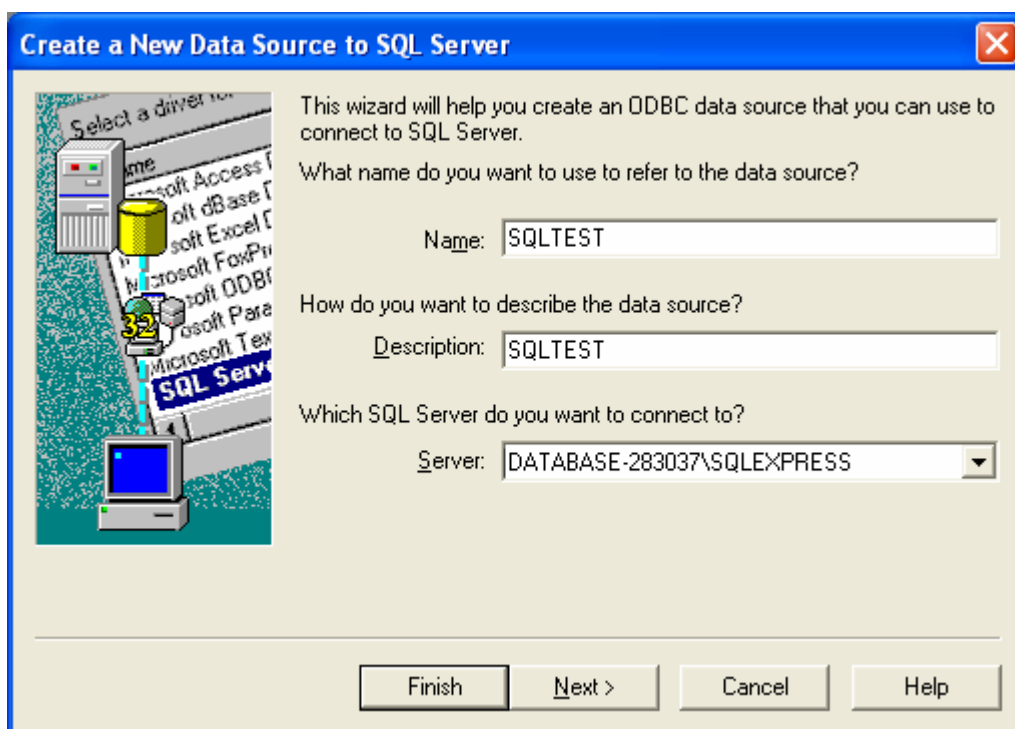
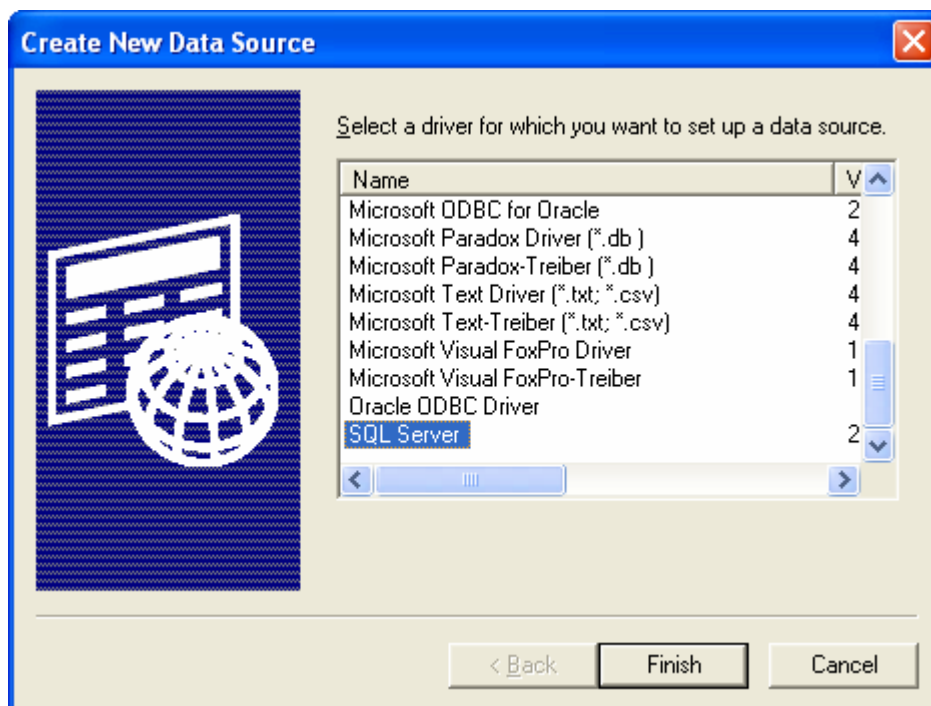
The connection dialog box appears.

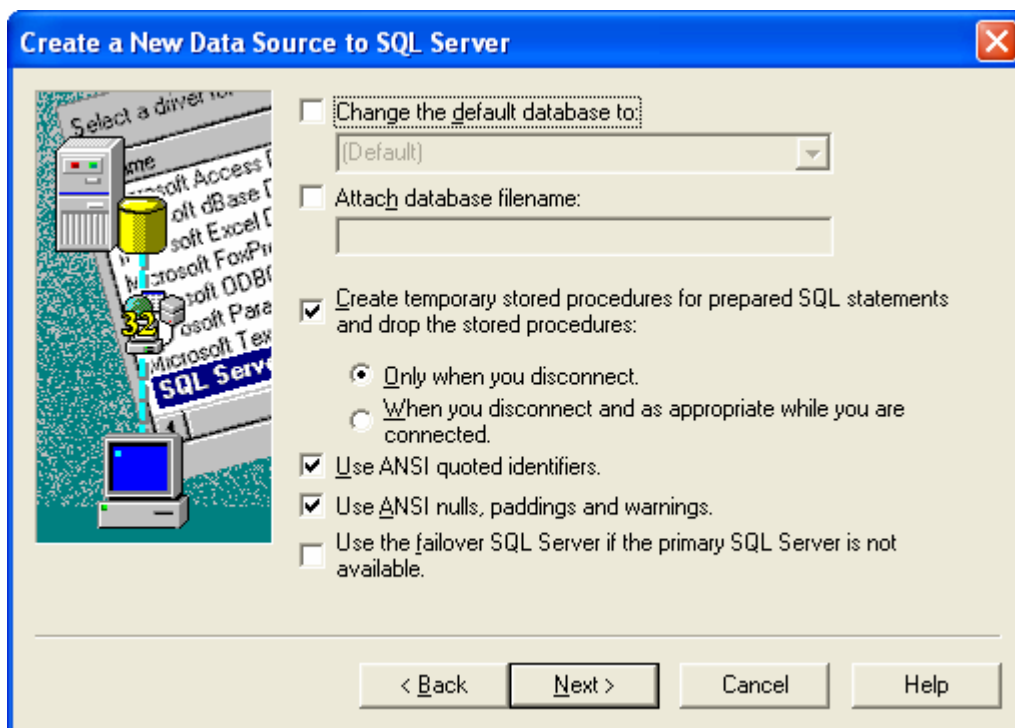
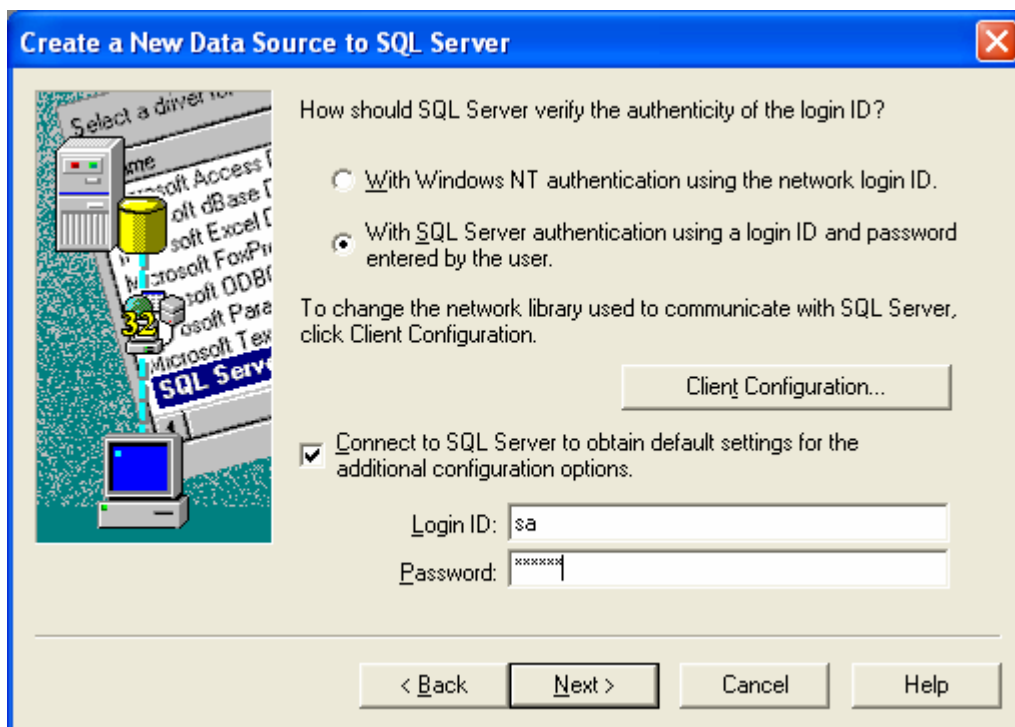


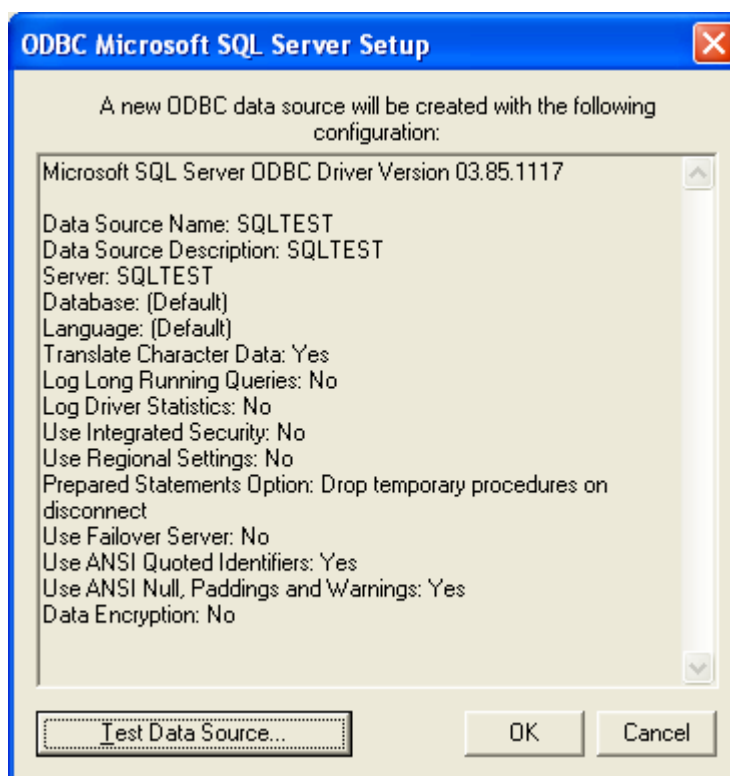
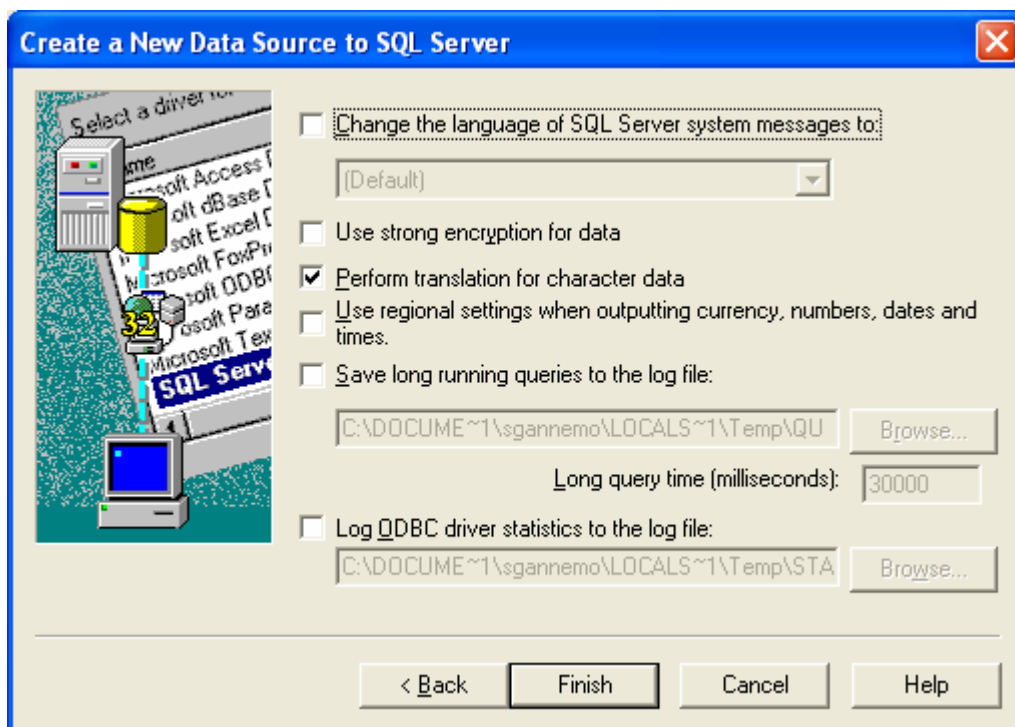
The SQL Server is the targeted instance, this must be ODBC Datasource.

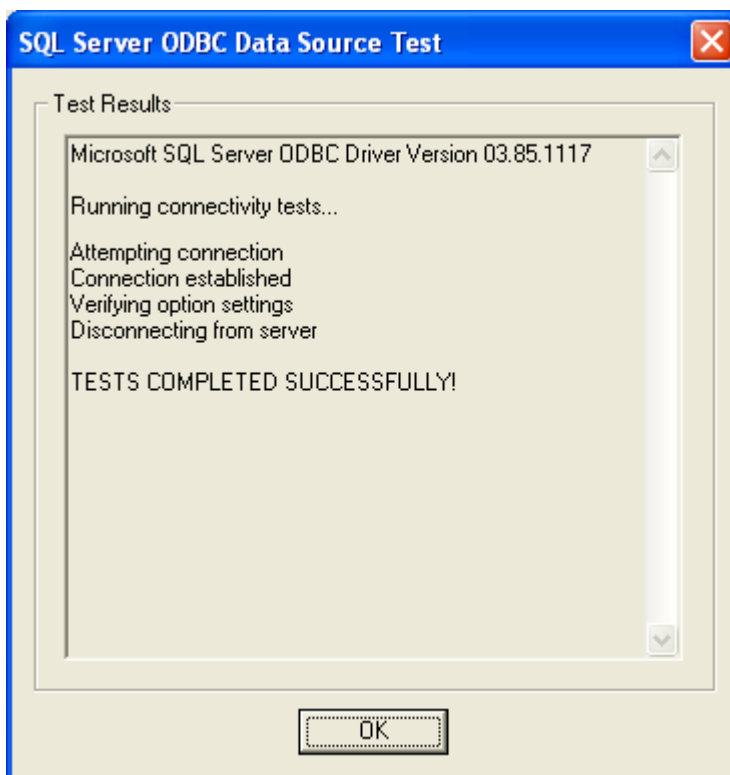
ODBC DataSource for SQL Server configuration

The DSN configuration must be the following.



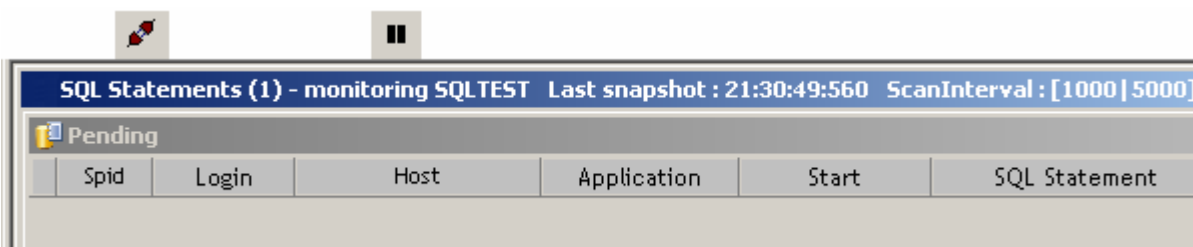






You must connect using a granted “sysadmin” login. You can use the system administrator’s «sa” login. Enter the snapshot frequencies for both SQL statements and SQL Server instance, caution: too high frequency will cause performance degradation on monitored SQL Server. Choose the license to be used for this monitoring session. The contents of this list depend on the acquired license and change based on the currently active monitoring sessions.

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 icon changes to red to indicate that the graphical user interface is displaying information in real time. The child window’s upper part (See Title of this window) changes to a «Pending» state.



Verification that SQL Shot is properly set up

As soon as the session is connected and the Automatic Refresh is activated, SQL Statements appear in the session’s report (based on the SQL activity of the targeted SQL Server). To make sure that the monitoring is properly set up, simply connect to the targeted SQL Server and run the following SQL Statement:

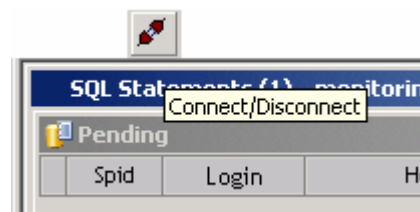
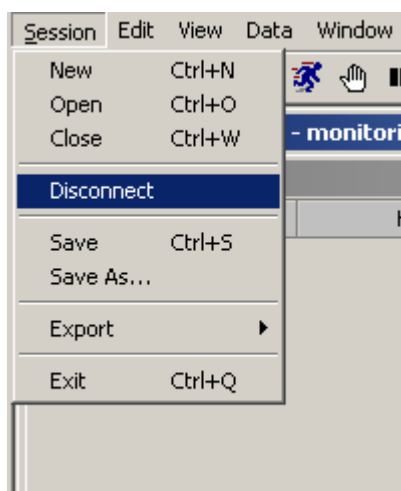
```
waitfor delay '00:01:00'
```

SQL Statements (1) - monitoring SQLTEST Last snapshot : 21:34:30:200 ScanInterval : [1000 5000]							
Pending							
Spid	Login	Host	Application	Start	SQL Statement	Batch Text	
51	sa	DATABASE-283037	OSQL-32	2009-12-30 21:34:14:107	waitfor delay '00:01:00'	waitfor delay '00:01:00'	

After a few seconds, you see appearing this SQL Statement in the session's report. This indicates that the monitoring is operational. If it fails to appear, see Chapter on [“SQL Shot's Technical Support”](#).

Monitoring session's shutdown

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



The window's title changes to display «SQL Statements (1)», meaning that the session is no longer active. The “Connection” and “Automatic Refresh” menu icons move to an inactive state and the title located in the upper part of the screen is deleted.



SQL Statements' monitoring

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](#)

[Session report's contents](#)

[SQL Statements' graphs](#)

[Filters](#)

[Display options](#)

[Sort](#)

[New session](#)

Session report's contents

See also:
SQL Statement's detail

The screenshot shows the SQL Shot application window. The title bar reads "SQL Shot". The menu bar includes "Session", "Edit", "View", "Data", "Window", and "Help". The toolbar contains various icons for session management and monitoring. The main window is titled "SQL Statements (1) - monitoring SQLTEST" and shows a "Last snapshot : 21:44:10:187" and "ScanInterval : [1000|5000]".

The window is divided into two main sections: "Pending" and "Completed".


Pending Section:

Spid	Login	Host	Application	Start	SQL Statement	Batch Text
53	Thierry	SSV_DEMO	OSQL-32	2009-12-30 21:44:06:310	waitfor delay @delay	exec sp_dbarc_lock_cnt
54	Yves	SSV_DEMO	OSQL-32	2009-12-30 21:44:07:310	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0),
55	Greg	SSV_DEMO	OSQL-32	2009-12-30 21:44:07:310	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0),
56	Paul	SSV_DEMO	OSQL-32	2009-12-30 21:44:07:310	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0),

Completed Section:

#	Spid	Login	Host	Application	End	SQL Statement	Batch Text
5	56	Yves	SSV_DEMO	OSQL-32	2009-12-30 21:44:04:357	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0),
4	55	Greg	SSV_DEMO	OSQL-32	2009-12-30 21:44:04:357	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0),
3	54	Paul	SSV_DEMO	OSQL-32	2009-12-30 21:44:04:357	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0),
2	53	Thierry	SSV_DEMO	OSQL-32	2009-12-30 21:44:04:357	waitfor delay @delay	exec sp_dbarc_lock_cnt

At the bottom of the window, there is a status bar that says "Connect or disconnect session".


The session's report is divided into 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. The bottom part of the window (completed) cannot be selected. The same goes for the SQL Statements' graphs. You need to switch off the "Automatic Refresh" option in order to be able to click on or scroll through the completed SQL statements' report. The monitoring remains active, but the captured pending SQL Statements are no longer refreshed, although they are 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:



Spid:	SQL Server Process Identifier.
Login:	Login name.
Host:	Hostname.
Application:	Application name.

Start:	Date and time of the pending SQL Statements. (These data vary according to the time zone of the SQL Server and they correspond to the beginning of the SQL Statement's execution).
End:	Date and time at which the execution of the SQL Statement ended. (These data vary according to the time zone of the SQL Server and they correspond to the completion of the SQL Statement).
Elapsed Time:	Time taken to execute the SQL Statement.
Batch Text:	Query sent by the client application.
SQL Statement:	SQL Statement captured by SQL Shot. This indicates the performance problem that has been identified.
Execution Plan:	The SQL Statement's execution plan.
Procedure - Trigger - Function:	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 «**NoDatabase**.**NoOwner**.**NoObject**» is displayed.
Executions count:	Number of executions of the SQL Statement (within a loop).
Logical reads:	Quantity of data page reads (in data cache) consumed by the SQL Statement.
Physical reads:	Quantity of data page reads (on physical device) consumed by the SQL Statement.
CPU Time:	Computation time used by the SQL Statement.
Blocking session:	Current identifier of session blocking the ongoing session (pending statements).
Last blocking session:	Last identifier of the session that blocked the ongoing session (completed statements).

Monitoring (See Paragraph on «How to start monitoring») and SQL Statements' Automatic Refresh displays (See Paragraph on «Session reports' contents») work independently from each other. This means that you cannot view the details regarding the SQL Statements contained in the completed statements' report when the option "Automatic Refresh" is

active (icon: ) and when the «Completed» report is grayed out. However, it is possible to interrupt the refreshing task

() 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. 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, used for the «View - SQL Statements - Pending» menu, opens up the session's report containing the pending SQL Statements. The  icon, used for the «View - SQL Statements - Completed» menu, opens up the session's report containing the SQL Statements whose execution is complete.

Each session's report can be split into two (at the left of the horizontal scrollbar), which enables you to view the pending SQL Statements' report as well as the completed SQL Statements' report. Each report comes with its own horizontal scrollbar. During the horizontal scrolling, however, the two parts of the window no longer have synchronized columns.

SQL Statement's detail

See also:

Detail - General

Detail - SQL Statement

Detail - Batch Text

Detail - Execution Plan

Detail - Procedure - Trigger - Function

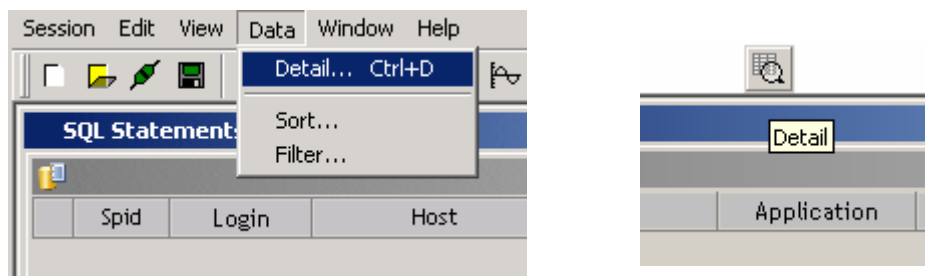
Detail - Server Performances

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.

#	Spid	Login	Host	Application	End	SQL Statement	Batch Text
8	55	Paul	SSV_DEMO	OSQL-32	2009-02-18 18:51:46:760	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0),
7	54	Yves	SSV_DEMO	OSQL-32	2009-02-18 18:51:46:760	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0),
6	53	Greg	SSV_DEMO	OSQL-32	2009-02-18 18:51:46:760	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0),
5	52	Thierry	SSV_DEMO	OSQL-32	2009-02-18 18:51:46:760	waitfor delay @delay	exec sp_dbarc_lock_cnt
4	55	Paul	SSV_DEMO	OSQL-32	2009-02-18	update Dbarc_cnt set id = id	begin tran

Once you have made your selection, choose the «Data - Detail...» menu item. Alternatively, click on the «Detail» menu icon (See Figures below), or use the right click on your mouse, to display the context menu. Next, choose the «Detail» menu item. It is possible to double-click on any column of the chosen line.

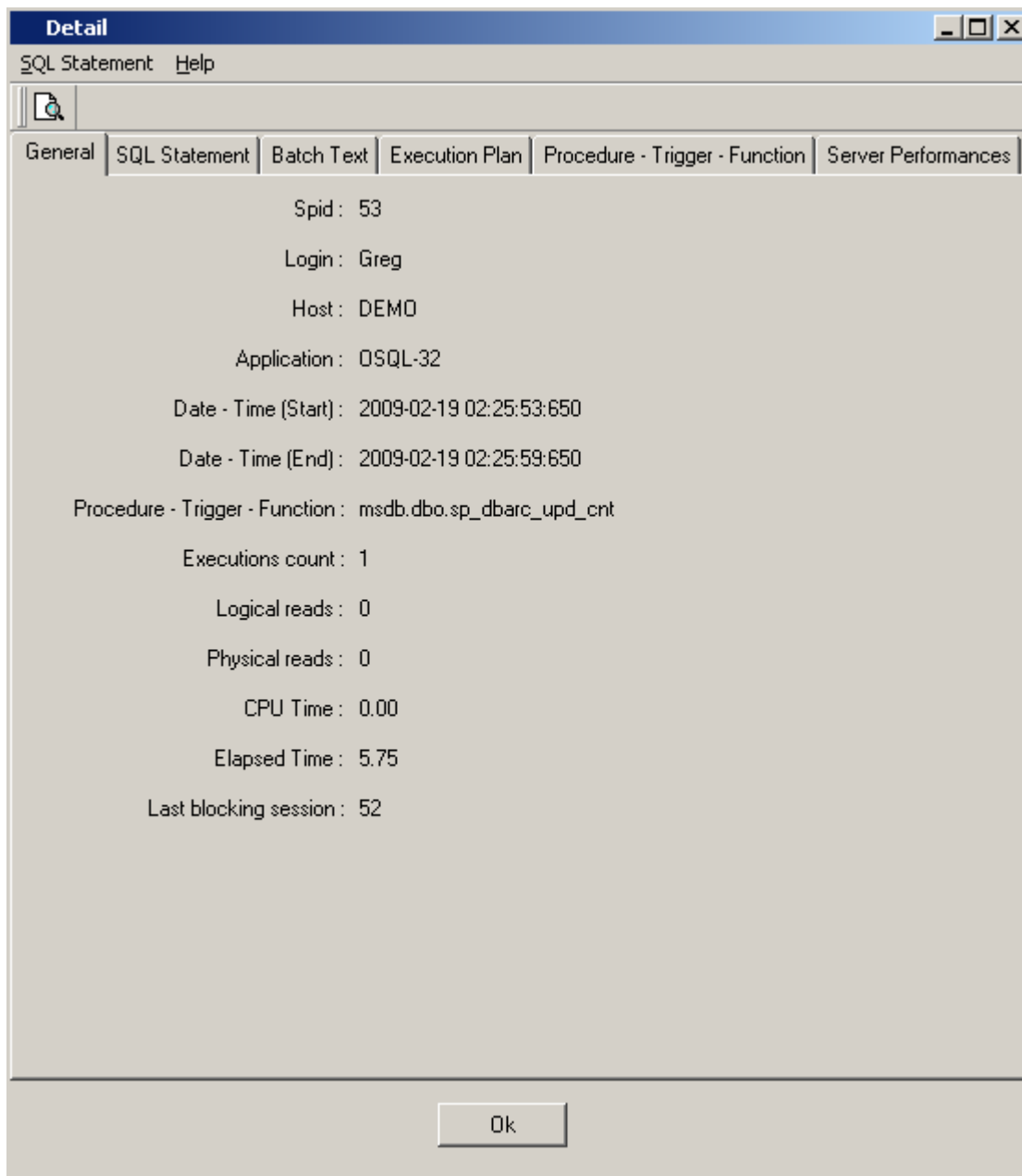


End	SQL Statement
2003-03-12 16:27:25	waitfor delay '00:00:2' rollback
2003-03-12 16:27:23	select @id = id from Dbarc_cpteur where end
2003-03-12 16:27:23	waitfor delay '00:00:3'
2003-03-12 16:27:20	wa
2003-03-12 16:27:20	select @id = id from Dbarc_cpteur where end
2003-03-12 16:27:16	waitfor delay '00:00:4' rollback

A window with six tabs is displayed.

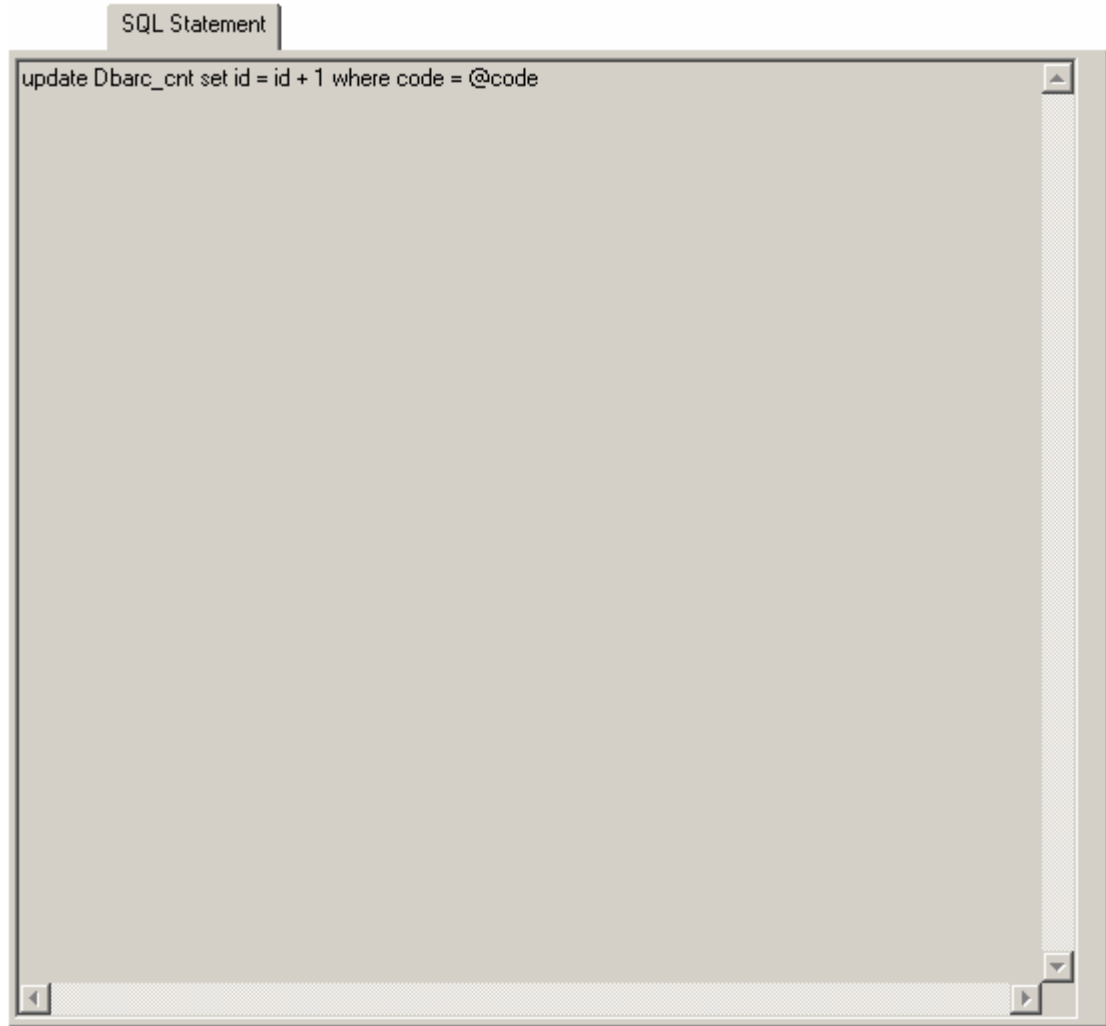
Detail - General

This window displays the following data: spid, login 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 Transact SQL object). Whenever there is a dynamic SQL statement, the detail tab reads: ****NoDatabase****. ****NoOwner****. ****NoObject****.



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, trigger, or function. 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 a stored procedure's or trigger's or by a function's duration.



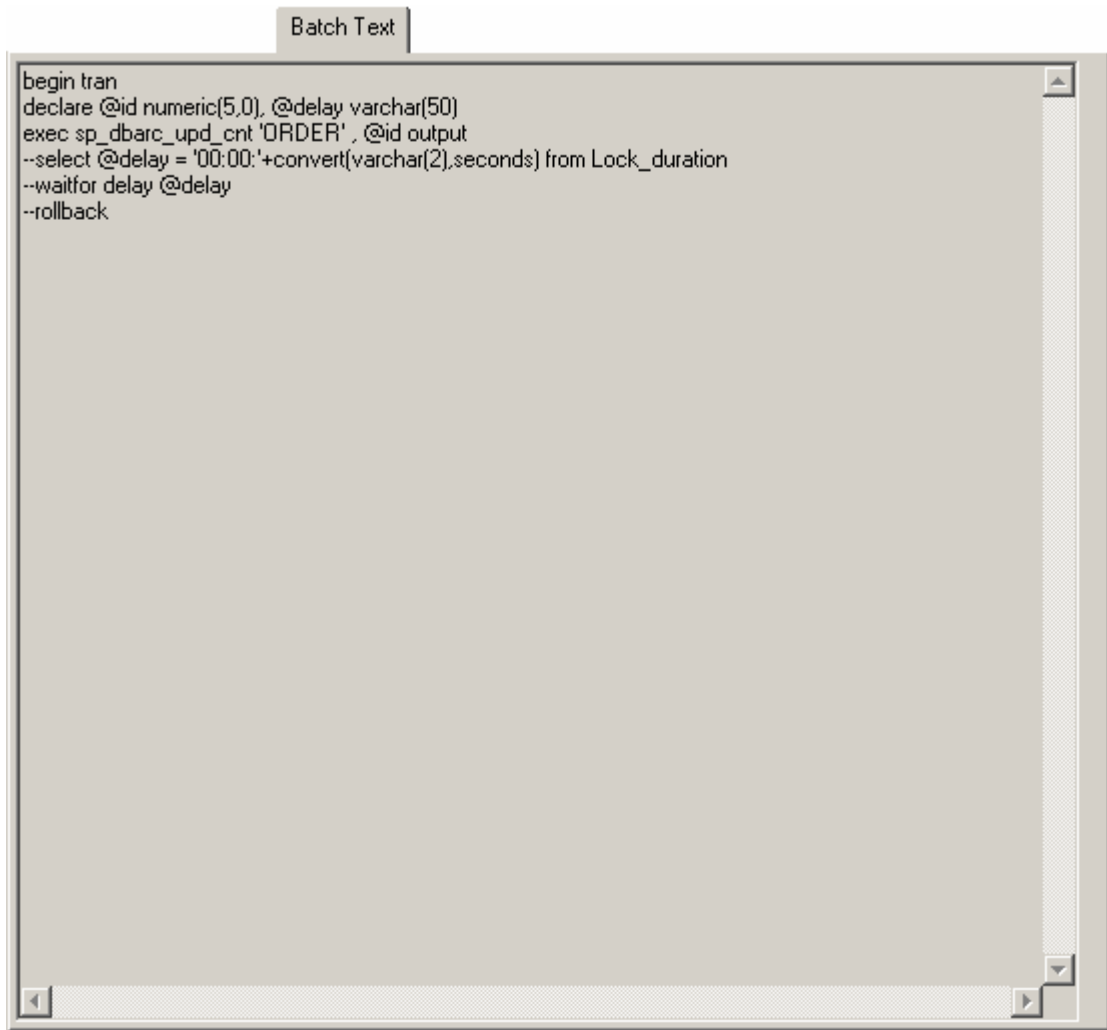
The screenshot shows a window titled "SQL Statement" with a text area containing the following SQL query:

```
update Dbarc_cnt set id = id + 1 where code = @code
```

The window has a standard Windows-style border with scrollbars on the right and bottom.

Detail - Batch Text

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

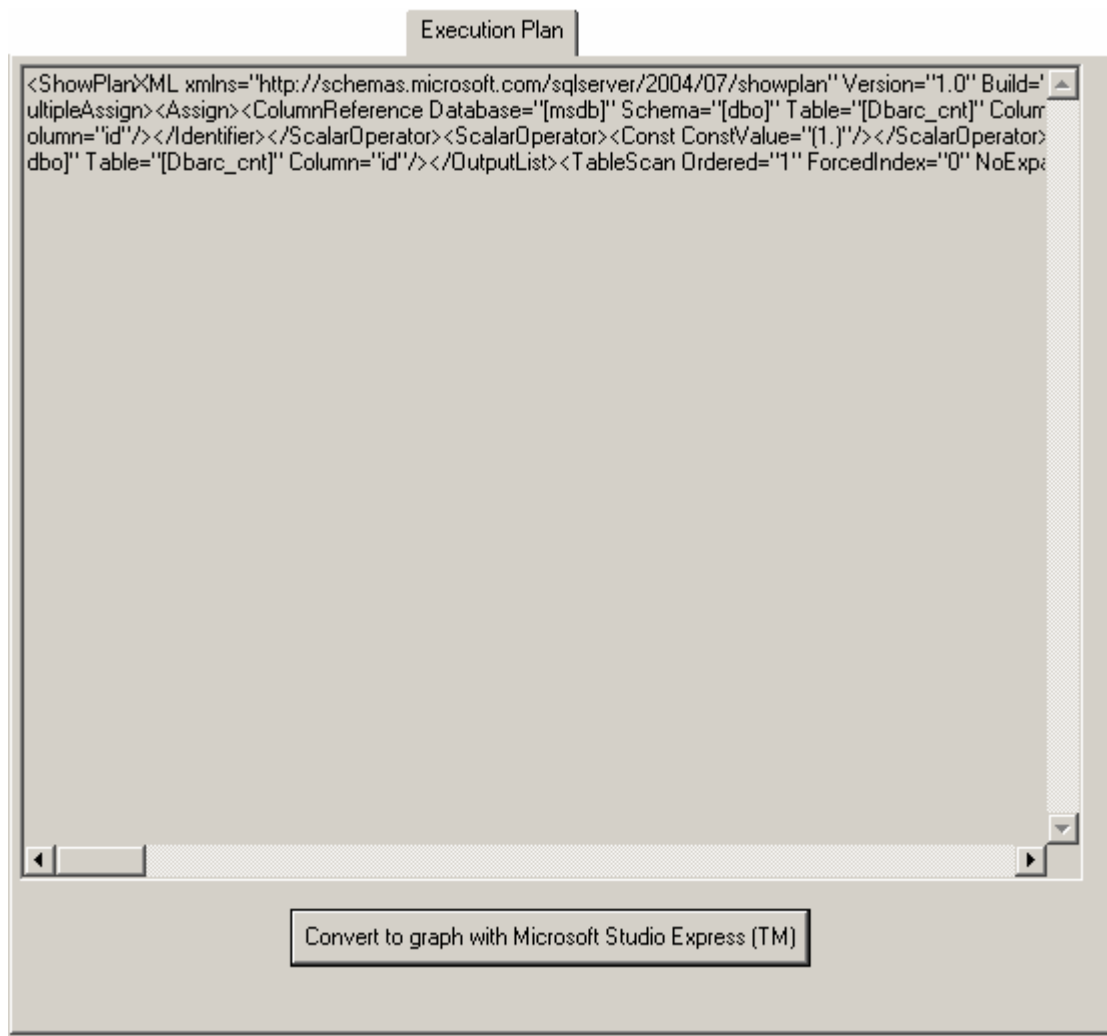
A screenshot of a window titled "Batch Text". The window contains the following SQL code:

```
begin tran
declare @id numeric(5,0), @delay varchar(50)
exec sp_dbarc_upd_cnt 'ORDER', @id output
--select @delay = '00:00:'+convert(varchar(2),seconds) from Lock_duration
--waitfor delay @delay
--rollback
```

The window has a standard Windows-style border with scrollbars on the right and bottom.

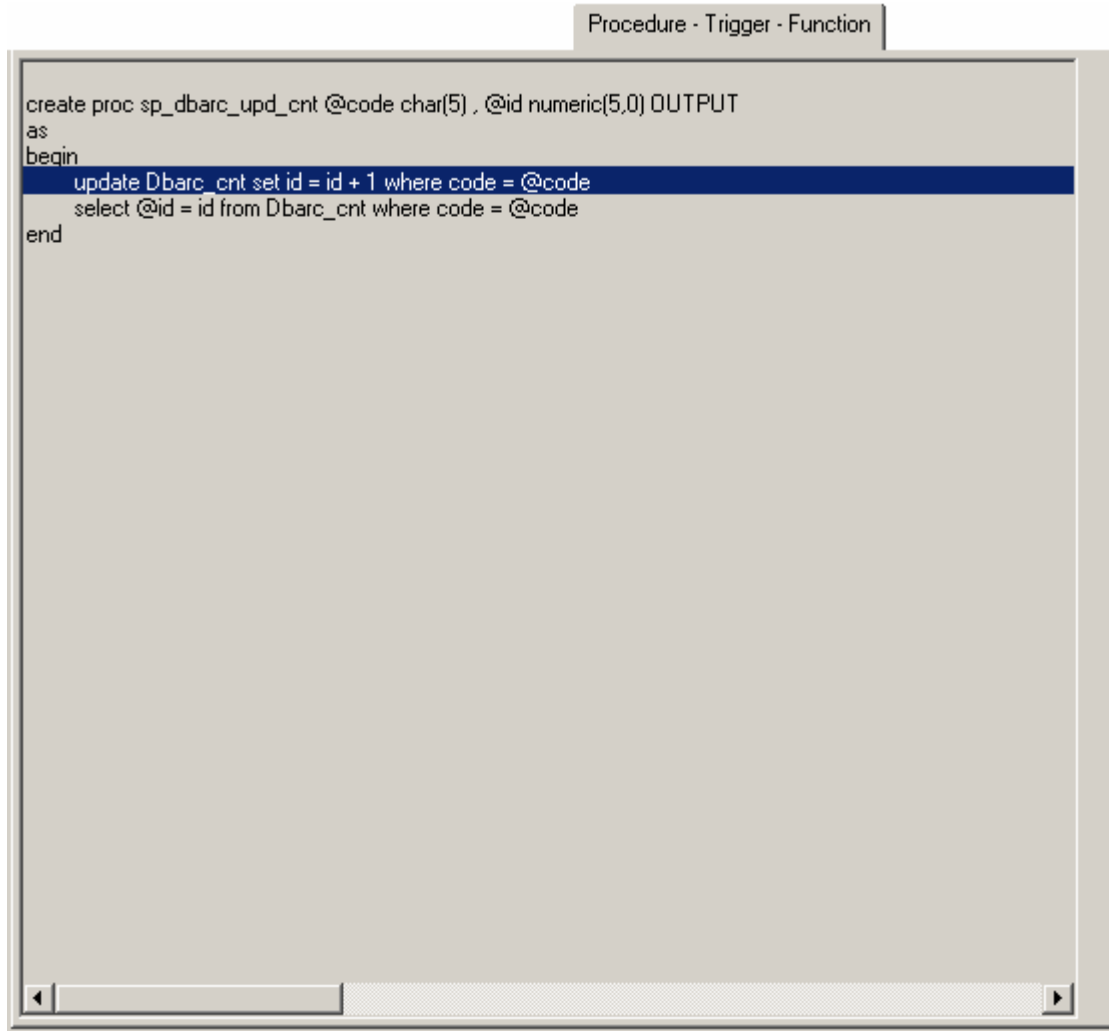
Detail - Execution Plan

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



Detail - Procedure - Trigger - Function

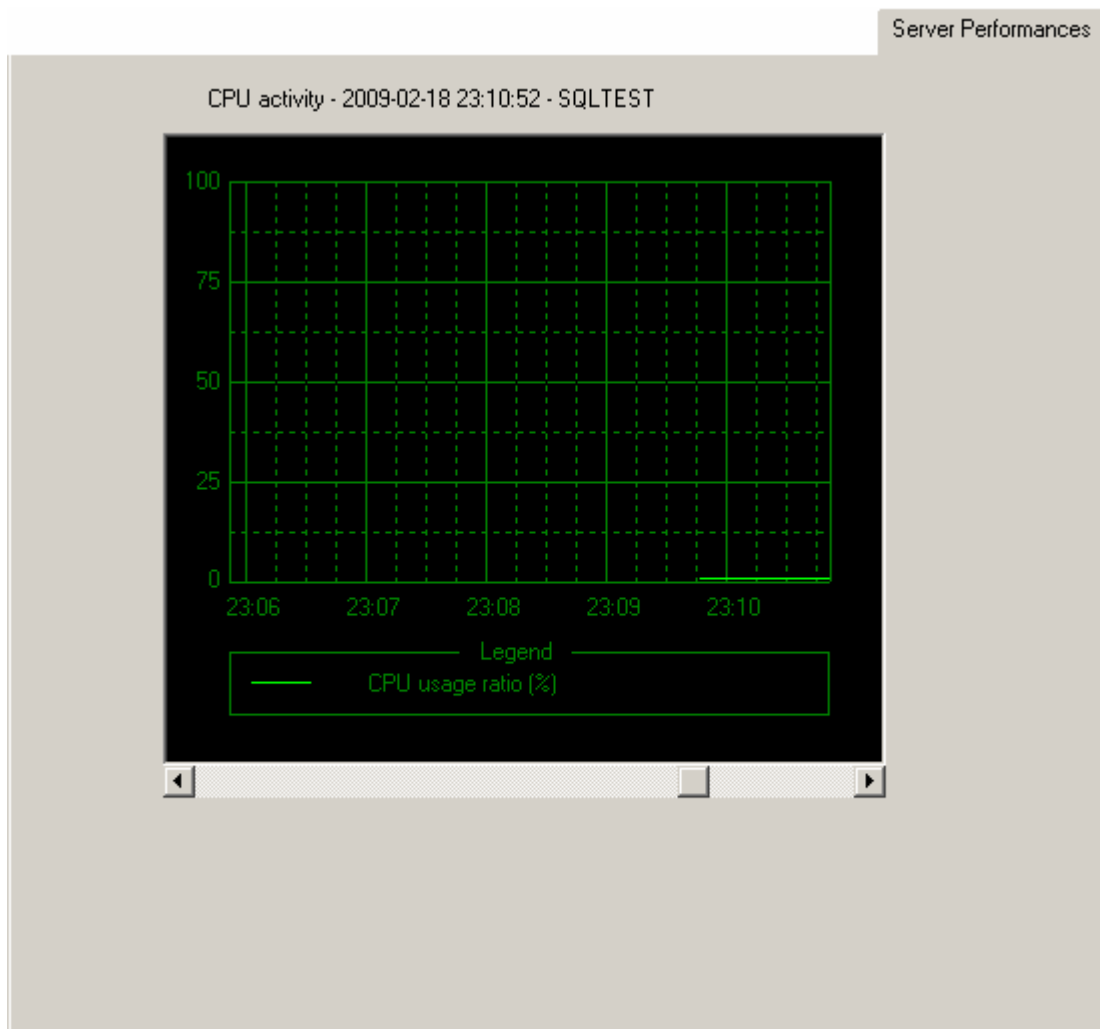
This is the stored procedure, trigger, or function source code containing the captured SQL Statement (provided the latter is part of a procedure, trigger, or function). In the source code text, the captured SQL Statement is selected. Whenever the SQL Statement is not a part of a SQL Server procedure, a trigger or a function, it is a part of the Batch Text, namely a dynamic SQL. In such a case, the tab displays the following character string: "***NoDatabase**NoOwner**NoObject**".



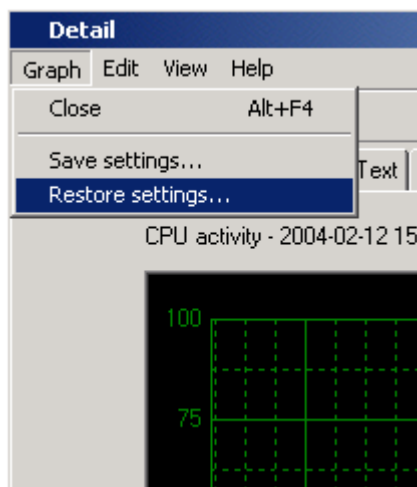
```
create proc sp_dbarc_upd_cnt @code char(5) , @id numeric(5,0) OUTPUT
as
begin
  update Dbarc_cnt set id = id + 1 where code = @code
  select @id = id from Dbarc_cnt where code = @code
end
```

Detail - Server Performances

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

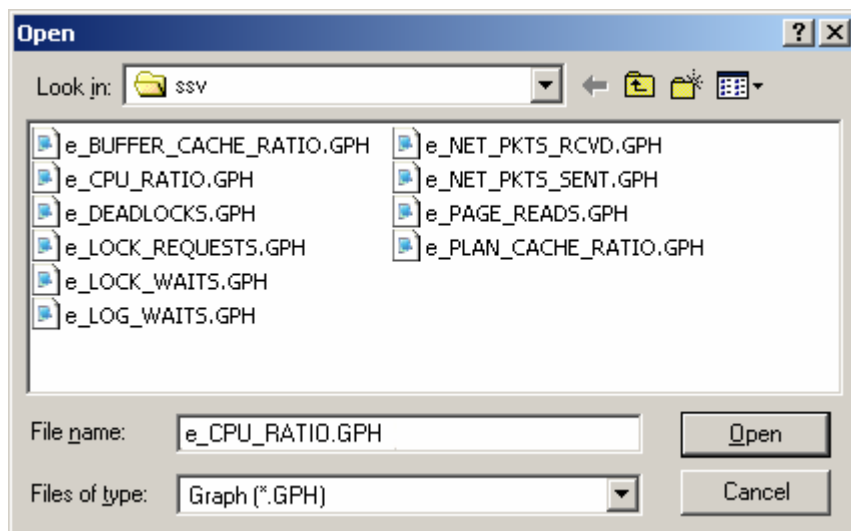


When selecting the «Graph - Restore settings...» menu item, you can view the various available performance indicators.

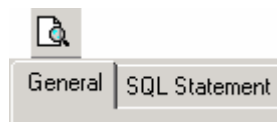
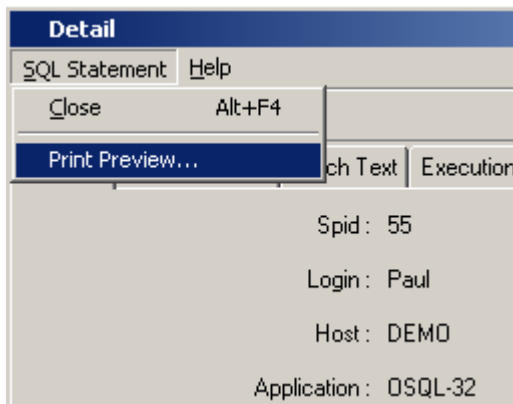


Select this by opening a configuration file previously²⁹configured with the following SQL Shot's feature: «SQL

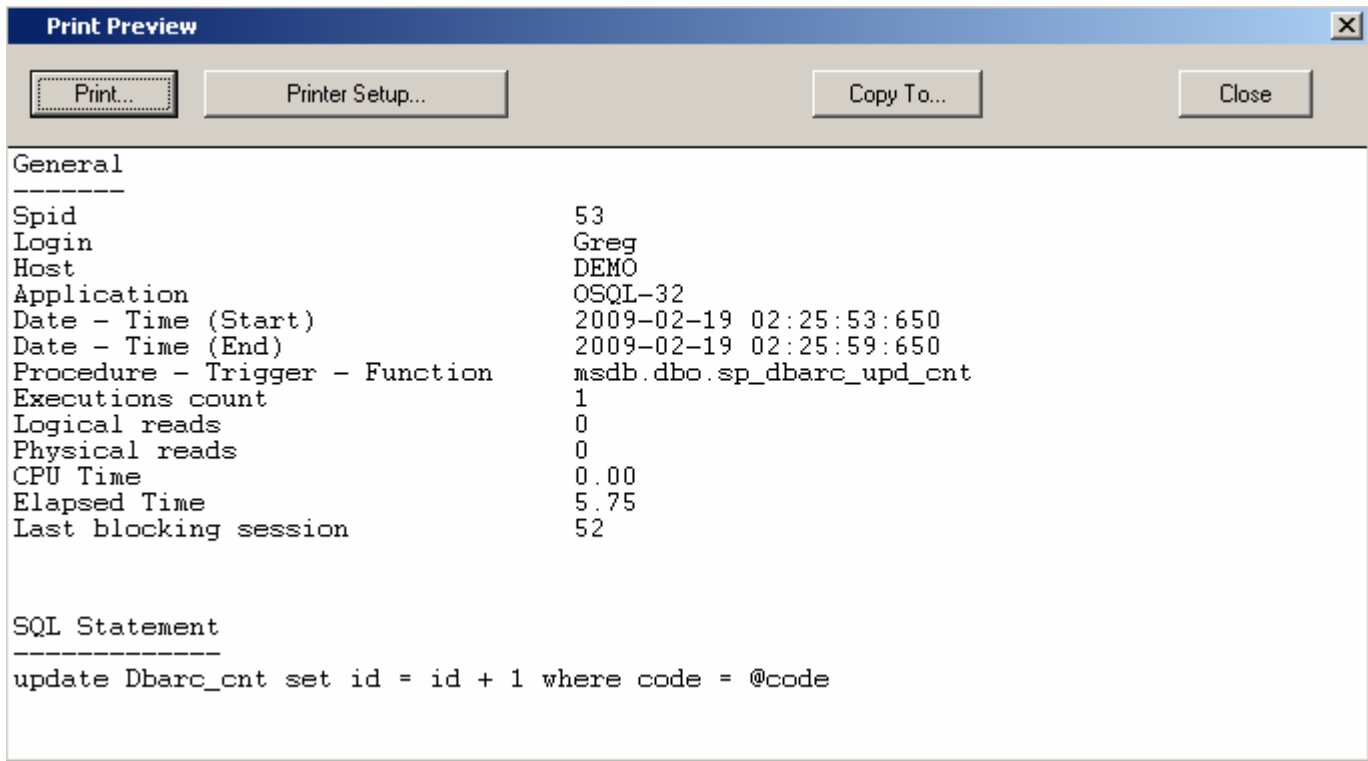
Server's performances' monitoring». To find out more about the way to save and change the graph configurations, See chapter on «Monitoring of the indicators' server». By default, «.GPH» files are stored in %APPDATA%\SQL Shot\cfg\gph directory.



Detail - Print

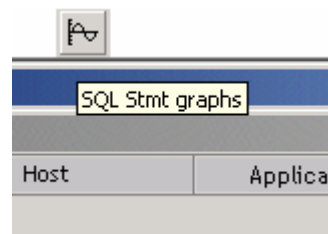
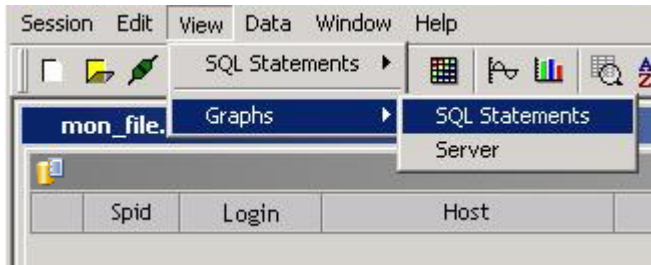


This feature allows you to print the contents of the «General», «SQL Statements», « Batch Text» and «Execution Plan» tabs. You simply select the «SQL Statement - Print Preview...» menu item, or click on “Print Preview” icon. The program displays the following print preview. You confirm while clicking on the «Print...» button, or choose to “Copy To...” a file.

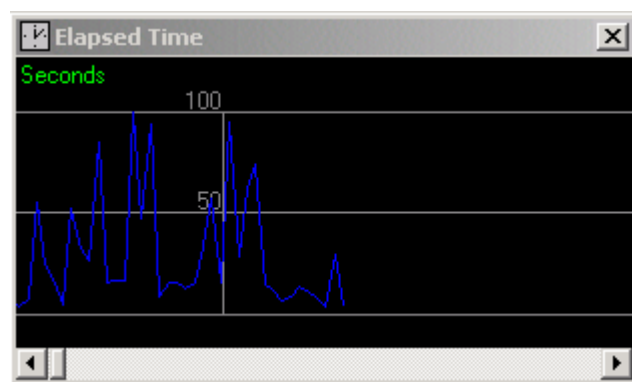
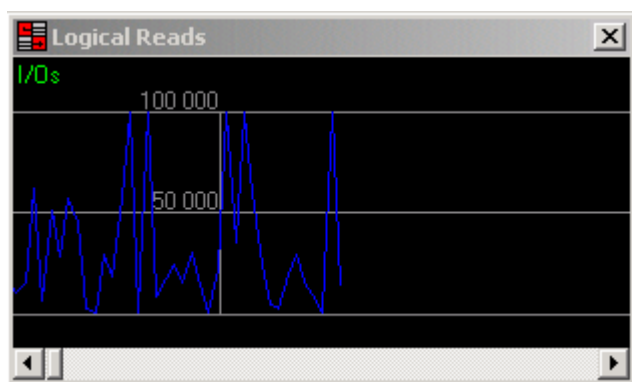


SQL Statements' graphs

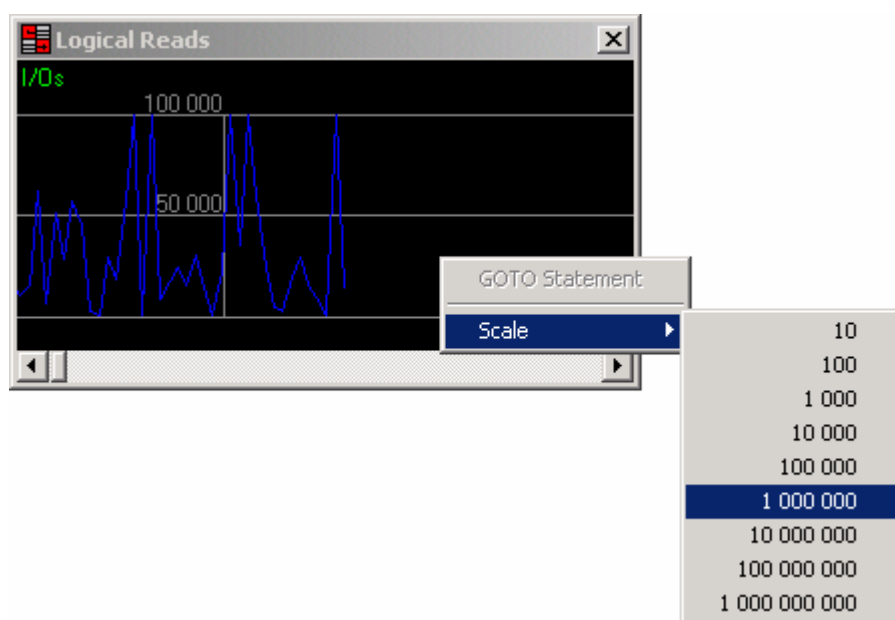
To display the SQL Statements' graphs, you need to select the «View - Graphs - SQL Statements» menu item. Alternatively, you can click on the «SQL Stmt graphs» menu icon (See Figures below). These graphs are only available for the completed SQL Statements.



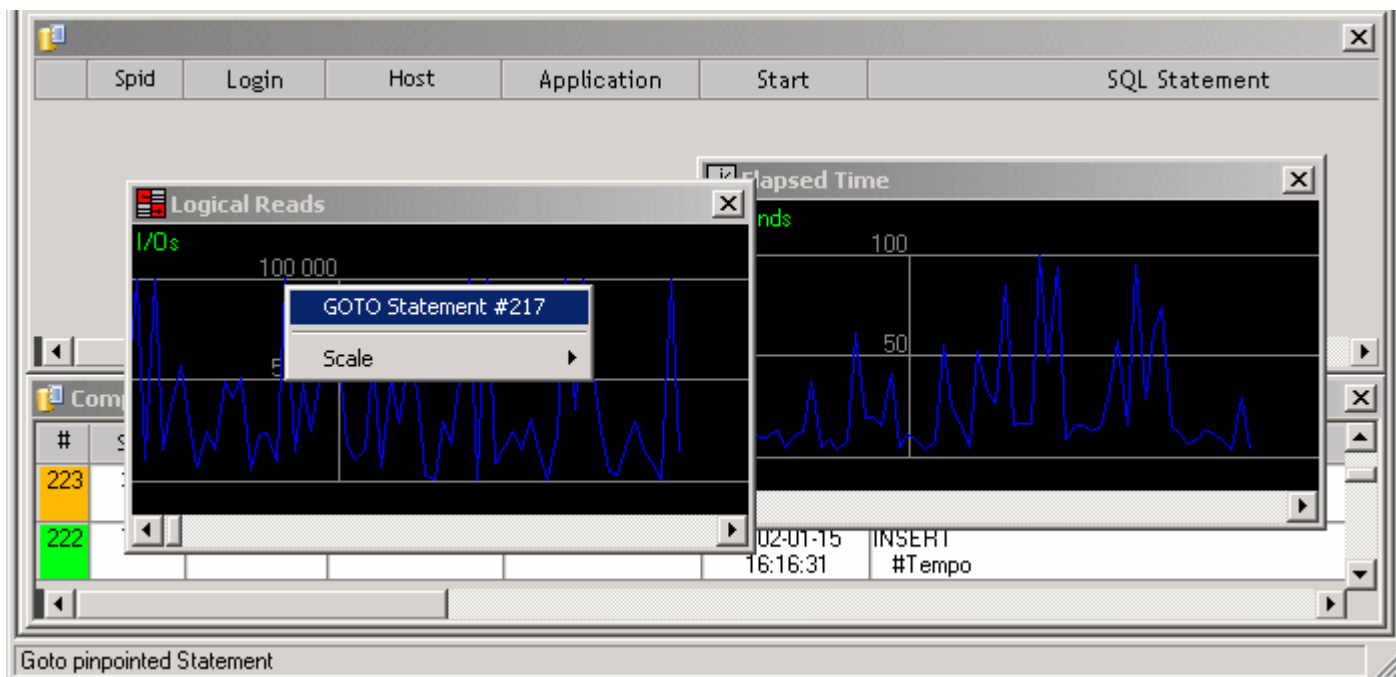
Two graphs are displayed:



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 chosen for the session's report. Each SQL Statement is displayed by a point in the curve.



According to the values obtained, you can adjust the scale by clicking on the right button. A contextual menu appears. Choose 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 which has generated a peak, right-click on the peak. Next, select the «GOTO Statement #(i)», in 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).

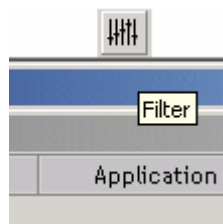
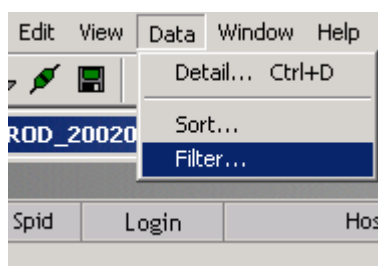


Filters

See also:

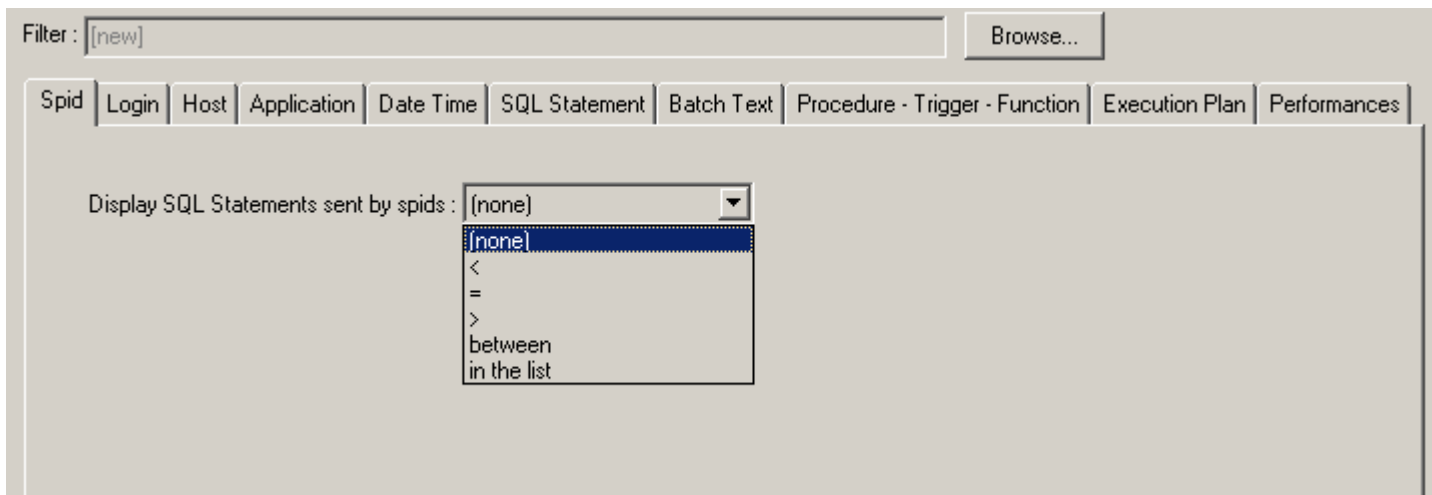
- Filter - Spid
- Filter - Login
- Filter - Host
- Filter - Application
- Filter - Date - Time
- Filter - SQL Statement
- Filter - Batch Text
- Filter - Procedure - Trigger - Function
- Filter - Execution plan
- Filter - Performances
- Filter - How to manage filter files

Filters are used to reduce the quantity of SQL Statements displayed in the session's report or captured by SQL Shot. 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, 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 have to fulfill all the defined filters (with an "AND" operator).

Example: A filter defines the «Paul» login and logical I/Os higher than 5000. The filtered SQL Statements will be related to SQL Queries sent by the «Paul» login and consuming more than 5000 logical I/Os.



Filter - Spid

Only the SQL Statements executed by the defined Spid(s) will be displayed. The list of spids can be set within a list or within a values interval.

Filter : [new]

Spid | Login | Host | Application | Date Time | SQL Statement | Batch Text | Procedure - Trigger - Function | Execution Plan | Performances

Display SQL Statements sent by spids : and and

Filter : [new]

Spid | Login | Host | Application | Date Time | SQL Statement | Batch Text | Procedure - Trigger - Function | Execution Plan | Performances

Display SQL Statements sent by spids :

Spids	Selected spids
1	8
2	9
3	10
4	
5	
6	
7	
11	
12	
13	
14	

Filter - Login

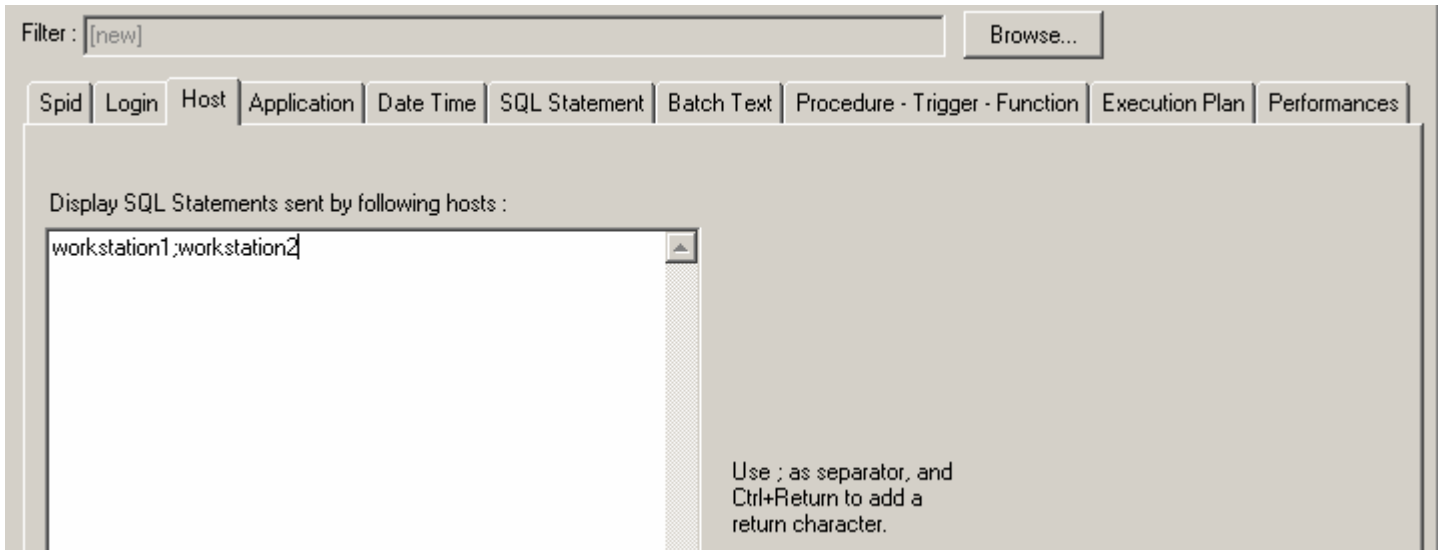
Only the SQL Statements executed by the defined logins will be displayed. The logins are entered in a list and are separated by a semicolon.

The screenshot shows a dialog box titled "Filter - Login". At the top, there is a text input field labeled "Filter:" containing the text "[new]", followed by a "Browse..." button. Below this is a horizontal tabbed menu with the following tabs: "Spid", "Login", "Host", "Application", "Date Time", "SQL Statement", "Batch Text", "Procedure - Trigger - Function", "Execution Plan", and "Performances". The "Login" tab is currently selected. The main area of the dialog contains the text "Display SQL Statements sent by following logins :". Below this text is a text input field containing the text "Paul;Greg". To the right of this field is a vertical scrollbar. Below the input field, there is a note: "Use ; as separator, and Ctrl+Return to add a return character."

Filter - Host

Only the SQL Statements sent by the defined workstations will be displayed. The hosts are entered in a list and they are separated by a semicolon.

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



Filter : [new]

Spid | Login | **Host** | Application | Date Time | SQL Statement | Batch Text | Procedure - Trigger - Function | Execution Plan | Performances

Display SQL Statements sent by following hosts :

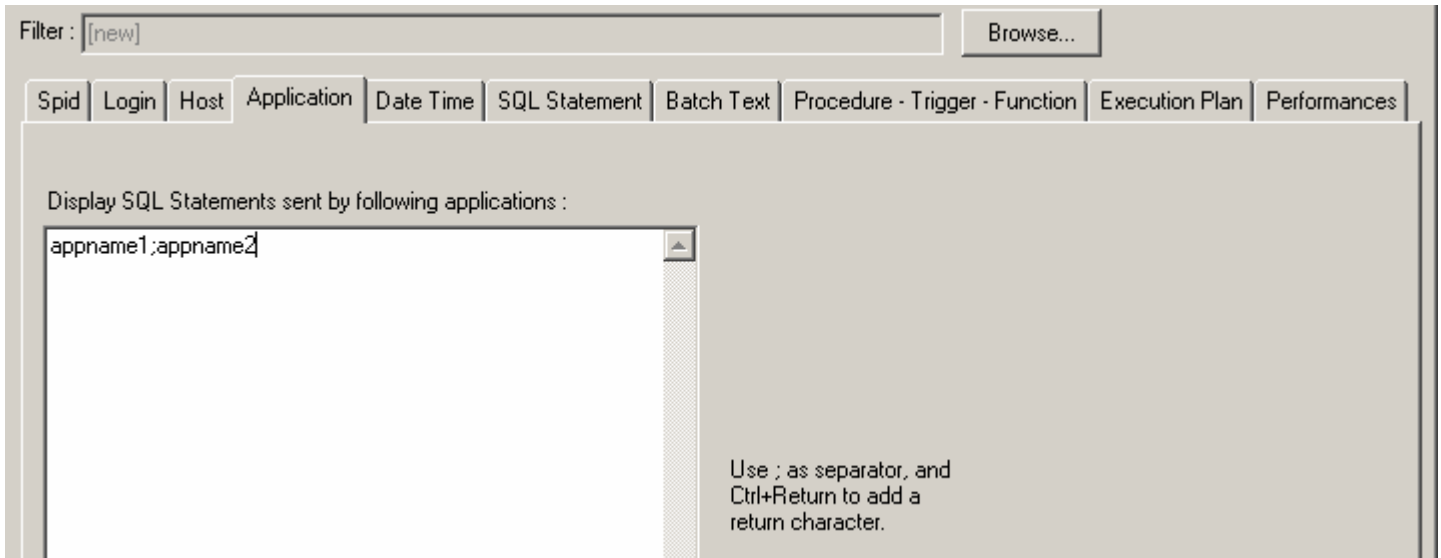
workstation1;workstation2

Use ; as separator, and
Ctrl+Return to add a
return character.

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 client application's connection's property. Since this property is optional, it may not always be mentioned. If you choose to filter with this value, you need to make sure that the applications being monitored mention this property.



Filter : [new]

Spid | Login | Host | **Application** | Date Time | SQL Statement | Batch Text | Procedure - Trigger - Function | Execution Plan | Performances

Display SQL Statements sent by following applications :

appname1;appname2

Use ; as separator, and
Ctrl+Return to add a
return character.

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 choose to display the SQL Statements starting from a particular date and time or you can select a particular segment between two time ranges.

Filter : [new]

Spid | Login | Host | Application | **Date Time** | SQL Statement | Batch Text | Procedure - Trigger - Function | Execution Plan | Performances

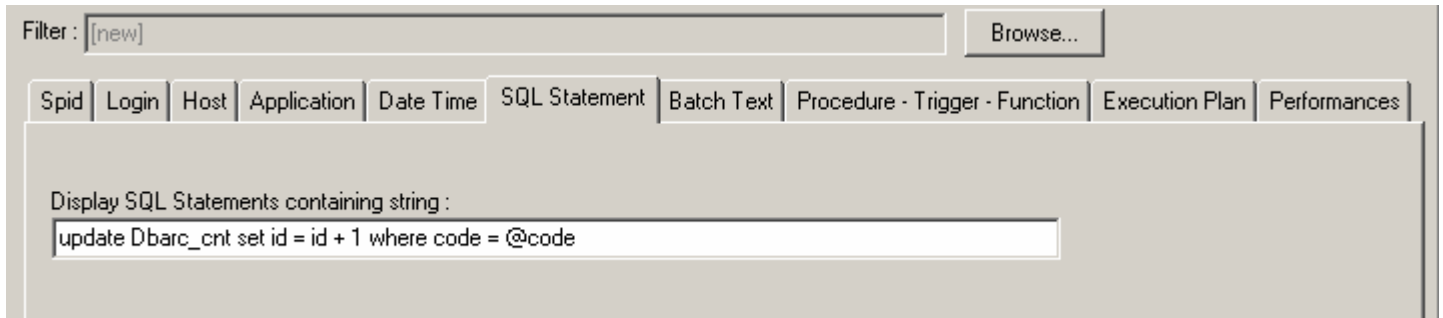
Display SQL Statements whose Date and Time of execution is

Date (yyyy-mm-dd) and Date (yyyy-mm-dd)

Time (hh:mm) Time (hh:mm)

Filter - SQL Statement

This filter allows you to only display the SQL Statements containing a defined character string (for example the name of a table). The character string entered needs to match the SQL Statement's text exactly.



The screenshot shows the 'Filter - SQL Statement' interface. At the top, there is a text input field labeled 'Filter:' containing the text '[new]', followed by a 'Browse...' button. Below this is a horizontal menu with several tabs: 'Spid', 'Login', 'Host', 'Application', 'Date Time', 'SQL Statement', 'Batch Text', 'Procedure - Trigger - Function', 'Execution Plan', and 'Performances'. The 'SQL Statement' tab is currently selected. Underneath the menu, the text 'Display SQL Statements containing string :' is followed by a text input field containing the SQL statement: 'update Dbarc_cnt set id = id + 1 where code = @code'.

Filter - Batch Text

This filter allows you to display the SQL Statements whose Batch Text contains a defined character string (for example a stored procedure's name). The character string entered needs to match the Batch Text exactly.

The screenshot shows the 'Filter - Batch Text' interface. At the top, there is a text input field labeled 'Filter:' containing the text '[new]', followed by a 'Browse...' button. Below this is a horizontal menu with several tabs: 'Spid', 'Login', 'Host', 'Application', 'Date Time', 'SQL Statement', 'Batch Text', 'Procedure - Trigger - Function', 'Execution Plan', and 'Performances'. The 'Batch Text' tab is currently selected. Below the menu, there is a label 'Display SQL Statements having Batch Text containing string :' followed by a text input field containing the SQL statement: `exec sp_dbarc_upd_cnt 'ORDER' , @id output`.

Filter - Procedure - Trigger - Function

This filter allows you to only display the SQL Statements which are part of a defined list of stored procedures, triggers or functions. To select the list, you first need to connect to the targeted SQL Server. Next, select the list of procedures, triggers and functions needed.

Filter : [new] Browse...

Spid Login Host Application Date Time SQL Statement Batch Text **Procedure - Trigger - Function** Execution Plan Performances

Connection
 SQL Server : (ODBC Data Source) SQLTEST
 Login : sa
 Password : •••••• Connect

Databases
 master.%.%
 model.%.%
msdb.%.%
 tempdb.%.%

Procedures - Triggers - Functions
 agent_datetime
 ConvertToInt
 cp1
 f_toto
 get_principal_id
 get_principal_sid
 sp_get_plantext
 sp_get_sqltext
 sp_add_alert

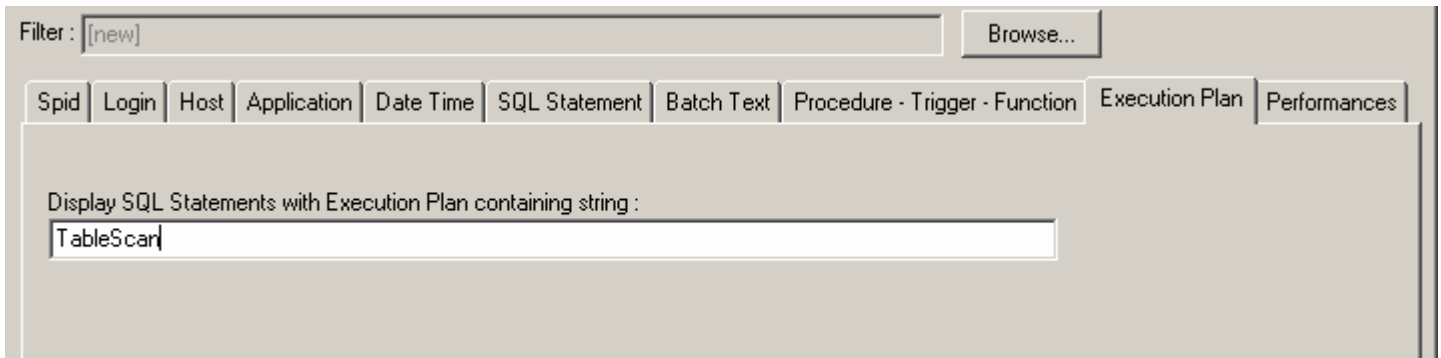
Display SQL Statements from :

Databases - Procedures - Triggers - Functions
 msdb.%.pp1
 msdb.%.pp2

Clear all Save Apply No Filter Close

Filter - Execution Plan

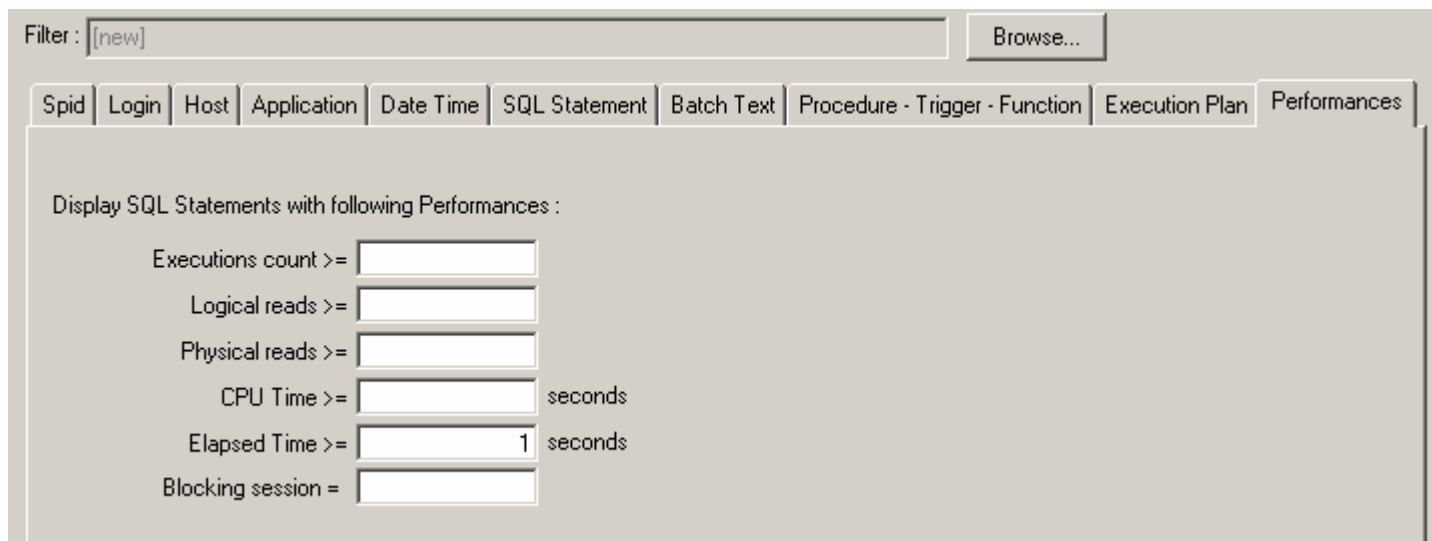
This filter allows you to only display the SQL Statements whose execution plan contains a defined character string (for example «Table Scan»). The character string entered needs to match the execution plan text exactly.



The screenshot shows a dialog box titled "Filter - Execution Plan". At the top left, there is a text input field labeled "Filter:" containing the text "[new]". To its right is a "Browse..." button. Below this is a horizontal row of tabs: "Spid", "Login", "Host", "Application", "Date Time", "SQL Statement", "Batch Text", "Procedure - Trigger - Function", "Execution Plan", and "Performances". The "Execution Plan" tab is currently selected. Below the tabs, there is a label "Display SQL Statements with Execution Plan containing string :" followed by a text input field containing the text "TableScan".

Filter - Performances

These various filters allow you to only display 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.



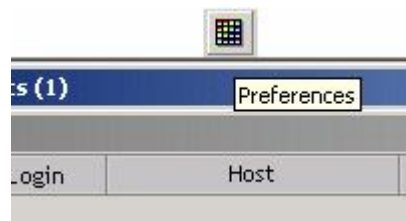
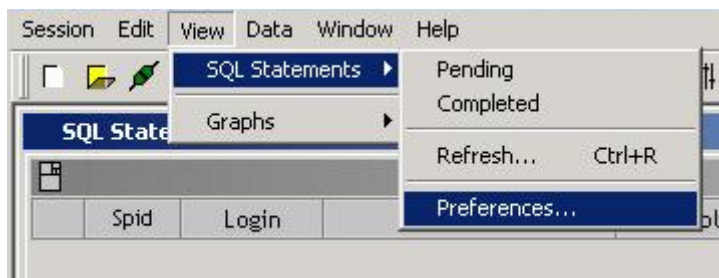
Filter - How to manage filter files

Once you have defined the chosen elements to filter, you need to save these settings in a file. To do this, click on the "Save" button and select a filename FLT. After you click 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. "New" button is used to clear all settings in the various tabs. The option "No Filter" unselects the filtering and displays the monitoring windows without applying any filter. By default, «.FLT» files are stored in %APPDATA%\SQL Shot\cfg\flt directory.

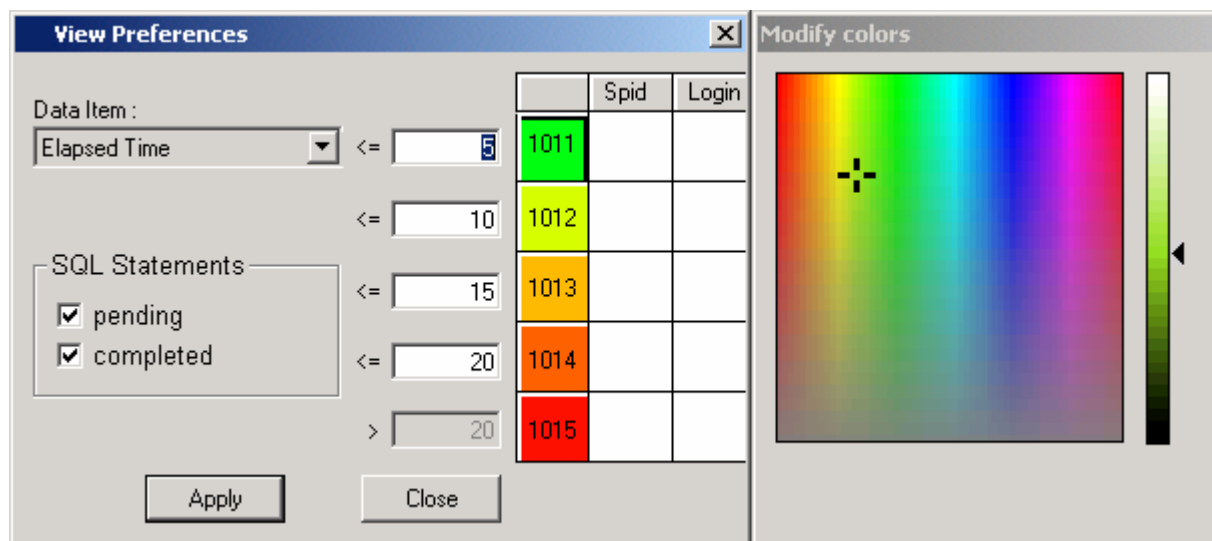
Display options

In order to easily identify the most consuming SQL Statements, SQL Shot 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.

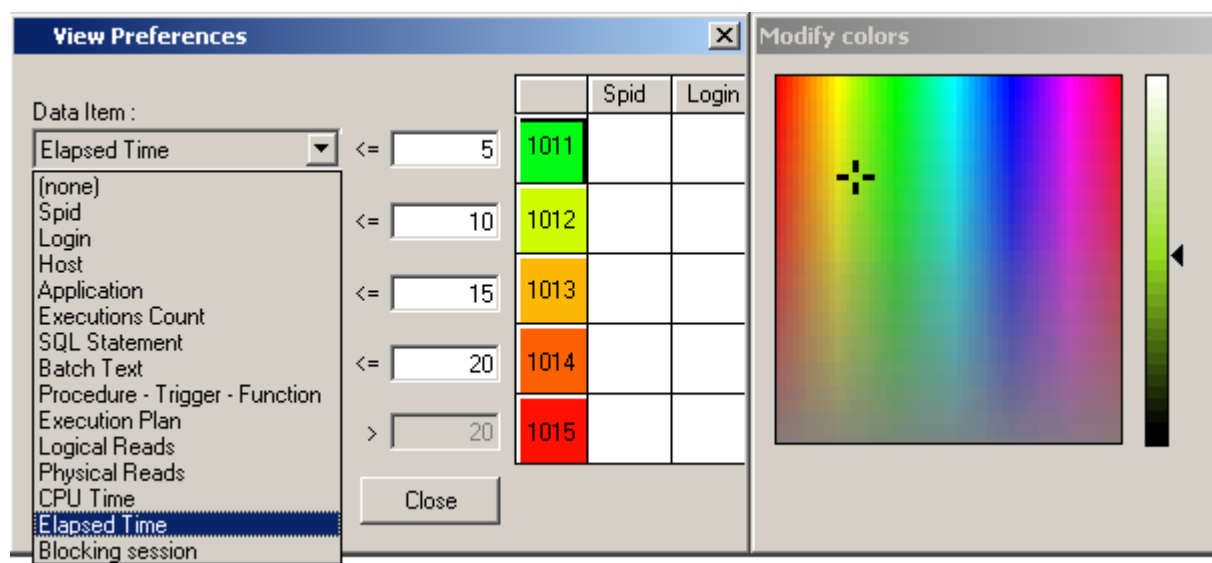
The display options make it possible to choose the performance indicator to which the thresholds will apply and to configure the values and colors associated with each threshold. To modify the display options, select the «View - SQL Statements - Preferences» menu item. Alternatively, click on the «Preferences» menu icon (See Figures below).



The following window is displayed:



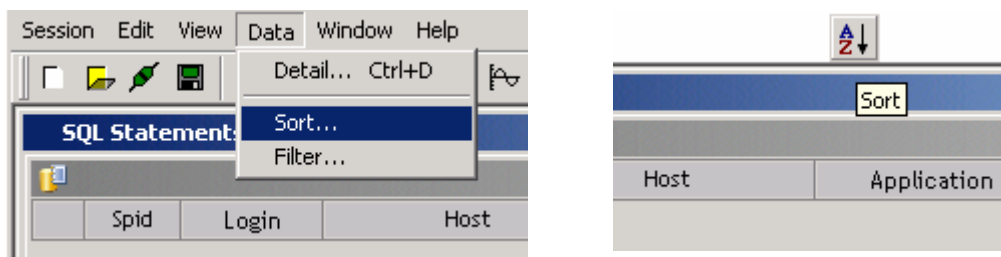
Choose the performance indicator to which the thresholds will apply.



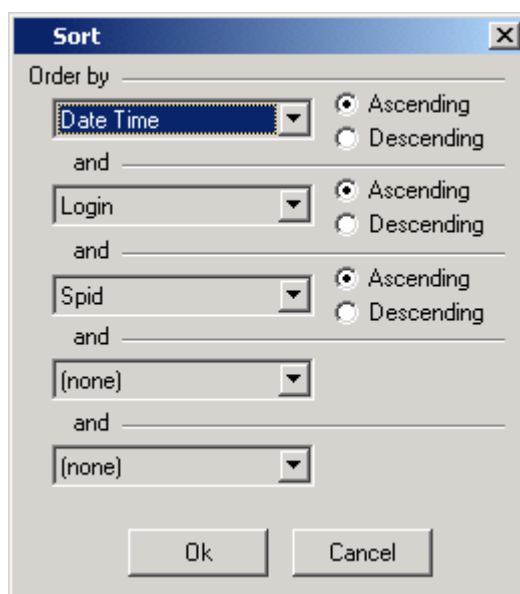
Choose colors as well as values for the various thresholds. Next, apply these display options to the pending SQL Statements, to the completed SQL Statements, or to both.

Sort

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



The following window is displayed:

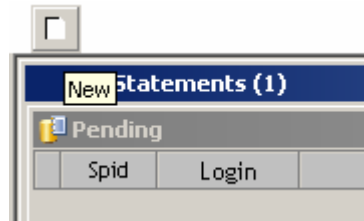
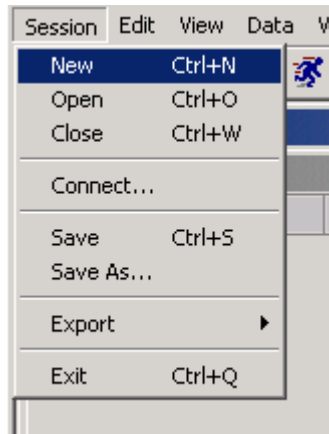


You need to choose the sort keys (up to five levels) as well as the direction (ascending or descending). Each sort key corresponds to a specific column of the session's report.

Caution: For performance's sake, it is not possible to do any sorting 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 SQL Shot 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. Thanks to your ability to create new sessions, you will be able to view several archive files simultaneously.



Within a new session, it is possible to create a new connection to the SQL 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).

SQL Server's performances' monitoring

See also:

How to start and stop monitoring

The SQL Server's 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

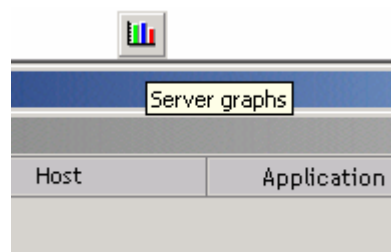
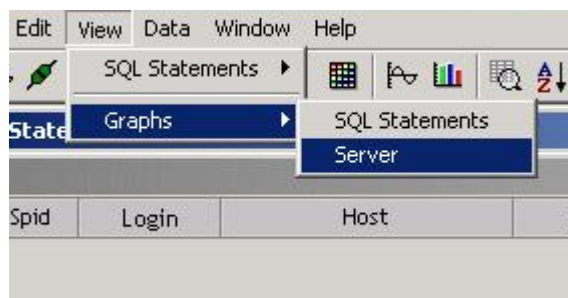
How to refresh the graph

SQL Statements related to a time period

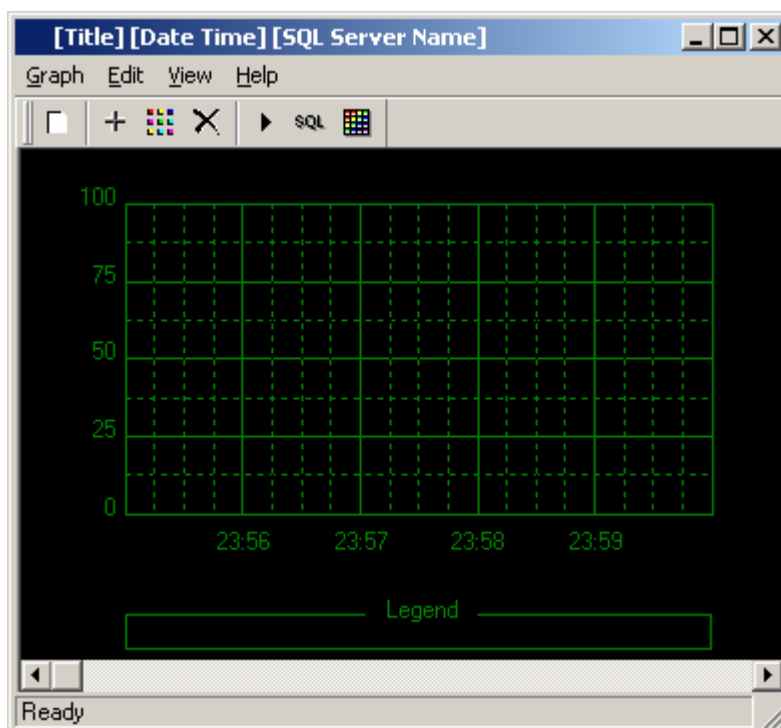
Graphs' display options

New graph

To view the monitoring results related to a specifically targeted SQL Server's performance indicators, 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, it is possible to view the SQL Server's performance curves. However, this window is not initialized. You first need to choose the indicators that you want to display. You must define the appropriate graph's scale to display the

chosen indicator(s) on the graph.

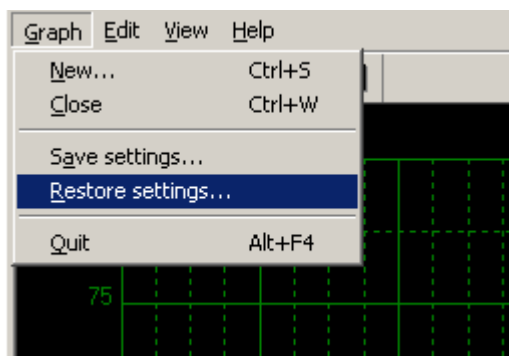
The SQL Server's indicators

Indicator's name	Description
CPU_RATIO	Reports the percentage of time that a process or set of processes running a given application was/were in operation.
DEADLOCKS	Reports the number of deadlocks per second.
LOCK_WAITS	Reports the number of locks per second that were not granted immediately and that required a waiting time before the release of another lock.
LOCK_REQUESTS	Reports the number of lock requests per second.
LOG_WAITS	Reports the number of times per second log writes were waiting.
NET_PKTS_RCVD	Reports the number of network packets received since last sample.
NET_PKTS_SENT	Reports the number of network packets sent since last sample.
BUFFER_CACHE_RATIO	Reports the percentage of times that a data page read could be satisfied from cache without requiring a physical page read.
PAGE_READS	Reports the number of page reads per second.
PLAN_CACHE_RATIO	Reports the percentage of times that a SQL execution found the query plan in cache and available for use.

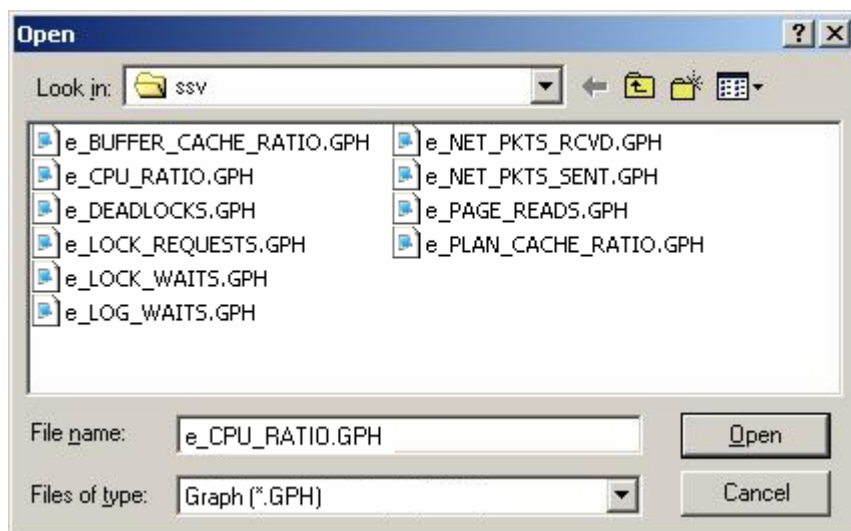
There are two ways to display an indicator. You can either use the preset graphs' configuration files or you can manually define the graph's aspect and its contents (See Save a graph's configuration).

How to restore a configuration

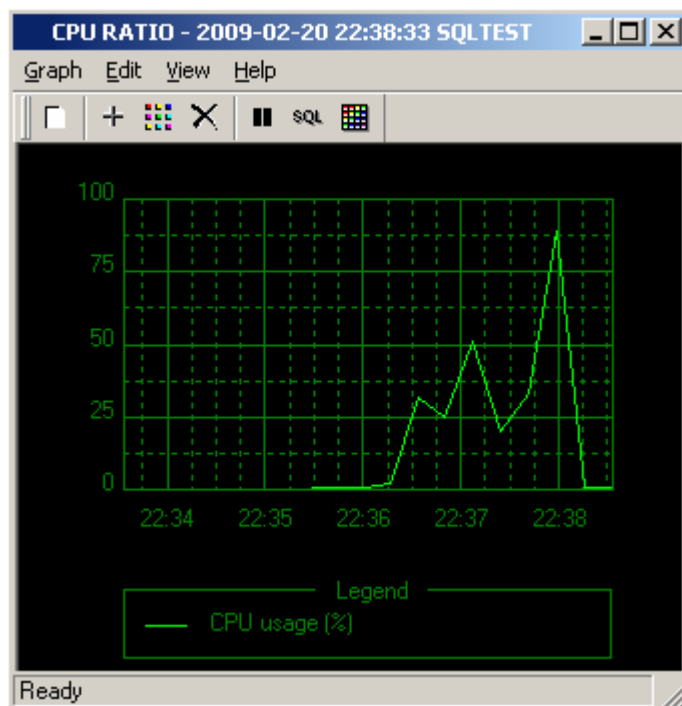
Select the «Graph - Restore settings...» menu item.



Next, determine the location and the name of the file containing the requested graph configuration. By default, «.GPH» files are stored in %APPDATA%\SQL Shot\cfg\gph directory.



As soon as you have chosen the indicator you would like 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 SQL Shot's installation. However, you can create your own configuration files or adapt those provided.

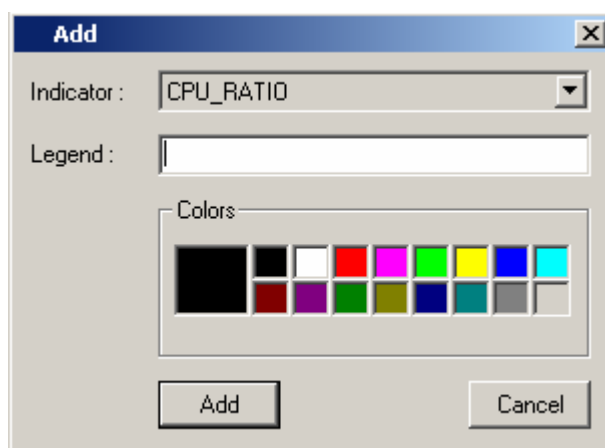


How to add an indicator

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, it is possible to define the indicator you would like to view. First, select the «Indicator» dropdown list box containing the various types described above (See «The server's indicators»). Next, enter the text you want to display in the lower part of the graph. You can then choose the color of the values' curve. By clicking on the "Add" button, the graph changes immediately and reflects the new values of the selected indicator.

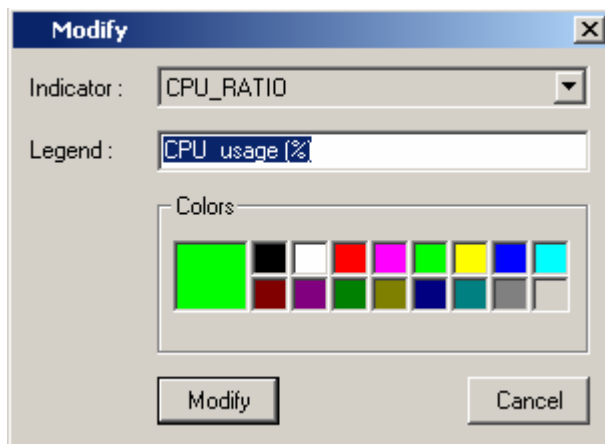
If the dropdown list box does not contain any value, it means that no monitoring session was created or that no values were read in the archive database or file. If the graph doesn't change after you have added an indicator, you need to change the graph's scale (See Paragraph below on «Graphs' display options»). The lack of change in the graph can also mean that there is no value for this indicator in the time range displayed in the session's report. (If this happens, please contact [SQL Shot's 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:



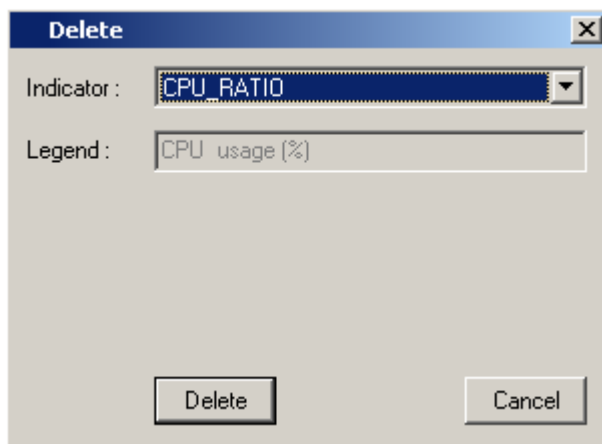
It is possible to modify the indicators' display characteristics that were previously added to the graph. You can modify the texts and/or colors by selecting the indicators you want to change from the «Indicator» dropdown list box. Next, click on the «Modify» button. The graph changes automatically to 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.



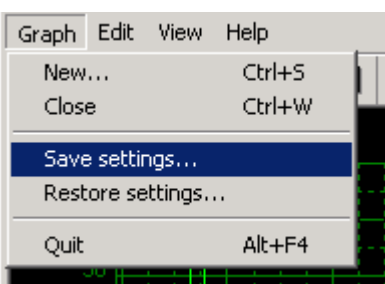
The following dialog box appears:



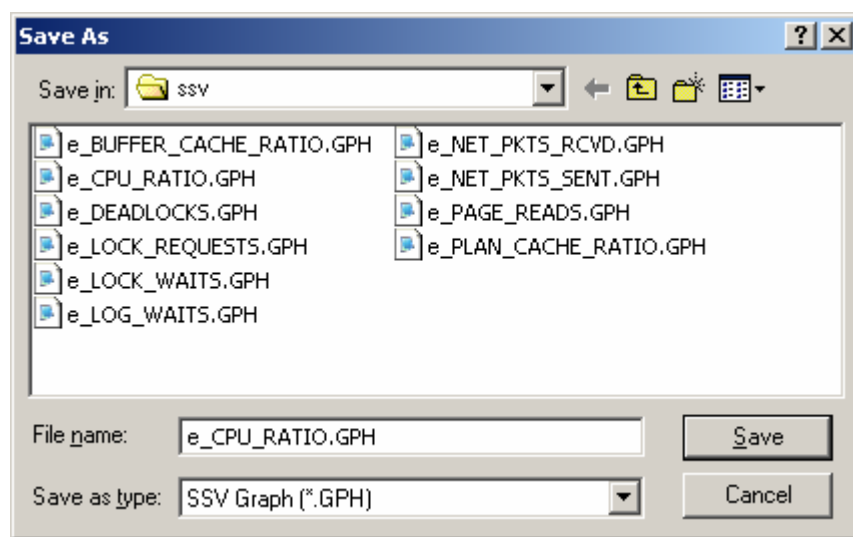
You can now remove indicators that were previously added to the graph. To do so, click on the «Delete» button. The selected indicator's values are immediately removed from the graph.

How to save the configuration

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

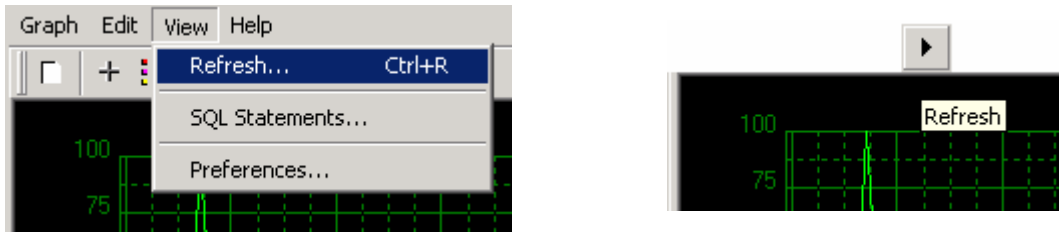


You will need to determine the location and the name of the file containing the configuration you have just saved. By default, «.GPH» files are stored in %APPDATA%\SQL Shot\cfg\gph directory.



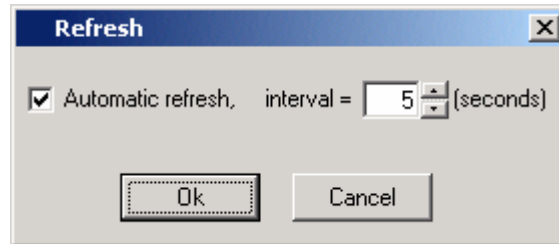
How to refresh the graph



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



This option makes it possible to refresh a report or a graph's displays, whether they are being produced locally or by a monitoring session in progress that is executed by another SQL Shot (See Save automatically). This option allows you to view from a distant client PC workstation the events captured by a workstation dedicated to monitoring. (See SQL Shot's executing in background task).

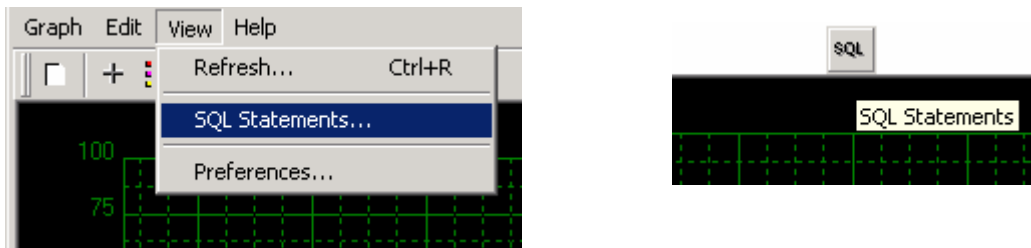
When you activate the automatic refresh function while reading a file or an archive database extract, a dialog box asks you to define the refresh frequency required. In direct monitoring mode (SQL Shot is connected to a SQL server), the automatic refresh function is performed with the same frequency as the monitoring; this means, in this case, that the refreshing frequency cannot be modified. However, you have the ability to interrupt the refreshing.



As soon as automatic refreshing is activated, the refresh icon goes from «read» mode  to «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).



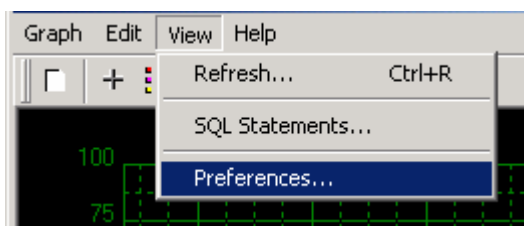
The following window appears. It contains only the SQL Statements captured from the beginning date-time of the graph to the ending date-time of the graph.

Captured SQL Statements between '2009-02-21 00:12:12:' and '2009-02-21 00:17:12'								
#	Spid	Login	Host	Application	End	SQL Statement	Batch Text	Elas
12	56	Paul	DEMO	OSQL-32	2009-02-21 00:16:03:187	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0).	
11	54	Yves	DEMO	OSQL-32	2009-02-21 00:16:03:187	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0).	
10	55	Greg	DEMO	OSQL-32	2009-02-21 00:16:03:187	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0).	
9	53	Thierry	DEMO	OSQL-32	2009-02-21 00:16:03:187	waitfor delay @delay	exec sp_dbarc_lock_cnt	
8	54	Yves	DEMO	OSQL-32	2009-02-21 00:15:43:280	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0).	
7	55	Greg	DEMO	OSQL-32	2009-02-21 00:15:43:280	update Dbarc_cnt set id = id + 1 where code = @code	begin tran declare @id numeric(5,0).	

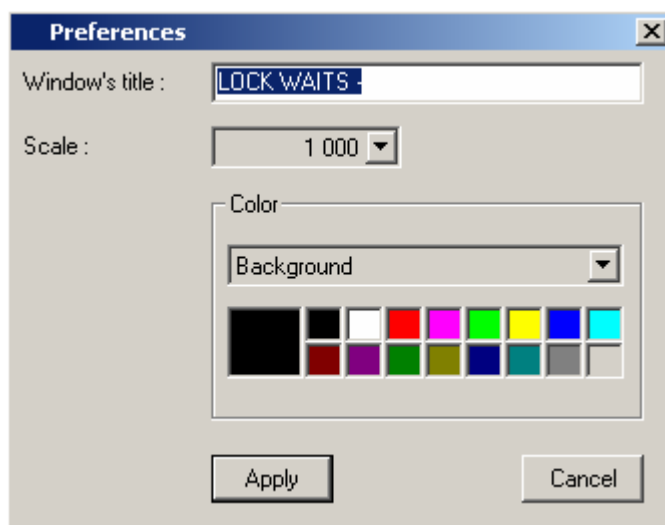
From this window, it is possible to export the data to Excel or to a .txt format file. It is possible to display SQL statements' graphs as well and to view the detail of a given SQL Statement, following the same principles as those applying to a monitoring session's report.

Graphs' display options

It is possible to modify the graphs' display options (graph's title and scale, window's fill color and graph grid's color). To do this, select the «View - Preferences...» menu item. Alternatively, click on the «Preferences» menu icon (See Figures below).



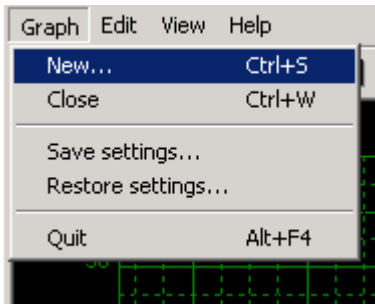
The following dialog box appears:



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:

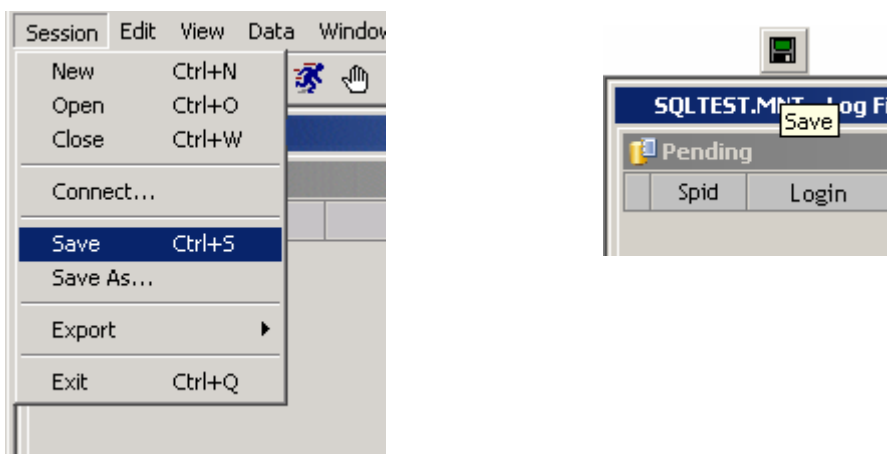
How to save to a database

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 options for saving your information: you can save it to a file or a database.

How to save to a database

You can save the contents of a session's report to a SQL Server's database. There is no technical prerequisite attached to this database. The database's physical model is provided in the appendix (See Database model).

This feature is only available in the SQL Shot's Enterprise Edition.

Technical recommendations

- 1) For performance's sake, the archive database should not be located on a WAN (from the SQL Shot collecting the monitoring results).
- 2) We advise you to not install this database on a production server.
- 3) Volume required for the archive database:

You need to allow for 10 KB per captured SQL Statement, on average, and for 1 Mo, for servers' indicators values, per monitoring hour. As an example, it is normal to capture more than 5000 SQL Statements during a given day on a targeted production SQL Server with performance problems. This represents a total volume of 50 Mo for the SQL Statements and of 24 Mo for the server's indicators values. Approximately 75 Mo are delivered, each monitoring day, for SQL Shot's stored delete procedures (See Chapter on «Results suppressions»). It is necessary to manage the archive database's free space. (A good method is to set alarm thresholds to the «default» setting and to «logsegment» segments. We advise you to create a database with a minimum size of 200 Mo for data and of 50 Mo for logs.

Permissions management

The login used to record the results in the archive database must be that of the database owner (DBO). The users who view the information can be simple «users» in the «public» group of this database. They will be able to execute logical deletes on the results (See Chapter on «Results suppressions»). However, they will not be allowed to physically remove information as it is the sole privilege of the DBO user.

The viewing of information within a centralized database can be of critical importance for your security. If this is what you are intending to do, you can create views as follows:

- 1) You create a login (sp_addlogin).
- 2) You give to him the permission to use the archive database (sp_adduser).

You modify the view "ssv_stmt_user":

```
Drop view ssv_stmt_user
go
create view ssv_stmt_user
as
select * from ssv_stmt
where < criteria of restriction >
```

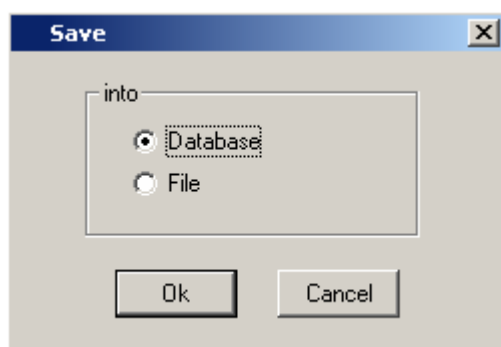
example:

```
drop view ssv_stmt_user
go
create view ssv_stmt_user
as
select * ssv_stmt
where (user_name() = "develop" and servname = "SERVPROD" and appname = "PORTFOLIO")
```

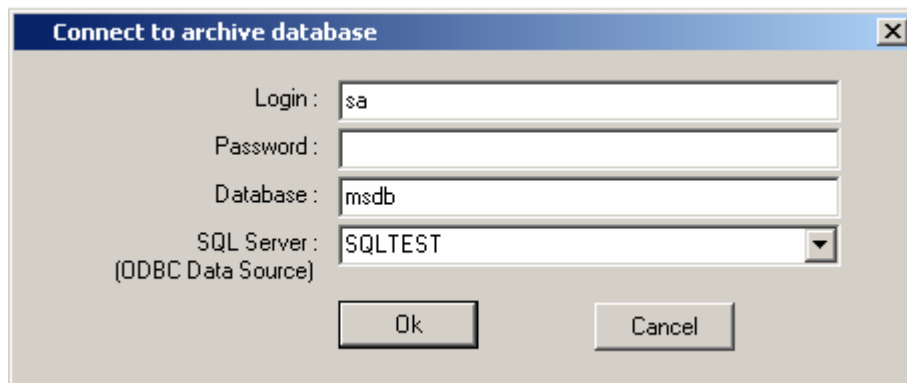
The "PORTFOLIO" application developers can only view the performance results of their own stored procedures. Do not hesitate to contact [SQL Shot's Technical Support](#) to validate your criteria concerning the SQL Shot's database model and performances using indexes.

Connect to the archive database

Choose the location for the archive. The following dialog box appears:



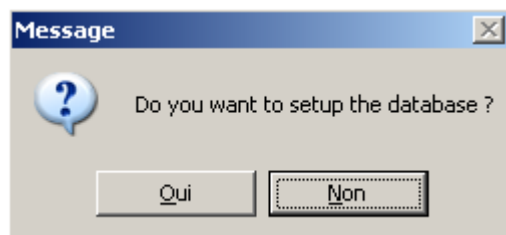
After selecting «Database», connect to it.



If the database is not yet formatted or if its structure has been modified, SQL Shot sends you the following message to let you know:



SQL Shot suggests that you execute the database setup. The connected login must be that of the DBO (database owner).



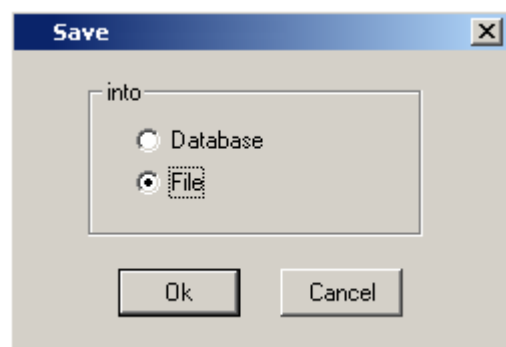
Caution: There is no specific restoration procedure delivered with the SQL Shot package. If you choose: «yes» (database setup), all previous data will be lost. While saving and restoring SQL Server databases, use the tools provided and certified by Microsoft.

How to save to a file

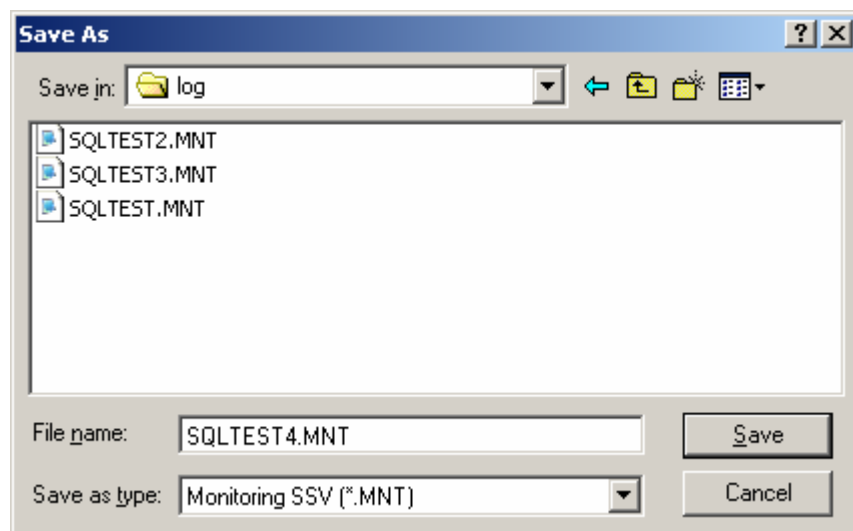
Recommendations:

- 1) For performance's sake, the archive file containing the monitoring results should not be located on a WAN (from the SQL Shot version recording the results).
- 2) We advise you to not create this file on a production server.
- 3) Volume required for the archive file:
You need to allow for 10 KB per captured SQL Statement, on average, and for 1 Mo, for servers' indicators values, per monitoring hour. As an example, it is normal to capture more than 5000 SQL Statements during a given day, on a targeted production SQL Server with performance problems. This represents a total volume of 50 Mo for the SQL Statements and of 24 Mo for the server's indicators values. Approximately 75 Mo are delivered, each monitoring day, for SQL Shot's stored delete procedures (See Chapter on «Results suppressions»). It is necessary to manage the free space of the archive file directories.

Choose the location for the archive. The following dialog box appears.



After selecting «File», define its location and name. The «.MNT» files containing performance data collected by SQL Shot are stored in the default directory named %APPDATA%\SQL Shot\Data Files.



How to save automatically

When a monitoring session is active, a window appears to setup the session's report automatic-save frequency. It is defined in seconds. The newly captured SQL Statements are automatically saved to the file at the selected frequency. The automatic-save frequency is independent from the monitoring frequency. (It cannot be changed.).



Exports

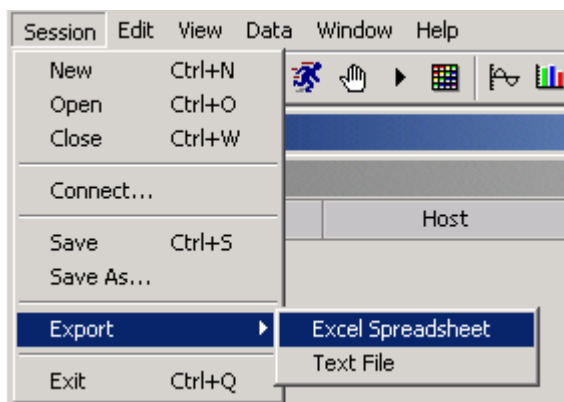
See also:

How to export to Excel

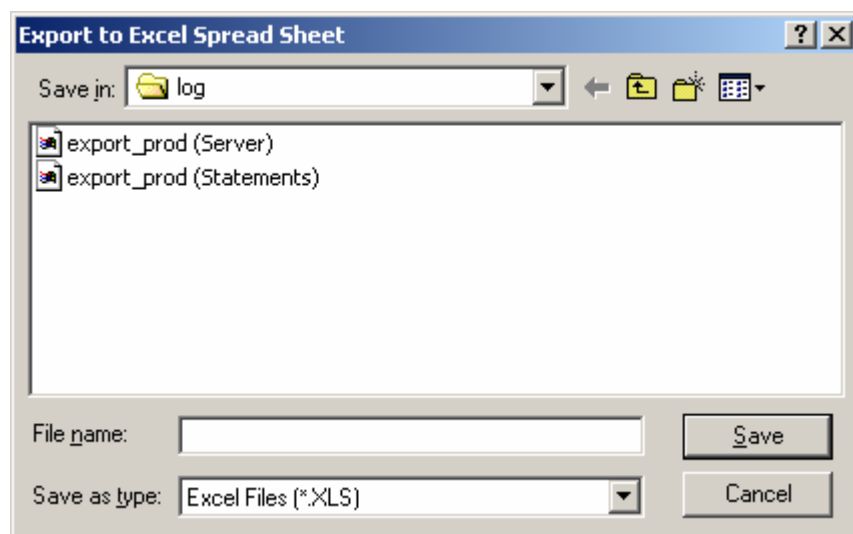
How to export to a text file

SQL Shot makes it possible to export all the SQL Statements or some of them to a file (Excel format or Text format). Instead of exporting results to a database, you can save them to a database (See Chapter on "How to save to a database"). To do that, select the SQL Statements you wish to export then select the «Session - Export» menu item and choose the file format required to export the data.

How to export to Excel

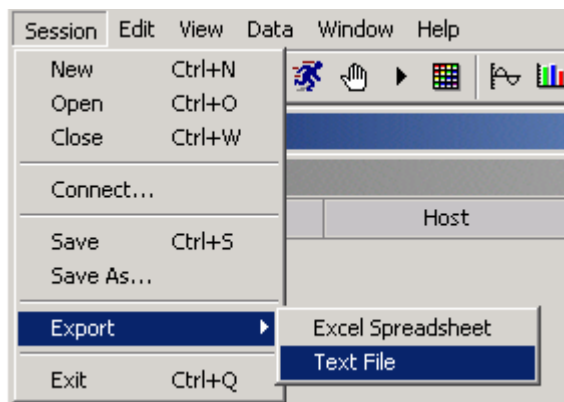


Enter a file name in the directory of your choice. The «.XLS» files containing performance data collected by SQL Shot, exported to Microsoft Excel format, are stored in the default directory named %APPDATA%\SQL Shot\Data Files.

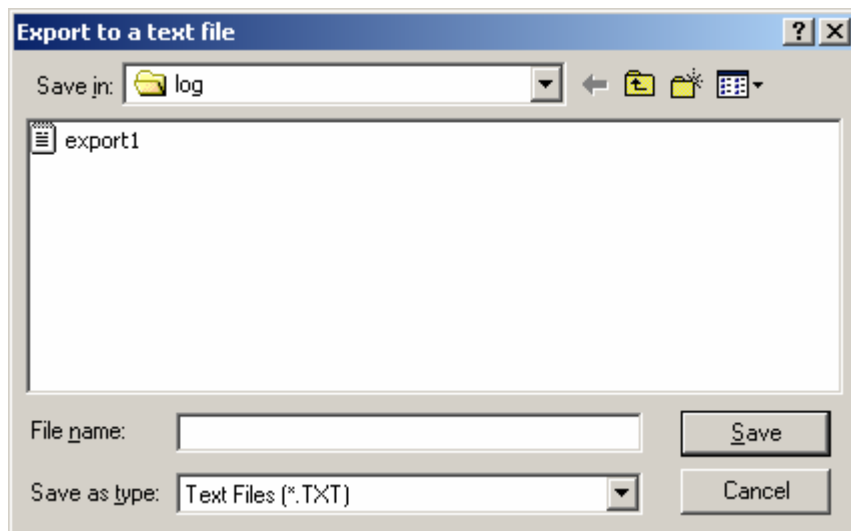


The export generates two files: < file name > (Statements).XLS, which contains the lines of the session's report and < file name > (Server).XLS, which contains the server's performance indicators' values.

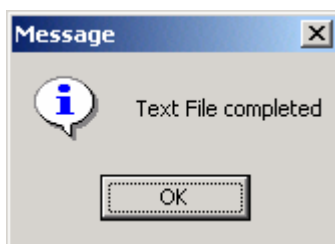
How to export to a text file



Enter a file name in the directory of your choice. The «.TXT» files containing performance data collected by SQL Shot, exported to text format, are stored in the default directory named %APPDATA%\SQL Shot\Data Files.



When the export operation is finished, 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 Server's performance indicators' values are not exported to the text file.

How to retrieve results

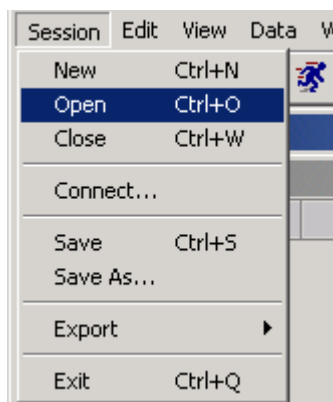
See also:

How to retrieve from a database

How to retrieve from a file

Automatic refresh

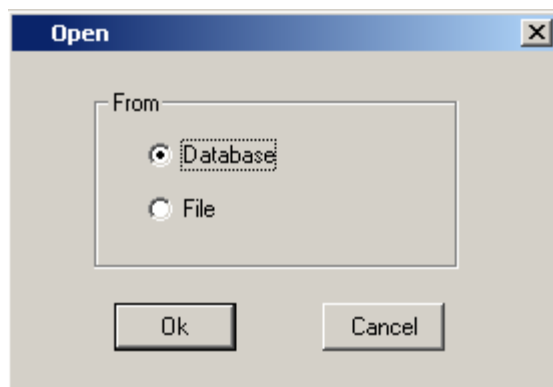
To open an archive, select the «Session - Open...» menu item. Alternatively, click on the «Open» menu icon (See Figures below).



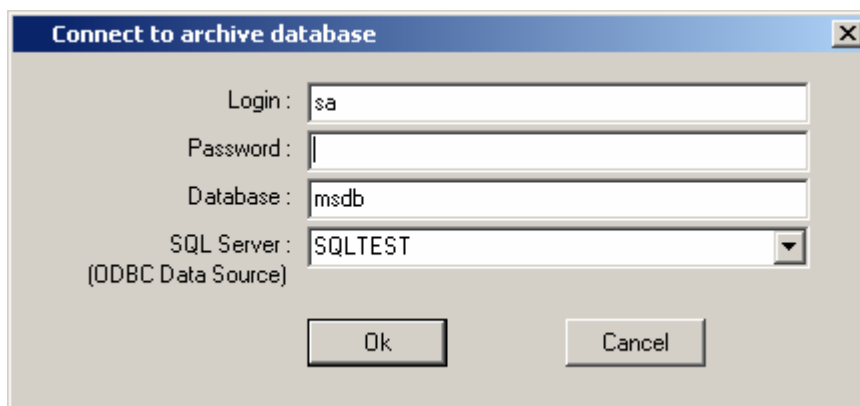
SQL Shot allows the two following archive formats: ASCII files and SQL Server dedicated databases.

How to retrieve from a database

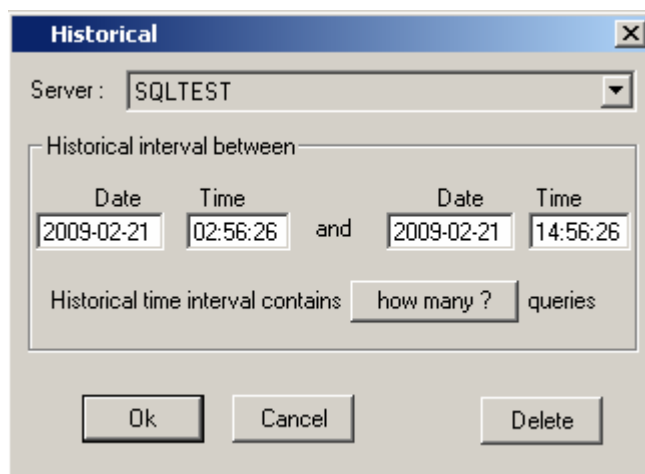
Choose the location for the results. The following dialog box appears:



After selecting «Database», connect to the archive database containing SQL Shot's monitoring results.



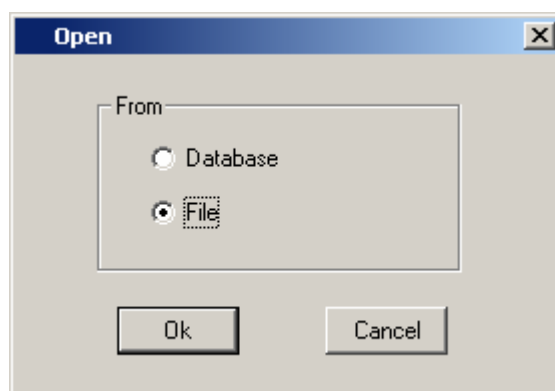
A dialog box enables you to define the targeted SQL Server and the time range selected to extract the performance information.



The «Server» dropdown list box contains the various SQL Server names for the monitoring data that exists in the database. Define the time interval needed. The «how many?» button enables you to know how many SQL Statements are present in the database according to these criteria. This function makes sure that your PC workstation's memory is not saturated with a bulky session's report. If you click on this button, the number of SQL Statements replaces the string «how many?». The «Delete» feature is explained in the paragraph on «Results suppressions».

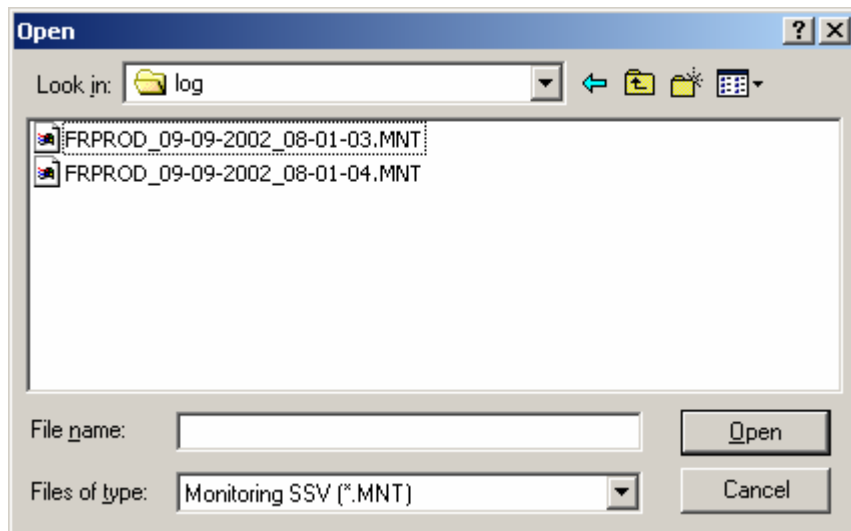
How to retrieve from a file

Choose the location from the following dialog box.



After selecting «File», define the directory and the file name containing the SQL Shot's monitoring results. The «.MNT»

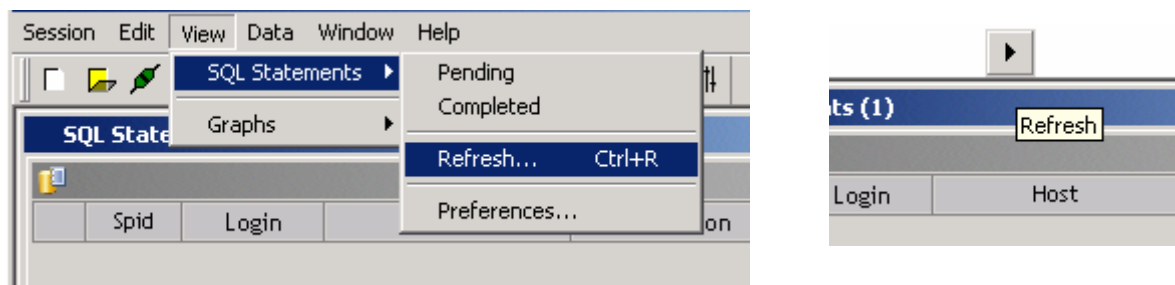
files containing performance data collected by SQL Shot are stored in the default directory named %APPDATA%\SQL Shot\Data Files.



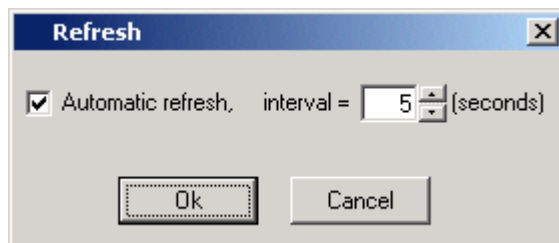
All SQL Statements contained in this file are loaded in the session's report.



Automatic refresh



It is possible to read a file currently being worked on by an active monitoring session running on a different PC workstation. To automatically refresh the display, select the «View - Refresh...» menu item. Alternatively, 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 refreshing frequency. In the direct monitoring mode (SQL Shot is connected to the SQL server), the automatic refresh function is performed as frequently as the monitoring. In this case, the refreshing frequency cannot be modified. However, it is possible to interrupt the refreshing.

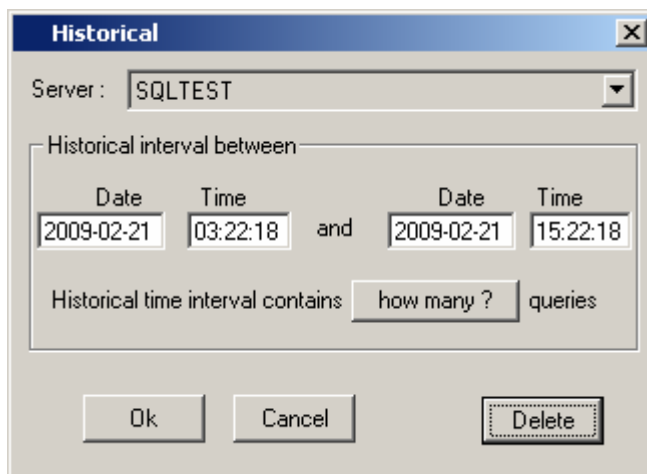


As soon as the automatic refresh function is activated, the refresh icon goes from «read» mode  to «pause»  mode. Starting with Release 3 of the software, the session's connection (See Paragraph on «How to start and stop

monitoring») and the automatic refresh function of SQL Statements' display (See Paragraph on «Session report's contents») are independent from each other. This means that you cannot view the SQL Statements' detail of the «Completed» report when the Automatic Refresh function is active (icon , and when the «Completed» report is grayed out). However, it is possible to interrupt the automatic refresh and to resume it () without losing the results of the monitoring session in progress. During this interruption, the captured SQL Statements are saved in the background and you can view the detail of the SQL Statements present in the reports. As for former releases, it is possible to view the SQL Statements' detail (that are in a pending state) in real time, when the Automatic Refresh function is active.

Results' suppressions

From this window, it is possible to activate the deletion of the monitoring results (SQL Statements and server's indicators' values) from an archive database (See below: «Delete» button). The SQL Statements and the server's performance indicators' values associated with the server and the selected time period are removed.



The image shows a dialog box titled "Historical" with a close button (X) in the top right corner. The dialog contains the following elements:

- A "Server:" label followed by a dropdown menu showing "SQLTEST".
- A section titled "Historical interval between" containing a table of date and time inputs.
- A "Historical time interval contains" label followed by a text input field containing "how many ?" and the word "queries".
- Three buttons at the bottom: "Ok", "Cancel", and "Delete". The "Delete" button is highlighted with a dashed border.

Historical interval between	
Date	Time
2009-02-21	03:22:18
and	
Date	Time
2009-02-21	15:22:18

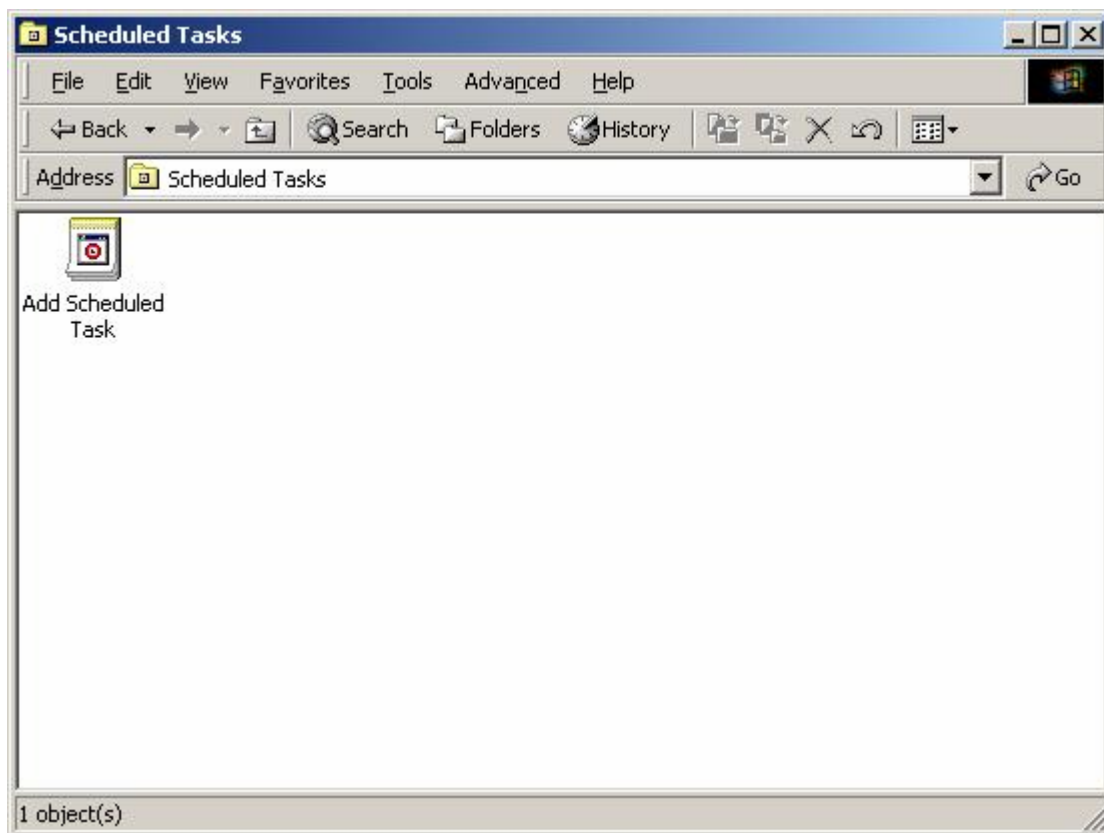
SQL Shot's execution in a background task

You can launch SQL Shot in a background task. To do so, create a scheduled task and use the background task's execution parameters.

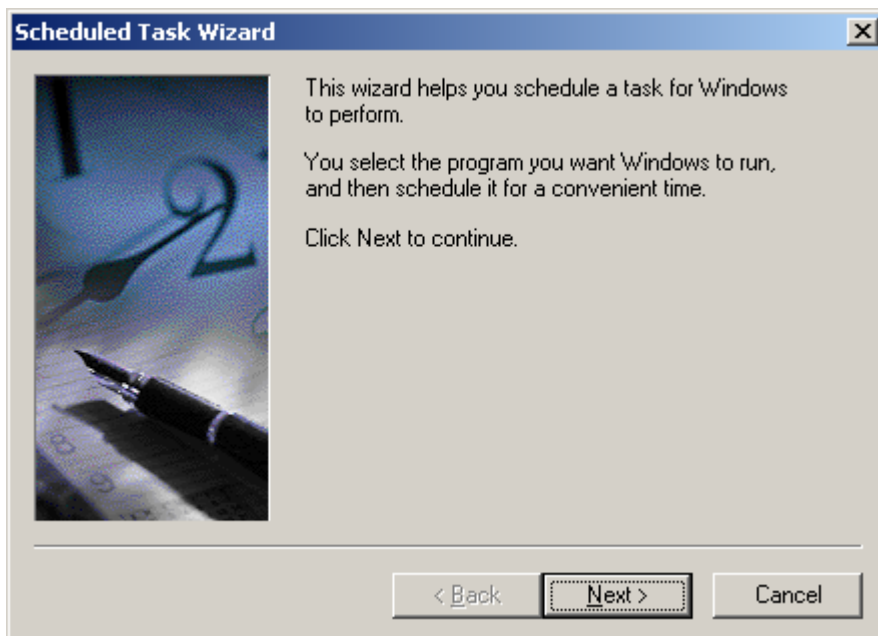
This feature is only available in the SQL Shot's Enterprise Edition.

Create a scheduled task

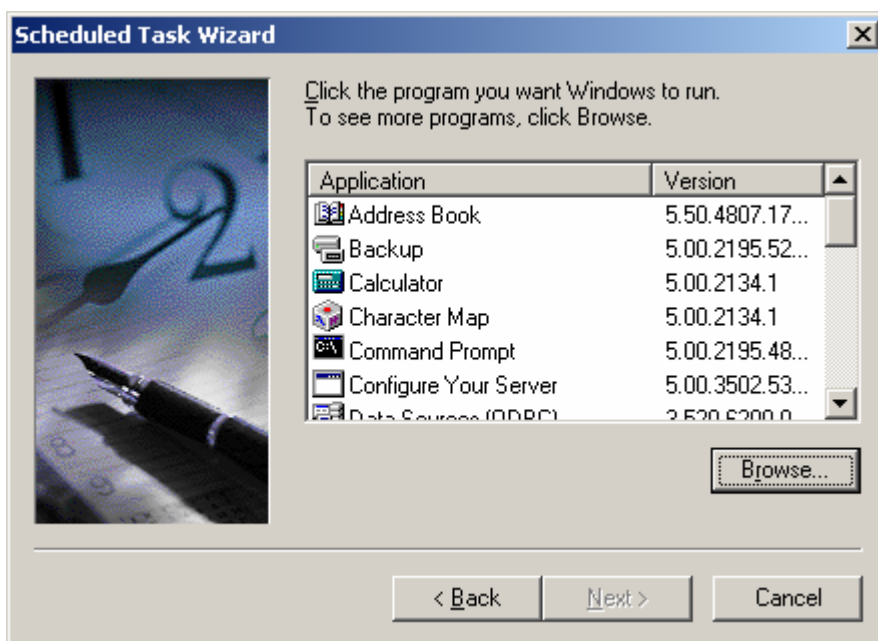
The following example illustrates the steps to follow. In the «Start - Settings - Control panel» menu, choose the «Scheduled Tasks» utility (to setup an automatic process execution in a background task of your computer). The window dedicated to the management of scheduled tasks enables you to create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



Next, click on «Add Scheduled Task».



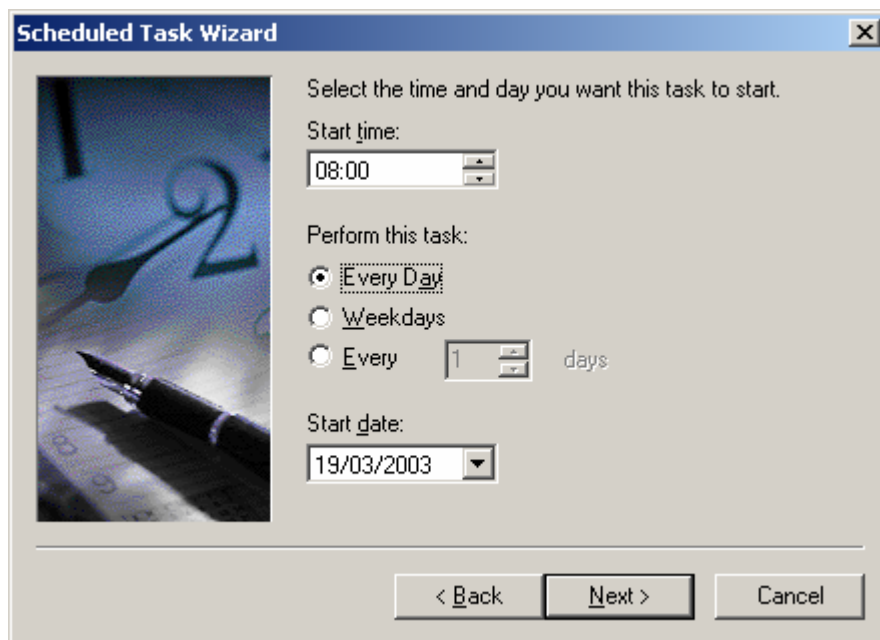
Now, click on the «Next>» button.



Click on the «Browse...» button to look for the SQL Shot's EXE file (SQL Shot.exe) which is in the installation directory. Next, choose the launch options. It is better to define a daily execution, especially if you intend to record the results to a file, to make sure that it is not too bulky.



You can now define a start time.



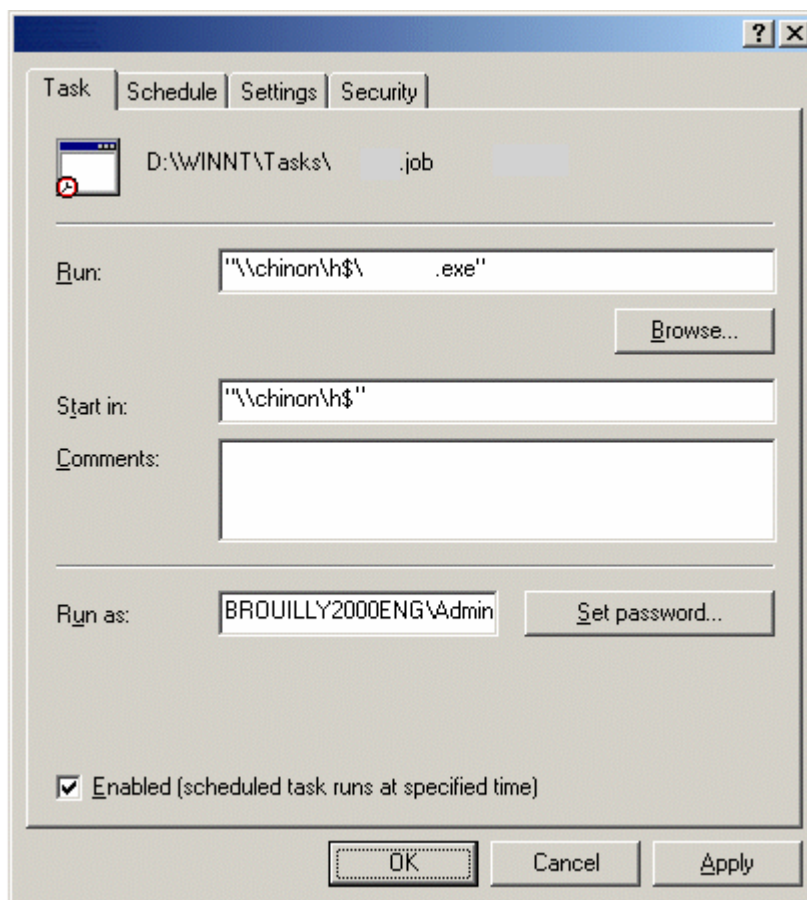
Enter the login of the background task's owner (the person who is launching the scheduled task). It is preferable to use a login that is not set to a specific person, in order to avoid any inappropriate password modifications.



The scheduled task creation is almost complete.



You now need to complete the command line (in the following «Run: « window) using SQL Shot's background task execution's options described below. The "Start in: "edit line defines the execution's directory, namely the place containing the SQL Shot's error log file named «SQL Shot log ». It contains SQL Shot's execution traces and all messages related to potential incidents.



Background task execution

- [-D] session's duration (in minutes)
- [-F] filters file
- [-L] license level
- [-O] archive file
- \$dd\$: day_month_year
- \$hh\$: hour-minute-second
- example:
SERVER_\$dd\$_\$hh\$.MNT --> SERVER_05-11-2009_15-01-03.MNT
- [-P] password of the login connected to the monitored SQL Server
- [-S] monitored SQL Server's name (ODBC datasource)
- [-U] login connected to the Monitored SQL Server
- [-V] application is visible (YES/NO)
- [-W] login connected to the SQL Server hosting the archive database
- [-X] password of the login connected to the SQL Server hosting the archive database
- [-Y] archive database
- [-Z] SQL Server hosting archive database (ODBC datasource)
- [-I] snapshot frequency for the SQL Statements (milliseconds), default = 1000
- [-J] snapshot frequency for the SQL Server instance (milliseconds), default = 15000
- [-K] this option is used to force a rate faster than one second, this for queries and SQL Server's performances' indicators. Thus, if you select a frequency faster than one second, the -K option, without any associated value, will be mandatory (see example below), see the warning in chapter «how to start and stop monitoring».

Example 1:

The following command creates a session whose task is to monitor the «SQLTEST» SQL Server 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\SQL Shot\sqls42.exe « -D600 -L5000 -OSQLTEST_$dd$_$hh$.MNT -P** -SSQLTEST -Usa -VNO
```

Example 2:

The following command creates a session whose task is to monitor the «SQLTEST» SQL Server for 10 hours. The results are recorded in a database named «ARCHIVE» located on another SQL Server called «SQL_ARCH». SQL Shot connects to the latter with the login «user1». The frequency for queries' performances' information is equal to 500 milliseconds and the -K option is present to confirm this frequency selection.

```
« C:\Program Files\SQL Shot\sql42.exe « -D600 -L5000 -P** -SSQLTEST -Usa -VNO -Wuser1 -X** -YARCHIVE -ZSQL_REC -I500 -K
```

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

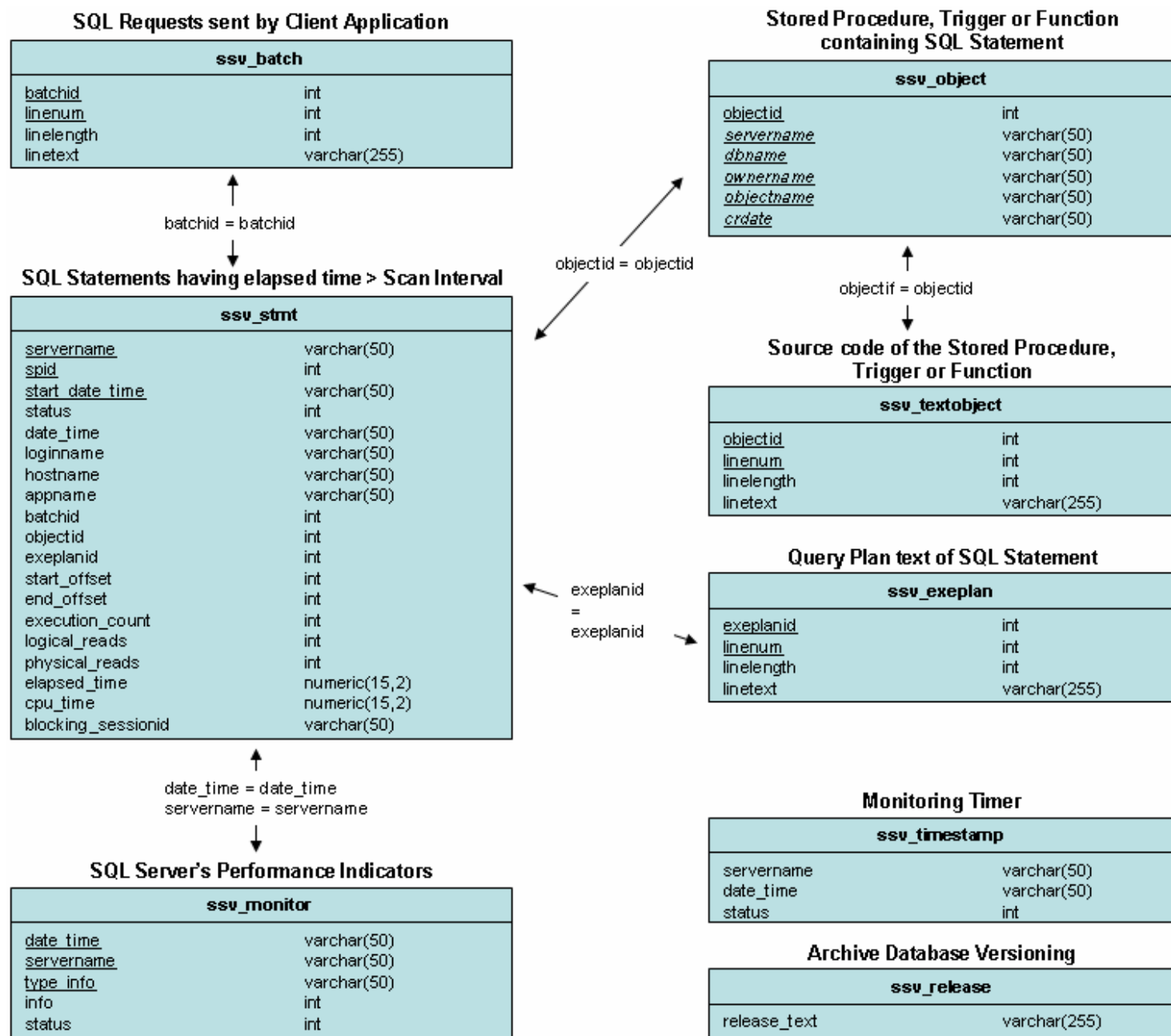
The «.MNT» files containing performance data collected by SQL Shot are stored in the default directory named %APPDATA%\SQL Shot\Data Files.

How to shutdown the background mode session

When it is launched in the background mode, SQL Shot creates a file called < Monitored SQL Server >.tmp in the %APPDATA%\SQL Shot\log directory. It is controlled by SQL Shot at every cycle of the Scan Interval. If this file is missing, SQL Shot stops the session started in the background mode. To stop a SQL Shot 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.

SQL Shot's archive database model

See also:
Tables



Tables

See:

ssv_stmt
ssv_batch
ssv_object
ssv_textobject
ssv_exeplan
ssv_monitor
ssv_release
ssv_timestamp

ssv_stmt

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

Table Column	In the GUI	Description
servername	Window's title	Monitored SQL Server's name (ODBC datasource).
spid	Spid	SQL Server Process Id executing the SQL Statement.
start_date_time		Date and time of the execution's start.
status		"1" = SQL Statement is pending, "0" = SQL Statement is completed.
loginname	Login	Login name of the connected user.
hostname	Host	Name of the client PC workstation that sent the query to the SQL Server.
appname	Application	Name of the application that sent the query to the SQL Server.
date_time	Date Time (Start) Date Time (End)	For pending SQL Statements, it is the date and time of the execution's start. For completed SQL statements, it is the date and time of the end of the execution.
batchid	Batch Text Join with "ssv_batch" table.	Foreign key used to establish the link with the "ssv_batch" table.
objectid	Procedure - Trigger - Function (join with "ssv_object" and "ssv_textobject" tables)	Foreign key used to establish the link with the "ssv_object" and "ssv_textobject" tables. If "objectid" = "0", the recorded SQL Statements belong neither to a stored procedure nor to a trigger, nor to a function. Therefore, it acts following a query sent in dynamic SQL.
exeplanid	Execution plan (join with "ssv_exeplan" table)	Foreign key used to establish the link with the "ssv_exeplan" table.
start_offset	SQL Statement (start of its position inside the source code)	Position start of the SQL Statement within the stored procedure, trigger or function or batch text (dynamic SQL).
end_offset	SQL Statement (end of its position inside the source code)	Position end of the SQL Statement within the stored procedure, trigger or function or batch text (dynamic SQL).
execution_count	Executions count	Number of times that the SQL Statements are executed within a loop.
logical_reads	Logical reads	Number of reads in BUFFER CACHE that are used by the SQL Statement.
physical_reads	Physical reads	Number of reads from disk that are used by the SQL Statement.
elapsed_time	Elapsed Time	Elapsed time used by the SQL Statement.
cpu_time	CPU Time	Computing time used by the SQL Statement.
blocking_sessionid	Blocking Session, Last Blocking Session	Session Id currently blocking or having blocked the present session.

ssv_batch

The lines in this table correspond to the SQL queries sent by the client PC workstations that prompt the execution of the SQL Statements captured by SQL Shot. Therefore, please note that there is a (1,N) association, where N is the number of lines in the SQL query that are present between the «ssv_stmt» table and the «ssv_batch» table.

Table Column	Session's Report Column	Description
batchid		This column is part of the primary key (batchid, linenum). This identifier is used to establish the link with the foreign key "batchid" of the table "ssv_stmt".
linenum		This column is part of the primary key (batchid, linenum). The text of the query is cut out in lines of 255 characters.
linelength		Length of the text line.
linetext	SQL Statement (If it is part of the batch text sent by the application – ie : dynamic SQL mode)	Query's text line.

ssv_object

The lines in this table correspond to the stored procedures or triggers containing the SQL Statements captured by SQL Shot. Therefore, please note there is a (0,1) association between the «ssv_stmt» table and the «ssv_object» table. Indeed, there is no correspondence (cardinality 0) between the two tables when the impacted SQL Statement is from a dynamic SQL.

Table Column	Session Report's Column	Description
objectid	Procedure – Trigger - Function (join with the "ssv_stmt" table)	Primary key of the table. This identifier is used to establish the link with the foreign key "objectid" of the "ssv_stmt" and "ssv_textobject" tables.
servername		Name of the Monitored SQL Server.
dbname		Database name containing the object.
ownername		Owner name of the object.
procname		Name of the stored procedure or trigger or function.
crdate		Creation date of the object. The versions are archived. It is thus possible to compare the performances recorded by SQL Shot on the various versions of the object.

ssv_textobject

The lines in this table correspond to the source codes of the stored procedures or triggers containing the SQL Statements captured by SQL Shot. There is a (1,N) association, where N illustrates the number of lines of the source code that are present between the «ssv_stmt» and the «ssv_object» tables.

Table Column	Session's Report Column	Description
objectid		This column is part of the primary key (objectid, linenum). This identifier is used to establish the link with the foreign key ("objectid") of the "ssv_stmt" table and the primary key ("objectid") of the "ssv_object" table.
linenum		This column is part of the primary key (objectid, linenum). The text of the source code (stored procedure or trigger or function) is cut out in lines of 255 characters.
linelength		Length of the text line.
linetext	SQL Statement (If it is part of a stored procedure, trigger or function)	Text Line of the source code of the stored procedure or the trigger or the function.

ssv_exeplan

The lines in this table correspond to the execution plans of the SQL Statements captured by SQL Shot. There is a (1,N) association, where N represents the number of lines of the execution plan that are present between the «ssv_stmt» and the «ssv_exeplan» tables.

Table Column	Session's Report Column	Description
exeplanid		This column is part of the primary key of the table (exeplanid, linenum). This identifier is used to establish the link with the foreign key ("exeplanid") of the "ssv_stmt" table.
linenum		This column is part of the primary key (exeplanid, linenum). The execution plan is cut out in lines of 255 characters.
linelength		Length of the text line.
linetext	Execution Plan (join with the "ssv_stmt" table)	Text Line of the Query Plan.

ssv_monitor

The lines in this table contain the values of the performance indicators collected at the SQL Server's level. These values can be viewed in conjunction with the graphs.

Table Column	Description
date_time	Timestamp of the indicator's value.
servername	Monitored SQL Server on which the measure was taken.
type_info	Indicator : BUFFER_CACHE_RATIO CPU_RATIO DEADLOCKS LOCK_REQUESTS LOCK_WAITS LOG_WAITS NET_PKTS_RCVD NET_PKTS_SENT PAGE_READS PHYSICAL_IOS PLAN_CACHE_RATIO (for further details, See Chapter on « server's indicators »)
info	Value
status	0.

ssv_release

The installed SQL Shot's archive database release.

Column of the table	Description
release_text	Character string which defines the version. You run in the database : Select * from ssv_release You obtain the following result: "<APPNAME> Database - X.X - date"

ssv_timestamp

It contains the monitoring sessions' status recorded in the base files.

Table Column	Description
servername	Monitored SQL Server's name.
date_time	Date, hour, minutes and seconds of the last monitoring's snapshot.
status	0 = not activated 1 = activated