VISUAL
message center
powered by thinkserver

SQL Server Security Agent

User Guide

1.6

VMC-TSS

tango04
Computing Group
Solutions for Advancing People

VISUAL Message Center SQL Server Security Agent User Guide

Copyright Notice

Document date: August 2012

Document version: 2.41

Product version: 1.6

Trademarks

Technical Support

For technical support visit our web site at www.tango04.com.

Tango/04 Computing Group S.L.
Avda. Meridiana 358, 5 A-B
Barcelona, 08027
Spain

Tel: +34 93 274 0051

Table of Contents

Chapter 9

Chapter 10

Chapter 11

Chapter 12

Chapter 13

Chapter 14

Chapter 15

Chapter 16

Chapter 17

Chapter 18

Appendices

How to Use this Guide

This chapter explains how to use Tango/04 User Guides and understand the typographical conventions used in all Tango/04 documentation.

Typographical Conventions

The following conventional terms, text formats, and symbols are used throughout Tango/04 printed documentation:

Convention	Description
Boldface	Commands, on-screen buttons and menu options.
Blue Italic	References and links to other sections in the manual or further documentation containing relevant information.
Italic	Text displayed on screen, or variables where the user must substitute their own details.
`Monospace`	Input commands such as System i commands or code, or text that users must type in.
UPPERCASE	Keyboard keys, such as CTRL for the Control key and F5 for the function key that is labeled F5.
	**Notes** and useful additional information.
	**Tips** and hints that will improve the users experience of working with this product.
	**Important** additional information that the user is strongly advised to note.
	**Warning** information. Failure to take note of this information could potentially lead to serious problems.

Chapter 1
Introduction

VISUAL Message Center offers a full range of monitoring capabilities for one or several SQL Server database engines.

It supports SQL Server 7, SQL Server 2000, SQL Server 2005 and Desktop Edition (MSDE) versions running on Windows 2000, Windows 2003 Server Edition, or Windows 2008. Our solution's wide range of monitoring capabilities includes control and management of:

- Database Availability
- Database Performance
- Database Security, Regulatory Compliance and Auditing
- Business Services that use SQL Server as an application component

SQL Server Security ThinAgents come pre-configured so that you can easily and quickly start monitoring out of the box (right away).

VISUAL Message Center can alert in real time or near real time of events coming from a comprehensive list of sources, including, among others:

- SQL Server WMI performance providers
- SQL Server internal performance indicators
- SQL Server logs
- Windows Event Log
- SQL Server Processes and Services statuses
- Synthetic transactions (see details below)
- Transaction Logs
- Auditing events
- SQL Transactions
- SNMP variables provided by SQL Server
- SQL Server objects included in System Monitor
- Other Microsoft and third-party monitoring products, such as MOM

- Critical Operating System and Server hardware performance, availability, and security indicators for the servers where SQL Server resides
- Networked devices and related infrastructure services that may affect SQL Server
- Business Services affected by database status

Usually there is no need to use all of these methods at once (in fact, some of them will retrieve exactly the same data), but is important to note that VISUAL Message Center is flexible enough to use the best monitoring strategies for each customer or scenario.

Non-relevant events can be filtered at source or at the SmartConsole to save CPU resources, network bandwidth and the need for operator-attended supervision. Only relevant events are highlighted and brought to the operator's attention, to avoid the difficulty of sorting out critical information amongst great amounts of not relevant technical data.

Note
This manual is intended for users with knowledge of SQL. The manual explains how VISUAL Message Center ThinkServer uses SQL variables to monitor SQL Performance and Security. This manual does not teach you how to use SQL.

Chapter 2
Auditing Method

This chapter explains how the SQL Server auditing process works and how VISUAL Message Center ThinkServer goes about retrieving the auditing information. We strongly recommend you upgrade to the latest SQL Server Service Packs before getting started.

The first step is to create a data source for the SQL Audit. When you create the data source, an auditing process (trace) is created on the server.

Once you have created a data source you can narrow down the auditing process by applying filters to the data source. Furthermore you can configure filters at monitor level and set conditions for setting health and actions to carry out.

Note that:

- The auditing process will run on the server, whether the ThinkServer service is running or not.
- The retrieved auditing events are stored in temporary files on the server.
  - In SQL Server 7 servers, data is stored in tables
  - In SQL Server 2000 servers and later, data is stored in binary files in a system folder.
- When the server or the SQL Server Service is restarted, the auditing process is also restarted. The SQL Audit ThinAgent creates a stored procedure called `tango_restart_traces` when the first audit DataSource is created. This stored procedure is called when SQL Server restarts. It reads the tango_traces table and restarts every auditing process running on server before starting SQL Server.
- The process created on a server receives events from the entire server, not just from a specific database. If you only need to audit a specific database set Data Source filters on `DatabaseName` or `DatabaseID` variables.
- Auditing a SQL Server can have important effects on a server system. The input/output operations can increase to dangerous levels if the appropriate filters are not defined. Note that a simple query could generate 30 events that will be written to a file or database. So it is very important to decide which users, databases, types of query you need to audit. It is not possible for an SQL Server with heavy load to catch and save all possible events.
- Note that retrieving a large amount of events may lead to storage problems on the server. If ThinkServer is not running or if the speed at which new events are generated is greater than the speed at which the ThinkServer reads the events storage problems are likely to occur.

The auditing events from the temporary files are retrieved from the data source recollection process using SQL queries.

Note
Audit processes on the server will run until the Data Source is removed, whether the ThinkServer service is running or not.

Chapter 3
Data Source Management

3.1 Creating a Data Source

When you create a monitor you will be prompted to create a data source if none exists yet for the ThinAgent you selected. The data source configuration screen will appear. Here you can enter the information required for retrieving data.

Data sources come partially pre-configured. Adjust the default values to suit your company's needs and enter the following fields, as required:

- DSN (ODBC connection name, attached to the SQL Server)
- User
- Password
- SQL Server version – select whether you are using SQL 7 or SQL 2000

The ThinAgent uses the following templates to automatically name the SQL Server auditing process and intermediate files:

- In SQL Server 7, temporary data is stored in a database named similar to:

  ```
  DefaultDatabase..ClientHostName_TangoTraceNNNNN,
  ```

  Where `DefaultDatabase` is the default database of the ODBC DSN used to connect to SQL Server (master is used by default), `ClientHostname` is the name of the system where ThinkServer is running, and `NNNNN` is a sequential number.

  The highest number plus 1 will be assigned to the next Data Source.

  For example:

  ```
  master..MICH_TangoTrace00001
  ```

- In SQL Server 2000 and later, temporary files are stored, by default, in the WinNT\system32 folder, with a name resembling:

  ```
  ClientHostName_TangoTraceNNNNN.trc
  ```

  Where ClientHostName is the name of the system where ThinkServer is running.

For example:

```
c:\winnt\system32\tangotraces\MICH_TangoTrace00002.trc
```

Note
In SQL server 2005 the temporary files path must be set.

3.2 Deleting a Data Source

The only way to stop an auditing process on the SQL Server is to delete the data source. When you delete the data source the trace file/table will also be removed.

Important
Please make sure you remove the data source in addition to deleting all the monitors attached to it. This is very important for avoiding performance problems.

If you do not remove the data source it will continue to generate events on the SQL Server potentially using significant amounts of disk.

Chapter 4
Special Considerations

This chapter covers important information you should read before starting to use SQL Security ThinAgents. The information presented here applies to all SQL Security ThinAgents and provides the user with the required knowledge to correctly use these monitors in addition to some useful problem solving tools.

4.1 Default Configuration of ThinAgents

Accurate filters configuration is a very important aspect of SQL Server ThinAgent configuration. We will insist on this point throughout this document.

SQL Security ThinAgents can be extremely useful for detecting suspicious security situations, errors and warnings produced on SQL Server, queries run on the SQL Server, object changes, and even performance information of the execution of a query. But it can also be a detrimental to your SQL Server system. The difference lies in the filter configuration of each monitor.

Although accurate filter configuration is even more important in SQL Server 7, all database systems require an accurate configuration. It is not possible for a system with an average workload to audit every cursor, every lock, every RPC involved on a query. For example if your monitors' configuration causes 60 events to be stored for each simple query, you will need 60 servers with the same capacity as the original to support the same workload.

We have created default configurations for each ThinAgent that filter those events that indicate there may be a problem on a server and the events that indicate important changes of permissions, such as login and user additions.

4.2 Audit Filters

Filters configuration for SQL Server 7 differs from the configuration of filters for SQL Server 2000 and later. In SQL Server 7 you simple enter the values to include or exclude for each event field. Filter configuration in SQL Server 2000 is more flexible and allows for a more specific definition of the filters.

4.2.1 SQL Server 7

Filter configuration in SQL Server 7 consists of defining all the values that you want to include or exclude for a particular event field in a single entry.

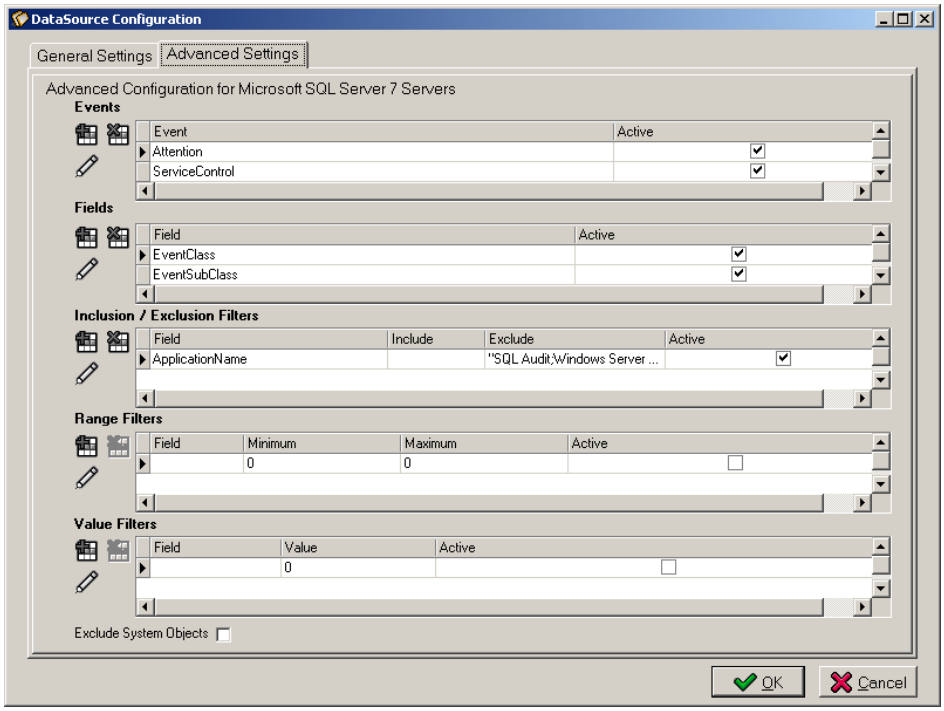

Figure 1 – Include and exclude filter configuration for SQL Server 7

In **Advanced Settings** tab click the **Add a filter** button to open the filter details window. Here you can browse the available variables using the insert button.

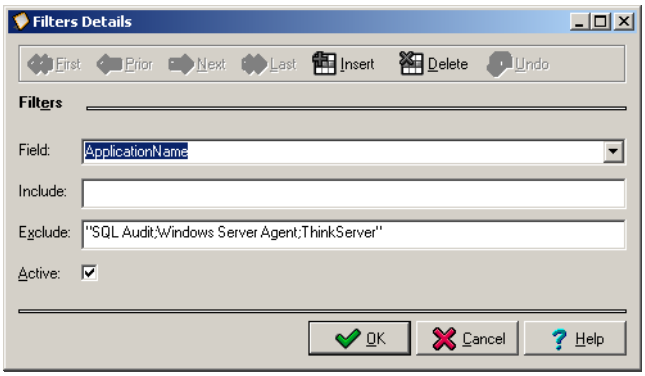

Figure 2 – Filter details

Select the desired variables to include or exclude for the event field. The variables appear as a list in the corresponding field.

4.2.2 SQL Server 2000 and Later

Filter configuration for SQL Server 2000 and later, requires one entry for each comparison of an event field.

To add a filter select the **Advanced** tab and click the **Add a filter** button.

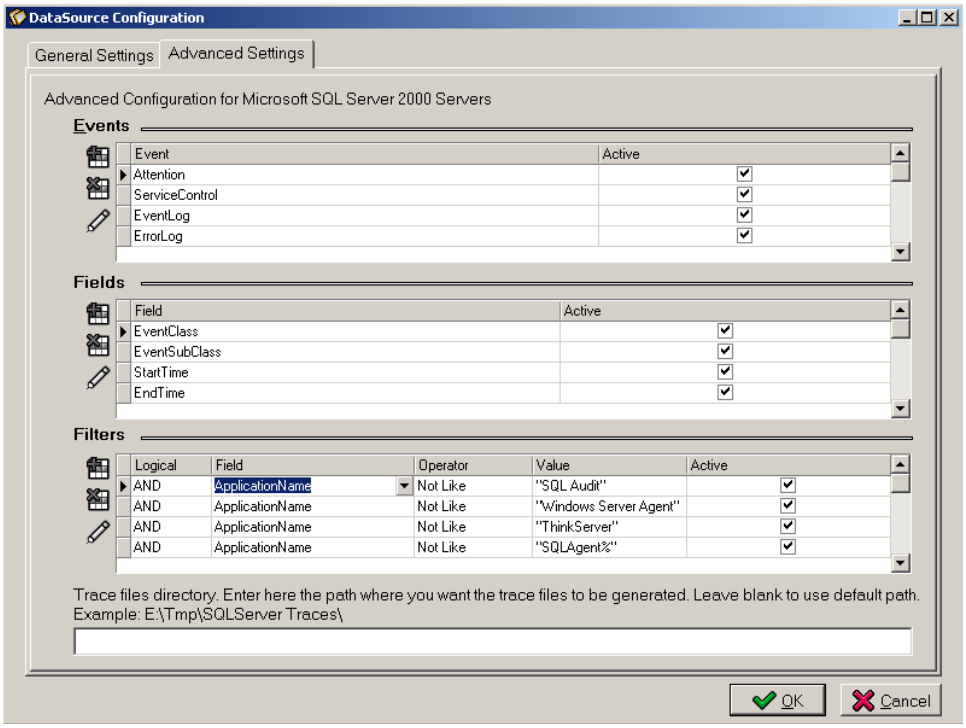

Figure 3 – Filter configuration for SQL Server 2000 and later.

The Filter Details window appears, where you can

- select the event field to which the filter applies,
- select the desired operator, and
- enter the required value.

To activate the filter remember to select Active. De-activating a filter allows you to temporarily stop a filter without losing the filter definition. To reactivate the filter simply select **Active**.

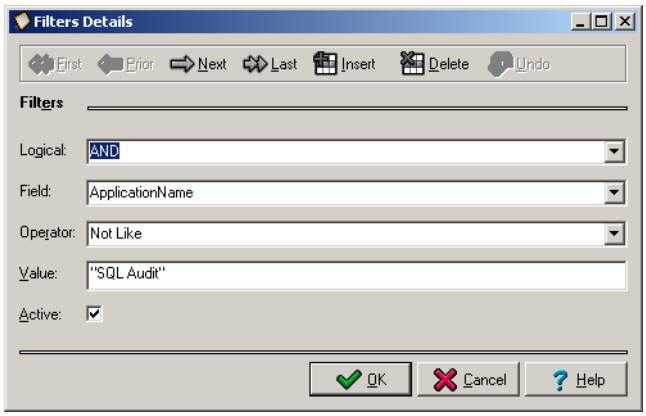

Figure 4 – Filter Details

Note
It is important to group all comparisons for a particular field.

4.3 Minimum User Profile

- In SQL Server 7, the user must be able to create a new trace and access the table that the trace creates to perform read and write operations.
- By default, Execute permissions are granted to members of the fixed server role sysadmin for:
  - `xp_trace_addnewqueue`
  - `xp_trace_setqueuedestination`
  - `xp_trace_restartqueue`
  - `xp_trace_pausequeue`
  - `xp_trace_destroyqueue`
  - `xp_trace_enumqueuehandles`
  - `xp_trace_getqueuedestination.`

  These permissions can also be granted to other users as the need arises.
- To perform select and delete queries on the table where event information will be stored you need either a user belonging to the sysadmin server role, or the user that created the trace by calling `xp_trace_setqueuedestination` extended stored procedure (i.e. the owner of the trace).
- To create a startup a stored procedure, you must be logged in as a member of the sysadmin fixed server role and create the stored procedure in the master database.
- To create the table `tango_traces`, The user must be granted `CREATE TABLE` permission. By default this permission is granted to the members of the `db_owner` and `db_ddladmin` fixed database roles. Members of the `db_owner` fixed database role and members of the sysadmin fixed server role can transfer `CREATE TABLE` permission to other users.
  To perform delete queries to table `tango_traces` `DELETE` permission must be granted. This permission is granted by default to members of the sysadmin fixed server role, the fixed database roles `db_owner` and `db_datawriter`, and the table owner. Members of the sysadmin, `db_owner`, and the `db_securityadmin` roles, in addition to the table owner, can transfer permissions to other users.
- In SQL Server 2005 and 2008, the server must permit the use of the stored procedure `xp_cmdshell`. This stored procedure is disabled by default and must be enabled using the SQL Server Surface Area Configuration see *section 4.3.2 - Enabling the xp_cmdshell Stored Procedure - SQL Server 2005* on *page 13* for further details.

  Unlike SQL Server 2000, SQL Server 2005 and 2008 do not have a strictly required `sysadmin` account which allows restriction of the given permissions to the DSN user. It is also possible to restrict the direct execution of the `xp_cmdshell` stored procedure. This results in two configuration options which are shown below. Configure these options in the Security tab by selecting the **"Avoid using xp_cmdshell directly"** parameter.

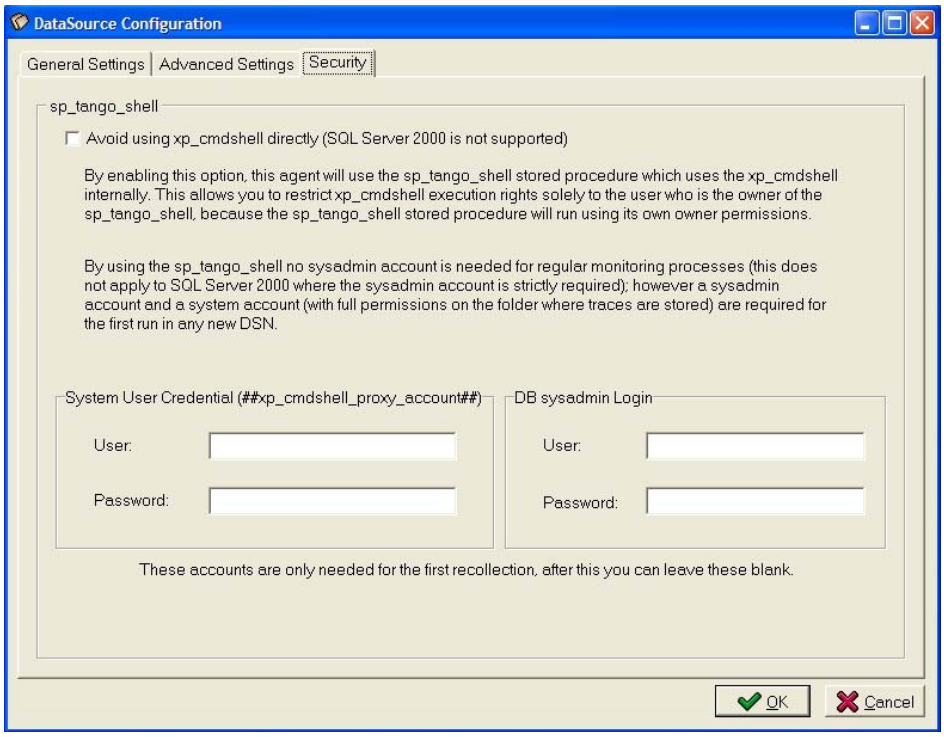

Figure 5 – DataSource Configuration

– **Option 1: Not using xp_cmdshell directly**

To enable this option, a system user (DOMAIN\USERNAME) and a DB sysadmin account are needed until the first data recollection. After the first data recollection, these accounts are no longer required and can be deleted from the data source configuration.

The system user must have *log on as a batch job* rights *(see section 4.3.1 - Grant Rights to Log On as a Batch Job)* and full control permissions on the directory where the traces are stored. This is used to configure `##xp_cmdshell_proxy_account##` which is the credential used to check permissions for accessing the `filesystem`. To reconfigure, the sysadmin must be entered and, as before, can be removed after the first recollection.

The sysadmin account is needed to create `sp_tango_shell` (which encapsulates `xp_cmdshell` and will be executed with its owner permissions), update, or create `##xp_cmdshell_proxy_account##` and grant necessary permissions to the DSN given login and user.

– **Option 2: Using xp_cmdshell directly**

To use this option, assign privileges to users or login names manually. These privileges are granted by default to members of the sysadmin fixed server role and can be granted to other users with the following commands:

```
USE [master]
GRANT ALTER ON SCHEMA :: [dbo] TO [tango_user]
GRANT SELECT ON SCHEMA :: [dbo] TO [tango_user]
GRANT INSERT ON SCHEMA :: [dbo] TO [tango_user]
GRANT UPDATE ON SCHEMA :: [dbo] TO [tango_user]
GRANT DELETE ON SCHEMA :: [dbo] TO [tango_user]
GRANT EXECUTE ON [master].[sys].[xp_cmdshell] TO [tango_user]
GRANT EXECUTE ON [master].[dbo].[sp_trace_setfilter] TO [tango_user]
GRANT EXECUTE ON [master].[dbo].[sp_trace_create] TO [tango_user]
GRANT EXECUTE ON [master].[dbo].[sp_trace_setevent] TO [tango_user]
GRANT EXECUTE ON [master].[dbo].[sp_trace_setstatus] TO [tango_user]
GRANT EXECUTE ON [master].[dbo].[xp_fileexist] TO [tango_user]
GRANT CREATE PROCEDURE TO [tango_user]
GRANT CREATE TABLE TO [tango_user]
GRANT ALTER TRACE TO [tango_login]
```

When `xp_cmdshell` is called by a user who is not a member of the sysadmin fixed server role, it connects to the operating system by using the account name and password stored in the credential named `##xp_cmdshell_proxy_account##`. If this proxy credential does not exist, `xp_cmdshell` will fail. Create the proxy account credential by executing the following command:

```
CREATE CREDENTIAL [##xp_cmdshell_proxy_account##] WITH IDENTITY =
N'DOMAIN\USER', SECRET = N'PASSWORD'
```

This user must have *log on as a batch job* rights and full control permissions on the directory where the traces are stored. For instructions, see *section 4.3.1 - Grant Rights to Log On as a Batch Job*.

SQL Audit ThinAgents also perform operations to check whether a file exists on the system and delete old temporary files where events were stored. This is done by executing advanced stored procedures `xp_cmdshell`, permission for which is granted by default to members of the `sysadmin` fixed server role, but can be granted to other users.

It is important to note that when using a Windows NT account that is not a member of the local administrator's group for the MSSQLServer service, users who are not members of the `sysadmin` fixed server role cannot execute `xp_cmdshell`.

4.3.1 Grant Rights to Log On as a Batch Job

It is necessary to add some permissions for the (new, non-fixed server role) user you assigned to the proxy account so that the user is able to log on as a batch job. This is done in the local security settings of the machine you wish to monitor.

To give the user "Log on as a batch job" privileges:

Step 1. Open the target SQL server.

Step 2. Click **Security Settings**, select **Local Policies** and click **User Rights Assignment**.

Step 3. Open **Log on as a batch job**, and add the user that you assigned to the `##xp_cmdshell_proxy_account##`. Click **Apply**, and then **OK**.

4.3.2 Enabling the xp_cmdshell Stored Procedure - SQL Server 2005

SQL Server 2005 has had several security improvements, one of which is forbidding any user to run the `xp_cmdshell` stored procedure. The SQL Server Security Agent uses this procedure; therefore it must be enabled.

Note
This is only applicable to SQL Server 2005 and later versions (not to SQL Server 2000).

To enable the xp_cmdshell stored procedure:

Step 1. On the task bar, click the **Start** menu, select **All Programs** > **Microsoft SQL Server 2005** > **Configuration Tools**, and then click **SQL Server Surface Area Configuration** to launch the Surface Area Configuration utility.

Detailed instructions are available at: `http://msdn2.microsoft.com/en-us/library/ms173748.aspx`

Step 2. Click the **Surface Area Configuration for Features** option

Further information is available at: `http://msdn2.microsoft.com/en-us/library/ms183753.aspx`

Figure 6 – Surface Area Configuration Utility- 'Surface Area Configuration for Features' option

Step 3. Select **xp_cmdshell** and click the **Enable xp_cmdshell** check box.

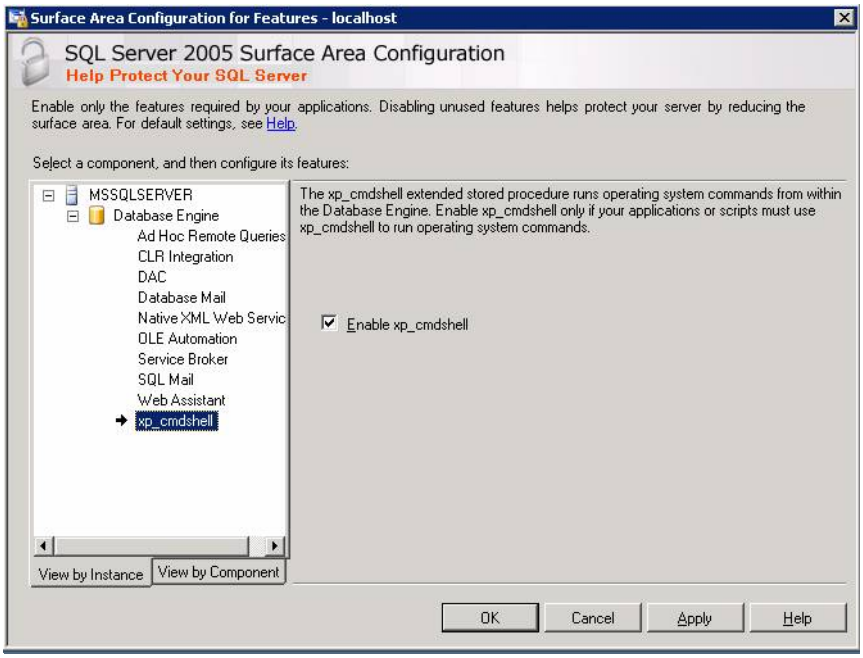

Figure 7 – SQL Server 2005 Surface Area Configuration – enable the xp_cmdshell procedure

4.3.3 Enabling the xp_cmdshell Stored Procedure - SQL Server 2008

In order to enable `xp_cmdshell` procedure in SQL Server 2008, you have to run the following sentences:

```
-- To allow advanced options to be changed.

EXEC sp_configure 'show advanced options', 1

GO

-- To update the currently configured value for advanced
options.

RECONFIGURE

GO

-- To enable the feature.

EXEC sp_configure 'xp_cmdshell', 1

GO

-- To update the currently configured value for this
feature.

RECONFIGURE

GO
```

4.4 Changing a Data Source Configuration

While a data source is busy retrieving data from an existing trace the user is allowed to make changes to the data source configuration, but these changes will not be applied untill the data source has completed retrieving all the data.

However the user can force the configuration changes to take effect immediately by stopping and starting the monitor(s) attached to the Data Source. Only once the changes have been applied will the new values be shown when editing the data source configuration.

Forcing changes can be useful in specific situations, for example if the recollection of events is continuous and a large number of events are generated. In this situation it is impossible to apply changes to the data source configuration, including any new filters configured to improve the situation, following the normal rules for updating data source configuration. In this case it is wise to force the changes to correct the situation.

4.5 Deleting Traces

There are many situations in which a trace continues to run and where it is not possible to delete it from the ThinkServer. For example when the ThinkServer is uninstalled from the client host or an error occurs when deleting the data source; the trace may continue to run while the user is left with no normal means of deleting it.

To ensure traces do not continue to run on the system unintended, two ThinAgents have been added to check which traces are running on server (Traces on SQL Server 2000 and Traces on SQL Server 7).

You should know how to delete unwanted traces and to prevent these traces from restarting the next time the SQL Server starts.

Note
The following queries should be run only in emergency situations. It is important to maintain the consistency between the data sources on the ThinkServer and the traces running on the server.

4.5.1 Deleting Traces in SQL Server 2000, 2005 and 2008

To delete traces in SQL Server 2000, 2005 or 2008:

Step 1. Retrieve the list of traces running on SQL Server by executing the query:

```
SELECT * FROM ::fn_trace_getinfo(default)
where property = 2
```

With this query you will retrieve the names and the identifiers of the traces running on the SQL Server

Step 2. Run the following queries on each trace identifier to pause and delete the traces running on the server

```
EXEC master..sp_trace_setstatus TraceIdentifier, 0

EXEC master..sp_trace_setstatus TraceIdentifier, 2
```

Step 3. This query displays the information of the traces that will be restarted when SQL Server restarts.

```
SELECT tracename, traceid FROM tango_traces
```

Step 4. Delete the information of the traces that shouldn't restart from table tango_traces

```
DELETE from tango_traces where traceid = TraceIdentifier
```

4.5.2 Deleting Traces in SQL Server 7

To delete traces in SQL Server 7:

Step 1. To retrieve the list of traces running on SQL Server execute the query:

```
EXEC master..xp_trace_enumqueuehandles
```

This query returns the list of trace identifiers that are running on server

```
EXEC master..xp_trace_getqueuedestination TraceIdentifier, 4
```

This query returns the table where a trace stores temporary data.

Step 2. To destroy the traces running on the server run the query

```
EXEC master..xp_trace_destroyqueue TraceIdentifier
```

Step 3. To list the definitions of traces on server that will be restarted the next time SQL Server starts run the query

```
EXEC master..xp_trace_enumqueuedefname 1
```

This query returns the list of trace identifiers that are running on server

Step 4. To delete the definitions of the queues so that the next time SQL Server starts these traces are not recreated execute the query:

```
EXEC master..xp_trace_deletequeuedefinition 'TraceName', 1
```

4.6 Real-Time Events

SQL Audit ThinAgents receive events to which they are subscribed in the order that the events are generated. If there are many events left on the server to be processed it could take long time for an event to reach ThinkServer and to be able to activate an alarm. This underscores the importance of configuring accurate filters in the data source as well as the monitor.

If the ThinkServer's capacity to read events is lower than the SQL Server's capacity to generate events, there will be an increasing time-lag in processing the events, which may end in the collapse of Server storage capacity.

In SQL Server 2000, where events are stored in a SQL Server memory buffer before they are stored in the files used by ThinkServer for processing, problems may arise if only very few events are generated. In this case it might take a long time for the buffer to fill, releasing the events to the ThinkServer files for processing.

When the ThinkServer detects that no data has been retrieved in 10 consecutive iterations, it forces the SQL Server buffer to write to the file, so that the ThinkServer can process the events. Note that by default this delay is only 10 minutes (10 iterations with a refresh time of 60 seconds). This delay changes according to your configuration of the refresh time in a data source.

Chapter 5
ThinAgents

All ThinAgents are based on the SQL Auditing (Generic) ThinAgent, but each ThinAgent has a purpose specific configuration built in to help you get started quickly. Of course you can change the default configurations of the ThinAgents or use the SQL Auditing (Generic) ThinAgent to configure monitors to suit your specific business needs.

The SQL Server ThinAgents currently available are:

- SQL Server Database Files Auditing
- SQL Server Errors and Warnings
- SQL Server Locks Auditing
- SQL Server Object Changes
- SQL Server Scans Auditing
- SQL Server Security Auditing
- SQL Auditing Monitor (Generic)
- SQL Server Statements Auditing
- SQL Server Stored Procedures Auditing
- SQL Server Transaction Auditing
- SQL Server User Auditing

The following two ThinAgents, based on the Data Adapter ThinAgent, are database requests to find what traces are running on the server.

- Traces on SQL Server 2000
- Traces on SQL Server 7

5.1 Events & Fields

ThinAgents retrieve their information from the data source. You can see the default fields and events available to a monitor in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events and fields you want the data source to retrieve, or add more events and fields to suit your organization's needs.

The events and fields that a data source can retrieve vary according to the ThinAgent. Each ThinAgent and the events and fields it retrieves are described individually in Chapters 10 through 15 later in this document.

In addition to the ThinAgent-specific fields, there are number of common fields that are retrieved by default for every ThinAgent. These fields are listed in the table below.

Field	Description
ApplicationName	Name of the client application that created the connection to an instance of SQL Server. This column is populated with the values passed by the application rather than the displayed name of the program.
ClientProcessID	ID assigned by the host computer to the process where the client application is running. This data column is populated if the client process ID is provided by the client.
DatabaseID	ID of the database specified by the USE database statement or the default database if no USE database statement has been issued for a given instance. SQL Profiler displays the name of the database if the Server Name data column is captured in the trace and the server is available. Determine the value for a database by using the DB_ID function.
DBUserName	SQL Server user name of the client.
EventClass	Type of event class captured.
EventSubClass	Type of event sub-class, providing further information about each event class. For example, event sub-class values for the Execution Warning event class represent the type of execution warning: 1 = Query wait. The query must wait for resources (for example, memory) before it can execute. 2 = Query time-out. The query timed out while waiting for required resources to execute. This data column is not populated for all event classes.
HostName	Name of the computer on which the client is running. This data column is populated if the client provides the host name. To determine the host name, use the HOST_NAME function.
LoginSid	Security identification number (SID) of the logged-in user. You can find this information in the `sysxlogins` table of the master database. Each SID is unique for each login in the server.
NTDomainName	Microsoft Windows NT® 4.0 or Windows 2000 domain to which the user belongs.
NTUserName	Windows NT 4.0 or Windows 2000 user name.
ServerName	Name of the instance of SQL Server being traced.
SPID	Server Process ID assigned by SQL Server to the process associated with the client.
StartTime	Time at which the event started, when available.
TextData	Text of the statement in the stored procedure.

In addition to the fields provided by SQL Server traces, we have added a set of descriptive variables that help users to understand the meaning of each event and detail the main information related to it.

Many of the variable names are not intuitive for the user. And in some cases a variable with the same name may mean something completely different depending on what event it is retrieved for.

It is very important to understand the meaning of each variable in order to create more intelligent filters to your data sources and monitors.

Field	Description
EventDescription	Short description of the SQL Server Event
EventInfo	Information about the class of the SQL Server Event
EventOrigin	Information about the user responsible (when available) for this SQL Server Event
EventPerf	Information about resources (when available) used by SQL Server related to this Event
EventText	Text information related to this SQL Server Event. It is usually the text of a TSQL query
EventTime	Time stamp of the action described by this SQL Server Event

Chapter 6
SQL Server Database Files Auditing ThinAgent (v2000 and later)

With the SQL Server Database Files Auditing monitor you can record all activity related to the automatic growth or shrinkage of data and log files.

6.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
DataFileAutoGrow	Indicates that the data file grew automatically. This event is not triggered if the data file is grown explicitly using ALTER DATABASE.
DataFileAutoShrink	Indicates that the data file has been shrunk.
LogFileAutoGrow	Indicates that the log file grew automatically. This event is not triggered if the log file is grown explicitly through ALTER DATABASE.
LogFileAutoShrink	Indicates that the log file has been shrunk.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

6.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each event. See *section 6.3 - Fields per Event* for details.

- EventClass
- StartTime
- EndTime
- Duration

- FileName

6.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server Database Files Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

Data File Auto Grow Event	
EventClass	Type of event recorded = 92.
EndTime	The time the data file auto grow ended.
Duration	The length of time (in milliseconds) necessary to extend the file.
FileName	The logical name of the file being extended.
IntegerData	The number of 8-kilobyte (KB) pages by which the file increased.

Data File Auto Shrink Event	
EventClass	Type of event recorded = 94.
EndTime	The time the auto shrink ended.
Duration	The time (in milliseconds) to shrink the file.
FileName	The logical name of the file being shrunk.
IntegerData	The number of 8 KB pages by which the file was reduced.

Log File Auto Grow Event	
EventClass	Type of event recorded = 93.
EndTime	The time at which the log file auto grow ended.
Duration	The time (in milliseconds) needed to extend the file.
FileName	The logical name of the file being extended.
IntegerData	The number of 8 KB pages by which the file increased.

Log File Auto Shrink Event	
EventClass	Type of event recorded = 95.
EndTime	The time the log file auto shrink ended.

Chapter 7
SQL Server Errors and Warnings ThinAgent

With this ThinAgent you can monitor all audit events related to the error and warning messages generated by the server. Note that some of these events indicate there is problem on SQL Server system, while others are informative only.

7.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
Attention	Occurs when attention events, such as client-interrupt requests or broken client connections, happen.
ServiceControl	Occurs when the SQL Server service state is modified.
EventLog	Indicates that events have been logged in the Microsoft Windows NT application log.
ErrorLog	Indicates that error events have been logged in the SQL Server error log.
Exception	Indicates that an exception has occurred in SQL Server.
Hash Warning	Indicates that a hashing operation (for example, hash join, hash aggregate, hash union, and hash distinct) that is not processing on a buffer partition has reverted to an alternate plan. This can occur because of recursion depth, data skew, trace flags, or bit counting.
Auto Update Stats	Indicates an automatic updating of index statistics has occurred.
OLE DB Errors	Indicates that an OLE DB error has occurred.
Execution Warnings	Indicates any warnings that occurred during the execution of a SQL Server statement or stored procedure.
Sort Warnings	Indicates sort operations that do not fit into memory. Does not include sort operations involving the creating of indexes; only sort operations within a query (such as an ORDER BY clause used in a SELECT statement).
Missing Column Statistics	Column statistics that could have been useful for the optimizer are not available.

Event	Description
Missing Join Predicate	Query that has no join predicate is being executed. This could result in a long-running query.
Server Memory Change	Microsoft SQL Server memory usage has increased or decreased by either 1 megabyte (MB) or 5 percent of the maximum server memory, whichever is greater.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

7.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 7.3 - Fields per Event* below for details.

- EventClass
- EventSubClass
- ClientHostName
- SQLSecurityLoginName
- TextData
- NTUserName
- NTDomainName
- StartTime
- EndTime
- ServerName
- DatabaseID
- ClientProcessID
- ApplicationName

7.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server User Activity Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

Attention	
EventClass	Type of event being recorded = 16

Service Control	
EventClass	Type of event being recorded = 18

ErrorLog	
EventClass	Type of event recorded = 22.
Error	Error number.
Severity	Severity of the error generated.
TextData	Text of the error message.

EventLog	
EventClass	Type of event recorded = 21.
BinaryData	Binary value dependent on the event class captured in the trace.
Error	Error number.
Severity	Error severity.
TextData	Text of the error message, if available.

Exception	
EventClass	Type of event recorded = 33.
Error	Error number.
State	Server state.
Severity	Error severity.

Hash Warning	
EventClass	Type of event recorded = 55.
EventSubClass	Type of hash operation. Can have these values: 0 = Hash recursion. 1 = Hash bail.
IntegerData	Recursion level (hash recursion only).
ObjectID	Node ID of the root of the hash involved in the repartition.

Auto Update Stats	
EventClass	Type of event being recorded = 58

OLE DB Errors	
EventClass	Type of event recorded = 61.
TextData	Error message from OLE DB.

Execution Warnings	
EventClass	Type of event recorded = 67.
EventSubClass	The type of execution warning. Can have these values: 1 = Query wait. The query must wait for resources (for example, memory) before it can execute. 2 = Query time-out. The query timed out while waiting for required resources to execute.
Error	Error number.

Sort Warnings	
EventClass	Type of event recorded = 69.
EventSubClass	Type of sort warning. Can have these values: 1 = Single pass. When the sort table was written to disk, only a single additional pass over the data to be sorted was required to obtain sorted output. 2 = Multiple pass. When the sort table was written to disk, multiple passes over the data were required to obtain sorted output.

Missing Column Statistics	
EventClass	Type of event recorded = 79.
TextData	List of the columns with missing statistics.

Missing Join Predicate	
EventClass	Type of event recorded = 80.

Server Memory Change	
EventClass	Type of event being recorded = 81

Chapter 8
SQL Server Locks Auditing ThinAgent

Use this ThinAgent to monitor locks caused by users and applications using the same database. For best results customize this ThinAgent to audit a specific database.

8.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
Lock:Deadlock	Two concurrent transactions have deadlocked each other by trying to obtain incompatible locks on resources that the other transaction owns.
Lock:Cancel	Acquisition of a lock on a resource has been cancelled (for example, due to a deadlock).
Lock:Timeout	A request for a lock on a resource, such as a page, has timed out due to another transaction holding a blocking lock on the required resource. Time-out is determined by the @@LOCK_TIMEOUT system function and can be set with the SET LOCK_TIMEOUT statement.
Lock:Deadlock Chain	This event is produced for each of the events leading up to the deadlock
Lock:Escalation	A finer-grained lock has been converted to a coarser-grained lock (for example, a row lock that is converted to a page lock).

It is possible to subscribe to the next events, though we do not recommend you do because of the large amount of dataflow they generate. Also take into account that it is normal for these events to occur for nearly every query run on the system.

Event	Description
Lock:Released	A lock on a resource, such as a page, has been released.
Lock:Acquired	Acquisition of a lock on a resource, such as a data page, has been achieved.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

8.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 8.3 - Fields per Event* below for details.

- EventClass
- ObjectID
- Mode
- Duration
- StartTime
- EndTime
- IntegerData
- BinaryData

8.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server Locks Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

Lock:Released Event	
EventClass	Type of event recorded = 23.
BinaryData	Resource type.
EndTime	End time of the event.
Duration	Wait time between the time the lock request was issued and the time the lock was released.
ObjectID	ID of the object on which the lock was released.
IndexID	ID of the index, if the object lock was on an index.

Lock:Acquired Event	
EventClass	Type of event recorded = 24.
Mode	Lock mode, such as intent exclusive, of the lock that was acquired.
BinaryData	Resource type.
EndTime	End time of the event.
Duration	Wait between the time the lock request was issued and the time the lock was acquired.
ObjectID	ID of the object on which the lock was acquired.
IndexID	ID of the index, if the object lock was on an index.

Lock:Deadlock Event	
EventClass	Type of event recorded = 25.
Mode	Lock mode of the lock that triggered the deadlock.
BinaryData	Resource type.
EndTime	End time of the deadlock.
IntegerData	Deadlock number. Numbers are assigned, beginning with zero, when the server is started and incremented for each deadlock.
Duration	Wait between the time the lock request was issued and the time the deadlock occurred.
ObjectID	ID of the object in contention.
IndexID	ID of the index, if the object lock was on an index.

Lock:Cancel Event	
EventClass	Type of event recorded = 26.
Mode	Mode of the lock that was canceled.
BinaryData	Resource type.
EndTime	End time of the event.
Duration	Wait between the time the lock requested was issued and the time the lock was canceled.
ObjectID	ID of the object on which the lock was canceled.
IndexID	ID of the index, if the object lock was on an index.

Lock:Timeout Event	
EventClass	Type of event recorded = 27.
Mode	Lock mode of the requested lock that has timed out.
BinaryData	Resource type.
EndTime	End time of the event.
Duration	Wait time between the time the lock request was issued and the time the lock was released.
ObjectID	ID of the object on which the lock was timed out.
IndexID	ID of the index, if the object lock was on an index.

Lock: Deadlock Chain Event	
EventClass	Type of event recorded = 59.
Mode	Lock mode of each lock in the deadlock chain.

Lock: Deadlock Chain Event	
BinaryData	Resource type.
IntegerData	Deadlock number. Numbers are assigned, beginning with zero, when the server is started and incremented for each deadlock.
ObjectID	ID of the object that was locked.
IndexID	ID of the index, if the object lock was on an index.

Lock: Escalation Event	
EventClass	Type of event recorded = 60.
ObjectID	ID of the object on which the lock was escalated.
IndexID	ID of the index, if the object lock was on an index.
Mode	Lock mode after the escalation.

Chapter 9
SQL Server Object Changes ThinAgent

With this ThinAgent you can monitor all changes to database objects, for example

- monitor the creation or destruction of database objects, such as index, table, or database
- monitor the execution of `SELECT`, `INSERT`, and `UPDATE` transactions, among others.

9.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
DOP Event	Occurs before a SELECT, INSERT, or UPDATE statement is executed.
Object:Created	Indicates that an object has been created, such as for CREATE INDEX, CREATE TABLE, and CREATE DATABASE statements.
Object:Deleted	Indicates that an object has been deleted, such as in DROP INDEX and DROP TABLE statements.
Audit Statement Permission	Occurs when a statement permission (such as CREATE TABLE) is used.
Audit Object Permission	Occurs when an object permission (such as SELECT) is used, either successfully or unsuccessfully.
Audit Backup/ Restore	Occurs when a BACKUP or RESTORE command is issued.
Audit DBCC	Occurs when DBCC commands are issued
Audit Object Derived Permission	Occurs when a CREATE, ALTER, and DROP object commands are issued.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

9.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 9.3 - Fields per Event* below for details.

- EventClass
- EventSubClass
- ClientHostName
- SQLSecurityLoginName
- TextData
- NTUserName
- NTDomainName
- StartTime
- EndTime
- ServerName
- DatabaseID
- ClientProcessID
- ApplicationName

9.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server User Activity Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

DOP Event	
EventClass	Type of event being recorded = 28
EventSubClass	If you are tracing a Microsoft® SQL Server™ 2000 server, the Event Sub Class can have these values, which reflect the type of statement: 1 = Select 2 = Insert 3 = Update 4 = Delete
BinaryData	Supplied binary data, which is the number of CPUs used to perform the statement.
IntegerData	Pages used in memory for the query plan.

Object:Created	
EventClass	Type of event being recorded = 46

Object:Created	
ObjectType	Type of object created.
ObjectName	Name of the object that was created
ObjectID	ID of the object that was created.
IndexID	Index ID, if an index was created

Object:Deleted	
EventClass	Type of event recorded = 47.
ObjectType	Type of object deleted
ObjectName	Name of the object that was deleted
ObjectID	ID of the object that was deleted
IndexID	Index ID, if an index was deleted

Audit Statement Permission	
EventClass	Type of event recorded = 113.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
Permissions	Type of statement issued. Values are: 1 = CREATE DATABASE (master database only) 2 = CREATE TABLE 4 = CREATE PROCEDURE 8 = CREATE VIEW 16 = CREATE RULE 32 = CREATE DEFAULT 64 = BACKUP DATABASE 128 = BACKUP LOG 512 = CREATE FUNCTION
TextData	Text value dependent on the event class captured.

Audit Object Permission	
EventClass	Type of event recorded = 114.

Audit Object Permission	
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
OwnerName	Owner name of the object for which the permissions are being checked.

Audit Backup/Restore	
EventClass	Type of event recorded = 115.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Backup 2 = Restore
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
TextData	The SQL text of the backup/restore statement.

Audit DBCC	
EventClass	Type of event recorded = 116.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
TextData	The SQL text of the DBCC command.

Audit Object Derived Permission	
Event Class	Type of event being recorded = 118.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success

Audit Object Derived Permission	
EventSubClass	Class of event within the event. Values are: 1 = Create object 2 = Alter object 3 = Drop object
DatabaseName	The name of the database in which the object is being created, altered, or dropped.
DBUserName	The issuer's user name in the database.
ObjectType	Type of object being created, altered, or dropped. Values are: 1 = Index 2 = Database 3 = User object 4 = CHECK constraint 5 = Default or DEFAULT constraint 6 = FOREIGN KEY constraint 7 = PRIMARY KEY constraint 8 = Stored procedure 9 = User-defined function (UDF) 10 = Rule 11 = Replication filter stored procedure 12 = System table 13 = Trigger 14 = Inline function 15 = Table valued UDF 16 = UNIQUE constraint 17 = User table 18 = View 19 = Extended stored procedure 20 = Ad-hoc query 21 = Prepared query 22 = Statistics
ObjectName	The name of the object that is being created, altered, or dropped.
OwnerName	The database username of the object owner of the object being created, altered, or dropped.
TextData	The SQL text of the statement.

Chapter 10
SQL Server Scans Auditing ThinAgent

SQL Server Scans Auditing Monitor allows you to audit all scans produced in a particular table or index. You can customize this ThinAgent for a specific table, or a specific query.

Note that scans are a common operation performed in database by most queries. A count statement, a select * statement, or a select conditioned by a non-indexed field cause a scan to start.

You should try to reduce the number of scans as much as possible, but in many queries they are inevitable.

10.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
Scan:Started	Table or index scan has started.
Scan:Stopped	Table or index scan has stopped.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

10.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 10.3 - Fields Per Event* below for details.

- EventClass
- StartTime
- Duration
- Mode
- EndTime
- Reads

- TransactionID
- Success
- ObjectID
- IndexID

10.3 Fields Per Event

This section details the fields that are retrieved for each event of the SQL Server Scans Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

Scan:Started Event	
EventClass	Type of event recorded = 51.
Mode	Scan mode. The following values are possible: 1 = Normal 2 = First 4 = Back 8 = Unordered 16 = No data 32 = Reserved 64 = Exlatch 128 = Index supplied 256 = Marker
ObjectID	ID of the object that is being scanned.
IndexID	ID of the index, if an index is being scanned.
TransactionID	ID of the transaction of which the scan is a part.

Scan:Stopped Event	
EventClass	Type of event recorded = 52.
Mode	Mode that was used to perform the scan. The following values are possible: 1 = Normal 2 = First 4 = Back 8 = Unordered 16 = No data 32 = Reserved 64 = Exlatch 128 = Index supplied 256 = Marker
EndTime	End time of the event.
Duration	Duration of the scan.
Reads	Number of logical pages read.

Scan:Stopped Event	
IndexID	ID of the index, if an index is being scanned.
ObjectID	ID of the object that is being scanned.

Chapter 11
SQL Server Security Auditing ThinAgent (v7)

With this ThinAgent you can monitor all security audit events related to login management, passwords, roles, permissions, and use of objects permissions.

11.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
SQL:StmtCompleted	Transact-SQL statement has completed.
SQL:StmtStarting	Transact-SQL statement has started.
SQL:BatchStarting	Transact-SQL batch has started.
SQL:BatchCompleted	Transact-SQL batch has completed.
SP:StmtCompleted	Statement within a stored procedure has completed.
SP:StmtStarting	Statement within a stored procedure has started.
SP:Starting	Stored Procedure has started
SP:Completed	Stored Procedure has completed

The following events can also be subscribed, but they add very little information and in a server with medium workload they may negatively affect the performance of the database as well as its storage capacity. We strongly recommend you do not use these if it is not necessary.

Event	Description
RPC:Starting	Occurs when an RPC has started.
RPC:Completed	Occurs when an RPC has been completed.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

11.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 11.3 - Fields Per Event* below for details.

- EventClass
- ObjectID
- HostName
- EventSubClass
- TextData
- Duration
- StartTime
- EndTime

11.3 Fields Per Event

This section details the fields that are retrieved for each event of the SQL Server Security Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

RPC:Starting Event	
EventClass	Type of event recorded = 11.
TextData	Text of the RPC.

RPC:Completed Event	
EventClass	Type of event recorded = 10.
EndTime	End time of the RPC.
Duration	Duration of the RPC.
CPU	Amount of CPU used by the RPC.
Reads	Number of page reads issued by the RPC.
Writes	Number of page writes issued by the RPC.
TextData	Text of the RPC.

SQL:StmtCompleted Event	
EventClass	Type of event recorded = 41.
Duration	The duration of the event.
EndTime	The end time of the event.

SQL:StmtCompleted Event	
Reads	The number of page reads issued by the SQL statement.
Writes	The number of page writes issued by the SQL statement.
CPU	The CPU used by the SQL statement.
IntegerData	The number of rows returned by the SQL statement.
ObjectID	The object ID of the parent stored procedure, if the SQL statement was run within a stored procedure.
NestLevel	The nest level of the stored procedure, if the SQL statement was run within a stored procedure.
TextData	The text of the statement that is about to be executed.

SQL:StmtStarting Event	
EventClass	Type of event recorded = 40.
ObjectID	The object ID of the parent stored procedure, if the SQL statement was run within a stored procedure.
NestLevel	The next level of the stored procedure, if the SQL statement was run within a stored procedure.
TextData	The text of the statement that is about to be executed.

SQL:BatchStarting Event	
EventClass	Type of event recorded = 13.
TextData	The text of the batch.

SQL:BatchCompleted Event	
EventClass	Type of event recorded = 12.
Duration	The duration of the event.
EndTime	The end time of the event.
Reads	The number of page read I/Os caused by the batch.
Writes	The number of page write I/O caused by the batch.
CPU	The CPU used during the batch.
TextData	The text of the batch.

SP:StmtCompleted Event	
EventClass	Type of event recorded = 45.
EventSubClass	Nesting level of the stored procedure.
IntegerData	Actual rows returned by the statement.

SP:StmtCompleted Event	
ObjectID	System-assigned ID of the stored procedure.
TextData	Text of the statement in the stored procedure.

SP:StmtStarting Event	
EventClass	Type of event recorded = 44.
EventSubClass	Nesting level of the stored procedure.
ObjectID	System-assigned ID of the stored procedure.
Duration	The time (in milliseconds) needed to shrink the file.
FileName	The logical name of the file being shrunk.
IntegerData	The number of 8 KB pages by which the file was reduced.

SP:Starting Event	
EventClass	Type of event recorded = 42.
NestLevel	Nesting level of the stored procedure.
ObjectID	The object ID of the stored procedure.
ObjectName	The name of the stored procedure found in the cache.
ObjectType	The type of stored procedure being started.
TextData	The text of the stored procedure call.

SP:Completed	
EventClass	Type of event recorded = 43.
NestLevel	Nesting level of the stored procedure.
EndTime	End time of the event.
Duration	Length of time the stored procedure ran.
ObjectID	Object ID of the stored procedure.
ObjectName	Name of the stored procedure found in the cache.
ObjectType	Type of stored procedure that was called.
TextData	Text of the stored procedure call.

11.4 Useful Filters

Logical	And
Field	TextData
Operator	Not like (exclude)
Value	'"-- network protocol: TCP/IP%"; "exec sp_MSadd_logreader%"; "exec sp_MSget_last_transaction%"; "exec sp_MSupdate_%"; "exec sp_replcmds %"; "exec sp_sproc_columns%"; "exec Iss_%"'

Chapter 12
SQL Server Security Auditing ThinAgent (v2000 and later)

The SQL Server Security Auditing ThinAgent allows you to monitor all security audit events related to login management, passwords, roles, permissions, use of objects permissions, and more.

12.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
SP:Starting	Stored Procedure has started
SP:Completed	Stored Procedure has completed
Audit Statement GDR	Records permission events for GRANT, DENY, REVOKE statements.
Audit Object GDR	Records permission events for GRANT, DENY, REVOKE objects.
Audit Add/Drop Login	Records add and drop actions on SQL Server logins for sp_addlogin and sp_droplogin.
Audit Login GDR	Records grant, revoke, and deny actions on Windows NT 4.0 or Windows 2000 account login rights for sp_grantlogin, sp_revokelogin, and sp_denylogin.
Audit Login Change Property	Records modifications on login property, except passwords for sp_defaultdb and sp_defaultlanguage.
Audit Login Change Password	Records SQL Server login password changes. Passwords are not recorded. If you are a member of the sysadmin or securityadmin fixed server role and you reset your own password by using sp_password with all three arguments specified ('old_password', 'new_password', 'login'), the audit record will reflect that you are changing someone else's password.
Audit Add Login to Server Role	Records the addition or removal of logins to and from a fixed server role for sp_addsrvrolemember and sp_dropsrvrolemember.

Event	Description
Audit Add DB User	Records the addition and removal of database users (Microsoft Windows NT® 4.0, Microsoft Windows® 2000, or Microsoft SQL Server™).
Audit Add Member to DB Role	Records the addition and removal of members to and from a database role (fixed or user-defined) for sp_addrolemember, sp_droprolemember, and sp_changegroup.
Audit Add/Drop Role	Records add or drop actions on database roles for sp_addrole and sp_droprole.
App Role Pass Change (Audit App Role Change Password)	Records changes to the password of an application.
Audit Statement Permission	Records the use of statement permissions.
Audit Object Permission	Records the successful or unsuccessful use of object permissions.
Audit Backup/Restore	Records BACKUP and RESTORE events.
Audit DBCC	Records DBCC commands that have been issued.
Audit Change Audit	Records AUDIT modifications.
Audit Object Derived Permission	Records when a CREATE, ALTER, or DROP command is issued for the specified object.

If these settings generate too many events, you can reduce them quickly by applying a new filter that retrieves only those events with variable Success value False (0).

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

12.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 12.3 - Fields per Event* below for details.

- EventClass
- Permissions
- Target
- RoleName CPU
- Database Username
- Duration
- EventSubClass
- ColumnPermissionsSet
- TargetUserName

- ObjectName
- TargetLoginName
- StartTime
- Success
- TextData
- Reads
- ObjectType
- TargetLoginSID
- EndTime
- Database name
- HostName
- Writes

12.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server Security Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

SP:Starting Event	
EventClass	Type of event recorded = 42.
NestLevel	Nesting level of the stored procedure.
ObjectID	The object ID of the stored procedure.
ObjectName	The name of the stored procedure found in the cache.
ObjectType	The type of stored procedure being started.
TextData	The text of the stored procedure call.

SP:Completed	
EventClass	Type of event recorded = 43.
NestLevel	Nesting level of the stored procedure.
EndTime	End time of the event.
Duration	Length of time the stored procedure ran.
ObjectID	Object ID of the stored procedure.
ObjectName	Name of the stored procedure found in the cache.
ObjectType	Type of stored procedure that was called.
TextData	Text of the stored procedure call.

Audit Statement GDR Event	
EventClass	Type of event being recorded = 102.
Success	The success or failure of the audit indicator. Values are: 0 = Failure, 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = GRANT, 2 = REVOKE, 3 = DENY
DatabaseName	Name of the database to which the GRANT/DENY/REVOKE statement permission is being applied.
DBUserName	The issuer's user name in the database.
Permissions	Type of statement issued. Values are: 1 = CREATE DATABASE (master database only), 2 = CREATE TABLE, 4 = CREATE PROCEDURE, 8 = CREATE VIEW, 16 = CREATE RULE, 32 = CREATE DEFAULT, 64 = BACKUP DATABASE, 128 = BACKUP LOG, 512 = CREATE FUNCTION
TextData	The SQL text of the GRANT/DENY/REVOKE statement.

Audit Object GDR Event	
EventClass	Type of event being recorded = 103.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Grant, 2 = Revoke, 3 = Deny
DatabaseName	Name of the database that the GRANT/DENY/REVOKE of the object permission is run in.
DBUserName	The issuer's user name in the database.
OwnerName	Name of the user who owns the object against which the GRANT/DENY/REVOKE statement is being run.
ObjectName	Name of the object to which the permissions are being applied.

Audit Object GDR Event	
Permissions	Type of statement issued. Values are: 1 = SELECT ALL, 2 = UPDATE ALL, 4 = REFERENCES ALL, 8 = INSERT, 16 = DELETE, 32 = EXECUTE (procedures only)
ColumnPermissions	Indicates whether a column permission was set. Values are: 0 = No, 1 = Yes
TextData	The SQL text of the GRANT/REVOKE/DENY statement.

Audit Add/Drop Login Event	
EventClass	Type of event being recorded = 104.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Add 2 = Drop
TargetLoginSID	Security identification number (SID) assigned to the login being added.
TargetLoginName	Name of the login being added.

Audit Login GDR Event	
EventClass	Type of event being recorded = 105.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Grant 2 = Revoke 3 = Deny
TargetLoginSID	Security identification number (SID) of the targeted Windows login.
TargetLoginName	Name of the targeted Windows login.

Audit Login Change Property Event	
EventClass	Type of event being recorded = 106.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success

Audit Login Change Property Event	
EventSubClass	Class of event within the event. Values are: 1 = Default database 2 = Default language
TargetLoginSID	Security identification number (SID) of the targeted Windows login.
TargetLoginName	Name of the targeted Windows login.

Audit Login Change Password Event	
EventClass	Type of event recorded = 107.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = User changed his or her own password 2 = User changed the password of another user
TargetLoginSID	Security identification number (SID) of the targeted Windows login.
TargetLoginName	Name of the targeted Windows login.

Audit Add Login to Server Role Event	
EventClass	Type of event recorded = 108.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Add 2 = Drop
TargetLoginSID	Security identification number (SID) of the targeted Windows login.
TargetLoginName	Name of the targeted Windows login.
RoleName	Name of the role to which the login is being added.

Audit Add DB User Event	
EventClass	Type of event recorded = 109.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = sp_adduser 2 = sp_dropuser 3 = grantdbaccess 4 = revokedbaccess

Audit Add DB User Event	
DatabaseName	Name of the database to which the user is being added.
DBUserName	The issuer's user name in the database.
TargetLoginSID	SID of the targeted Microsoft® Windows® login.
TargetLoginName	Name of the targeted Windows login.
TargetUserName	Name of the database user being added to the database.
RoleName	Name of a role to which the new database user is being added.

Audit Add Member to DB Event	
EventClass	Type of event recorded = 110.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Add 2 = Drop
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
TargetLoginSID	The SID of the targeted login.
TargetLoginName	The name of the login that is having role membership modified.
TargetUserName	Name of the user that is having role membership modified.

Audit Add/Drop Role Event	
EventClass	Type of event recorded = 111.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Add 2 = Drop
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
RoleName	Name of the role being created in the database.

App Role Pass Change – Audit App Role Change Password Event	
EventClass	Type of event recorded = 112.

App Role Pass Change – Audit App Role Change Password Event	
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
RoleName	Database application role name whose password is being changed.

Audit Statement Permission	
EventClass	Type of event recorded = 113.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
Permissions	Type of statement issued. Values are: 1 = CREATE DATABASE (master database only) 2 = CREATE TABLE 4 = CREATE PROCEDURE 8 = CREATE VIEW 16 = CREATE RULE 32 = CREATE DEFAULT 64 = BACKUP DATABASE 128 = BACKUP LOG 512 = CREATE FUNCTION
TextData	Text value dependent on the event class captured.

Audit Object Permission	
EventClass	Type of event recorded = 114.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.

Audit Object Permission	
OwnerName	Owner name of the object for which the permissions are being checked.
ObjectName	Name of the object whose permissions are being checked.
Permissions	Type of statement issued. Values are: 1 = SELECT ALL 2 = UPDATE ALL 4 = REFERENCES ALL 8 = INSERT 16 = DELETE 32 = EXECUTE (procedures only)
ColumnPermissions	Indicates whether a column permission was used. Parse the statement text to determine which permissions were applied to which columns.
TextData	Text value dependent on the event class captured.

Audit Backup/Restore	
EventClass	Type of event recorded = 115.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Backup 2 = Restore
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
TextData	The SQL text of the backup/restore statement.

Audit DBCC	
EventClass	Type of event recorded = 116.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
TextData	The SQL text of the DBCC command.

Audit Change Audit	
EventClass	Type of event recorded = 117.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = New audit started 2 = Audit stopped

Audit Object Derived Permission	
EventClass	Type of event being recorded = 118.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Create object 2 = Alter object 3 = Drop object
DatabaseName	The name of the database in which the object is being created, altered, or dropped.
DBUserName	The issuer's user name in the database.
ObjectType	Type of object being created, altered, or dropped. Values are: 1 = Index 2 = Database 3 = User object 4 = CHECK constraint 5 = Default or DEFAULT constraint 6 = FOREIGN KEY constraint 7 = PRIMARY KEY constraint 8 = Stored procedure 9 = User-defined function (UDF) 10 = Rule 11 = Replication filter stored procedure 12 = System table 13 = Trigger 14 = Inline function 15 = Table valued UDF 16 = UNIQUE constraint 17 = User table 18 = View 19 = Extended stored procedure 20 = Ad-hoc query 21 = Prepared query 22 = Statistics
ObjectName	The name of the object that is being created, altered, or dropped.

Audit Object Derived Permission	
OwnerName	The database username of the object owner of the object being created, altered, or dropped.
TextData	The SQL text of the statement.

12.4 Useful Filters

Logical	And
Field	TextData
Operator	Not like (*exclude)*
Value	'"-- network protocol: TCP/IP%"; "exec sp_MSadd_logreader%"; "exec sp_MSget_last_transaction%"; "exec sp_MSupdate_%"; "exec sp_replcmds %"; "exec sproc_columns%"; "exec Iss_%'"

Chapter 13
SQL Server Audit (Generic) ThinAgent

Visual Message Center ThinkServer comes with a number of pre-configured SQL Server ThinAgents, which are all based on the Generic SQL Server Audit ThinAgent.

The Generic SQL Server ThinAgent is able of monitor all of the variables mentioned in the other ThinAgents and more. You can use this Generic ThinAgent to create monitors for any special SQL monitoring needs you may have that are not already covered by the pre-configured SQL ThinAgents.

13.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events in SQL Server 2000:

Event	Description
Audit Change Audit	Records AUDIT modifications.

In SQL Server 7 by default retrieves the following events

Event	Description
TraceStart	Records an AUDIT start
TraceStop	Records an AUDIT stop

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

13.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 13.3 - Fields per Event* below for details.

- EventClass
- Permissions
- Target

- RoleName CPU
- Database Username
- Duration
- EventSubClass
- ColumnPermissionsSet
- TargetUserName
- ObjectName
- TargetLoginName
- StartTime
- Success
- TextData
- Reads
- ObjectType
- TargetLoginSID
- EndTime
- Database name
- HostName
- Writes

13.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server Security Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

Audit Change Audit	
EventClass	Type of event recorded = 117.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
Event Sub Class	Class of event within the event. Values are: 1 = New audit started 2 = Audit stopped

Chapter 14
SQL Server Statements Auditing ThinAgent

Use this ThinAgent to monitor the execution and completion of your application queries, in addition to the Transact-SQL queries submitted in a batch or in an individual statement

14.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
Exec Prepared SQL	Indicates when a prepared SQL statement or statements have been executed by ODBC, OLEDB, or DB-Library.
Prepare SQL	Indicates when an SQL statement or statements have been prepared for use by ODBC, OLEDB, or DB-Library.
SQL:BatchStarting	Transact-SQL batch has started.
SQL:BatchCompleted	Transact-SQL batch has completed.
SQL:StmtStarting	Transact-SQL statement has started.
SQL:StmtCompleted	Transact-SQL statement has completed.
Unprepare SQL	Indicates when a prepared SQL statement or statements have been unprepared by ODBC, OLEDB, or DB-Library.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

14.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 14.3 - Fields per Event* below for details.

- EventClass
- TextData
- Duration

- EndTime
- StartTime
- Reads
- Writes
- CPU
- Handle
- ObjectID
- NestLevel

14.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server TSQL Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

Exec Prepared SQL Event	
EventClass	Type of event recorded = 72.
Handle	Handle of the prepared TSQL statement.

Prepare SQL Event	
EventClass	Type of event recorded = 71.
Handle	Handle of the prepared TSQL statement.

SQL:BatchStarting Event	
EventClass	Type of event recorded = 13.
TextData	The text of the batch.

SQL:BatchCompleted Event	
EventClass	Type of event recorded = 12.
Duration	The duration of the event.
EndTime	The end time of the event.
Reads	The number of page read I/Os caused by the batch.
Writes	The number of page write I/O caused by the batch.
CPU	The CPU used during the batch.
TextData	The text of the batch.

SQL:StmtStarting Event	
EventClass	Type of event recorded = 40.
ObjectID	The object ID of the parent stored procedure, if the SQL statement was run within a stored procedure.
NestLevel	The next level of the stored procedure, if the SQL statement was run within a stored procedure.
TextData	The text of the statement that is about to be executed.

SQL:StmtCompleted Event	
EventClass	Type of event recorded = 41.
Duration	The duration of the event.
EndTime	The end time of the event.
Reads	The number of page reads issued by the SQL statement.
Writes	The number of page writes issued by the SQL statement.
CPU	The CPU used by the SQL statement.
IntegerData	The number of rows returned by the SQL statement.
ObjectID	The object ID of the parent stored procedure, if the SQL statement was run within a stored procedure.
NestLevel	The nest level of the stored procedure, if the SQL statement was run within a stored procedure.
TextData	The text of the statement that is about to be executed.

Unprepare SQL Event	
EventClass	Type of event recorded = 73.
Handle	The handle of the prepared TSQL statement.

Chapter 15
SQL Server Stored Procedures Auditing ThinAgent

The SQL Server Stored Procedures Auditing monitor allows you to record all stored procedures executed on an SQL Server.

15.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
SP:Starting	Stored procedure has started.
SP:Completed	Stored procedure has completed.
SP:StmtStarting	Statement within a stored procedure has started.
SP:StmtCompleted	Statement within a stored procedure has completed.
SP:CacheMiss	Stored procedure is not found in the procedure cache.
SP:CacheInsert	Item is inserted into the procedure cache.
SP:CacheRemove	Item has been removed from the procedure cache.
SP:Recompile	Stored procedure has been recompiled.
SP:CacheHit	Procedure is found in the cache.
SP:ExecContextHit	Execution version of a stored procedure has been found in the cache.

The following events can also be subscribed, but they add very little information and in a server with medium weight they can negatively affect database performance as well as its storage capacity. We strongly recommend you do not use these unless really necessary.

Event	Description
RPC:Starting	Occurs when an RPC has started.
RPC:Completed	Occurs when an RPC has been completed.

Event	Description
RPC Output Parameter	Displays information about output parameters of a previously executed remote procedure call (RPC).

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

15.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 15.3 - Fields per Event* below for details.

- EventClass
- EventSubClass
- TextData
- StartTime
- EndTime
- Duration
- Reads
- Writes
- CPU
- ObjectID
- ObjectName
- ObjectType

15.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server Stored Procedures Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

SP:Starting Event	
EventClass	Type of event recorded = 42.
NestLevel	Nesting level of the stored procedure.
ObjectID	The object ID of the stored procedure.
ObjectName	The name of the stored procedure found in the cache.
ObjectType	The type of stored procedure being started.
TextData	The text of the stored procedure call.

SP:Completed Event	
EventClass	Type of event recorded = 43.
NestLevel	Nesting level of the stored procedure.
EndTime	End time of the event.
Duration	Length of time the stored procedure ran.
ObjectID	Object ID of the stored procedure.
ObjectName	Name of the stored procedure found in the cache.
ObjectType	Type of stored procedure that was called.
TextData	Text of the stored procedure call.

SP:StmtStarting Event	
EventClass	Type of event recorded = 44.
EventSubClass	Nesting level of the stored procedure.
ObjectID	System-assigned ID of the stored procedure.
Duration	The time (in milliseconds) needed to shrink the file.
FileName	The logical name of the file being shrunk.
IntegerData	The number of 8 KB pages by which the file was reduced.

SP:StmtCompleted Event	
EventClass	Type of event recorded = 45.
EventSubClass	Nesting level of the stored procedure.
IntegerData	Actual rows returned by the statement.
ObjectID	System-assigned ID of the stored procedure.
TextData	Text of the statement in the stored procedure.

SP:CacheMiss Event	
EventClass	Type of event recorded = 34.
EventSub Class	Nesting level of the stored procedure.
ObjectName	The name of the stored procedure found in the cache.

SP:CacheInsert Event	
EventClass	Type of event recorded = 35.

SP:CacheInsert Event	
ObjectID	Object ID of the stored procedure.
ObjectName	Name of the stored procedure found in the cache.
TextData	Text of the SQL statement that is being cached.

SP:CacheRemove Event	
EventClass	Type of event recorded = 36.
ObjectID	Object ID of the stored procedure.
ObjectName	Name of the stored procedure found in the cache.
TextData	Text of the SQL statement being removed from the cache.

SP:Recompile Event	
EventClass	Type of event recorded = 37.
NestLevel	Nesting level of the stored procedure.
ObjectID	The object ID of the stored procedure.
ObjectName	The name of the stored procedure found in the cache.
TextData	The text of the stored procedure call that triggered the recompile.

SP:CacheHit Event	
EventData	Type of event recorded = 38.
ObjectID	Object ID of the stored procedure found in the cache.
ObjectName	Name of the stored procedure found in the cache.
TextData	Text of the SQL statement that was found in the cache.

SP:ExecContextHit Event	
EventClass	Type of event recorded = 39.
ObjectID	Object ID of the stored procedure.
ObjectName	The name of the stored procedure found in the cache.
TextData	The text of the stored procedure call found in the cache.

RPC:Starting Event	
EventClass	Type of event recorded = 11.
TextData	Text of the RPC.

RPC:Completed Event	
EventClass	Type of event recorded = 10.
EndTime	End time of the RPC.
Duration	Duration of the RPC.
CPU	Amount of CPU used by the RPC.
Reads	Number of page reads issued by the RPC.
Writes	Number of page writes issued by the RPC.
TextData	Text of the RPC.

RPC Output Parameter Event	
EventClass	Type of event recorded = 100.
ObjectName	Name of the output parameter from the RPC event (for example, handle).
TextData	Value of the parameter named in object name that was returned by the remote procedure call (RPC).

Chapter 16
SQL Server Transactions Auditing ThinAgent

Use this ThinAgent to monitor the status of transactions (i.e. to monitor Distributed Transactions, SQL Transactions, or the activity in the Transaction Log)

16.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
DTCTransaction	Tracks Microsoft® Distributed Transaction Coordinator (MS DTC) coordinated transactions between two or more databases.
SQL Transaction	Tracks Transact-SQL BEGIN, COMMIT, SAVE, and ROLLBACK TRANSACTION statements.
Transaction Log	Tracks when transactions are written to the transaction log

This default configuration may cause heavy load on a Server that mainly runs transactions. Furthermore it does not indicate distinguish normal from suspect situations, it simply works as a log of transactions run on server. Use it occasionally and be sure to add restrictive filters to these monitors.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

16.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 16.3 - Fields per Event* below for details.

- EventClass
- EndTime
- Binary Data
- EventSubClass
- Reads

- ObjectID
- TextData
- Writes
- IndexID
- HostName
- CPU
- Transaction ID
- Duration
- Integer Data
- Mode
- StartTime

16.3 Fields per Event

This section details the fields retrieved for each event of the SQL Server Transactions Auditing monitor. The events in this section follow the order in which they are presented in the ThinkServer configuration.

DTCTransaction Event	
EventClass	Type of event recorded = 19.
EventSubClass	Microsoft® Distributed Transaction Coordinator (MS DTC) state. For more information, see the MS DTC documentation. Possible values include: 0 = GET_DTC_ADDRESS_SUB_CLASS 1 = PROPAGATE_XACT_SUB_CLASS 2 = DOWORK_SUB_CLASS 3 = CLOSE_CONN_SUB_CLASS 4 = DTC_VIRGIN_SUB_CLASS 5 = DTC_IDLE_SUB_CLASS 6 = DTC_BEG_DIST_SUB_CLASS 7 = DTC_ENLISTING_SUB_CLASS 8 = DTC_INT_ACTIVE_SUB_CLASS 9 = DTC_INT_COMMIT_SUB_CLASS 10 = DTC_INT_ABORT_SUB_CLASS 11 = DTC_INT_ASYNC_ABORT_SUB_CLASS 12 = DTC_ACTIVE_SUB_CLASS 13 = DTC_INIT_PREPARE_SUB_CLASS 14 = DTC_PREPARING_SUB_CLASS 15 = DTC_PREPARED_SUB_CLASS 16 = DTC_ABORTING_SUB_CLASS 17 = DTC_COMMITTING_SUB_CLASS 18 = DTC_DO_ASYNC_ABORT_SUB_CLASS 19 = DTC_DISASTER_SUB_CLASS 20 = DTC_DRAIN_ABORT_SUB_CLASS 21 = DTC_ASYNC_ABORT_SUB_CLASS 22 = DTC_TM_RECOVERY_SUB_CLASS

DTCTransaction Event	
EndTime	The end time of the event.
Duration	The length of the DTC transaction.
Reads	The number of page reads generated locally by the DTC transaction.
Writes	The number of page writes generated locally by the DTC transaction.
CPU	The amount of CPU used by the DTC transaction.
IntegerData	Transaction isolation level. Possible values are: 256 = Read uncommitted 4096 = Read Committed 65536 = Repeatable read 1048576 = Serializable 4294967295 = Unspecified
BinaryData	Globally unique ID (GUID), in hexadecimal form, of the transaction, if available. For possible values of the Binary Data, see Table 2 below.

SQL Transaction Event	
EventClass	Type of event recorded = 50.
EventSubClass	Type of SQL transaction event. Possible values include: 0 = Begin Transaction 1 = Commit Transaction 2 = Rollback Transaction 3 = A Savepoint was issued
EndTime	The end time of the event. This option is only for a COMMIT or ROLLBACK.
Duration	How long the transaction ran for. This option is only for a COMMIT or ROLLBACK.
TransactionID	The internal ID number of the transaction.
TextData	The savepoint or rollback name, if provided.
ObjectName	The transaction name, if provided.

TransactionLog Event	
EventClass	Type of event recorded = 54.
EventSubClass	Type of transaction log event, such as BEGINXACT(null).
IntegerData	The length of the log record.
BinaryData	The Replication log_pubid is the publication ID that is currently being worked on. If you are using replication and look in the table for MSPublications there is a column of publication_id. This is the value represented in Binary Data. You can use this ID to find the publication and any articles associated with it.

TransactionLog Event	
EndTime	The end time of the event.
Reads	The number of read I/Os issued to perform the log entry.
Writes	The number of I/Os issued to perform the log entry.
CPU	The amount of CPU used to write the transaction entry.
TransactionID	The internal ID number of the transaction.
ObjectID	The ID of the object that has logged modifications.
IndexID	The ID of the index that has logged modifications.

Chapter 17
SQL Server User Auditing ThinAgent (v7)

With this ThinAgent you can monitor all audit events related to login management, passwords, roles, permissions, use of objects permissions, and more. This Monitor works best when you apply the filter mentioned in *section 17.4 - Useful Filters* on *page 69*.

17.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
Login Failed	Indicates that a login attempt to SQL Server from a client failed.

The following events can also be subscribed, however they do not differentiate between normal and suspect situations and, in a server with a high number of logons, they may even complicate the detection of suspect behavior of a user or an application. We strongly recommend you do not use these unless really necessary.

Event	Description
Disconnect	Collects all disconnect events, such as when a client issues a disconnect command
Connect	Collects all connection events, such as when a client requests a connection to a server running Microsoft® SQL Server™.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

17.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 17.3 - Fields per Event* below for details.

- EventClass
- EventSubClass

- ClientHostName
- SQLSecurityLoginName
- TextData
- NTUserName
- NTDomainName
- StartTime
- EndTime
- ServerName
- DatabaseID
- ClientProcessID
- ApplicationName

17.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server User Activity Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

Disconnect Event	
EventClass	Type of event being recorded = 15
EndTime	End time of the connection

Connect Event	
EventClass	Type of event being recorded = 14
StartTime	Start time of the connection

Audit Login Failed Event	
EventClass	Type of event being recorded = 20
Success	The success or failure of the audit indicator. Value will always be: 0 = Failure

17.4 Useful Filters

To get the most out of this monitor apply the following filter to the data source.

Logical	And
Field	SQLSecurityLoginName
Operator	Like (include)
Value	'"username to monitor"

Chapter 18
SQL Server User Auditing ThinAgent (v2000 and later)

With this ThinAgent you can monitor all audit events related to login management, passwords, roles, permissions, use of objects permissions. This Monitor works best when you apply the filter mentioned in *section 18.4 - Useful Filters* on *page 78*.

18.1 Retrieved Events

The default configuration of this ThinAgent retrieves the following events:

Event	Description
Audit Add/Drop Login	Records add and drop actions on SQL Server logins for sp_addlogin and sp_droplogin.
Audit Login GDR	Records grant, revoke, and deny actions on Windows NT 4.0 or Windows 2000 account login rights for sp_grantlogin, sp_revokelogin, and sp_denylogin.
Audit Login Change Property	Records modifications on login property, except passwords for sp_defaultdb and sp_defaultlanguage.
Audit Login Change Password	Records SQL Server login password changes. Passwords are not recorded. If you are a member of the sysadmin or securityadmin fixed server role and you reset your own password by using sp_password with all three arguments specified ('old_password', 'new_password', 'login'), the audit record will reflect that you are changing someone else's password.
Audit Add Login to Server Role	Records the addition or removal of logins to and from a fixed server role for sp_addsrvrolemember and sp_dropsrvrolemember.
Audit Add DB User	Records the addition and removal of database users (Microsoft Windows NT® 4.0, Microsoft Windows® 2000, or Microsoft SQL Server™).
Audit Add Member to DB	Records the addition and removal of members to and from a database role (fixed or user-defined) for sp_addrolemember, sp_droprolemember, and sp_changegroup.
Audit Add/Drop Role	Records add or drop actions on database roles for sp_addrole and sp_droprole.

Event	Description
App Role Pass Change (Audit App Role Change Password)	Records changes to the password of an application.
Audit Statement Permission	Records the use of statement permissions.
Audit Object Permission	Records the successful or unsuccessful use of object permissions.
Audit Backup/Restore	Records BACKUP and RESTORE events.
Audit DBCC	Records DBCC commands that have been issued.
Audit Change Audit	Records AUDIT modifications.
Audit Object Derived Permission	Records when a CREATE, ALTER, or DROP command is issued for the specified object.

The following events can also be subscribed, but they do not differentiate normal form suspect situations and in a server with a high number of logons can complicate the detection of suspect behavior of a user or an application. We strongly recommend you do not use these unless really necessary.

Event	Description
Audit Login	Collects all new connection events since the trace was started (for example, a client requesting a connection to a server running an instance of SQL Server).
Audit Logout	Collects all new disconnect events since the trace was started, such as when a client issues a disconnect command.
ExistingConnection	Detects activity by all users connected to Microsoft SQL Server before the trace was started.
Audit Statement GDR	Records permission events for GRANT, DENY, REVOKE statements.
Audit Object GDR	Records grant, revoke, and deny actions on Windows NT 4.0 or Windows 2000 account login rights for sp_grantlogin, sp_revokelogin, and sp_denylogin.

You will find these events in the Advanced Data Source Configuration tab of the data source configuration. Here you can select which events you want the data source to retrieve.

18.2 Fields

In the Advanced Data Source Configuration tab of the data source configuration you will find a section with fields to retrieve when updating the monitor. Here you can select which fields you want the data source to retrieve. Note that the fields available are different for each Event. See *section 18.3 - Fields per Event* below for details.

- EventClass
- Permissions
- TargetRoleName
- CPU

- Database
- Username
- Duration
- EventSubClass
- Column
- PermissionsSet
- TargetUserName
- ObjectName
- TargetLoginName
- StartTime
- Success
- TextData
- Reads
- ObjectType
- TargetLoginSID
- EndTime
- Database name
- HostName
- Writes

18.3 Fields per Event

This section details the fields that are retrieved for each event of the SQL Server User Activity Auditing monitor. The events in this section follow the order in which events are presented in the ThinkServer configuration.

Audit Statement GDR Event	
EventClass	Type of event being recorded = 102.
Success	The success or failure of the audit indicator. Values are: 0 = Failure, 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = GRANT, 2 = REVOKE, 3 = DENY
DatabaseName	Name of the database to which the GRANT/DENY/REVOKE statement permission is being applied.
DBUserName	The issuer's user name in the database.

Audit Statement GDR Event	
Permissions	Type of statement issued. Values are: 1 = CREATE DATABASE (master database only), 2 = CREATE TABLE, 4 = CREATE PROCEDURE, 8 = CREATE VIEW, 16 = CREATE RULE, 32 = CREATE DEFAULT, 64 = BACKUP DATABASE, 128 = BACKUP LOG, 512 = CREATE FUNCTION
TextData	The SQL text of the GRANT/DENY/REVOKE statement.

Audit Object GDR Event	
EventClass	Type of event being recorded = 103.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Grant, 2 = Revoke, 3 = Deny
DatabaseName	Name of the database that the GRANT/DENY/REVOKE of the object permission is run in.
DBUserName	The issuer's user name in the database.
OwnerName	Name of the user who owns the object against which the GRANT/DENY/REVOKE statement is being run.
ObjectName	Name of the object to which the permissions are being applied.
Permissions	Type of statement issued. Values are: 1 = SELECT ALL 2 = UPDATE ALL 4 = REFERENCES ALL 8 = INSERT 16 = DELETE 32 = EXECUTE (procedures only)
ColumnPermissions	Indicates whether a column permission was set. Values are: 0 = No 1 = Yes
TextData	The SQL text of the GRANT/REVOKE/DENY statement.

Audit Add/Drop Login Event	
EventClass	Type of event being recorded = 104.

Audit Add/Drop Login Event	
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Add 2 = Drop
TargetLoginSID	Security identification number (SID) assigned to the login being added.
TargetLoginName	Name of the login being added.

Audit Login GDR Event	
EventClass	Type of event being recorded = 105.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Grant 2 = Revoke 3 = Deny
TargetLoginSID	Security identification number (SID) of the targeted Windows login.
TargetLoginName	Name of the targeted Windows login.

Audit Login Change Property Event	
EventClass	Type of event being recorded = 106.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Default database 2 = Default language
TargetLoginSID	Security identification number (SID) of the targeted Windows login.
TargetLoginName	Name of the targeted Windows login.

Audit Login Change Password Event	
EventClass	Type of event recorded = 107.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success

Audit Login Change Password Event	
EventSubClass	Class of event within the event. Values are: 1 = User changed his or her own password 2 = User changed the password of another user
TargetLoginSID	Security identification number (SID) of the targeted Windows login.
TargetLoginName	Name of the targeted Windows login.

Audit Add Login to Server Role Event	
EventClass	Type of event recorded = 108.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Add 2 = Drop
TargetLoginSID	Security identification number (SID) of the targeted Windows login.
TargetLoginName	Name of the targeted Windows login.
RoleName	Name of the role to which the login is being added.

Audit Add DB User Event	
EventClass	Type of event recorded = 109.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = sp_adduser 2 = sp_dropuser 3 = grantdbaccess 4 = revokedbaccess
DatabaseName	Name of the database to which the user is being added.
DBUserName	The issuer's user name in the database.
TargetLoginSID	SID of the targeted Microsoft® Windows® login.
TargetLoginName	Name of the targeted Windows login.
TargetUserName	Name of the database user being added to the database.
RoleName	Name of a role to which the new database user is being added.

Audit Add Member to DB Event	
EventClass	Type of event recorded = 110.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Add 2 = Drop
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
TargetLoginSID	The SID of the targeted login.
TargetLoginName	The name of the login that is having role membership modified.
TargetUserName	Name of the user that is having role membership modified.

Audit Add/Drop Role Event	
EventClass	Type of event recorded = 111.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Values are: 1 = Add 2 = Drop
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
RoleName	Name of the role being created in the database.

App Role Pass Change – Audit App Role Change Password Event	
EventClass	Type of event recorded = 112.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
RoleName	Database application role name whose password is being changed.

Audit Statement Permission	
EventClass	Type of event recorded = 113.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
Database Name	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
Permissions	Type of statement issued. Values are: 1 = CREATE DATABASE (master database only) 2 = CREATE TABLE 4 = CREATE PROCEDURE 8 = CREATE VIEW 16 = CREATE RULE 32 = CREATE DEFAULT 64 = BACKUP DATABASE 128 = BACKUP LOG 512 = CREATE FUNCTION
TextData	Text value dependent on the event class captured.

Audit Object Permission	
EventClass	Type of event recorded = 114.
Success	The success or failure of the audit indicator. Values are: 0 = Failure 1 = Success
EventSubClass	Class of event within the event. Value is: Always = 1
DatabaseName	Name of the database in which the command is being run.
DBUserName	The issuer's user name in the database.
OwnerName	Owner name of the object for which the permissions are being checked.
ObjectName	Name of the object whose permissions are being checked.
Permissions	Type of statement issued. Values are: 1 = SELECT ALL 2 = UPDATE ALL 4 = REFERENCES ALL 8 = INSERT 16 = DELETE 32 = EXECUTE (procedures only)
ColumnPermissions	Indicates whether a column permission was used. Parse the statement text to determine which permissions were applied to which columns.

Audit Object Permission	
TextData	Text value dependent on the event class captured.

Audit Login	
EventClass	Type of event being recorded = 14.
Success	The success or failure of the audit indicator. Values are: 0 = Failure, 1 = Success
BinaryData	Session level settings, including ANSI nulls, ANSI padding, cursor close on commit, null concatenation, and quoted identifiers.
TextData	A delimited list of all set options.

Audit Logout	
EventClass	Type of event being recorded = 15
Success	The success or failure of the audit indicator. Values are: 0 = Failure, 1 = Success
EndTime	The end time of the log out.
Duration	The approximate amount of time since the user logged in.
Reads	The amount of logical read I/Os issued by this user during the connection.
Writes	The amount of logical write I/Os issued by this user during the connection.
CPU	The amount of CPU used by this user during the connection.

ExistingConnection	
EventClass	Type of event being recorded = 17.
BinaryData	Session level settings, including ANSI nulls, ANSI padding, cursor close on commit, null concatenation, and quoted identifiers.

18.4 Useful Filters

To get the most out of this monitor apply the following filter to the data source.

Logical	And
Field	DataBaseUserName
Operator	Like (include)
Value	'"username to monitor"

Appendix A
Field Map ThinkServer – SmartConsole

When messages arrive in VISUAL Message Center SmartConsole the variable names are not always the same as the original SQL variable name. In the following table shows the names of the variables in the applications.

SmartConsole	SQL v7	SQL v2000
SourceName	VSMMonitorName	VSMMonitorName
EventID	EventClass	EventClass
Category	VSMMonitorClass	VSMMonitorClass
ComputerName	Host	Host
User	DBUserName	DBUserName
HourGenerated	MessageTime	MessageTime
DayGenerated	MessageDate	MessageDate

Appendix B
Further Information

B.1 Using Tango/04 PDF Documentation

Tango/04 documentation is available directly from the Tango/04 solutions DVD.

To open the Tango/04 documentation that is provided in PDF files use Adobe Acrobat Reader. Acrobat Reader lets you view, search, and print the documentation. You can download Acrobat Reader for free from the Adobe Web site (`http://www.adobe.com`).

Tip
We advise printing PDF documentation for easy reference. Please ensure you familiarize yourself with a products user guide before attempting to use the product.

To access PDF documents on the DVD:

Step 1. Navigate to a *product suite* (VISUAL Message Center for example) and click on the **Product Documentation** link to open a list of all the User Guides available for that product suite. The list contains direct links to the documents in PDF format.

Step 2. Alternatively, you can navigate within the DVD menu to a particular *product* and click on the **Product Documentation** link to open the User Guide in PDF format for that product.

B.2 Tango/04 University

In a continuous effort to provide all users of Tango/04 technologies with high quality training and education, Tango/04 Computing Group presents the new training program open to partners and users worldwide.

Tango/04 University is aimed at providing Tango/04 users and partners with the most effective tools and knowledge to manage Tango/04 technologies and products and use them at their highest potential.

Attendance of the training course and passing the related exams is mandatory in order to qualify as Tango/04 Business Partner for the technology area covered by the course, and will offer you important benefits such as:

- Tango/04 Official Certifications - Tango/04 partners will be required to have a number of certified consultants, depending on the Business Partner Level

- Exploit the full potential of Tango/04 technologies - Solutions such as VISUAL Message Center and VISUAL Security Suite are very broad solutions that feature much functionality. Knowing all these functions and how to use them is key to getting the most out of the product
- Integration with other solutions - Tango/04 is constantly growing: knowing the new products and agents may allow you to integrate other parts of the IT infrastructure into Tango/04 Solutions
- Tango/04 Business Partners will learn how to effectively deploy a monitoring project in order to obtain the maximum effectiveness and customer satisfaction.

Participants' profile: Consultants, System Administrators, operators and technical staff, with knowledge of Windows, iSeries, Linux and Unix systems who will be involved in managing or deploying Tango/04 technology.

Pre-requisites: Being Tango/04 Business Partner or Tango/04 Customer.

B.3 Contacting Tango/04

North America

Tango/04 North America
PO BOX 3301
NH 03458 Peterborough
USA

Phone: 1-800-304-6872 / 603-924-7391
Fax: 858-428-2864
sales@tango04.net
www.tango04.com

Italy

Tango/04 Italy
Viale Garibaldi 51/53
13100 Vercelli
Italy

Phone: +39 0161 56922
Fax: +39 0161 259277
info@tango04.it
www.tango04.it

Sales Office in Switzerland

Tango/04 Switzerland
18, Avenue Louis Casaï
CH-1209 Genève
Switzerland

Phone: +41 (0)22 747 7866
Fax: +41 (0)22 747 7999
contact@tango04.net
www.tango04.fr

Sales Office in Peru

Barcelona/04 PERÚ
Centro Empresarial Real
Av. Víctor A. Belaúnde 147, Vía Principal 140
Edificio Real Seis, Piso 6
L 27 Lima
Perú

Phone: +51 1 211-2690
Fax: +51 1 211-2526
info@barcelona04.net
www.barcelona04.com

EMEA

Tango/04 Computing Group S.L.
Avda. Meridiana 358, 5 A-B
08027 Barcelona
Spain

Phone: +34 93 274 0051
Fax: +34 93 345 1329
info@tango04.net
www.tango04.com

Sales Office in France

Tango/04 France
La Grande Arche
Paroi Nord 15ème étage
92044 Paris La Défense
France

Phone: +33 01 40 90 34 49
Fax: +33 01 40 90 31 01
contact@tango04.net
www.tango04.fr

Latin American Headquarters

Barcelona/04 Computing Group SRL (Argentina)
Avda. Federico Lacroze 2252, Piso 6
1426 Buenos Aires Capital Federal
Argentina

Phone: +54 11 4774-0112
Fax: +54 11 4773-9163
info@barcelona04.net
www.barcelona04.com

Sales Office in Chile

Barcelona/04 Chile
Nueva de Lyon 096 Oficina 702,
Providencia
Santiago
Chile

Phone: +56 2 234-0898
Fax: +56 2 2340865
info@barcelona04.net
www.barcelona04.com

About Tango/04 Computing Group

Tango/04 Computing Group is one of the leading developers of systems management and automation software. Tango/04 software helps companies maintain the operating health of all their business processes, improve service levels, increase productivity, and reduce costs through intelligent management of their IT infrastructure.

Founded in 1991 in Barcelona, Spain, Tango/04 is an IBM Business Partner and a key member of IBM's Autonomic Computing initiative. Tango/04 has more than a thousand customers who are served by over 35 authorized Business Partners around the world.

Alliances

Partnerships

IBM Business Partner

IBM Autonomic Computing Business Partner

IBM PartnerWorld for Developers Advanced Membership

IBM ISV Advantage Agreement

IBM Early code release

IBM Direct Technical Liaison

Microsoft Developer Network

Microsoft Early Code Release

Awards

Legal Notice

The information in this document was created using certain specific equipment and environments, and it is limited in application to those specific hardware and software products and version and releases levels.

Any references in this document regarding Tango/04 Computing Group products, software or services do not mean that Tango/04 Computing Group intends to make these available in all countries in which Tango/04 Computing Group operates. Any reference to a Tango/04 Computing Group product, software, or service may be used. Any functionally equivalent product that does not infringe any of Tango/04 Computing Group's intellectual property rights may be used instead of the Tango/04 Computing Group product, software or service

Tango/04 Computing Group may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.

The information contained in this document has not been submitted to any formal Tango/04 Computing Group test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility, and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. Despite the fact that Tango/04 Computing Group could have reviewed each item for accurateness in a specific situation, there is no guarantee that the same or similar results will be obtained somewhere else. Customers attempting to adapt these techniques to their own environments do so at their own risk. Tango/04 Computing Group shall not be liable for any damages arising out of your use of the techniques depicted on this document, even if they have been advised of the possibility of such damages. This document could contain technical inaccuracies or typographical errors.

Any pointers in this publication to external web sites are provided for your convenience only and do not, in any manner, serve as an endorsement of these web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries: iSeries, iSeriese, iSeries, i5, DB2, e (logo)®Server IBM ®, Operating System/400, OS/400, i5/OS.

Microsoft, SQL Server, Windows, Windows NT, Windows XP and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries. UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group. Oracle is a registered trade mark of Oracle Corporation.

Other company, product, and service names may be trademarks or service marks of other companies.