Cisco Tidal Enterprise Scheduler Microsoft SQL Server Adapter Guide

Version: 6.2.1

May 4, 2016

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Cisco Tidal Enterprise Scheduler Microsoft SQL Server Adapter Guide

CONTENTS

Preface

This guide describes the installation, configuration, and usage of the Microsoft SQL Server Adapter with Cisco Tidal Enterprise Scheduler (TES).

Audience

This guide is for administrators who install and configure the Microsoft SQL Server Adapter for use with TES, and who troubleshoot TES installation and requirements issues.

Related Documentation

See the *Cisco Tidal Enterprise Scheduler Documentation Overview* for your release on cisco.com at:

http://www.cisco.com/c/en/us/support/cloud-systems-management/tidal-enterprise-scheduler/products-documentation-roadmaps-list.html

...for a list of all TES guides.

Note We sometimes update the documentation after original publication. Therefore, you should also review the documentation on Cisco.com for any updates.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see What’s New in Cisco Product Documentation at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html.

Subscribe to What’s New in Cisco Product Documentation, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

Document Change History

The table below provides the revision history for the *Cisco Tidal Enterprise Scheduler Microsoft SQL Server Adapter Guide*.

Version Number	Issue Date	Reason for Change
6.1.0	October 2012	• New Cisco version.
6.2.1	June 2014	• Available in online Help only.
6.2.1 SP2	June 2015	• Configuration provided in the *TES Installation Guide*; usage provided in online Help only.
6.2.1 SP3	May 2016	• Consolidated all Microsoft SQL Server Adapter documentation into one document.

CHAPTER 1

Introducing the Microsoft SQL Server Adapter

This chapter provides an overview of the Microsoft SQL Server Adapter and its requirements:

- Overview
- Prerequisites

Overview

The Microsoft SQL Server (MSSQL) adapter integrates Tidal Enterprise Scheduler (Scheduler) with MSSQL Server allowing you to:

- Create, schedule and run MSSQL queries and MSSQL Agent jobs through Scheduler.
- Monitor events within a MSSQL database, including table, index, and row creation, modification and deletion.

The MSSQL Adapter supports running jobs under both Microsoft SQL Server 2000, 2005, 2008, and 2012. However, event monitoring is supported on 2005, 2008 and 2012 only.

If you are installing Microsoft SQL Server 2000 and 2005 on the same machine, you will need to have separate names for each instance. Or, alternatively, you can assign each instance its own unique listening port. The default database port is usually 1433.

If you are using different versions of Microsoft SQL Server, it is a good idea to place all jobs under a SQL group. This way, when switching between versions, you only need to switch the connection at the group level.

Prerequisites

Refer to your *Cisco Tidal Enterprise Scheduler User Guide* for a complete list of hardware and software prerequisites. These sections describe the prerequisite Microsoft SQL Server user permissions and roles:

- Database Jobs
- Database Events

Database Jobs

Creating, viewing and running database jobs requires a variety of permissions. The easiest way to accomplish this is to assign the **sysadmin** role to your user.

To assign the sysadmin role and grant permissions:

Step 1 Run the following query to determine if your user is assigned the sysadmin role:

```
select is_srvrolemember('sysadmin')
```

If assigning the sysadmin role is not possible, then your user must be granted the permissions listed in the next steps.

Step 2 Run the queries below to verify these have been granted:

```
select has_perms_by_name('msdb..sysobjects', 'object', 'select')
select has_perms_by_name('msdb..sysproxies', 'object', 'select')
select has_perms_by_name('master..syslogins', 'object', 'select')
select has_perms_by_name('master..sysdatabases', 'object', 'select')
select has_perms_by_name('sys.objects', 'object', 'select')
select has_perms_by_name('sys.schemas', 'object', 'select')
select has_perms_by_name('sys.sysusers', 'object', 'select')
select has_perms_by_name('sys.syslogins', 'object', 'select')
```

Step 3 If one or more of the above permissions are missing, issue the following statements to grant them to your user. Assume your user is **TIDALSOFT\myuser**.

```
grant select on object::msdb..sysobjects to [TIDALSOFT\myuser];
grant select on object::msdb..sysproxies to [TIDALSOFT\myuser];
grant select on object::master..syslogins to [TIDALSOFT\myuser];
grant select on object::master..sysdatabases to [TIDALSOFT\myuser];
grant select on object::sys.objects to [TIDALSOFT\myuser];
grant select on object::sys.schemas to [TIDALSOFT\myuser];
grant select on object::sys.sysusers to [TIDALSOFT\myuser];
grant select on object::sys.syslogins to [TIDALSOFT\myuser];
```

Step 4 Ensure that your user has the following permissions for each database your SQL Adapter jobs reference. In the example below the database is called **XYZ**.

```
select has_perms_by_name('[XYZ]..sysobjects', 'object', 'select')
select has_perms_by_name('[XYZ]..sysusers', 'object', 'select')
select has_perms_by_name('[XYZ]', 'database', 'showplan')
```

Step 5 If any of the above permissions are missing, issue the following statements:

```
grant select on object::[XYZ]..sysobjects to [TIDALSOFT\myuser];
grant select on object::[XYZ]..sysusers to [TIDALSOFT\myuser];
grant showplan on database::[XYZ] to [TIDALSOFT\myuser];
```

Step 6 Ensure that your user can call these database functions. You do not need to assign any permissions by default.

```
getDate()
getUtcDate()
getServerProperty()
suser_sname()
```

Your user must be assigned the **sysadmin** role or one of the following SQL Server Agent roles: **SQLAgentUserRole**, **SQLAgentReaderRole** or **SQLAgentOperatorRole**. Only the **sysadmin** role can create, view and run all jobs in the database. Refer to SQL Server documentation about the restrictions placed on each of the SQL Server Agent roles with regard to jobs. Refer to the documentation about how to check which roles are assigned to your user and how to assign one of these roles to your user.

Database Events

Monitoring database events requires your user be assigned to the **sysadmin** role and the **db_owner** role for each database being monitored.

Configuring the Microsoft SQL Server Adapter

Overview

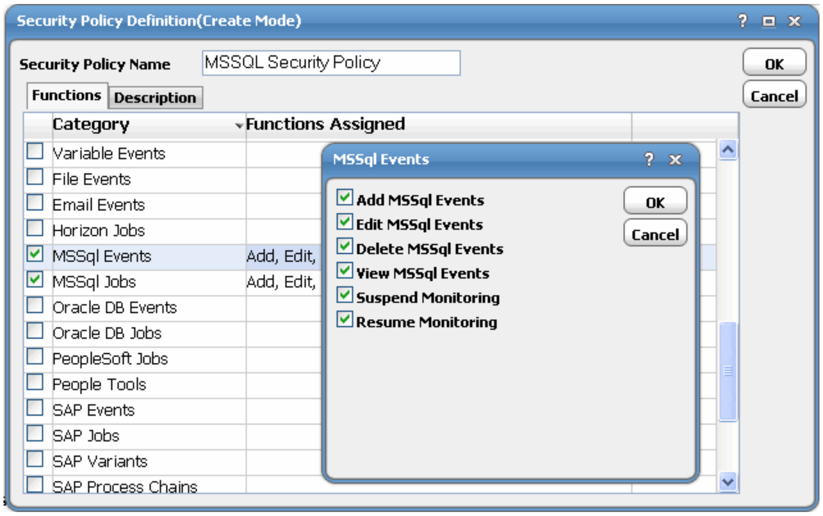
The Microsoft SQL Server adapter software is already installed as part of a normal installation of Scheduler. However, you must perform the following steps to license and configure the adapter before you can run Microsoft SQL jobs:

- Defining a Microsoft SQL Security Policy – Create a security policy to assign to the users that will run Microsoft SQL jobs. You may want to create different security policies to differentiate between users who just run Microsoft SQL jobs versus users who can run and edit Microsoft SQL jobs.
- Licensing an Adapter – License the connection(s) to the Microsoft SQL instance. You cannot define a Microsoft SQL connection until you have applied the Microsoft SQL license from Tidal Software.
- Securing the Microsoft SQL Database Adapter – Create one or more user definitions in Scheduler with access to the Microsoft SQL instance using Microsoft SQL user names and passwords provided by the Microsoft SQL administrator. Users in Scheduler are authorized to run Microsoft SQL jobs on behalf of these Microsoft SQL runtime users.
- Define a Microsoft SQL Connection – Define a Microsoft SQL connection so the master can communicate with the Microsoft SQL instance.

See also Configuring service.props for information about general and adapter-specific properties that can be set to control things like logging and connection properties.

Defining a Microsoft SQL Security Policy

You can create a new security policy or modify an existing security policy in TES that includes the authority to add, edit, view and delete Microsoft SQL jobs and events. In the **Security Policy Definition** dialog, there is an **MSSql Jobs** and **MSSql Events** category.



Security Policy Name: MSSQL Security Policy

	Category	Functions Assigned
□	Variable Events	
□	File Events	
□	Email Events	
□	Horizon Jobs	
☑	MSSql Events	Add, Edit,
☑	MSSql Jobs	Add, Edit,
□	Oracle DB Events	
□	Oracle DB Jobs	
□	PeopleSoft Jobs	
□	People Tools	
□	SAP Events	
□	SAP Jobs	
□	SAP Variants	
□	SAP Process Chains	

- ☑ Add MSSql Events
- ☑ Edit MSSql Events
- ☑ Delete MSSql Events
- ☑ View MSSql Events
- ☑ Suspend Monitoring
- ☑ Resume Monitoring

To grant access privileges:

Step 1 In the **Navigator** pane, select **Security Policies** to display the **Security Policies** pane.

Step 2 Select a security policy for the Microsoft SQL job privileges and double-click on it to display its **Security Policy Definition** dialog.

Step 3 Scroll down the list of function categories and double-click on the **MSSql Events** or **MSSql Jobs** category to display the available functions.

Step 4 Select the desired job privileges, then click **OK**.

A check mark appears next to the **MSSql** function category indicating that one or more functions are selected within the category.

If needed, different security policies with varying authorized functions can be created to provide different levels of access for a variety of users.

For more information about the Security Policy Definition dialog, refer to the “Users” chapter in your *Cisco TES User Guide*.

Licensing an Adapter

Each TES Adapter must be separately licensed. You cannot use an Adapter until you apply the license file. If you purchase the Adapter after the original installation of TES, you will receive a new license file authorizing the use of the Adapter.

You might have a Demo license which is good for 30 days, or you might have a Permanent license. The procedures to install these license files are described below.

To license an Adapter:

Step 1 Stop the master:

Windows:

a. Click **Start** and select **Programs>TIDAL Software>Scheduler>Master>Service Control Manager**.

b. Verify that the master is displayed in the **Service** list and click on the **Stop** button to stop the master.

UNIX:

Enter **tesm stop**

Step 2 Create the license file:

- For a Permanent license, rename your Permanent license file to *master.lic*.
- For a Demo license, create a file called *demo.lic*, then type the demo code into the *demo.lic* file.

Step 3 Place the file in the **C:\Program File\TIDAL\Scheduler\Master\config** directory.

Step 4 Restart the master:

Windows:

Click **Start** in the Service Control Manager.

UNIX:

Enter **tesm start**

The master will read and apply the license when it starts.

Step 5 To validate that the license was applied, select **Registered License** from **Activities** main menu.

Securing the Microsoft SQL Database Adapter

There are two types of users associated with the Microsoft SQL database adapter, **Runtime Users** and **Schedulers**. You maintain definitions for both types of users from the **Users** pane.

- **Runtime Users**

  Runtime users in the context of Microsoft SQL jobs represent those users and passwords required to connect to the SQL Server database to run queries and jobs. Runtime users are also for connecting to SQL Servers to monitor database events.

- **Schedulers**

  Schedulers are those users who will define and/or manage Microsoft SQL jobs and events. There are three aspects of a user profile that grant and/or limit access to scheduling jobs that invoke Microsoft SQL:

  - Security policy that grants or denies add, edit, delete and view capabilities for Microsoft SQL jobs and events.
  - Authorized runtime user list that grants or denies access to specific database user accounts for use with database connections to monitor for events and with database jobs.
  - Authorized agent list that grants or denies access to specific Microsoft SQL adapter connections for use when defining Microsoft SQL jobs and events.

Defining Runtime Users (Database Users)

To define a database runtime user to run SQL jobs and monitor events:

Step 1 From the **Navigator** pane, select **Administration>Runtime Users** to display the **Users** pane.

Step 2 Right-click and select **Add Runtime User** from the context menu, or select and existing user and choose **Edit** to display the **User Definition** dialog.

Step 3 If this is a new user definition, enter the new user name in the **User Name** field.

Step 4 For documentation, enter the **Full Name** or description associated with this user.

Step 5 In the **Domain** field, select a Windows domain associated with the user account required for authentication, if necessary.

Step 6 On the **Passwords** tab, click **Add** to display the **Change Password** dialog.

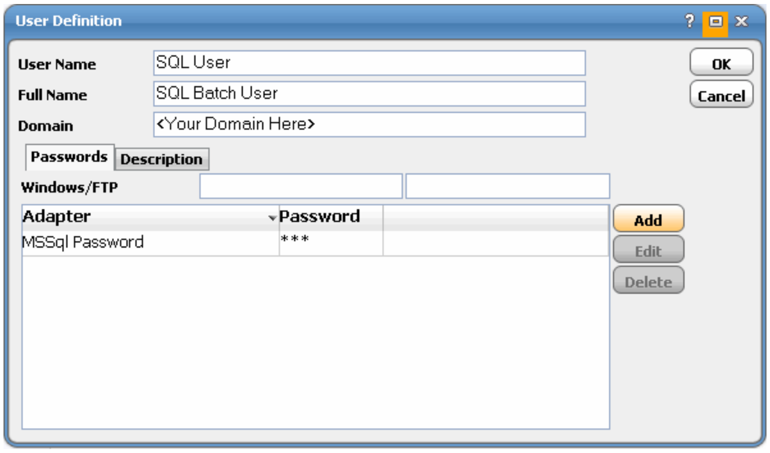
Step 7 From the **Password Type** list, select **MSSql**.

Step 8 Enter a password (along with confirmation) in the **Password** and **Confirm Password** fields.

Only those users with a password specified for Microsoft SQL will be available for use with Microsoft SQL jobs and events. The password may be the same as the one specified for Windows/FTP jobs.

Step 9 Click **OK** on the **Change Password** dialog.

The **User Definition** dialog displays the new record.



User Name: SQL User

Full Name: SQL Batch User

Domain: <Your Domain Here>

Windows/FTP

Adapter	Password
MSSql Password	***

Step 10 Click **OK** to add or save the user recorded in the Enterprise Scheduler database.

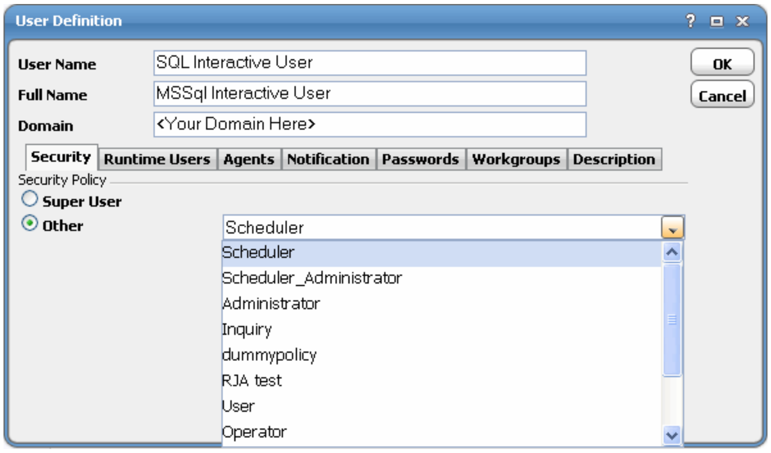
Authorizing Schedulers to Work with Microsoft SQL Jobs and Events

To define a Scheduler user to work with Microsoft SQL jobs and events:

Step 1 From the **Navigator** pane, select **Administration>Interactive Users** to display the **Users** pane, listing all defined users.

Step 2 Right-click and select **Add Interactive User** from the context menu, or select an existing user and choose **Edit Interactive User** to display the **User Definition** dialog.

Note Refer to the *Tidal Enterprise Scheduler User Guide* for a general discussion on setting up a user to work with Scheduler.



User Definition

User Name: SQL Interactive User

Full Name: MSSql Interactive User

Domain: <Your Domain Here>

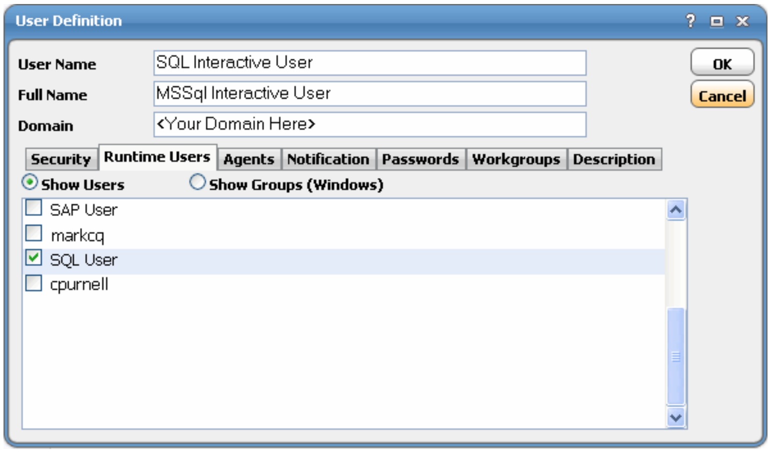
Security Policy

○ Super User

● Other: Scheduler

Step 3 On the **Security** tab, select a security policy that includes authorization for Microsoft SQL jobs and/or events.

Step 4 Click the **Runtime Users** tab.



User Definition

User Name: SQL Interactive User

Full Name: MSSql Interactive User

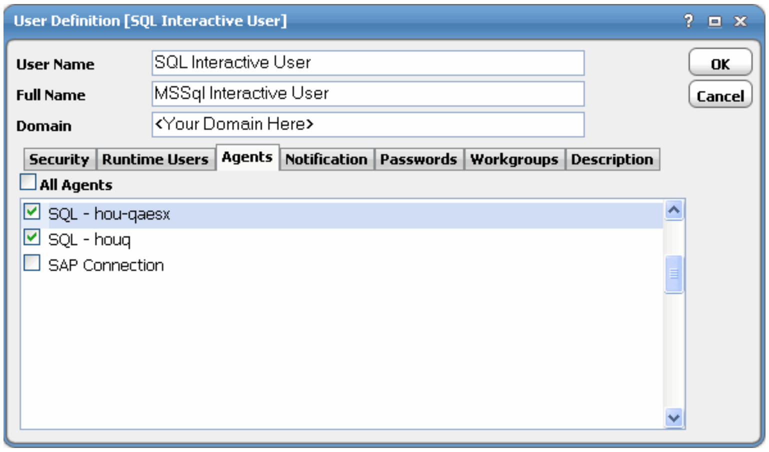
Domain: <Your Domain Here>

● Show Users ○ Show Groups (Windows)

□ SAP User
□ markcq
☑ SQL User
□ cpurnell

Step 5 Select the database users that this scheduling user may use when defining Microsoft SQL jobs and/or connections for database event monitoring.

Step 6 Click the **Agents** tab.



User Name: SQL Interactive User
Full Name: MSSql Interactive User
Domain: <Your Domain Here>

☐ All Agents
☑ SQL - hou-qaesx
☑ SQL - houq
☐ SAP Connection

Step 7 Select which Microsoft SQL connections that this scheduling user can access when scheduling jobs or events.

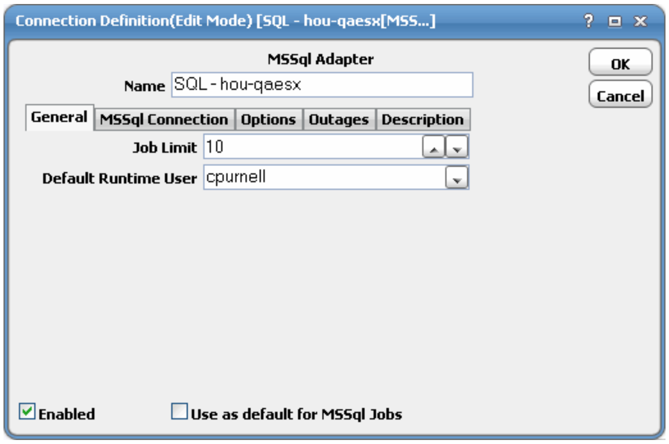
Step 8 Click **OK** to save the user definition.

Define a Microsoft SQL Connection

You must create a connection to the Microsoft SQL instance(s) before Scheduler can run your Microsoft SQL jobs or monitor database events. These connections also must be licensed before Scheduler can use them.

The connection for the Microsoft SQL adapter is defined like other Scheduler connections except it requires configuration information unique to Microsoft SQL instances.

You can create connections to one or more Microsoft SQL instances and/or databases. Microsoft SQL access can be managed by limiting a user account to use only certain connections to particular servers. A connection is created through the **Connection Definition** dialog.



Name: SQL - hou-qaesx
Job Limit: 10
Default Runtime User: cpurnell

☑ Enabled ☐ Use as default for MSSql Jobs

This dialog contains the following elements:

- **Name** – Enter a name for this Microsoft SQL adapter connection.
- **Enabled** – Activates or disables (deactivates) the connection, shutting down the adapter.

A job cannot run on an adapter if its connection is disabled. If jobs try to run on a disabled connection, the job enters Agent Disabled status. Jobs already running on an agent that is disabled (or if the connection to the agent is lost) run to completion; however, their completion status is not returned until the agent is enabled.

- **Use as default for MSSQL Jobs** – Select to use the connection as your default connection when setting up future Microsoft SQL jobs.
- **General** – This tab designates a job limit and default runtime user.
- **MSSQL Connection** – This tab provides information to connect to the designated Microsoft SQL instance.
- **Options** – This tab provides configuration options for accessing the Microsoft SQL instance.
- **Description** – This tab describes the connection and provides a convenient place for notes about its purpose and use.

Adding a Microsoft SQL Connection

Note Modifying an adapters connection configuration should only be done when there are no jobs running that utilize the adapter connection. If the connection is created while jobs are active, the jobs may go into an orphan state.

To add a connection:

Step 1 From the **Navigator** pane, select **Administration>Connections** to display the **Connections** pane.

Step 2 Click the **Add** button or right-click and select **Add Connection>MSSQL Adapter** from the context menu to display the **MSSQL Connection Definition** dialog.

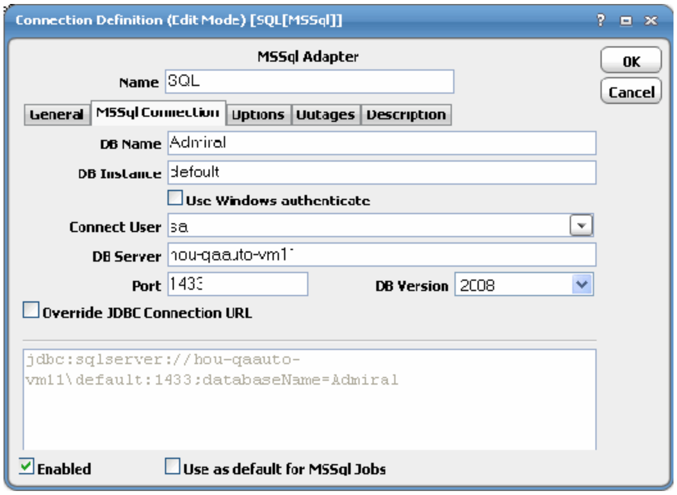
Step 3 Enter a name for the new connection in the **Name** field.

Step 4 In the **Job Limit** field of the **General** tab, select the maximum number of concurrent active processes that Scheduler should submit to this Microsoft SQL instance and/or database at one time.

Step 5 Optionally, in the **Default Runtime User** list, select the name of a default user for Microsoft SQL jobs.

Only users that have been defined with Microsoft SQL passwords display in this list. The user selected is automatically supplied as the runtime user in Scheduler Microsoft SQL job definitions.

Step 6 Click the **MSSQL Connection** tab.



Connection Definition (Edit Mode) [SQL[MSSql]]

MSSql Adapter

Name: SQL

DB Name: Admiral

DB Instance: default

□ Use Windows authenticate

Connect User: sa

DB Server: hou-qaauto-vm11

Port: 1433

DB Version: 2008

□ Override JDBC Connection URL

```
jdbc:sqlserver://hou-qaauto-
vm11\default:1433:databaseName=Admiral
```

☑ Enabled

□ Use as default for MSSql Jobs

This tab defines how to access a Microsoft SQL instance. There are several different ways to connect to a Microsoft SQL instance. SIf you wish to connect to a specific database, you can designate that Microsoft SQL database for the connection.

Step 7 In the **DB Name** field, enter the name of the default database for this connection.

This is identical to the **DatabaseName** value in a JDBC connection string.

This field is optional. If not specified here, you will need to select a database for each job and/or event definition that uses this connection.

Step 8 Optionally, in the **DB Instance** field, enter the name of the database instance to connect to (if there are multiple instances installed on the same server).

This field is optional. Leave blank to use the default, unnamed instance.

Step 9 (Optional) Check the **User Windows authenticate** checkbox to enable the use Windows Authentication feature.

Note When using Windows Authentication, verify that the MSSql Server for that instance is running as a domain account. MSSql Server requires AD access for login account verification when the adapter intends to connect to MSSql Server.

Note If Windows Authentication is checked, the connection user and all runtime users associated with jobs for the connection will be authenticated with Windows Authentication. Similarly, if Windows Authentication is left unchecked, the connection user and all runtime users will be authenticated using SQL Authentication. Mixing the two authentication methods is not allowed for a single connection.

The jTDS JDBC driver is used to support Windows Authentication. You can download this driver from http://jtds.sourceforge.net.

To apply the driver:

a. Stop the master.

b. Unzip the download driver file, then copy the jTDS JDBC driver (*jtds-1.2.5.jar*) into **{TIDAL_HOME}/services/{CAFA6E81-E29F-4263-8E80-4928FFF416A2}/lib**.

c. Restart the master.

Step 10 From the **Connect User** list, select the appropriate Microsoft SQL user.

Step 11 In the **DB Server** field, enter the machine name where the Microsoft SQL database is located.

Step 12 In the **Port** field, specify the port the database instance listens on.

Step 13 In the **DB Version** list, verify the correct version of the database you are connecting to.

Step 14 Select the **Override Jdbc URL** option if you need to override the JDBC connection string.

Step 15 Click the **Options** tab to configure parameters for this connection.

Step 16 Click the **Outages** tab to define time windows when the agent/adapter will not be available.

Step 17 Click **OK** to add the connection to the Scheduler database.

Microsoft SQL Connection Status Light

If the Microsoft SQL connection is not available to the master, the connection status light in the first column of the display is red (or yellow if other agents are connected to the master).

The status light indicates the health of the adapter connection to the SQL server.

If the Enterprise Scheduler master cannot connect or loses its connection to a Microsoft SQL instance, you will see a red status light next to your Microsoft SQL connection in the **Connections** pane of the Tidal Web client.

Note If the SQL master adapter connection is not available, you will not be able to define Microsoft SQL jobs and extract information on job steps from your Microsoft SQL instance.

Defining Microsoft SQL Jobs

Overview

There are three types of Microsoft SQL Server Adapter jobs that you can define in TES:

- **MSSQL Job** – Direct inclusion of T-SQL statements.
- **Database Job** – Includes one or more steps of various database task types, including the following:
  - T-SQL
  - ActiveX Scripts
  - Operating system command
  - Replication distributor
  - Replication merge
  - Replication queue reader
  - Replication snapshot
  - Replication transaction-log reader
  - SQL Server analysis services command
  - SQL Server analysis services query
- **Existing Database Job** – Includes jobs already defined to run under the SQL Server agent.

Note Before creating and scheduling Microsoft SQL jobs, you must have:
- added the Microsoft SQL database user to Scheduler.
- defined your Microsoft SQL connection(s).
These topics are covered in Configuring the Microsoft SQL Server Adapter.

This chapter covers these topics:

- Adding a Microsoft SQL Job
- Adding a Database Job
- Linking to an Existing Database Job

Adding a Microsoft SQL Job

You can create a Microsoft SQL job using the context menu within the **Jobs** pane. You can also edit, copy and delete an existing Microsoft SQL job. If you add a Microsoft SQL job to a Scheduler job group, items common between the job group and the Microsoft SQL job are inheritable.

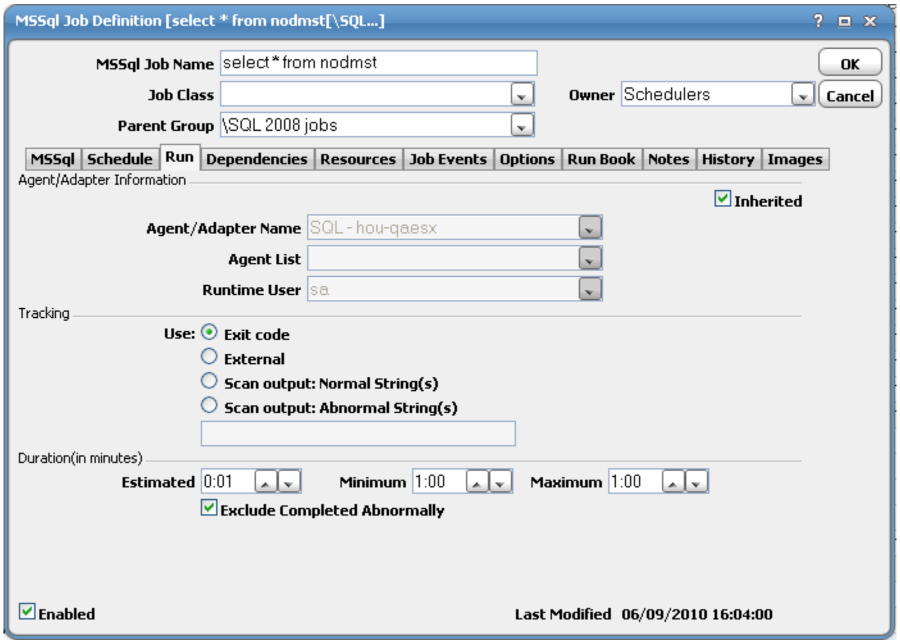
However, unless the parent group has a Microsoft SQL agent assigned to it, you must clear the **Inherited** option and choose an appropriate Microsoft SQL connection.

Selecting the **Add MSSQL Job** option from the Scheduler **Jobs** pane displays the **MSSql Job Definition** dialog.

To add a Microsoft SQL job:

Step 1 From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.

Step 2 Click the **Add** button from the Scheduler toolbar or right-click in either the **Navigator** or **Jobs** pane and select **Add MSSQL Job** from the context menu to display the **MSSQL Job Definition** dialog.



MSSql Job Definition [select * from nodmst[\SQL...]

MSSql Job Name: select * from nodmst
Job Class:
Owner: Schedulers
Parent Group: \SQL 2008 jobs

Agent/Adapter Information

☑ Inherited

Agent/Adapter Name: SQL - hou-qaesx
Agent List:
Runtime User: sa

Tracking

Use: ● Exit code
○ External
○ Scan output: Normal String(s)
○ Scan output: Abnormal String(s)

Duration(in minutes)

Estimated 0:01 Minimum 1:00 Maximum 1:00

☑ Exclude Completed Abnormally

☑ Enabled

Last Modified 06/09/2010 16:04:00

Step 3 In the **MSSql Job Name** field, enter a name up to 50 characters in length for your job.

The Microsoft SQL Job Name is an identifier for Scheduler only. All of the other job definition information, such as Job Class, Owner and Parent Group, is also the same as a standard Scheduler job and is used in the same way.

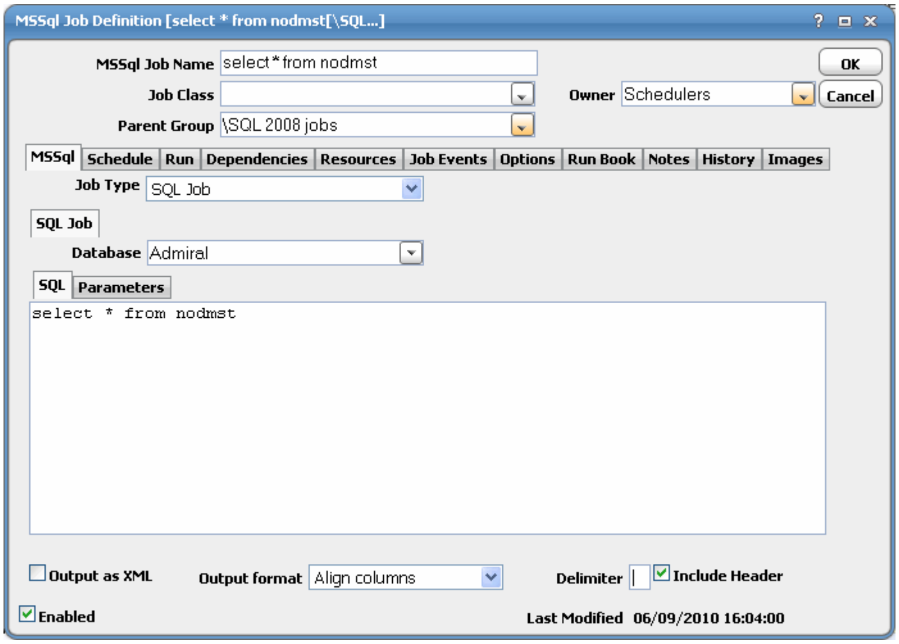
If you are putting your Microsoft SQL job into a group, note that unless the parent group selected has a Microsoft SQL agent connection assigned, you must clear the **Inherited** option on the **Run** tab before you can select a Microsoft SQL connection.

Step 4 On the **Run** tab, select an agent/adapter name from the **Agent/Adapter Name** list.

Step 5 Select a runtime user from the **Runtime User** list.

For information on authorizing a runtime user, see *"Defining Runtime Users (Database Users)"*.

Step 6 Click the **MSSql Job** tab.



MSSql Job Definition [select * from nodmst[\SQL...]

MSSql Job Name: select * from nodmst
Job Class:
Owner: Schedulers
Parent Group: \SQL 2008 jobs
Job Type: SQL Job
Database: Admiral

```
select * from nodmst
```

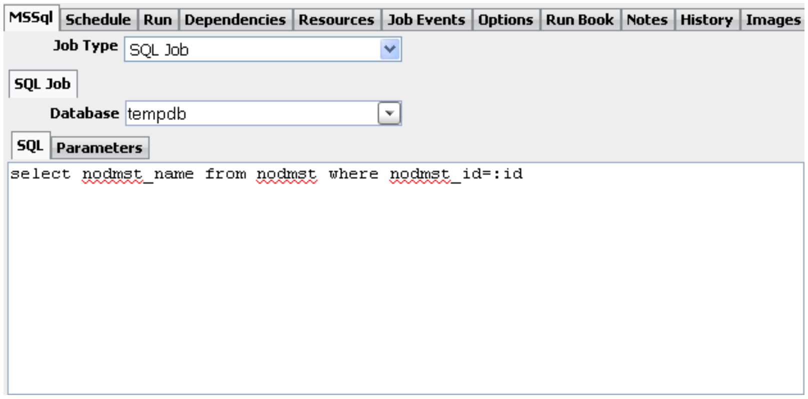
□ Output as XML Output format: Align columns Delimiter: | ☑ Include Header
☑ Enabled Last Modified 06/09/2010 16:04:00

Step 7 Select the job type (**SQL Job**) from the **Type** list.

Step 8 Select the database associated with the job from the **Database** list.

Step 9 In the **SQL** tab field, type in SQL statements to execute here (multiple statements separated by semi-colons).

To include parameters that are replaced at runtime, use a parameter name of your choice preceded by a colon (i.e. **:id**).



Job Type: SQL Job
Database: tempdb

```
select nodmst_name from nodmst where nodmst_id=:id
```

Step 10 Select the **Output as XML** option to write the query results in XML format, if desired.

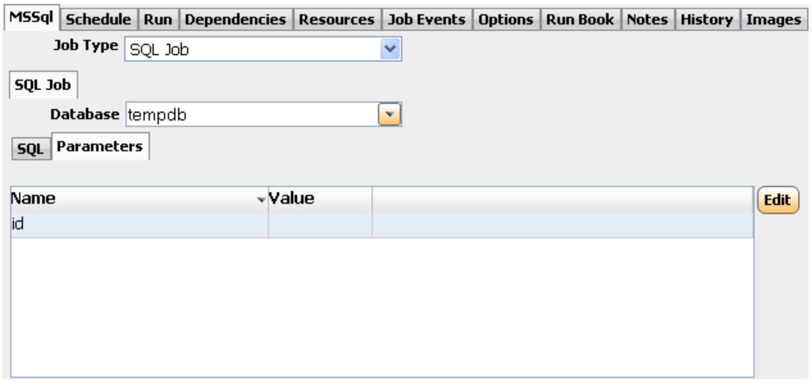
Step 11 In the **Output Format** list, select how the query results are formatted, if **Output as XML** is not selected.

- **Align Columns** – Displays the values in the most readable format.
- **CSV Format** – Separates values with commas.
- **Raw** –Separates values with a user-defined character.

Step 12 In the **Delimiter** field, specify the custom character to use for delimiting the query results, if **Raw** is selected for **Output Format**.

Step 13 Select the **Include Headers** option to write out the column headers of the results as well.

Step 14 If you are using parameters to be replaced at runtime, click the **Parameters** tab view a list of parameters that have been preceded by a colon where you can provide values.

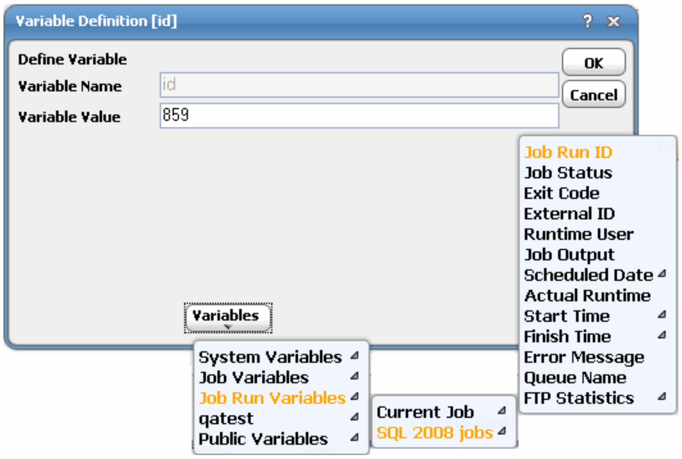


Job Type: SQL Job

Database: tempdb

Name	Value
id	

Step 15 Click **Edit** display the **Variable Definition** dialog.



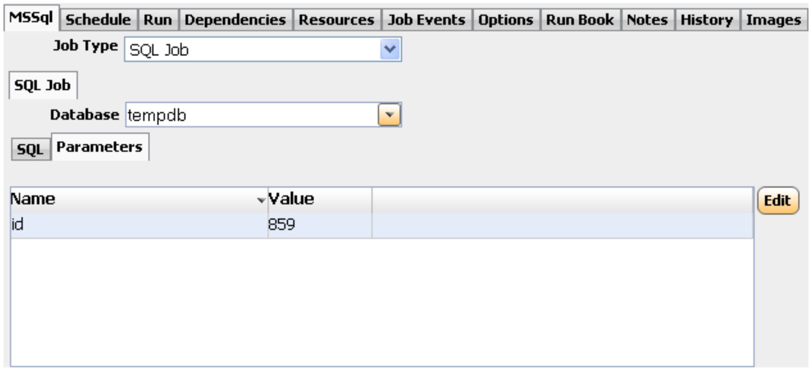
Define Variable

Variable Name: id

Variable Value: 859

Step 16 Enter a parameter value, then click **OK** to save the value.

The value displays in the variable row.



Job Type: SQL Job

Database: tempdb

Name	Value
id	859

Step 17 Click **OK** to save the job.

Adding a Database Job

To add a database job:

Step 1 From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.

Step 2 Click the **Add** button from the Scheduler toolbar or right-click in either the **Navigator** or **Jobs** pane and select **Add MSSQL Job** from the context menu to display the **MSSQL Job Definition** dialog.

Step 3 In the **MSSql Job Name** field, enter a name up to 50 characters in length for your job.

The Microsoft SQL Job Name is an identifier for Scheduler only. All of the other job definition information, such as Job Class, Owner and Parent Group, is also the same as a standard Scheduler job and is used in the same way.

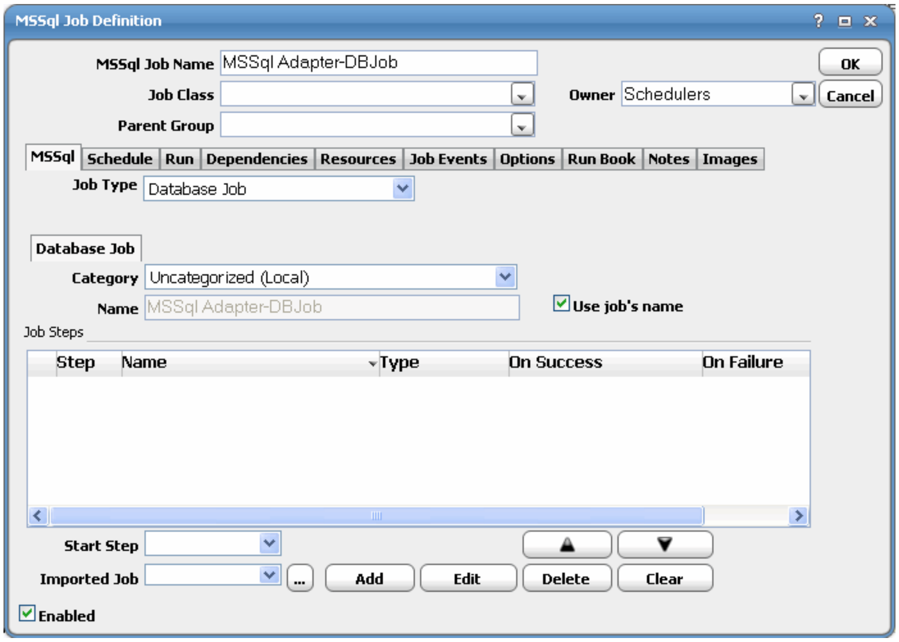
If you are putting your Microsoft SQL job into a group, note that unless the parent group selected has a Microsoft SQL agent connection assigned, you must clear the **Inherited** option on the **Run** tab before you can select a Microsoft SQL connection.

Step 4 On the **Run** tab, select an agent/adapter name from the **Agent/Adapter Name** list.

Step 5 Select a runtime user from the **Runtime User** list.

For information on authorizing a runtime user, see *Defining Runtime Users (Database Users)*.

Step 6 Click the **MSSql Job** tab.



MSSql Job Name: MSSql Adapter-DBJob

Job Class:

Owner: Schedulers

Parent Group:

Job Type: Database Job

Category: Uncategorized (Local)

Name: MSSql Adapter-DBJob ☑ Use job's name

Step	Name	Type	On Success	On Failure

Start Step:

Imported Job:

☑ Enabled

Step 7 Select the job type (**Database Job**) from the **Job Type** list.

Step 8 In the **Name** field, enter the name you want to use to identify the job within SQL Server.

If you would like the SQL job name to match the Scheduler job name, leave this field blank and select the **Use Job's Name** option.

When this option is selected, the **Name** field will be disabled.

Step 9 On the **Database Job** tab, select the SQL Server category to classify job under from the **Category** list.

The default is **Uncategorized**.

Step 10 From the **Start Step** list, select the first step to execute when job is run.

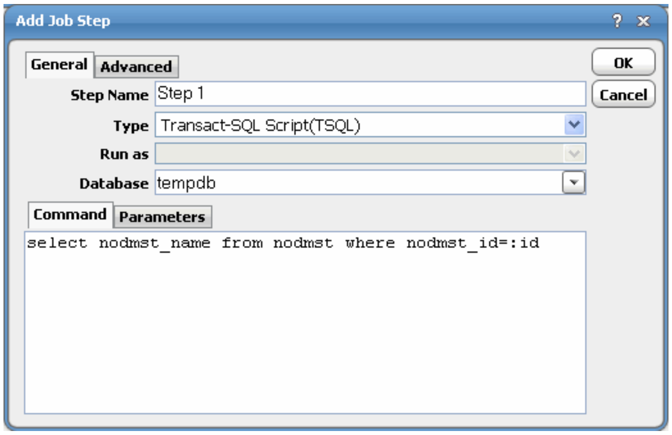
Step 11 Use the **Import Job** field to select a pre-existing job to import as a template.

See "Importing Steps from an Existing Database Job".

Step 12 Select the **Enabled** option to enable/disable job definition.

Step 13 Click **OK**.

Step 14 Click **Add** to display the **Job Step** dialog.



Step Name: Step 1
Type: Transact-SQL Script(TSQL)
Run as:
Database: tempdb

```
select nodmst_name from nodmst where nodmst_id=:id
```

Step 15 Enter a name for the step in the **Step Name** field.

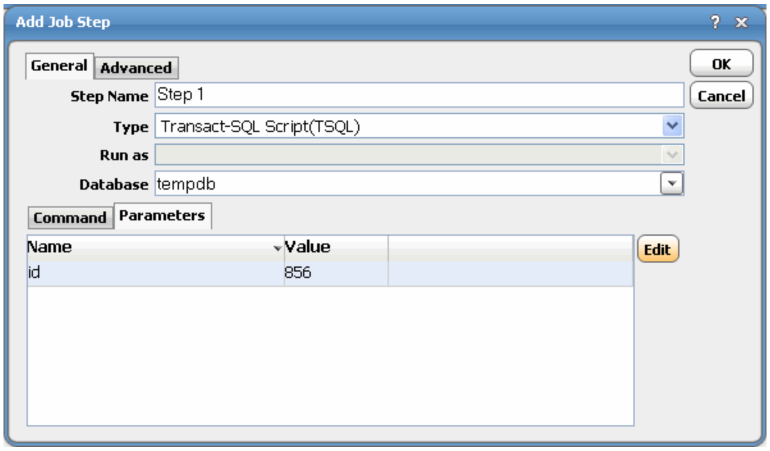
Step 16 Select the type of step from the **Type** field.

Database jobs support a variety of step types, from ActiveX scripts to T-SQL procedures.

Step 17 Select the database associated with the job from the **Database** list.

Step 18 In the **Command** tab field, type in a command to execute.

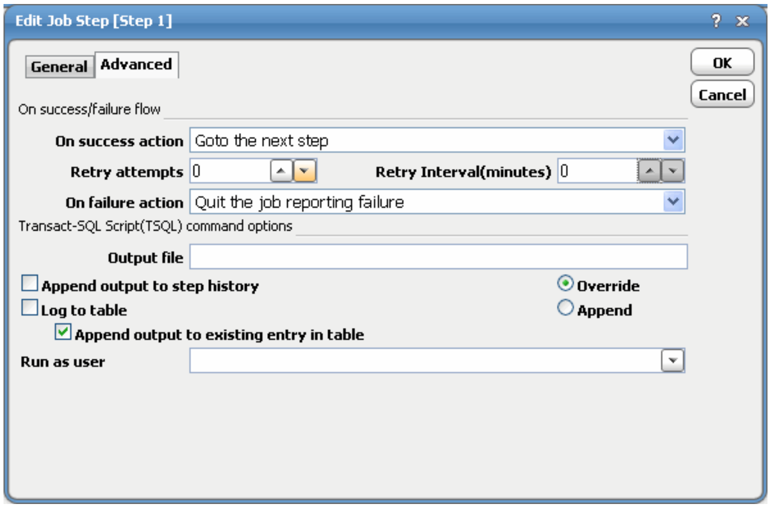
Step 19 If you are using parameters to be replaced at runtime, click the **Parameters** tab to view a list of parameters that have been preceded by a colon where you can provide values.



Step Name: Step 1
Type: Transact-SQL Script(TSQL)
Run as:
Database: tempdb

Name	Value
id	856

Step 20 Click the **Advanced** tab to configure options for this step.



On success/failure flow

On success action: Goto the next step

Retry attempts: 0 Retry Interval(minutes): 0

On failure action: Quit the job reporting failure

Transact-SQL Script(TSQL) command options

Output file:

□ Append output to step history ● Override

□ Log to table ○ Append

☑ Append output to existing entry in table

Run as user:

Note Depending on the option selected from the Type list of the General tab, this dialog displays different options.

Step 21 Select the **Append output to step history** option.

You can also re-direct the output to a local file or log it to a database table.

In the advanced configuration, you can also specify what action to take when a step succeeds or fails. You can stop the entire job, reporting either success or failure. Or, you can jump to any previous or next step. For the last step of the job, you usually want to specify Quit the job reporting success from the **On success action** list.

Step 22 Click **OK** to return to the **MSSql Job** tab.

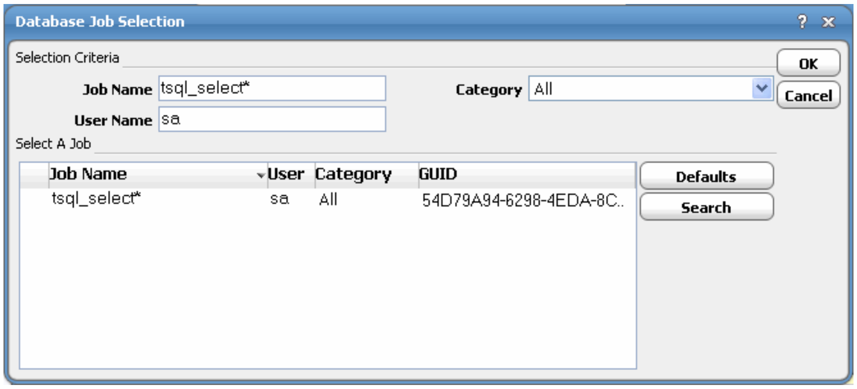
Importing Steps from an Existing Database Job

As an alternative to adding steps one-by-one, you can import steps from an existing Microsoft SQL server job definition.

To import steps:

Step 1 On the **MSSql** tab of the **MSSql Job Definition** dialog, select **Database Job** from the **Type** list.

Step 2 In the **Import Job** field, click the ellipsis to display the **Database Job Selection** dialog.



Selection Criteria

Job Name: tsql_select*

Category: All

User Name: sa

Select A Job

Job Name	User	Category	GUID
tsql_select*	sa	All	54D79A94-6298-4EDA-8C..

Step 3 In the **Select A Job** field, select the job that contains the steps you want to import.

Step 4 Click **OK**.

The steps display on in the **Job Steps** field of the **MSSql Job Definition** dialog.

Step 5 If necessary, you can edit the imported step by clicking the **Edit Step** button to display the **Edit Job Step** dialog and you can add a new step by clicking the **Add Step** button to display the **New Job Step** dialog.

Step 6 Click **OK**.

Linking to an Existing Database Job

To link to an existing database job:

Step 1 From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.

Step 2 Click the **Add** button from the Scheduler toolbar or right-click in either the **Navigator** or **Jobs** pane and select **Add MSSQL Job** from the context menu to display the **MSSQL Job Definition** dialog.

Step 3 In the **MSSql Job Name** field, enter a name up to 50 characters in length for your job.

The Microsoft SQL Job Name is an identifier for Scheduler only. All of the other job definition information, such as Job Class, Owner and Parent Group, is also the same as a standard Scheduler job and is used in the same way.

If you are putting your Microsoft SQL job into a group, note that unless the parent group selected has a Microsoft SQL agent connection assigned, you must clear the **Inherited** option on the **Run** tab before you can select a Microsoft SQL connection.

Step 4 On the **Run** tab, select an agent/adapter name from the **Agent/Adapter Name** list.

Step 5 Select a runtime user from the **Runtime User** list.

For information on authorizing a runtime user, see *"Defining Runtime Users (Database Users)"*.

Step 6 Click the **MSSql Job** tab.

Step 7 Select the job type (**Existing Database Job**) from the **Type** list.

Step 8 In the **Job** field, click the ellipsis to create a link to an existing job.

The **Database Job Selection** dialog displays.

Step 9 From the **Results** section, select the job.

Step 10 Click **OK** to return to the **MSSql Job** tab where the select job steps are displayed in the **Job Steps** field.

Step 11 From the **Start Step** field, select the first step to execute when the job is run.

Step 12 If necessary, you can edit this step by clicking **Edit** or you can add a new step by clicking **Add** to display the **Job Step** dialog.

Step 13 If adding a step, enter a name for the step in the **Step Name** field.

Step 14 Select the type of step from the **Type** field.

Database jobs support a variety of step types, from ActiveX scripts to TSQL procedures.

Step 15 Select the database associated with the job from the **Database** list.

Step 16 In the **Command** tab field, type in a command to execute.

Step 17 Click the **Parameters** tab to enter variables inside the query for parameter replacement.

Step 18 Click the **Advanced** tab to collect output for this step.

Note Depending on the option selected from the On success action list, this dialog displays different options.

Step 19 Select the **Append output to step history** option.

You can also re-direct the output to a local file or log it to a database table.

In the advanced configuration, you can specify what action to take when a step succeeds or fails. You can stop the entire job, reporting either success or failure.

Or, you can jump to any previous or next step. For the last step of the job, you usually want to specify Quit the job reporting success from the **On success action** list.

Step 20 Click **OK** to return to the MSSql Job tab.

Step 21 Click **OK** to save the job.

CHAPTER 4

Working with Microsoft SQL Jobs

Overview

This chapter describes how to work with Microsoft SQL jobs in TES.

- Understanding Microsoft SQL Job Output
- Controlling Adapter and Agent Jobs

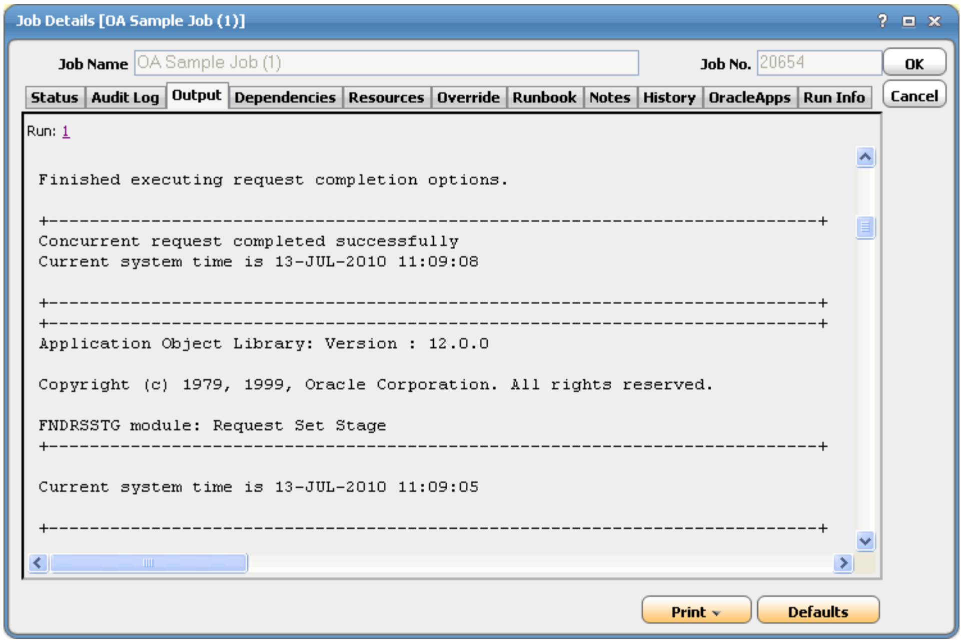
Understanding Microsoft SQL Job Output

The **Job Details** dialog displays by double-clicking on a job instance record in the **Job Activity** pane or by right-clicking and selecting the **Details** option from the context menu. The **Job Detail** dialog provides information on the job after it has completed or as it is still running. The tabs of this dialog specific to the Microsoft SQL Server Adapter are the **Output**, **MSSQL**, and **Run Info** tabs.

Output Tab

The **Output** tab of the **Job Detail** dialog, if job is configured to save output, the Microsoft SQL logs generated for the job and any output if available. Scheduler can be configured to save or discard job output by default from the **Defaults** tab of the **System Configuration** dialog. Regardless of the system default, any individual job instance can be configured from its job definition to override the system default. Each time a job is rerun that run's output is separated by a block of number signs (#).

Note Scheduler's default is to discard job output. If you want to be able to view job output, you must select the Save Output option on the Options tab in the Job Definition dialog or change the system default on the Defaults tab in the System Configuration dialog.



Job Name: OA Sample Job (1)

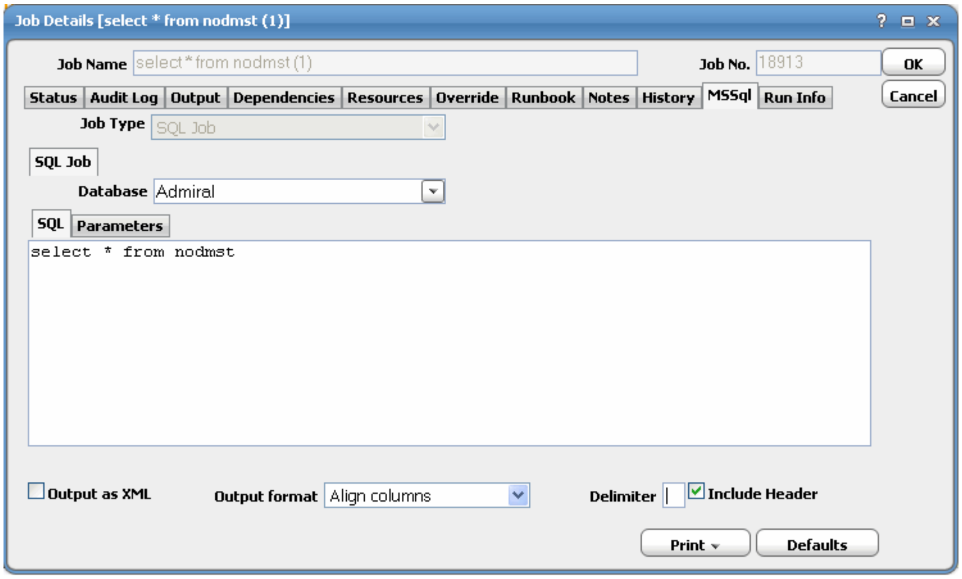
Job No.: 20654

Run: 1

```
Finished executing request completion options.

+---------------------------------------------------------------------------+
Concurrent request completed successfully
Current system time is 13-JUL-2010 11:09:08

+---------------------------------------------------------------------------+
+---------------------------------------------------------------------------+
Application Object Library: Version : 12.0.0

Copyright (c) 1979, 1999, Oracle Corporation. All rights reserved.

FNDRSSTG module: Request Set Stage
+---------------------------------------------------------------------------+

Current system time is 13-JUL-2010 11:09:05

+---------------------------------------------------------------------------+
```

MSSql Tab

The **MSSql** tab of the **Job Detail** dialog contains the request with the variables used when this job was submitted. This tab allows you to override the parameter values listed.



Job Name: select * from nodmst (1)

Job No.: 18913

Job Type: SQL Job

Database: Admiral

```
select * from nodmst
```

□ Output as XML

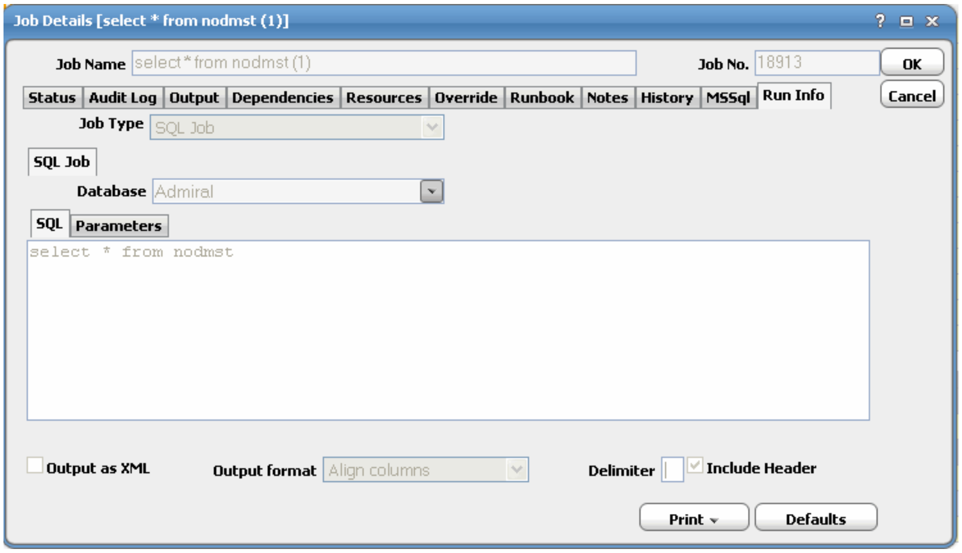
Output format: Align columns

Delimiter: |

☑ Include Header

Run Info Tab

The **Run Info** tab of the **Job Detail** dialog contains the request that was submitted to Microsoft SQL. Each tab reflects the last run of this Microsoft SQL instance. This may or may not be the same thing you see on the **MSSQL** tab depending on whether you have made any changes to this instance since the last run. This tab is read-only.


Job Details [select * from nodmst (1)]

Job Name: select * from nodmst (1)

Job No. 18913

Job Type: SQL Job

Database: Admiral

```
select * from nodmst
```

□ Output as XML

Output format: Align columns

Delimiter:

☑ Include Header

Controlling Adapter and Agent Jobs

Scheduler provides the following job control capabilities for either the process currently running or the job as a whole:

- Holding a Job—Hold a job waiting to run.
- Aborting a Job—Abort an active job.
- Rerunning a Job—Rerun a job that completed.
- Making One Time Changes to an Adapter or Agent Job Instance—Make last minute changes to a job.
- Deleting a Job Instance before It Has Run—Delete a job instance before it has run.

Holding a Job

Adapter/agent jobs are held in the same way as any other Scheduler jobs.

Adapter/agent jobs can only be held before they are launched. Once a job reaches the Adapter/Agent system, it cannot be held or suspended.

To hold a job:

Step 1 From the **Job Activity** pane, right-click on the job.

Step 2 Select **Job Control>Hold/Stop**.

Aborting a Job

Adapter/agent jobs are aborted in the same way as any other Scheduler jobs.

To abort a job:

Step 1 From the **Job Activity** pane, right-click on the job.

Step 2 Select **Job Control>Cancel/Abort**.

Rerunning a Job

On occasion, you may need to rerun an Adapter/Agent job. You can override parameter values first, if necessary, from the Adapter/Agent tab.

To rerun a job:

Step 1 From the **Job Activity** pane, right-click the Adapter/Agent job you need to rerun.

Step 2 Select **Job Control>Rerun** option from the context menu.

Making One Time Changes to an Adapter or Agent Job Instance

Prior to a run or rerun, you can edit data on the specific **Adapter/Agent** tab. To ensure that there is an opportunity to edit the job prior to its run, you can set the **Require operator release** option on the **Options** tab in the Adapter **Job Definition** dialog. Use this function to make changes to an Adapter job after it enters Waiting on Operator status as described in the following procedure.

To make last minute changes:

Step 1 From the **Job Activity** pane, double-click the Adapter/Agent job to display the **Job Details** dialog.

Step 2 Click the Adapter tab.

Step 3 Make the desired changes to the job and click **OK** to close the **Job Details** dialog.

Step 4 If this job is Waiting on Operator, perform one of the following tasks:

- To release the job, select **Job Control->Release**.
- To rerun the job with changes, select **Job Control->Rerun**.

Deleting a Job Instance before It Has Run

Adapter/Agent job instances are deleted in the same way as any other Scheduler job.

Deleting a job from the **Job Activity** pane removes the job from the Scheduler job activity only. The original definition is left in tact.

To delete a job instance:

Step 1 From the **Job Activity** pane, right-click the Adapter/Agent job to be deleted.

Step 2 Select **Remove Job(s) From Schedule**.

CHAPTER 5

Working with Microsoft SQL Events

Overview

You can add new events that are triggered on Windows SQL Server for table, index, and row creation, modification and deletion.

This chapter describes how to work with Microsoft SQL events in TES.

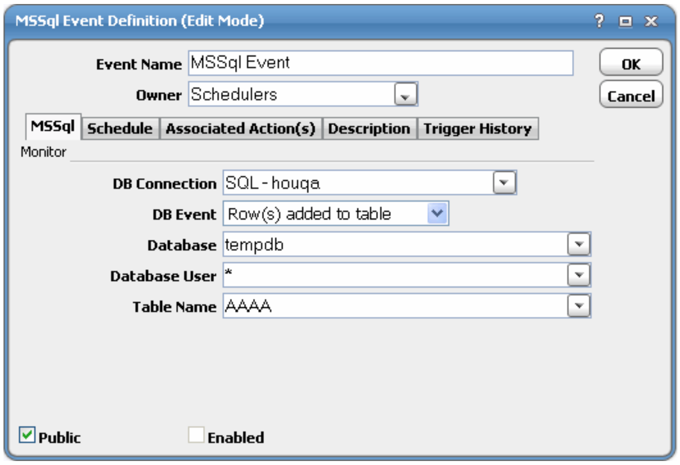
- Adding a Microsoft SQL Event
- Assigning an Alert Action to an Event

Adding a Microsoft SQL Event

To add an event:

Step 1 From the **Navigator** pane, select **Events>MSSql Events** to display the **MSSql Events** pane.

Step 2 Click the **Add** button or right-click on the **MSSql Events** pane and select **Add Event** from the context menu to display the **MSSql Event Definition** dialog.



MSSql Event Definition (Edit Mode)

Event Name: MSSql Event

Owner: Schedulers

Monitor

DB Connection: SQL - houqa

DB Event: Row(s) added to table

Database: tempdb

Database User: *

Table Name: AAAA

☑ Public

□ Enabled

Step 3 In the **Event Name** field, enter a name for the event.

Step 4 From the **Owner** list, select the owner of the event.

Step 5 On the **MSSql** tab, specify which database connection to monitor from the list of defined connections from the DB Connection list.

This tab designates the data table and indexes in the Microsoft SQL database to be monitored and defines the condition that will raise the event. Once the designated change to the data table is detected, the event triggers any linked action(s) as an automatic response to the change.

Step 6 From the **DB Connection** list, specify which database connection to monitor from the list of defined connections.

Step 7 From the **DB Event** list, designate one of the listed database changes from the list as the event trigger.

Step 8 From the **Database** list, specify the database to be monitored.

Initially this text field defaults to the database in the Connection Definition.

Step 9 From the **Database User** list, select the name of the user making the change in the database.

Initially this text field defaults to the user in the Connection Definition.

The value entered here must be all uppercase with without any spaces. The wildcard characters ***** and **?** can be used in this text field.

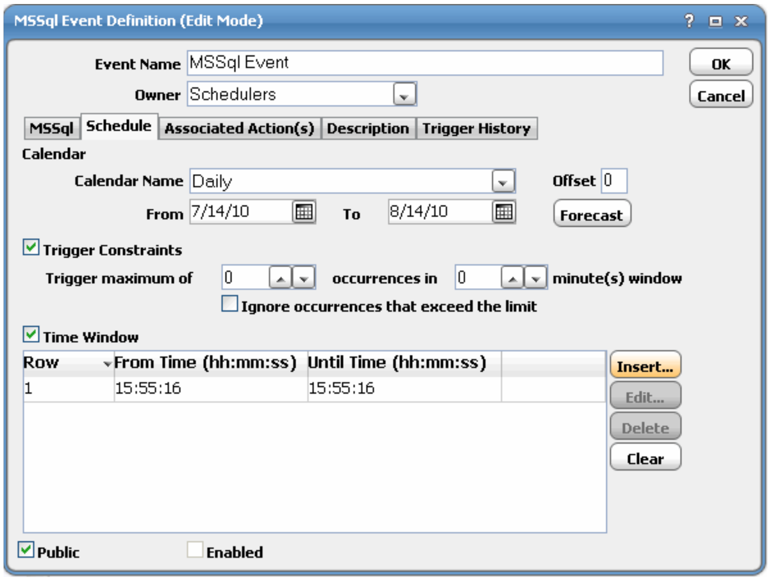
Step 10 From the **Table Name** list, select the name of the data table to be monitored.

The wildcard characters ***** and **?** can be used in this text field, but only if the database event being monitored is one of the following:

- Table created
- Table deleted
- Index created
- Index deleted

All other database events require a specific table name. The value entered here must be all uppercase without any spaces.

Step 11 Click the **Schedule** tab to assign a schedule to the event.



MSSql Event Definition (Edit Mode)

Event Name: MSSql Event

Owner: Schedulers

Calendar

Calendar Name: Daily Offset: 0

From: 7/14/10 To: 8/14/10

☑ Trigger Constraints

Trigger maximum of 0 occurrences in 0 minute(s) window

□ Ignore occurrences that exceed the limit

☑ Time Window

Row	From Time (hh:mm:ss)	Until Time (hh:mm:ss)
1	15:55:16	15:55:16

☑ Public □ Enabled

Step 12 From the **Calendar Name** field, select the calendar or calendar group indicating which days the event monitor should run.

You must select a calendar for the event monitor to be scheduled automatically. A calendar can encompass one or more days. If you do not select a calendar than the event monitor will run every day.

You can clear a calendar by right-clicking on the **Calendar** field and selecting **Clear** from the context menu or by pressing the **Delete** key.

Step 13 In the **Offset** field, enter the number of days before or after every day in the calendar that the event will be monitored.

Each day in the calendar is adjusted by the offset you provide. For example, if the event monitor is using the Fiscal Month End calendar with an offset of 2, the event monitor will be inserted into the schedule 2 days after Fiscal Month End date. If you use an offset of -2, the event monitor will be inserted 2 days before the Fiscal Month End date.

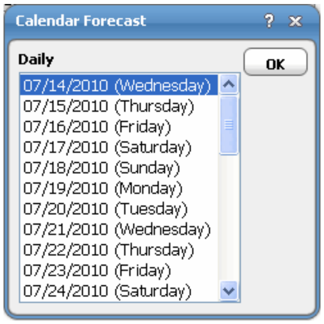
Step 14 In the **From** field, select the earliest date that the event monitor should be active.

The event monitor will operate starting from the first date in the calendar that is on or after this date. You can choose a date using the built-in calendar dialog by clicking the adjacent drop-down arrow.

Step 15 In the **To** field, select the last date the event monitor can operate based on the calendar.

If not specified, and your calendar includes repetitive dates, your event monitor will run indefinitely. You can choose a date using the built-in calendar dialog by clicking the adjacent drop-down arrow.

Step 16 Click **Forecast** to display the **Calendar Forecast** dialog for the selected calendar (adjusted by offset, if provided) showing all the dates for which the event monitor will be active.



Daily

07/14/2010 (Wednesday)
07/15/2010 (Thursday)
07/16/2010 (Friday)
07/17/2010 (Saturday)
07/18/2010 (Sunday)
07/19/2010 (Monday)
07/20/2010 (Tuesday)
07/21/2010 (Wednesday)
07/22/2010 (Thursday)
07/23/2010 (Friday)
07/24/2010 (Saturday)

Step 17 On the **Schedule** tab, select the **Trigger Constraints** option.

The **Trigger Constraints** section defines constraints on actions associated with the event monitor. The constraints allow you to throttle the execution of actions. At times, the event condition we are searching for may trigger in rapid successions. Rather than act on each and every event in the short interval, it may be preferable to limit the action execution frequency.

Note While the event may be suppressed from occurring each time the event conditions are detected, the event monitor will still perform the associated action each time it detects the specified condition. The configured action always occurs whenever an event condition is detected whether the event "counts" or not.

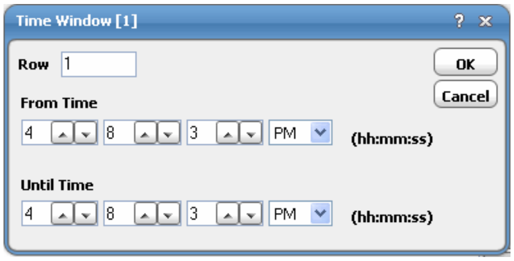
Step 18 In the **Trigger maximum of** field, specify the maximum number of times that the event monitor will react to the defined conditions within a certain time window.

Step 19 In the **Occurrences in __ minute (s)** field, designate the time window within which the event monitor will react to the defined conditions.

Step 20 Select the **Ignore occurrences that exceed the limit** option if you want to ignore any occurrences that exceed configured value.

Step 21 Select the **Time Window** option.

Selecting this option will limit the event monitor to being active only during the defined time windows that are listed here. If this option is not selected then the event monitor will always be active. This section lists multiple time windows that the event monitor will monitor for the defined conditions. Any number of time windows can be created via the **Time Window** dialog.



Time Window [1]

Row: 1

From Time: 4 8 3 PM (hh:mm:ss)

Until Time: 4 8 3 PM (hh:mm:ss)

- **Row** – A sequence number for the defined time window.
- **From Time (hh:mm:ss)** – Designates the time that the event monitor will begin actively monitoring for the defined conditions. This is expressed in a 24 hour clock format of hours, minutes and seconds.
- **Until Time (hh:mm:ss)** – Designates the time that the event monitor will stop actively monitoring for the defined conditions. This is expressed in a 24 hour clock format of hours, minutes and seconds.

Step 22 Click **OK** to add the new event.

Assigning an Alert Action to an Event

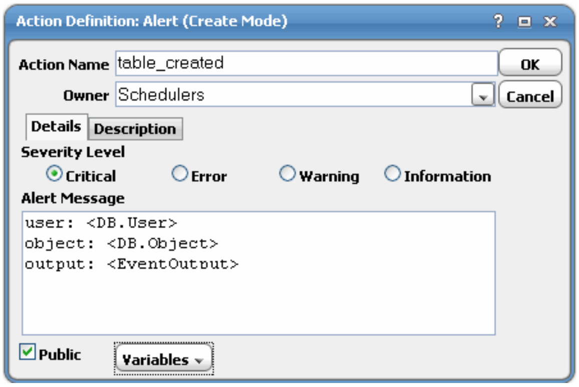
You can assign an Alert action to an event so you know when the event has triggered. You can use the following Microsoft SQL event variables:

- **DB.User** – The database user who triggered the event.
- **DB.Object** – The database object that triggered the event.
- **EventOutput** –The output of the event selected.

To assign an Alert action:

Step 1 In the **Navigator** pane, select **Definitions>Actions>Alert** to display the **Alerts** pane.

Step 2 Click the **Add** button or right-click on the **Alerts** pane and select **Add Action** from the context menu to display the **Action Definition** dialog.



Action Name table_created

Owner Schedulers

Severity Level

● Critical ○ Error ○ Warning ○ Information

Alert Message

```
user: <DB.User>
object: <DB.Object>
output: <EventOutput>
```

☑ Public

Step 3 In the **Action Name** field, enter a name for the alert.

Step 4 From the **Owner** list, select the owner name.

Step 5 In the **Severity Level** section, select the severity level of the alert.

You can select from four different severity levels:

- **Critical** – Sends the highest priority alert to the **Job Activity** pane when the action is triggered.
- **Error** – Sends a high priority alert to the Job Activity pane when the action is triggered.
- **Warning** – Sends a medium priority alert to the Job Activity pane when the action is triggered.
- **Information** – Sends the lowest priority alert to the Job Activity pane when the action is triggered.

Step 6 In the **Alert Message** field, enter the message the operator receives when viewing the alert from the **Alert Details** dialog.

You can enter a combination of text and/or variables in your message. Click the **Variable** button to choose variables from the variables context menu.

Step 7 Select the **Public** option.

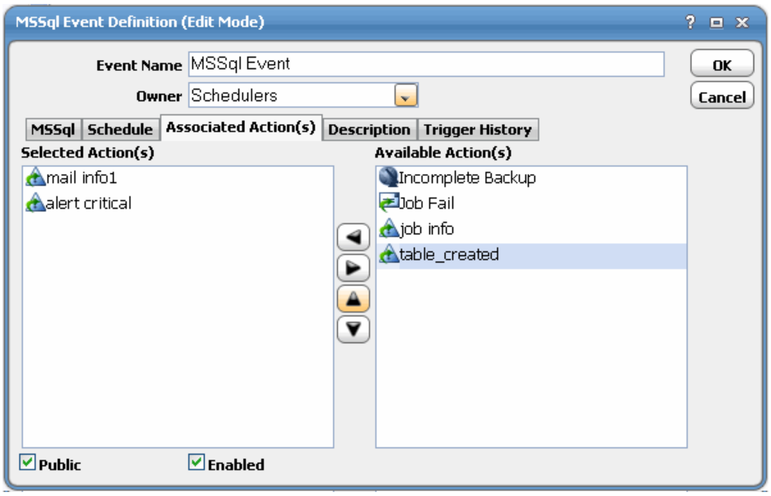
When selected, the action is available to all Scheduler users (within the constraints of their security policy). You can select the default condition for this option from the **Defaults** tab of the **System Configuration** dialog.

Step 8 Click **OK**.

Step 9 From the **Navigator** pane, select **Events>MSSql Events** to display the **MSSql Events** pane.

Step 10 Double-click on the event that you would like to assign the **Alert** action to display the **MSSql Event Definition** dialog.

Step 11 Click the **Associated Action(s)** tab.



Event Name: MSSql Event

Owner: Schedulers

Selected Action(s):
- mail info1
- alert critical

Available Action(s):
- Incomplete Backup
- Job Fail
- job info
- table_created

☑ Public ☑ Enabled

Step 12 In the **Available Action(s)** field, select the appropriate alert.

Note You can also add an alert from this field by right-clicking and selecting an option from the context menu.

Step 13 Click the left arrow button to move the event into the **Selected Action(s)** field.

Step 14 Click **OK**.

Configuring service.props

About Configuring service.props

The **service.props** file is used to configure adapter behavior. **service.props** is located in the \config directory located under the Adapter's GUID directory, You can create both the directory and file if it does not yet exist. Properties that can be specified in service.props control things like logging and connection configuration. Many of the properties are specific to certain adapters; others are common across all adapters.

service.props Properties

The table below lists many of the parameters that can be specified in service.props. Some properties apply to all adapters (shaded in the table) and some properties are adapter-specific as indicated by the **Applicable Adapter(s)** column. The properties are listed in alphabetical order.

Property	Applicable Adapter(s)	Default	What It Controls
BYPASS_SEC_VALIDATION	Oracle Apps	N	If set to Y, the secondary user validation is bypassed. If not, secondary user validation is performed.
CLASSPATH	All	<none>	(Optional) – The path to the JDBC driver. If the default CLASSPATH used when the Adapter process is started does not include an appropriate JDBC driver jar required to connect to the PowerCenter Repository Database, you will need to specify this *service.props* configuration
CONN_SYNC	All	N	Setting this flag to Y allows synchronous connections without overloading the RDOnly Thread. If set to N, the adapter might stop trying to reconnect after an outage or downtime.
DISCONN_ON_LOSTCONN	Informatica	N	Setting this flag to Y avoids an unnecessary logout call to the Informatica server when the connection is lost. This logout call usually hangs.

Property	Applicable Adapter(s)	Default	What It Controls
EnableDynamicPollingInterval	All	N	Use to avoid frequent polling on long-running jobs. When set to Y in service.props of a particular adapter, these properties are enabled: MinDynamicPollInterval—Minimum value should be 5 seconds. MaxDynamicPollIntervalInMin—Maximum value should be 5 minutes. PercentOfEstDuration—Default value is 5.
IGNORE_CODES	Informatica	<none>	This parameter can be set in service.props, job configuration and connection configuration parameters. The order of precedence is service.props (applicable for all jobs running in all connections), job level (only for that particular job), and connection (applicable for all jobs in the connection). This parameter is used to specify Informatica-specific error codes, separated by commas (,), that you want to ignore while running a job.
IGNORESUBREQ	Oracle Apps	N	Y or N. Setting this flag to Y stops huge job xml file transfers back and forth between the adapter and the AdapterHost during polls when a single request set has multiple sub-requests of more than 100. The default value is N or empty.
jarlib	Hive and MapReduce	<none>	Specifies the specific Java library to use for the adapter: • For Apache 1.1.2, add: **`jarlib=apache1.1.2`** • For Cloudera 3, add: **`jarlib=cloudera`** • For Cloudera 4, add: **`jarlib=cdh4`** • For MapR add: **`jarlib=apache1.1.2`**
kerbkdc	MapReduce	<none>	If the Hadoop cluster is Kerberos secured, use this value to specify the KDC Server. For example, **`kerbkdc=172.25.6.112`**
kerbrealm	MapReduce	<none>	If the Hadoop cluster is Kerberos secured, use this value to specify the Kerberos Realm. For example, **`kerbrealm=TIDALSOFT.LOCAL`**

Property	Applicable Adapter(s)	Default	What It Controls
Keystore	BusinessObjects, BusinessObjects BI, BusinessObjects DS, Cognos, JD Edwards, Oracle Applications, UCS Manager, VMware, Web Service	<none>	Specify `Keystore=c:\\<adapter_certificate_directory>\\<your_trusted_keystore>.keystore` when importing certificates into a Java keystore.
LAUNCH_DELAY (in milliseconds)	Informatica	<none>	This parameter can be set in service.props, job configuration and connection configuration parameters. The order of precedence is service.props (applicable for all jobs running in all connections), job level (only for that particular job), and connection (applicable for all jobs in the connection). If a non-zero value is set for this parameter, then the jobs are delayed for the specified number of milliseconds before being submitted to Informatica.
LoginConfig	BusinessObjects BI Platform, BusinessObjects Data Services	<none>	Specifies the location of the login configuration if using WinAD or LDAP authentication. For example: `LoginConfig=c:\\windows\\bscLogin.conf` where "`c:\\windows\\bscLogin.conf`" is the location of the login configuration information. Note the use of \\ if this is a Windows location.
MaxLogFiles	Informatica, JDBC	50	(Optional) – Number of logs to retain.
OUTPUT_ASYNC_LOGOUT	Informatica	N	Setting this flag to Y avoids jobs getting stuck in Gathering Output status.
OUTPUT_SYNC	All	Y	Enables concurrent output gathering on a connection. To enable this feature, set the value to N.
POLL_SYNC	All	Y	Enables concurrent polling on connections of the same type. This is helpful when there is a heavily load on one connection of an adapter. The heavily loaded connection will not affect the other adapter connection. To enable this feature, set the value to N.
QUERY_TIMEOUT	Oracle Apps	N	Y or N. If set to Y, the timeout value defined using the parameter QUERY_TIMEOUT_VALUE is applied to the SQL queries. Default value is N or empty.

Property	Applicable Adapter(s)	Default	What It Controls
QUERY_TIMEOUT_VALUE	Oracle Apps	unset	The time period in seconds that SQL queries wait before timeout. If 0 or not set, there is no timeout.
READPCHAINLOG	SAP	Y	Used to control the log gathering in SAP Process Chain jobs. This property depends on the Summary Only check box of the job definition Options tab.
SCANFOR_SESSIONSTATS	Informatica	Y	Y or N - Set this parameter to N to turn off the default behavior of Informatica jobs collecting the session statistics during the job run.
SCANFOR_SESSIONSTATS_AFTER_WF_ENDS	Informatica	N	Y or N - Set this parameter to Y to turn off the gathering of session statistics during each poll for the status of Informatica jobs.
TDLINFA_LOCALE	Informatica	<none>	Points to the Load Manager Library locale directory. See "Configuring the Informatica Adapter" in the *Informatica Adapter Guide* for how to set this for Windows and Unix environments.
TDLINFA_REQUESTTIMEOUT	Informatica	<none>	(Optional) – The number of seconds before an API request times out. The default is 120 seconds, if not specified.
TDLJDBC_LIBPATH	JDBC	<none>	(Windows only, optional) An alternate path to the JDBC library files. The library file path should have been configured given system environment variables. This option is available in case you wish to use an alternate set of libraries and may be helpful for trouble-shooting purposes.
TDLJDBC_LOCALE	JDBC	<none>	The path to the JDBC locale files.
TRANSACTION_LOG_BATCH_SIZE	MS SQL	5000	Set this parameter if more than 5000 lines need to be read from the transaction table.
version_pre898	JD Edwards	N	If running on a JD Edwards server version that is less than 8.9.8, set version_pre898=Y.