



**TaskCentre® v4.5  
MS SQL Server Trigger Tool  
White Paper**

- COPYRIGHT ..... 1**
- TRADEMARKS..... 1**
- INTRODUCTION ..... 2**
  - Overview ..... 2
  - Features ..... 2
- GLOBAL CONFIGURATION..... 2**
- TECHNICAL SUMMARY ..... 2**
  - Working with other Steps..... 2
  - Dependencies..... 2
    - MS SQL Server Trigger Agent ..... 3
  - Objects Consumed ..... 3
  - Objects Exposed ..... 3
- CONFIGURATION ..... 3**
  - General ..... 4
  - Main..... 5
  - Trigger ..... 6
    - Configure..... 6
    - Script.....10

## Copyright

The copyright in this document is owned by Orbis Software 2010. All rights reserved.

This publication may not, in whole or part, be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form or by any means without the prior written consent of Orbis Software Limited.

Head Office:

Orbis Software Limited (UK)

2 Nuffield Road

Nuffield Industrial Estate

Poole, Dorset

BH17 0RB

United Kingdom

Tel: +44 (0) 1202 241124

Fax: +44 (0) 1202 241125

Email: [enquiries@orbis-software.com](mailto:enquiries@orbis-software.com)

## Trademarks

Orbis Software Ltd owns the registered Trademark 'TaskCentre'.

All other Trademarks used within this documentation are acknowledged as the property of their respective owners.

The information provided in this publication may contain errors, omissions, or typographical errors or may be out of date. Orbis Software may change, delete, or update any published information at any time and without prior notice. The information published in this document is provided for informational purposes only and is not binding on Orbis Software in any way except to the extent that it is specifically indicated to be so.

# Introduction

## Overview

The MS SQL Server Trigger Tool is used to create a Task Step that utilises the Microsoft SQL Server trigger functionality to create a trigger on a table of a specified SQL data source. When a specified data modification is attempted, such as an attempt to add or delete a table row, the trigger then causes a Task to run.

The trigger creates virtual tables for inserted/updated and deleted rows which may be mapped to Task variables so that the data may be used in other Steps within the Task.

If a Database Query (ODBC) or Database Query (OLEDB) Step is used within the Task to retrieve a recordset from the SQL data source when the data is modified, then the memory feature may be used in a subsequent Step to memorise all of the data into a Repository.

## Features

- A Trigger may be activated when any row is updated, deleted or a new row inserted.
- A Trigger may be activated when specified columns are updated.
- Values of updated, deleted or new rows in the virtual tables may be mapped to variables for use in other Steps within the Task.

## Global Configuration

The MS SQL Server Trigger Tool requires no global configuration. However, SQL Server 2000, 2005 or 2008 data source is required and the MS SQL Server Trigger Agent must be installed on the same machine as the data source.

## Technical Summary

### Working with other Steps

The Schedule Step is independent of any other Steps used in a Task. However, it accesses parameters that the Task exposes through the Task Browser dialog.

### Dependencies

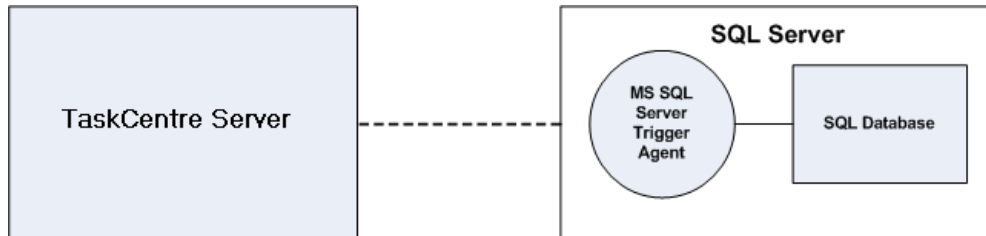
The products, technologies, protocols or systems that are required for the Schedule Step to work are listed below.

- **Microsoft SQL Server 2000, 2005 or 2008 Data Source**
- **MS SQL Server Trigger Agent** - The agent must be installed on the SQL Server.

## MS SQL Server Trigger Agent

The MS SQL Server Trigger agent is a service that sits between TaskCentre and SQL Server and may be installed onto any SQL Server. It is responsible for both maintaining the Triggers in SQL Server and for handling the running of a Task once a Trigger has been fired.

When a SQL Server Trigger is fired, the agent service will queue the applicable Task.



## Objects Consumed

The Schedule Step consumes no objects exposed by other Steps.

## Objects Exposed


The objects that the MS SQL Server Trigger Step exposes that can be utilised by other Steps are listed below.

- **Variables (Text)** - Textual data is mapped to selected Parameter Variables for use in other Steps



## Configuration

A MS SQL Server Trigger Step is configured through the MS SQL Server Trigger dialog which is displayed through the following methods:

To create a new Step:

- In the Task Browser dialog, select the 'Tools' tab, click on the MS SQL Server Trigger icon  under the 'Event' node and then drag and drop- it onto the Task Plan. The MS SQL Server Trigger dialog will open and once created an icon for the Step will be displayed on the Task Plan.
- In the Task Browser dialog, right-click on the Steps node and select the 'MS SQL Server Trigger' option from the 'New' > 'Event' drop down menu. The MS SQL Server Trigger dialog will open and once created the Step is displayed under the Steps node in the Task Browser. It may then be dragged and dropped onto the Task Plan.
- Right-click anywhere on the Task Plan in the Design tab of the Task dialog and selected the 'MS SQL Server Trigger Event' option from the 'New' > 'Event' drop down menu. The MS SQL Server Trigger dialog will open and once created an icon for the Step will be displayed on the Task Plan.

To open an existing Step:

- On the Task Plan in the Design tab of the Task dialog, double-click on the MS SQL Server Trigger icon . The MS SQL Server Trigger dialog will open to enable the Step details to be edited.
- In the Task Browser dialog, double-click on the MS SQL Server Trigger node  displayed under the Steps node. The MS SQL Server Trigger dialog will open to enable the Step details to be edited.
- In the Task Browser dialog, right-click on the MS SQL Server Trigger node displayed under the Steps node and select the 'Edit Step' option from the drop down menu. The MS SQL Server Trigger dialog will open to enable the Step details to be edited.

The dialog has three tabbed panes described below through which an MS SQL Server Trigger Step may be created and configured.

## General

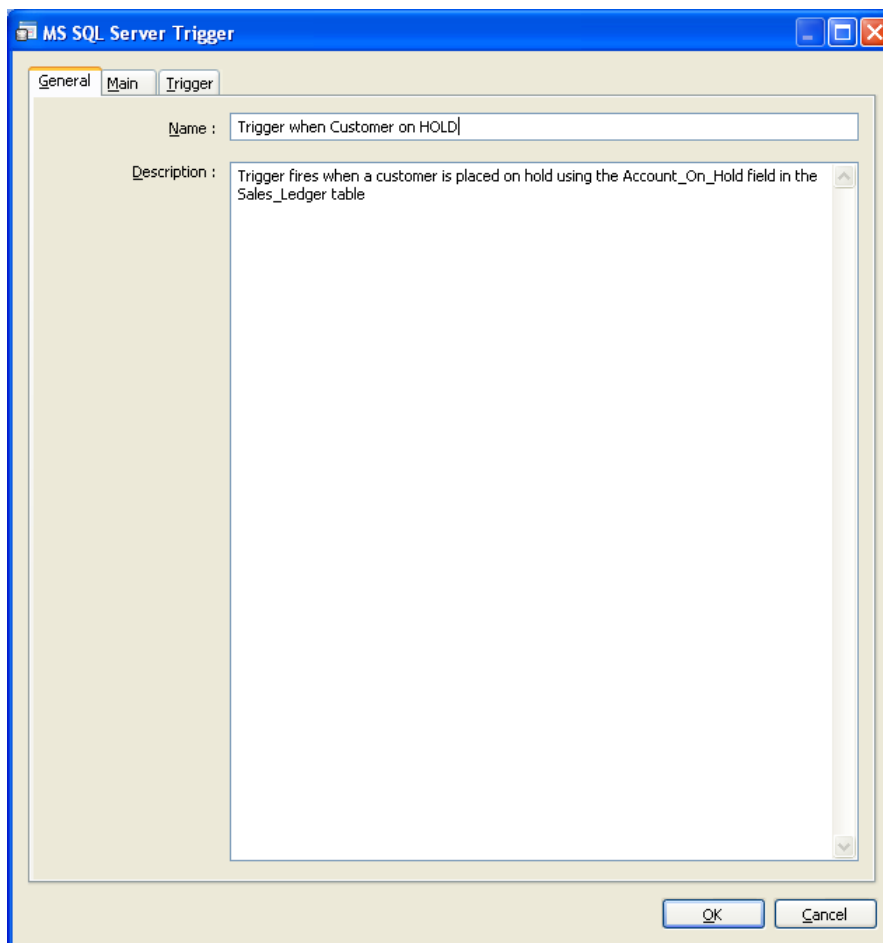


Figure 1. MS SQL Server Trigger – General tab.

The contents of the '**General**' tab (Ref: Figure 1) are as follows:

- '**Name**' (field) – A unique name for the Step may be manually entered.
- '**Description**' (field) – A description for the Step may be manually entered.

## Main

Figure 2. MS SQL Server Trigger – Main tab.

The Main tab (Ref: Figure 2) provides options for configuring the connection to the required database table to be used by the trigger and where more than one trigger has been created for the Task the order in which it is run.

The contents of the '**Main**' tab are as follows:

- '**Server**' (area) - This area contains the following for selecting a server which is hosting the required SQL database:
  - '**Select Server:**' (drop down list) - Displays a list of all available SQL Servers that have the MS SQL Server Trigger Agent installed for selection.
  - '**Select Server Instance:**' (drop down list) – If more than one instance of SQL Server is installed on the selected Server, then they are displayed in the list for selection.

- **'Credentials'** (area) - This area contains the following for setting the selected server login credentials:
  - **'Login Name'** (field) - A valid login name for connecting to the selected server may be manually entered.
  - **'Password'** (field) - A valid password for the selected login name may be manually entered.
  - **'Use Trusted Connection'** (check box) - If this option is selected then the currently logged in Users credentials are used to access the selected server. A different User name may be selected to run the Task if this is required to access the required server / database table through the Impersonation tab of the Task Options dialog.
- **'Select a Table / View'** (area) - This are contains the following for selecting the required SQL database table or view:
  - **'Create Trigger On:'** (field) - Displays the table / column names selected from the SQL database.
  - **'Browse...'** (button) - Opens the SQL Server Browser dialog to enable a SQL database table or view to be selected.
- **'Trigger Order'** (area) - This area provides the following 'Fire Trigger' options for selection:
  - **'First'** (option radio button) - If more than one trigger has been created for the Task, selecting this option fires this trigger first.
  - **'Last'** (option radio button) - As above, if more than one trigger has been created then selecting this option fires this trigger last.
  - **'No order specified'** (option radio button) - This option is selected by default.

## Trigger

The Trigger tab provides a comprehensive set of options for configuring the trigger. The dialog has two tabbed panes listed below through which a MS SQL Server Trigger may be created and configured.

## Configure

The Configure tab (Ref: Figure 3) provides configuration options to enable TaskCentre to automatically create the necessary T-SQL Script for the trigger. The script created may be viewed by clicking on the Script tab.

The Configure tab contains the following:

- **'Trigger Type:'** (option check boxes) - The following options are provided for selection:
  - **UPDATE** - When this trigger type is selected, the trigger is fired when a column in the selected table is updated.
  - **INSERT** - When this trigger type is selected, the trigger is fired when a new row is added to the table.



- **DELETE** - When this trigger type is selected, the trigger is fired when an existing row is deleted from the table.
- **'Fire Trigger:'** (option radio buttons) - The following options are provided for selection:
  - **AFTER** - This is the default trigger option which fires the trigger after the table has been updated.
  - **INSTEAD OF** - When this trigger option is selected, the trigger fires and performs its task instead of the table being updated.
- **'Run Task once for each Trigger'** (option radio button) - If this option is selected then the Task is run once for each trigger created for the Task. Numerous updates may therefore be made to the table but the trigger will only fire once. No mapping of results set columns to Task Parameters is available, but any changes to the database that caused the trigger to fire may be accessed through the TaskCentre memory feature.
- **'Run Task once for each Row in Results Set in:'** (option radio button) - When the trigger is fired, an 'Inserted' and a 'Deleted' table are produced containing a record set depending on the updates made to the original table. Selecting this option enables you to select which of the tables is to be used for running the Task.
  - **'Inserted Table'** (option radio button) - If this table option is selected then the Task will run once for each row created in the Inserted table when the SQL trigger was fired. Each row represents the addition of a new row in the original table and in addition will include any updated rows if the 'UPDATE' trigger type has been previously selected.
  - **'Deleted Table'** (option radio button) - If this table option is selected then the Task will run once for each row created in the Deleted table when the SQL trigger was fired. Each row represents the editing or deletion of an existing row in the original table.
- **'Map Columns to Task Parameters:'** (table) - Displays a list of variables created for the Task. The 'Columns' column contains a drop down list of columns for selection, from the 'Inserted' and 'Deleted' tables created by the SQL Trigger, to be mapped against the variable. The variable may then be used to expose the selected column value to subsequent Steps.
- **'Only Fire Trigger when a selected Column is updated:'** (option check box) - Selecting this option activates the 'Columns' table displayed below it.
- **'Columns'** (table) - Displays a list of columns available in the selected database table. A check box is provided to the left of each column name for selecting / de-selecting the column. The trigger is then fired when any of the selected columns is updated.
- **'Filter Table'** (option check box) - Selecting this option activates the **'Filter'** button so that it may be selected.

- **'Filter'** (button) - Clicking on the Filter button opens the Selecting the **'Table Filter Expression'** dialog (Ref: Figure 4) to enable a new expression to be entered.

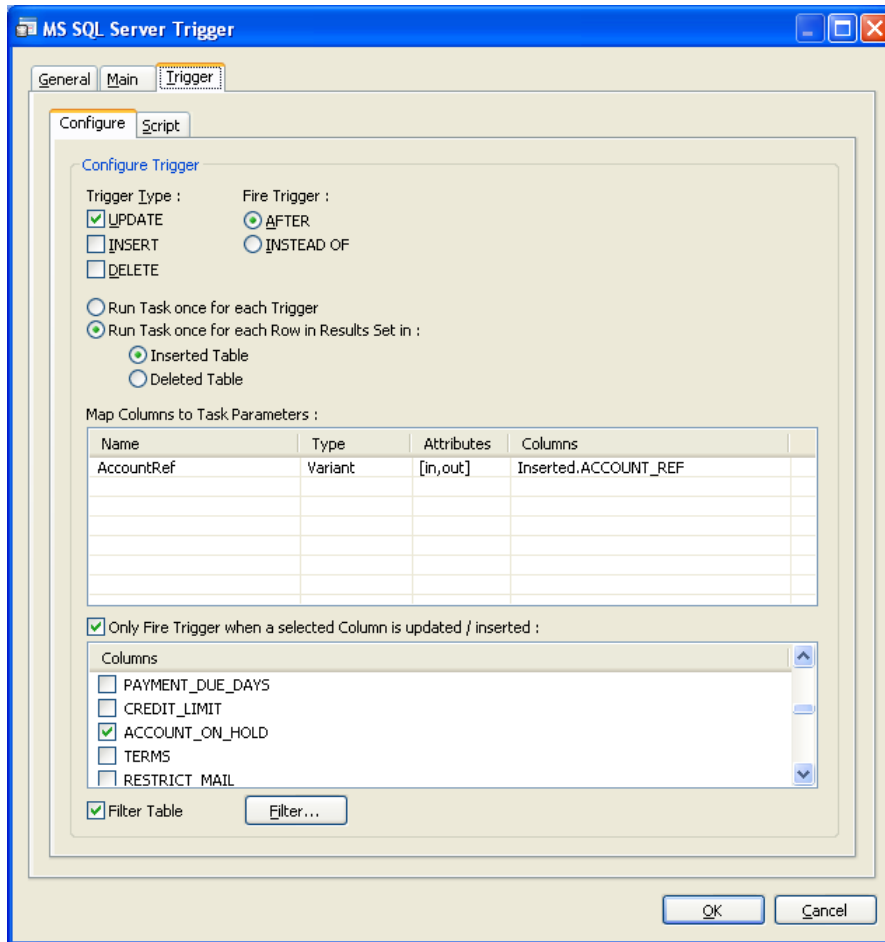


Figure 3. MS SQL Server Trigger – Trigger tab - Configure.

## Table Filter Expression dialog

The **'Table Filter Expression'** dialog (Ref: Figure 4) is displayed when the **'Filter'** button is selected in the **'Trigger'** tab of the **'MS SQL Server Trigger'** dialog.

In the **'Main'** tab of the Tool, a SQL database and table are selected and when a trigger fires it creates virtual tables for inserted/updated and deleted rows in the selected table.

You can configure the trigger to fire when a specific column is updated but through the **'Table Filter Expression'** dialog you may also create a filter expression on that column value.

For example, a table is used to store the value of orders received in a column called 'Value'. The Task runs a report to list all orders received with a value in excess of £10,000 and then sends it to the Sales Manager.

The report could be scheduled to run on a regular basis but the manager requires the report to be run each time a new large value order is received.

The following expression is therefore created for the trigger 'Inserted.Value > = 10000' to cover occurrences where the value is equal to or exceeds £10,000.

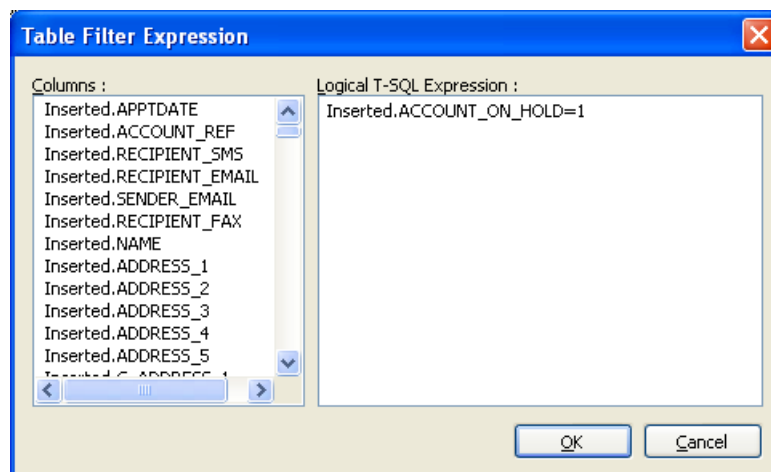


Figure 4. Table Filter Expression dialog.

The dialog contains the following for creating a filter:

- **'Columns'** table - Displays a list of tables/columns that will be created when the trigger is fired. Inserted and deleted tables/columns will be created for each available column in the selected table.
- **'Logical T-SQL Expression'** text box - The expression is manually entered into this area. Any table/column listed in the 'Columns' table may be dragged and dropped into the text box for inclusion in the filter expression.

## Script

The Script tab (Ref: Figure 5) is provided to enable the T-SQL Script for a trigger to be manually created.

When the tab is first selected it will display the T-SQL Script automatically created for the trigger through the Configure tab using the default options selected in the tab.

Selecting the **'Edit'** button switches the tab to script mode so that the T-SQL Script may be manually edited.

Once the script mode has been selected it cannot be reversed and the trigger may only be edited by updating the T-SQL Script.

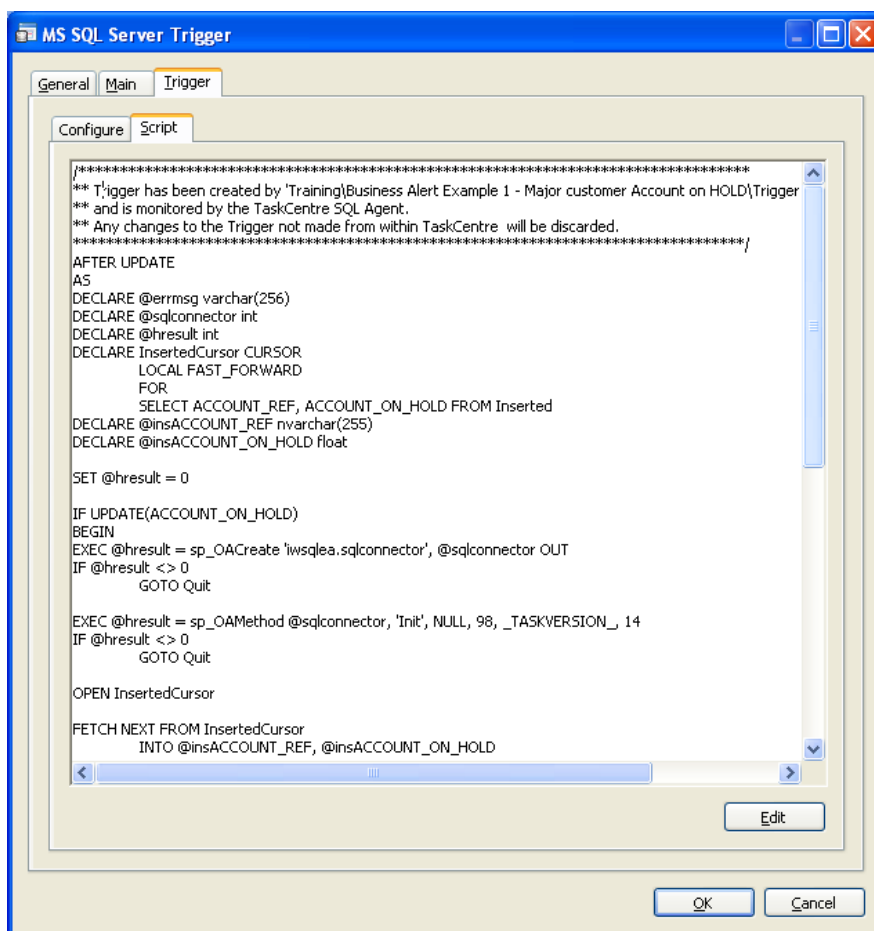


Figure 5. MS SQL Server Trigger – Trigger tab - Script.