Legal Notices

Copyright © 2013, CA. All rights reserved.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Nimsoft LLC disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Nimsoft LLC shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Nimsoft LLC and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Nimsoft LLC as governed by United States and international copyright laws.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Nimsoft LLC's standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Trademarks

Nimsoft is a trademark of CA.
Adobe®, Acrobat®, Acrobat Reader®, and Acrobat Exchange® are registered trademarks of Adobe Systems Incorporated.
Intel® and Pentium® are U.S. registered trademarks of Intel Corporation.
Java(TM) is a U.S. trademark of Sun Microsystems, Inc.
Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.
Netscape(TM) is a U.S. trademark of Netscape Communications Corporation.
Oracle® is a U.S. registered trademark of Oracle Corporation, Redwood City, California.
UNIX® is a registered trademark of the Open Group.
ITIL® is a Registered Trade Mark of the Office of Government Commerce in the United Kingdom and other countries.
All other trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

Contact Nimsoft

For your convenience, Nimsoft provides a single site where you can access information about Nimsoft products.

At http://support.nimsoft.com/, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- Nimsoft Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about Nimsoft product documentation, you can send a message to support@nimsoft.com.
Contents

Chapter 1: sql_response 1.6 ................................. 7
  sql_response Overview .................................................. 7

Chapter 2: sql_response Probe Deployment .............. 9
  Supported Platforms ...................................................... 9
  System Requirements ..................................................... 9
  Software Requirements ................................................... 9
  Monitoring System Requirements ............................ 10
  Probe Deployment Information ....................................... 10

Chapter 3: sql_response Configuration ..................... 11
  Probe Defaults ............................................................ 11
  Probe Configuration Interface Installation .................. 11
  Probe Configuration ..................................................... 11
    Setup Tab ...................................................................... 12
    Connections Tab ........................................................... 12
    Profiles Tab ................................................................. 13

Chapter 4: QoS Threshold Metrics ....................... 37
  sql_response QoS Metrics .............................................. 37
  sql_response Alert Metrics Default Settings .............. 38

Chapter 5: Troubleshooting and FAQs ................... 39
  Tips .............................................................................. 39
This description applies to sql_response probe version 1.6x.

This section contains the following topics:
sql_response Overview (see page 7)
Documentation Changes (see page 8)

sql_response Overview

The sql_response probe executes SQL queries, using ADO or ODBC connectivity, and evaluates its response time, number of returned rows, and returned value.
Documentation Changes

This table describes the version history for this document.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>What's New?</th>
</tr>
</thead>
</table>
| 1.6     | February 2013 | ■ Added Probe Defaults for the probe  
■ Added callback function to fetch active profile using wild card or regex expressions in profile_name.  
■ Resolved issue where "datetime" type variables were not properly displayed in alarm messages. |
| 1.5     | June 2011   | Added support for reading alarm tokens from configuration.                                                                                                                                                 |

Related Documentation

Documentation for other versions of the sql_response probe  
(../../../sql_response.html)

The Release Notes for the sql_response probe

Getting Started with CA Nimsoft® Probes

Monitor Metrics Reference Information for CA Nimsoft Probes
Chapter 2: sql_response Probe Deployment

This section contains the prerequisites, system requirements and deployment information for the sql_response probe.

This section contains the following topics:

- Supported Platforms (see page 9)
- System Requirements (see page 9)
- Software Requirements (see page 9)
- Monitoring System Requirements (see page 10)
- Probe Deployment Information (see page 10)

Supported Platforms

The sql_response probe supports the same set of operating systems and databases as supported by the Nimsoft Server solution. Please refer to the Nimsoft Compatibility Support Matrix for the latest information on supported platforms.

System Requirements

The sql_response probe should be installed on systems with the following minimum resources:

- Memory: 2-4 GB of RAM. This probe OOTB configuration requires 256 MB of RAM.
- CPU: 3 GHz dual-core processor, 32-bit or 64-bit

Software Requirements

The sql_response probe requires the following software environment:

- Nimsoft Monitor Server 5.1.1 or later
- Nimsoft robot version 5.23 or later
- Java Virtual Machine version 1.6 or later (deployed as part of the probe package)

Note: For SOC functionality, NM Server 5.6 or later and UMP 2.5.2 or later is required.
Monitoring System Requirements

The sql_response probe allows the user to monitor response of user defined queries for Oracle, SQL Server, Informix, DB2 and Sybase databases.

Probe Deployment Information

There are two ways to distribute archive packages. You can distribute the package within Infrastructure Manager or use the standalone Nimsoft Distribution application.

See Probe Deployment for more information on deploying probes.
Chapter 3: sql_response Configuration

You can change the configuration for sql_response probe for changing the default settings for the probe, defining new connections and adding new profiles or editing the existing profiles.

The sql_response probe is configured by double-clicking the probe in the Infrastructure Manager. This brings up the configuration tool for the probe.

This section contains the following topics:

- **Probe Defaults** (see page 11)
- **Probe Configuration Interface Installation** (see page 11)
- **Probe Configuration** (see page 12)

## Probe Defaults

At the time of deploying a probe for the first time on robot, some default configuration will be deployed. For this you have to drag and drop the test template to the profile. These probe defaults could be Alarms, QoS, Profiles and so on which save time to configure the default settings. These probe defaults will be seen on a fresh install, that is no instance of that probe is already available on that robot in activated or deactivated state.

## Probe Configuration Interface Installation

The probe configuration interface is automatically downloaded and installed by the Nimsoft Infrastructure Manager when the probe is deployed on a robot.
探头配置

设置选项卡

设置选项卡下的控件配置探头的各个部分。

- **日志**
  - **日志文件**
    - 识别探头日志中关于其内部活动的信息的文件。
  - **日志大小（KB）**
    - 表示日志文件的最大大小。
  - **日志级别**
    - 设置写入日志文件的详细程度。在正常操作期间尽可能少写日志以减少磁盘消耗，并在调试时增加详细程度。

- **默认值**
  - **间隔**
    - 定义探头应运行定义的配置文件的频率。
  - **CCM错误**
    - 定义错误级别。

此选项卡包含以下字段：

**日志**

- **日志文件**
  - 识别探头日志中关于其内部活动的信息的文件。

- **日志大小（KB）**
  - 表示日志文件的最大大小。

- **日志级别**
  - 设置写入日志文件的详细程度。在正常操作期间尽可能少写日志以减少磁盘消耗，并在调试时增加详细程度。

**默认值**

- **间隔**
  - 定义探头应运行定义的配置文件的频率。注意，此值可能被单个运行间隔（Run Interval）参数覆盖，该参数可单独为每个配置文件定义。

- **CCM错误**
  - 定义错误级别。
Connection error

Specifies the severity level for the alarms issued when communication errors occur.

Connections Tab

The Connections tab lists all database connections that can be used by all the profiles. Initially this list only contains a sample connection (sample_conn).

<table>
<thead>
<tr>
<th>Setup</th>
<th>Connections</th>
<th>Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Provider</td>
<td>Data Source</td>
</tr>
<tr>
<td>sample_conn</td>
<td>SQLLOEDB:1</td>
<td>sample_server</td>
</tr>
</tbody>
</table>

Add New Connection

This functionality allows you to add a new connection.

Follow these steps:
1. Right-click and select New under the Connections tab.

The Add New Connection dialog appears.
2. Enter the name of the new connection in the **Name** box and click **OK**.

The **New Connection** dialog appears.

![New Connection dialog](image)

This dialog contains the following fields:

**Provider / DSN**

- If **OLEDB**: Database provider.
- If **ODBC**: DSN. See below how to define a DSN.

**Authorization**

- If checked, Windows domain authorization scheme will be used.
- If unchecked, SQL Server authorization scheme will be used.
**Initial Catalog**

- **DB2**: empty.
- **Informix**: empty.
- **Oracle**: empty.
- **SQL Server**: Database name.
- **Sybase**: Database name.

**Data Source**

- **DB2**: database name.
- **Informix**: database@server.
- **Oracle**: NET Service name or empty.
- **SQL Server**: Database server.
- **Sybase**: Data Source name.

**User ID**

Defines the Database user.

**Password**

Defines the database user’s password.

**Timeout**

Specifies the connection timeout. In case the probe cannot establish database connection in the given time, a connection alarm is issued.

**Subsystem**

Defines the alarm subsystem. All profiles using one connection will use this ID as their alarm subsystem.

**Parameters**

Lets you add additional parameters for the connection. This is for advanced use only. Leave the field blank unless you have detailed knowledge and know how to use these parameters.

**Test button**

Allows you to test the defined connection.

**Connection Error**

Indicates the alarm message (variables may be used) to be issued if the connection to the database cannot be established. You can also select the severity of the alarm message.
Connection Established

Specifies the clear message (variables may be used) to be issued when the connection to the database is re-established after a connection error. You can also select the severity of the clear message.

Similarly you can also perform the edit and delete functions in the Connections tab.

Define an ODBC Connection

To be able to select a DSN from the drop-down menu when defining a new connection, you need to create ODBC connections.

Follow these steps:

1. Open the Control Panel on your computer and select Administrative Tools.
2. Select Data Sources (ODBC).

The ODBC Data Source Administrator dialog appears.

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.
3. Select the **System DSN** tab and click the **Add** button. A wizard (Create New Data source) is launched.

4. Select the driver you want to use (e.g. SQL server) and click the **Next** button.

5. Follow the steps through the wizard to finish the definition. When finished, the new definition will appear in the drop-down menu on the probe when you define a new ADO connection, selecting ODBC.

**Set up an Oracle ADO Connection**

You can create an ADO connection using the **Edit Connection** dialog.
Follow these steps:

1. Click **OLEDB**.
2. Select the Oracle provider you want to use from the **Provider** drop-down.
3. Leave the **Initial Catalog** field blank.
4. Enter the TNS you want to connect to in the **Data Source** field.
5. Enter the **User ID** and **Password** for database connection.
6. Leave the **Parameters** field blank unless you have detailed knowledge and know how to use these parameters.
7. Click **Test** to test the connection created.

### Profiles Tab

The **Profiles** tab lists all the profiles. Initially this list contains only a sample profile (sample_prof).

<table>
<thead>
<tr>
<th>Setup</th>
<th>Connections</th>
<th>Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Description</td>
<td>Connection</td>
</tr>
<tr>
<td>sample_prof</td>
<td>sample profile</td>
<td>sample_conn</td>
</tr>
</tbody>
</table>
Add New Profile

This functionality allows you to add a new profile. A profile is a definition of one specific sql_response task.

Follow these steps:

1. Right-click and select New under the Profiles tab.

   The Add New Profile dialog appears.

2. Enter the name of the new profile in the Name box and click OK.

   The Edit Profile dialog appears. This dialog contains General, SQL Query, Response time, Row count and Value tabs.

   Similarly you can also perform the copy, edit, run now and delete functions in the Profiles tab.
**Edit Profile**

Click a profile appearing in the **Profiles** tab.

The **Edit Profile** dialog appears that contains the following tabs:

- **General tab** (see page 21)
- **SQL Query tab** (see page 25)
- **Response time tab** (see page 27)
- **Row Count tab** (see page 30)
- **Value tab** (see page 32)
General Tab

The **General** tab allows you to set the general properties for a profile.

![General Tab Image]

This tab contains the following fields:

- **Name**
  
  Indicates the name you have given to the new profile.

- **Description**
  
  Provides a short description of the profile.

- **Connection**
  
  Specifies one of the connections from the connection pool.

- **Alarm source**
  
  Indicates an alternative alarm source (default is the robot name).
Run Interval

Specifies how often the SQL query and tests should run. This value overrides the default value set on the Setup tab.

Query Timeout

Specifies the time the probe should wait for the query to finish. If this timeout is exceeded, the probe cancels the whole process (since it seems to hang). The probe then needs to be restarted.

Timeout error

Specifies the severity of the alarm message issued when a query timeout occurs.

Scheduling

Lets you select how to use the Schedules settings, if any.

rules

Selecting rules means to run according to the rules described in the Schedules settings.

exceptions

Selecting exceptions means to run except the rules described in the Schedules settings

Schedules

Lists the schedules defined for the profile. Right-clicking in the list, you can add, edit, copy or delete schedules.
Schedules

Right-click the **Schedules** field at the bottom of the **Profile** tab to open the **Schedules** dialog. It enables you to define a schedule.
If the schedules list is empty, the profile will be executed in interval matter, 24 hours a day. In addition, there can be defined number of schedules per profile, which can define additional rules to the check interval or exceptions of it. The rules and exceptions cannot be mixed in one profile.

In principle, a Schedule is a definition of an execution period (or execution break if exceptions used) with specified days, time from/to and date from/to values. Additionally, if only Date from and Time from is defined, the first execution can be defined. Clicking Run once will cause the checkpoint to run only once a day in the defined period (unlike multiple times if Run interval is used).
The **SQL Query** tab allows you to send SQL queries and check the output of your query. You can also specify the alarm message and the severity of the alarm message if no record is returned after the query.

This tab contains the following fields:

**Cursor**

- **Server**
  
  Creates the cursor in the database server. This setting delivers realistic data about the database performance.

  **Disadvantage:** Reduces performance when the net-traffic increases.
Client

Transfers the record set to the client machine by ADO and manage the cursor there.

**Advantage:** Better performance.

**Disadvantage:** The results are buffered, depend on ADO provider and can hide real database performance.

Simple Query

Used for simple one-line queries. Click the **Test** button to run the query defined.

From File

Used to store multi-line queries in a file on the location you specify. This way a query can be shared between different profiles. It also makes it possible to create queries with other tools.

**Note:** If the From File field is filled, the probe will always read the query from the file, regardless of the content in Simple Query.

Test button

Checks the output of your query.

No Record Returned

Specifies the alarm message (variables may be used) to be issued if no record was returned after the query. You can also select the severity of the alarm message.

Suppression key

Enables to clear the alarm if the query returns data the next time. Alarms may be issued in case the query doesn't return any data. Select the parameter to be checked in the **Suppression key** drop-down box.

Default parameter to check is the **value** returned (more rarely used is response or count). You can also decide not to have the alarm cleared by using other.

Severity

Selects the severity of the alarm message to be issued.

Send QoS as

In case the query returns no row, depending on the option selected from the drop-down box, the probe will issue either a QoS null value, zero value, or no value at all into the QoS database.

Message

Specifies the alarm message to be returned as a regular expression.
Response Time Tab

The **Response time** tab measures how long (in milliseconds) it took to run the SQL query.

This tab contains the following fields:

**Alarm**

Sets the alarm message to be issued if the threshold values defined below are exceeded.

**QoS**

Enables the **Response time** QoS messages to be issued.
Network

inclusive
Indicates that the network delay is included in the response time.

exclusive
Indicates that the probe removes the network delay from the response time.
The network delay is calculated by sending one or more pings.

Pings
Defines the number of pings to be used to find the reasonable network approximation (valid only if selecting exclusive as described above).

Checked Value
Specifies what the checked value comprises:

total
Includes all sql phases from connect to close connection.

connect
Includes create_connection instance + open connection.

prepare
Includes create rs_instance, open record set.

recordset
Includes prepare and all fetches.

fetch
Includes the time per one fetch (average over all records).

High Threshold
Specifies the high threshold value in ms. An alarm message with the specified alarm message text (supports variable expansion) and the selected severity level will be issued if this value is exceeded and the Alarm option (above) is selected.

Note: If the severity level Inactive is selected, no checking will be done. This also applies if the Value field is left empty.

Message variables that can be used are:

$description, $profile, $server, $database, $time (value depends on the parameter selected for Checked Value) $threshold, $total, $connect, $prepare, $fetch, $rs_close (recordset close), $cn_close (connection close), $ping.
Low Threshold

Specifies the low threshold value in ms.

An alarm message with the specified alarm message text (supports variable expansion) and the selected severity level will be issued if this value is exceeded and the Alarm option (above) is checked.

Note: If the severity level Inactive is selected, no checking will be done. This also applies if the Value field is left empty.

Message variables that can be used are:

$description, $profile, $server, $database, $time (value depends on the parameter selected for Checked Value) $threshold, $total, $connect, $prepare, $fetch, $rs_close (recordset close), $cn_close (connection close), $ping.

Clear

Issues the defined Clear message if the measured value does not breach the High or Low thresholds and alarm messages on breached thresholds have previously been issued.

Note: If the severity level Inactive is selected, no checking will be done.

Message variables that can be used are:

$description, $profile, $server, $database, $time (value depends on the parameter selected for Checked Value) $threshold, $total, $connect, $prepare, $fetch, $rs_close (recordset close), $cn_close (connection close), $ping.
Row Count Tab

The **Row count** tab measures the number of rows returned by SQL query.

This tab contains the following fields:

**Alarm**

Sets an alarm message to be issued if the threshold values defined below are exceeded.

**QoS**

Enables the **Row count** QoS messages to be issued.

**Operator**

Used to compare the value found against the thresholds defined.
High Threshold

Defines the high threshold value. An alarm message with the specified alarm message text (supports variable expansion) and the selected severity level will be issued if this value is exceeded and the Alarm option (above) is selected.

Note: If the severity level Inactive is selected, no checking will be done. This also applies if the Value field is left empty.

Message variables that can be used are:
$description, $profile, $server, $database, $rows.

Low Threshold

Defines the low threshold value. An alarm message with the specified alarm message text (supports variable expansion) and the selected severity level will be issued if this value is exceeded and the Alarm option (above) is selected.

Note: If the severity level Inactive is selected, no checking will be done. This also applies if the Value field is left empty.

Message variables that can be used are:
$description, $profile, $server, $database, $rows.

Clear

Issues the defined Clear message if the measured value does not breach the High or Low thresholds and alarm messages on breached thresholds have previously been issued.

Note: If the severity level Inactive is selected, no checking will be done.

Message variables that can be used are:
$description, $profile, $server, $database, $rows.
Value Tab

The **Value** tab sends the value of a selected column (must be a numeric value) returned by the SQL query.

This tab contains the following fields:

**Alarm**

Sets an alarm message to be issued if the threshold values defined below are exceeded.

**QoS**

Enables you to issue **Value** QoS messages.

**QoS details**

Opens the **QoS list** with/for QoS definitions.
Row key
Defines one or more columns which will compose an unique identifier for every row in case the query returns more than one row. In this manner, you can distinguish alarm and QoS messages issued by the profile for every row.

Column
Specifies the column from which you can read the value.
To populate the list, you have to run "Test query"!

Operator
Used to compare the value found against the thresholds defined.

Comparison
Specifies the type of comparison to evaluate query results. The available values are numeric, character and regular expression.

NULLs
Defines how to handle NULL values returned from the query.

as_zero
If the query returns a NULL value, it will be replaced by 0 (zero).

ignore
If the query returns a NULL value, no checking occurs.

alarm
If the query returns a NULL value, the probe issues an alarm (without further checking).

High Threshold
Defines the high threshold value. An alarm message with the specified alarm message text (supports variable expansion) and the selected severity level will be issued if this value is exceeded and the Alarm option (above) is checked.

Note: If the severity level Inactive is selected, no checking will be done. This also applies if the Value field is left empty.

Message variables that can be used are:
$description, $profile, $server, $database, $value.
Low Threshold

Defines the low threshold value. An alarm message with the specified alarm message text (supports variable expansion) and the selected severity level will be issued if this value is exceeded and the Alarm option (above) is checked.

**Note:** If the severity level Inactive is selected, no checking will be done. This also applies if the Value field is left empty.

Message variables that can be used are:
$description, $profile, $server, $database, $value.

Clear

Issues the defined Clear message if the measured value does not breach the High or Low thresholds and alarm messages on breached thresholds have previously been issued.

**Note:** If the severity level Inactive is selected, no alarm checking will be done.

Message variables that can be used are:
$description, $profile, $server, $database, $value.

QoS List

To open the QoS list with/for QoS definitions, click the QoS details icon.
The **QoS list** displays all defined QoS definitions for the profile. You can delete old QoS definitions or create new ones from this list.

![QoS List](image)

See the properties dialog for the QoS definitions in the section **Edit QoS**.

### Edit QoS Dialog

QoS data based on query results have to be defined using this panel.

![Edit QoS Dialog](image)

This dialog contains the following fields:

**Name**

Defines the name used for the QoS database. Cannot be changed.

**Description**

Provides the description used for QoS database. Cannot be changed.
**Probe Configuration**

Unit
- Defines the long name for QoS data unit.

Abbreviation
- Defines the short name for QoS data unit.

Metric
- Specifies the column name, used as QoS value. After the query has been executed, the drop-down list is filled with suitable candidates.

Max value
- Defines maximum value the metric can have (if available). Default is "0", means no max value.

Object
- (Optional) Defines the name to describe the value in the QoS database.
Chapter 4: QoS Threshold Metrics

Many Nimsoft Monitor probes ship with default QoS threshold values set. The default threshold values provide an idea of the type of values to be entered in the fields and are not necessarily recommended best practice values. To aid in tuning thresholds and reducing false-positive alarms, this section describes the QoS metrics and provides the default QoS thresholds.

This section contains the following topics:

- sql_response QoS Metrics (see page 37)
- sql_response Alert Metrics Default Settings (see page 38)

sql_response QoS Metrics

The following table describes the checkpoint metrics that can be configured using the sql_response probe.

<table>
<thead>
<tr>
<th>Monitor Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QOS_SQL_RESPONSE</td>
<td>Milliseconds</td>
<td>SQL Query Response</td>
</tr>
<tr>
<td>QOS_SQL_ROWS</td>
<td>Rows</td>
<td>SQL Query Rows</td>
</tr>
<tr>
<td>QOS_SQL_VALUE</td>
<td>User defined</td>
<td>SQL Query Value</td>
</tr>
</tbody>
</table>
## sql_response Alert Metrics Default Settings

This section contains the QoS metric default settings for the sql_response probe.

<table>
<thead>
<tr>
<th>QoS Metric</th>
<th>Warning Threshold</th>
<th>Warning Severity</th>
<th>Error Threshold</th>
<th>Error Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response time</td>
<td>90</td>
<td>Warning</td>
<td>150</td>
<td>Minor</td>
<td>Monitors the alarm value that will be sent if the response time of the profile exceeds the Value setting in milliseconds</td>
</tr>
<tr>
<td>Row count</td>
<td>5</td>
<td>Warning</td>
<td>10</td>
<td>Minor</td>
<td>Monitors the alarm value that will be sent if the number of rows of the profile exceeds the Condition and Value settings</td>
</tr>
</tbody>
</table>
Chapter 5: Troubleshooting and FAQs

This section contains the following topics:

Tips (see page 39)

Tips

There are some rules that you should follow when you create a query:

1. To avoid excessive alarms, try to limit the number of rows that is returned by the query. If possible, use the following select:

   \[
   \text{SELECT a, b FROM table1 WHERE somedate < DATEADD(n,\text{-}10,\text{GETDATE}())}
   \]

2. Use queries that return one row if you can. For example, \text{SELECT count(*) as rows FROM table1.}

3. Remember that each row returned by a query results in one alarm and one or more QoS messages.