

# The Sentry-go Monitoring System Configuring Logging Options

Last Updated Wednesday, 04 November 2009

© 3Ds (UK) Limited  
<http://www.Sentry-go.com>

*Be Proactive, Not Reactive!*

---

## Table of Contents

- Symbols .....2
- Background.....2
- Types of Logging .....2
- Configuring Sentry-go Logging Options.....3
- Logging Alerts to a File .....3
- Setting up the Local File .....4
- Logging Alerts to a Database .....5
- Setting up the Database .....7
- Logging Performance Information .....8
- More Information, Help & Support.....9

---

## Symbols

Thank you for choosing Sentry-go® as your monitoring solution for Windows. In this guide, the following symbols are used to denote specific items ...



Important information which should be noted – it may affect what you are trying to do.



Additional information relating to the operation being described is shown.

---

## Background

This guide gives full details of how you can configure the various logging options within Sentry-go.

---

## Types of Logging

With Sentry-go there are a number of different types of logging available. Some allow you to capture alert details centrally while others are typically used for internal use.

### The Sentry-go Log File

From time to time Sentry-go may need to write out information, either due to an error or to record a specific event such as the service being started or reconfigured. To do this, errors & diagnostic information are written to the a central log file, called "Sentry-go Monitoring System.Log" and located in the directory in which the monitor was installed. By default, this will be located in a path similar to "C:\Program Files\Sentry-go Monitoring System\Sentry-go Monitoring System.Log".

- All Sentry-go components use this log file to record errors, configuration issues and general information.
- To prevent this file becoming excessively large, the size of this file is automatically monitored by the logging system and the oldest records removed periodically when required.



Logging to the Sentry-go log file is an automatic process & does not require further configuration.

### Logging alerts to a text file

When Sentry-go triggers an alert, you can record details to a text-based or delimited file – e.g. a CSV file. This can then be used for further processing, statistical analysis or later reporting.

### Logging alerts to an ODBC database

In addition to a delimited text file, you can also record alert information in a database such as Microsoft SQL Server, Access or Oracle. Again, this may then be used for further processing, statistical analysis or reporting etc.

## Logging performance data to a file

If you are monitoring system or software performance (using performance counters), data can also be recorded to a file for further trend analysis through the Performance Optimiser.

---

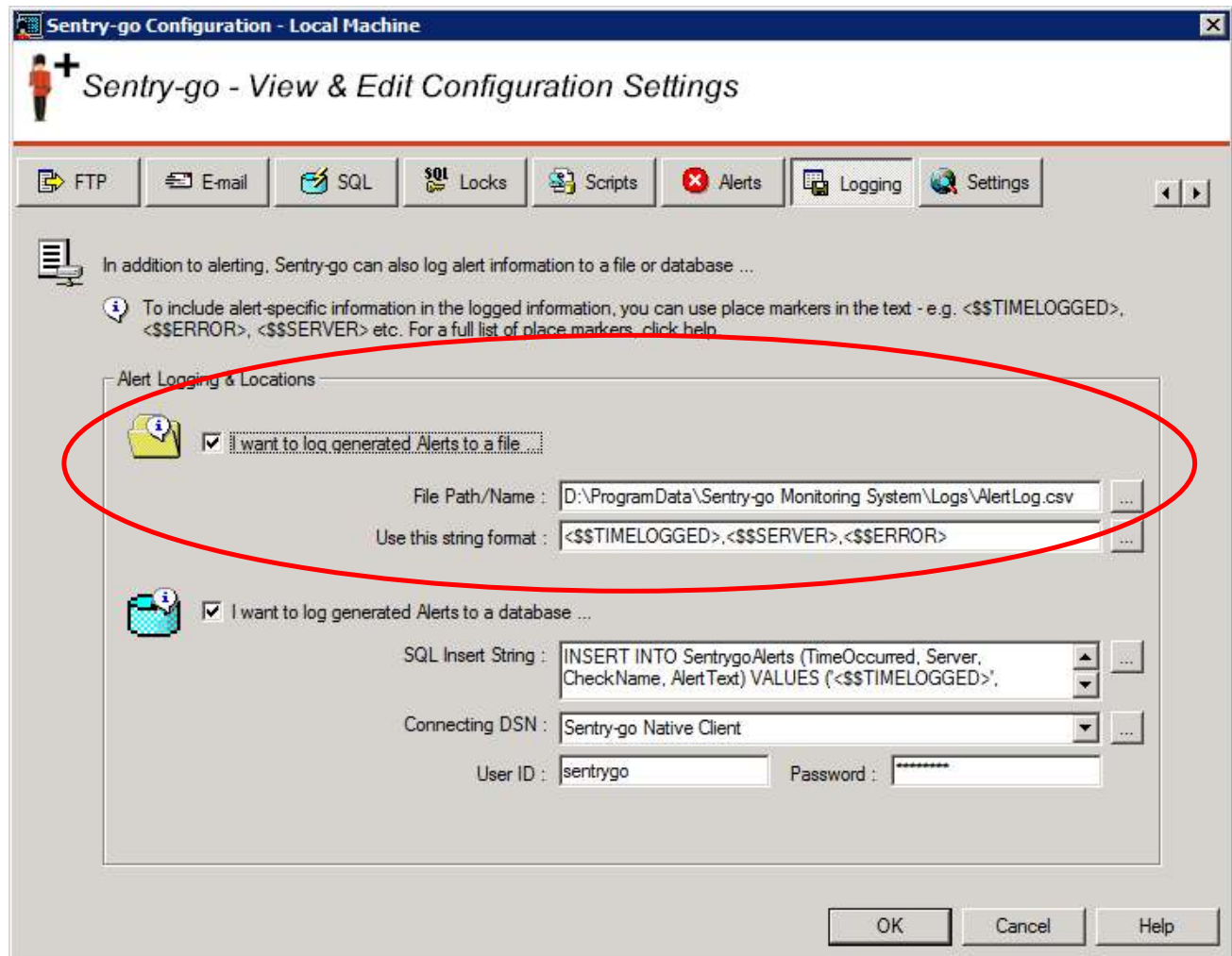
## Configuring Sentry-go Logging Options

To configure Sentry-go logging options, simply select the Sentry-go monitor from the Client Console with the right mouse button and click "Configure". A window containing a number of tabs will be displayed. Now select the "Logging" tab to display & configure logging options.

---

## Logging Alerts to a File

From the logging tab, select the top set of options to define the logging of alerts to a text or delimited file ...



## Log Alerts to a File

Check (tick) this option if you wish to record alert information to a text or delimited file.

### File Record String

This is the mask for the record that will be written to the file. It can be made up of one or more place-markers which will be substituted before the record is written. These [place-markers](#) are the same as used by the Alert Engine.


For example, for comma-separated records ...

```
<$$TIMELOGGED>, <$$ALERTLEVEL>, <$$SERVER>, <$$TEST>, <$$ERROR>
```

For help in defining this message text, click the appropriate “...” button to launch the message builder. See [“Sentry-go– Place-markers”](#) for more information.


### File Path/Name

This field is used to specify the path/name of the log file you wish to create. To choose a name/path, click the “...” button to the right of the field.

-  Note that the path selected/entered must be relative to the local server. If configuring a remote machine, navigate to that machine and select a UNC path - e.g. \\SERVER\Path, or ensure the path entered is a local name that is valid for the remote server being configured.

---

## Setting up the Local File

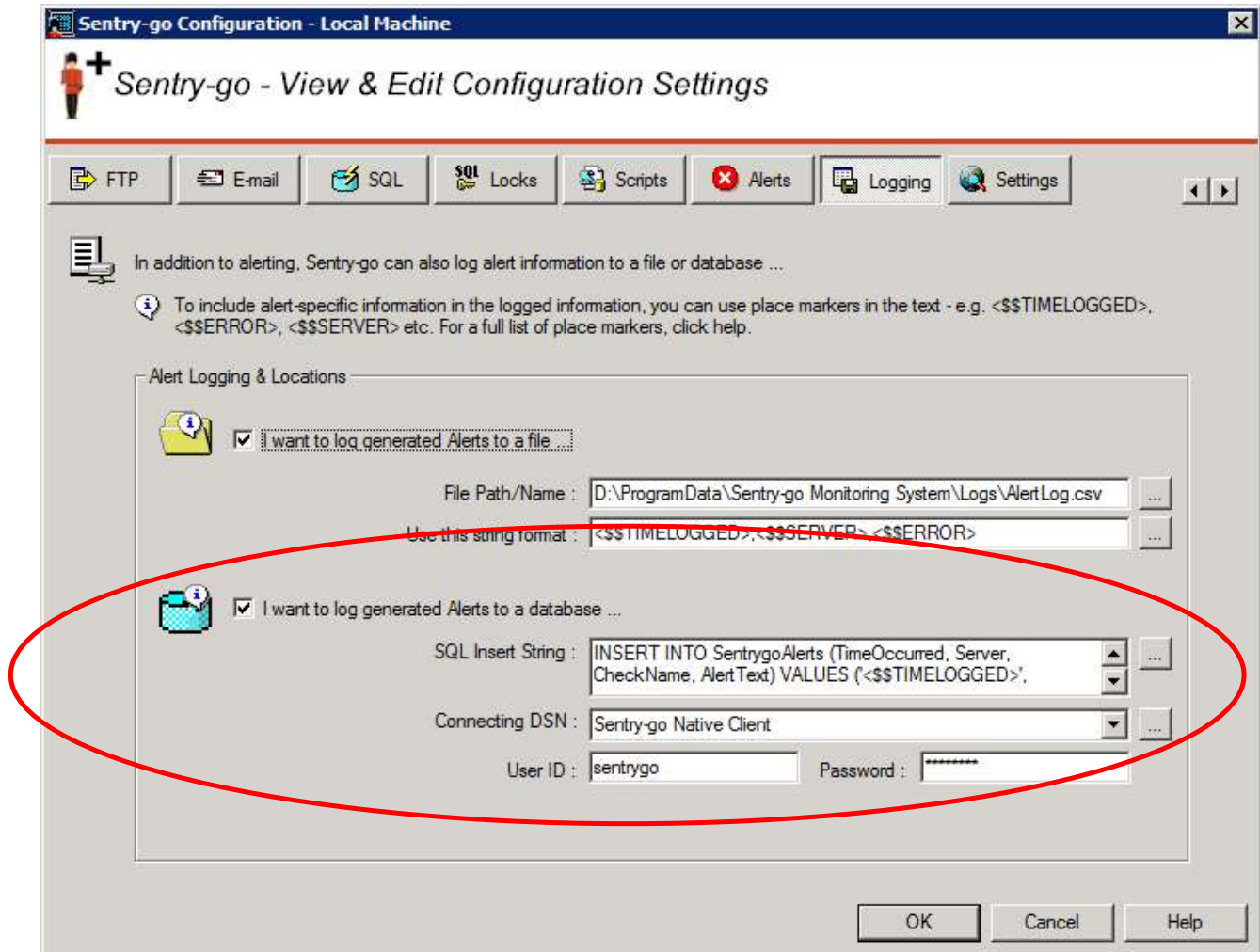
-  There is no external setup required when logging to a text file. The file will be created automatically by Sentry-go if it doesn't already exist.

## Logging Alerts to a Database

From the logging tab, select the second set of options to define the logging of alerts to an ODBC databases.

This option is similar to the above file log, except that triggered alerts are inserted into an SQL database. Any database that supports ODBC can be used. Simply install a suitable ODBC driver, create an ODBC entry to connect to your database and reference it in the configuration window.

You can insert the data into any database table, file or record using the "Insert string" defined. This allows you to define the SQL statement, substituting the actual alert values at run time.



## Log Alerts to a database

Check (tick) this option if you wish to record alert information to an ODBC data source (database).

## SQL Insert String

This is the mask for the SQL statement (normally an INSERT) that will be run in order to insert the alert into the database. It can be made up of one or more system variables, which will be substituted before the database connection is made and the statement run. The variables available are the same as the Alert Engine's [place-markers](#).

For example, ...

```
INSERT INTO SentrygoAlerts (TimeOccurred, AlertLevel, Server, CheckName,
AlertText) VALUES ('<$$TIMELOGGED>', <$$ALERTLEVEL>, '<$$SERVER>',
'<$$TEST>', '<$$ERROR>')
```

The statement itself is dependent on the target table definition and the data types (e.g. string vs numeric etc.) See "Setting up the Database " below for more information.

For help in defining this message text, click the appropriate "..." button to launch the message builder. See ["Sentry-go- Place-markers"](#) for more information.

## Connecting DSN

Select the data source that Sentry-go should use in order to connect to the target database/table. Click the "..." button to create a new DSN through ODBC.



Note that this option is only available if you are configuring the local server.

## User ID

This value is used to specify the SQL Server User ID that is to be used with the above ODBC connection in order to logon to the database (if you are logging to a SQL Server database).

- For databases such as Microsoft Access that do not require a logon, leave this field blank.
- To use a Trusted SQL Server Connection, leave this field blank. See ["Using a Trusted SQL Server Connection with Sentry-go"](#) for more information.

## Password


This is the password associated with the above SQL Server User ID. If no password is required, simply leave this entry blank.




---

## Setting up the Database


To log details to a database, the target database must contain a known table that Sentry-go can reference. You can create your Alert table using any format you require in any ODBC-compliant database such as SQL Server, Oracle, Microsoft Access - e.g. to fit in with other systems etc.

-  Example databases/scripts for Microsoft SQL Server and Microsoft Access can be found in the "Alert Logging" folder on the installation disk or download file.

Typically your installation will work something like this ...

- Create a database that will contain the new table (or choose an existing database if appropriate).
  -  Ensure the ODBC DSN configured above is connecting with this database (e.g. it is its default database). If not, reconfigure the ODBC DSN.
- Create a table with the same name as in the INSERT statement above – e.g. SentrygoAlerts.
- Ensure the columns & database types match those shown in the statement configured above. Typically these will be ...

Variable	Data type
<\$\$SERVER>	CHAR/VARCHAR (255)
<\$\$ERROR>	VARCHAR (<Length of Message>). It is recommended that either a text or memo-type field, or a VARCHAR field which is trimmed (cut to this length or less) is used within the SQL statement.
<\$\$TEST> & <\$\$SOURCE>	CHAR/VARCHAR (255)
<\$\$INFO>	CHAR/VARCHAR (255)
<\$\$TITLE>	CHAR/VARCHAR (255)
<\$\$TIMELOGGED>	CHAR/VARCHAR (30)
<\$\$TIMENOW>	CHAR/VARCHAR (30)
<\$\$OSNAME>	CHAR/VARCHAR (255)
<\$\$SERVICEPACK>	CHAR/VARCHAR (255)
<\$\$ALERTLEVEL>	Numeric 1-10

- If required, create a database user account that Sentry-go will use to access the table and ensure permissions allow this user both read & write access to this table. For example, create a "sentrygo" user with an appropriate password.
  -  Ensure this SQL Server user is also configured on the logging tab so Sentry-go uses it when logging an alert to the database.

---

## Logging Performance Information

If you have purchased the Performance Monitoring component, then you have the option of logging performance data to a file for use with trend analysis and for export into other tools.

 For more information, please refer to the [Sentry-go - Monitoring System Performance](#) guide.

---

## More Information, Help & Support

More information can be found in the guides that accompany the Sentry-go software. You can also access the following resources ...

- For the very latest information & product updates, please visit <http://www.Sentry-go.com>
- For sales advice, please e-mail [Sales@Sentry-go.com](mailto:Sales@Sentry-go.com)
- For technical support, please e-mail [Support@Sentry-go.com](mailto:Support@Sentry-go.com)



**3Ds (UK) Limited**  
*Design, Develop, Deliver Solutions!*

69, Esher Road,  
East Molesey,  
Surrey.  
KT8 0AQ

<http://www.3Ds.co.uk>