



Cisco Workload Automation JDBC Adapter Guide

Version 6.3.2

First Published: October 2017

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies are considered un-Controlled copies and the original on-line version should be referred to for latest version.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

© 2017 Cisco Systems, Inc. All rights reserved.

Contents 3

Preface 5

- Audience 5
- Related Documentation 5
- Obtaining Documentation and Submitting a Service Request 5
- Document Change History 5

Introducing the JDBC Adapter 7

- Overview 7
- Requirements 7

Configuring the JDBC Adapter 9

- Overview 9
- Licensing an Adapter 9
- Configuring the JDBC Adapter 10
- Securing the JDBC Adapter 12
 - Defining Runtime Users (Database Users) 12
 - Authorizing Schedulers to Work With JDBC Jobs and Events 13
 - Defining a JDBC Security Policy 13
 - Defining a CWA User of JDBC Jobs and Events 15
- Defining a Connection Using JDBC 16
 - Adding a Connection Using JDBC 17
 - JDBC Connection Status Light 20

Defining JDBC Jobs 21

- Overview 21
- Adding a JDBC SQL Job 21
- Adding a JDBC Stored Procedure Job 25

Working with JDBC Jobs 29

- Overview 29
- Understanding JDBC Job Runtime Details 29
 - Output Tab 29
 - JDBC Tab 30
 - Run Info Tab 31
- Controlling Adapter and Agent Jobs 32
 - Holding a Job 32
 - Aborting a Job 32
 - Rerunning a Job 32
 - Making One Time Changes to an Adapter or Agent Job Instance 32
 - Deleting a Job Instance before It Has Run 33

33

Working with JDBC Events 35

- Overview 35
- Adding a JDBC Event 35
- Assigning an Action to an Event 40

Configuring service.props 43

- About Configuring service.props 43
- service.props Properties 43



Preface

This guide describes the installation, configuration, and usage of the JDBC Adapter with Cisco Workload Automation (CWA).

Audience

This guide is for administrators who install and configure the JDBC Adapter for use with CWA, and who troubleshoot CWA installation and requirements issues.

Related Documentation

See the *Cisco Workload Automation Documentation Overview* for your release on [cisco.com](http://www.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 CWA guides.

Note: We sometimes update the documentation after original publication. Therefore, you should also review the documentation on [Cisco.com](http://www.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 Cognos 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 <i>TES Installation Guide</i> ; usage provided in online Help only.

Document Change History

Version Number	Issue Date	Reason for Change
6.2.1 SP3	May 2016	Consolidated all JDBC Adapter documentation into one document.
6.3	August 2016	<ul style="list-style-type: none">■ Rebranded “Cisco Tidal Enterprise Scheduler (TES)” to “Cisco Workload Automation (CWA)”.■ Miscellaneous edits for the 6.3 release.
6.3.2	Oct 2017	<ul style="list-style-type: none">■ Added Runtime user definition for Azure SQL database connection.■ Added configuration parameter for Azure SQL database connection.



1

Introducing the JDBC Adapter

This chapter provides an overview of the JDBC Adapter and its requirements:

- [Overview, page 7](#)
- [Requirements, page 7](#)

Overview

The Cisco Workload Automation Adapter for JDBC provides automation of database operations as part of cross platform process organization between CWA and the enterprise. The consolidation of scheduling automation is accomplished as follows:

- Designed using the same user interface approach as other types of jobs, seamlessly integrating JDBC data management into existing operational processes.
- Integrated security management features of the CWA can be applied to the implementation of connections using JDBC and associated jobs to ensure secure management and use.

CWA's centralized console brings complex scheduling functions to an existing database management system. Users can define, launch, control, and monitor SQL commands, and stored procedures submitted to a database via JDBC on a scheduled basis. The JDBC adapter also provides easy to use access for connectivity setup, job definition, and monitoring.

The JDBC adapter includes the following features:

- Connection management to monitor system status with a live connection to the database via JDBC.
- JDBC job and event management includes the following:
 - Schedule and monitor SQL queries and stored procedures from a centralized work console with CWA
 - Provide dynamic runtime overrides for SQL parameters and values passed to stored procedures
 - Provide output formatting options to control the results, including table, XML, and CSV
 - Associate defined dependencies and events with CWA for scheduling control
 - Provide job execution control including terminating, holding, and releasing scheduled SQL

Requirements

Refer to your *Cisco Workload Automation User Guide* for a complete list of hardware and software requirements.

The JDBC driver(s) for the target database must be installed on the CWA master server with a reference in the system CLASSPATH. Refer to the documentation for the driver that you are using.

Requirements

Note: If you change the CLASSPATH environment variable to include a reference to a JDBC library, you will need to restart the master or the master system if the master runs under the Local System account (on Windows).



2

Configuring the JDBC Adapter

Overview

The JDBC adapter software is installed as part of a normal installation of CWA. However, you must perform the following steps to license, configure, secure, and deploy the adapter before you can run JDBC jobs:

- [Licensing an Adapter, page 9](#) - License the connection(s) using JDBC. You cannot define a connection using JDBC until you have applied the JDBC license from Cisco.
- [Configuring the JDBC Adapter, page 10](#) - Configure the JDBC libraries for use with CWA. Also describes some configuration properties that can be defined in `service.props`.

See also [Configuring `service.props`, page 43](#) for additional information about general and JDBC-specific properties that can be set to control things like logging and connection properties.

- [Securing the JDBC Adapter, page 12](#) - Create one or more user definitions in CWA with access to the database that you are connecting to via JDBC using user names and passwords provided by the database administrator. Users in CWA are authorized to run JDBC jobs on behalf of these database runtime users.
- [Defining a Connection Using JDBC, page 16](#) - Define connections using JDBC so that the master can communicate with your databases.

Licensing an Adapter

Each CWA 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 CWA, 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:

1. Stop the master:

Windows:

- a. Click on **Start** and select **All Programs>Cisco Workload Automation>Scheduler>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**

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.

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

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

Configuring the JDBC Adapter

To configure the adapter:

1. Stop the master:

For Windows:

- a.** Click the **Start** button and then select **All Programs>Cisco Workload Automation>Scheduler>Service Control Manager**.
- b.** In the Service list, verify that the master is displayed and click **Stop** to stop the master.

For Unix:

- a.** Stop the master by entering **tesm stop**.
- b.** Verify the master is stopped by entering **tesm status**.

The Master deploys the *jdbc.pkg* file and moves it from the Master's **config** folder to the Adapter's **GUID** (**{68597802-1828-464D-BEDC-2B6E1D3C8C0C}**) folder located under the Master's **Services** folder.

For example:

For Windows:

```
C:\Program Files\TIDAL\Scheduler\master\services\{68597802-1828-464D-BEDC-2B6E1D3C8C0C}
```

For Unix:

```
/opt/TIDAL/Scheduler/master/services/{68597802-1828-464D-BEDC-2B6E1D3C8C0C}
```

To Install and configure the JDBC Libraries:

1. For connectivity to the database, you need access to the JDBC Drivers.

Note: You must enable High Debug to view Adapter startup information in the AdapterHost log files. Refer to the “Diagnostic Logging for the Master” in the Troubleshooting chapter in the *Cisco Workload Automation User Guide* for setting debug levels.

The JDBC Adapter includes the Master's CLASSPATH, which contains references to the JDBC driver jars for operation with both MSSQL and SQL Server databases. These drivers are NOT distributed as part of the Master installation but are usually sufficient for connectivity to either SQL Server (*sqljdbc4.jar*) or Oracle Databases (*ojdbc7.jar*).

Note: JDBC adapter does not ship any JDBC drivers. Download the drivers from the JDBC driver website and specify the driver path as follows:

1. Create a config directory under `/master/services/{68597802-1828-464D-BEDC-2B6E1D3C8C0C}`
2. Create `service.props` under config directory.
3. In `service.props`, specify the driver path following the example:

In windows:

```
CLASSPATH=c:\\mylib\\jconn4.jar;c:\\DB2\\sqlj.zip;${CLASSPATH}
```

In Linux/Unix:

```
CLASSPATH=/opt/unixsa/DB_Drivers/jconn4.jar:/opt/unixsa/6.0.2/master/41/TIDAL/master/lib/ojdbc6.jar:${CLASSPATH}
```

When the Adapter process is started, its CLASSPATH includes the Master's CLASSPATH. You can determine the CLASSPATH for the JDBC Adapter by opening the *AdapterHost-xxx.log* located in the `..\\Master\\Log` directory and searching for the following words:

```
*** Starting Service:JDBC
```

The following is an example of the startup parameters for the JDBC Adapter on Windows. Notice the inclusion of the CLASSPATH (`-cp`) with a delimited listed of jar files. This list includes Master distributed JDBC drivers *sqljdbc.jar* (SQLServer) and *ojdbc14.jar* (Oracle).

Note: Should the CLASSPATH not include the driver needed to connect to the database, you may need to do one or both of the following:

1. Obtain the appropriate driver for your configuration
2. Specify the CLASSPATH option in `Service.props`, referencing the fully qualified name of the JDBC jar file.

```
*** Starting Service:JDBC Command is:java -Xms16m -Xmx256m -cp "C:\\Program
Files\\TIDAL\\Scheduler\\master\\services\\jdbcservice.pkg;C:\\Program
Files\\TIDAL\\Scheduler\\master\\services\\{68597802-1828-464D-BEDC-2B6E1D3C8C0C}\\lib\\jniLM.jar;C:\\Program
Files\\TIDAL\\Scheduler\\master\\services\\{68597802-1828-464D-BEDC-2B6E1D3C8C0C}\\lib\\commons-net-1.1.0.
jar;C:\\Program Files\\TIDAL\\Scheduler\\master\\lib\\AdapterHost.jar;C:\\Program
Files\\TIDAL\\Scheduler\\master\\lib\\Scheduler.jar;C:\\Program
Files\\TIDAL\\Scheduler\\master\\lib\\sqljdbc.jar;C:\\Program
Files\\TIDAL\\Scheduler\\master\\lib\\ojdbc14.jar;C:\\SAPJCO\\sapjco.jar"
com.tidalsoft.service.jdbc.JDBCApp -p 6980 -id {7640B420-5530-11DE-8812-7B8656D89593}
```

Note: For operation with MSSQL 2008, you will need Microsoft SQL Server JDBC Driver 2.0 or Microsoft SQL Server JDBC Driver 3.0, which is backwards compatible with SQL Server 2000, SQL Server 2005, and SQL Server 2008. Version 3.0 has additional support for SQL Server 2008. The Microsoft SQL Server JDBC Driver 3.0 download contains two jar files, *sqljdbc.jar* and *sqljdbc4.jar*. Use *sqljdbc.jar* with JRE 5.0 and *sqljdbc4.jar* for JRE version 6.0 or later.

Windows Example:

```
C:\\Program Files\\TIDAL\\Scheduler\\master\\services\\{7640B420-5530-11DE-8812-7B8656D89593}\\lib
```

Unix Example:

```
/opt/tidal/master/services/{7640B420-5530-11DE-8812-7B8656D89593}/lib
```

You will now update the `service.props` to include a reference to the CLASSPATH for the JDBC driver for use in your configuration.

`service.props` entry example for Windows:

```
CLASSPATH= C:\\Program Files\\TIDAL\\Scheduler\\master\\services\\{7640B420-5530-11DE-8812-7B8656D89593}\\lib\\sqljdbc.jar;${CLASSPATH}
```

Note: This example is for SQLServer JDBC driver on Windows. The .jar file will refer to the appropriate driver for your configuration. Also note the use of ";" as a classpath separator on Windows systems. You will use the appropriate separator for your operation system, for example ":" on Unix/Linux systems.

service.props entry example for Unix:

```
CLASSPATH=/opt/unixsa/6.0.1/master/152/TIDAL/master/lib/sqljdbc.jar:${CLASSPATH}
```

2. Reboot the Master machine on Windows as needed. Source the profile file as needed on Unix.

3. Restart the master.

Note: For Windows installations: The JDBC Libraries requires Microsoft Visual C++ runtime components. If your system does not include these components and there is an error starting the JDBC Adapter, you may need to install the runtime components from Microsoft.

Download the installer from Microsoft, as follows for the version of the JDBC Adapter being installed: Download "Microsoft Visual C++ 2005 Redistributable Package (x86)" or "Microsoft Visual C++ 2005 SP1 Redistributable Package (x64)" as appropriate. These downloads can be found from Microsoft's site given an internet search from your browser.

Securing the JDBC Adapter

Access to databases using JDBC is controlled by assigning a JDBC security policy with specified privileges to user accounts. The system administrator should create a new security policy or edit an existing policy in CWA as described in the *Users* chapter of the *Cisco Workload Automation User Guide*. In addition to the normal user privileges, this policy should include the capability to add or edit JDBC jobs. A user whose assigned security policy does not include JDBC privileges cannot create or run JDBC jobs.

The user types associated with the JDBC Adapter are described as follows and are defined in the **Users** pane:

■ Runtime Users

Runtime users in the context of JDBC jobs represent those users and passwords required to connect to the database using JDBC to run queries.

■ Schedulers

Schedulers are those users who will define or manage JDBC jobs and events. There are three aspects of a user profile that grant or limit access to scheduling jobs that interact with JDBC:

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

Defining Runtime Users (Database Users)

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

1. From the **Navigator** pane, select **Administration>Runtime Users** to display the defined users in the **Users** pane.

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

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

Note: If the **DB Server** is **Azure SQL database**, append the server name to the user name in the format <username>@<server name>. For example, if the SQL database server name is cwaserver.database.windows.net and username is testuser, enter the **User Name** as testuser@cwaserver.

4. For documentation, enter the **Full Name** or description associated with this user.
5. In the **Domain** field, select a Windows domain associated with the user account required for authentication, if necessary.
6. On the **Passwords** tab, click **Add** to display the **Change Password** dialog.
7. From the **Password Type** list, select **JDBC**.
8. Enter a password (along with confirmation) in the **Password** and **Confirm Password** fields.

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

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

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

10. Click **OK** to add or save the user record in the CWA database.

Authorizing Schedulers to Work With JDBC Jobs and Events

Authorizing schedulers to work with JDBC jobs and events requires:

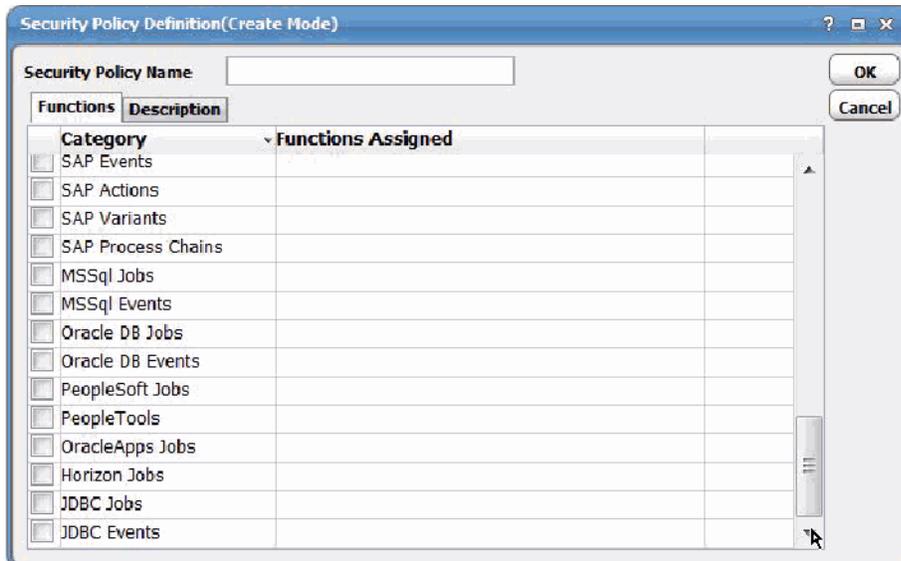
- [Defining a JDBC Security Policy, page 13](#)
- [Defining a CWA User of JDBC Jobs and Events, page 15](#)

Defining a JDBC Security Policy

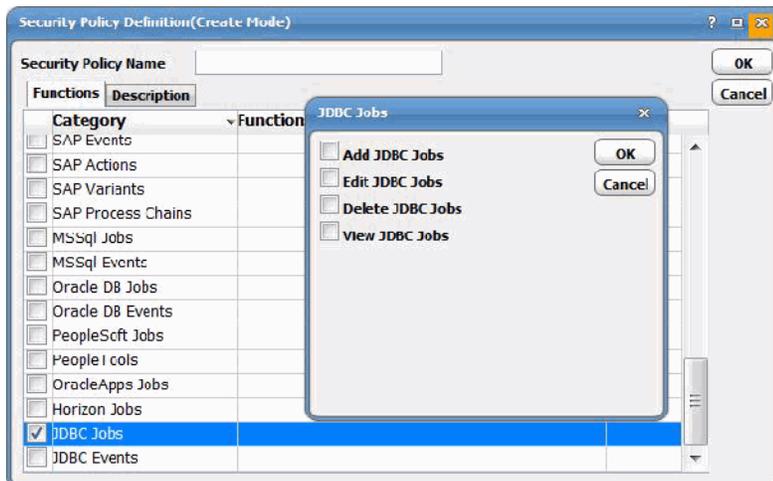
To define a Security Policy that authorizes access to JDBC jobs and events:

1. To define a Security Policy that authorizes access to JDBC jobs and events, select **Administration>Security Policies** from the **Navigator** pane to display the **Security Policies** pane, listing all defined users.

- Right-click anywhere in the **Security Policies** pane and select **Add Security Policy** from the context menu or select an existing policy and choose **Edit** to display the **Security Policy Definition** dialog.



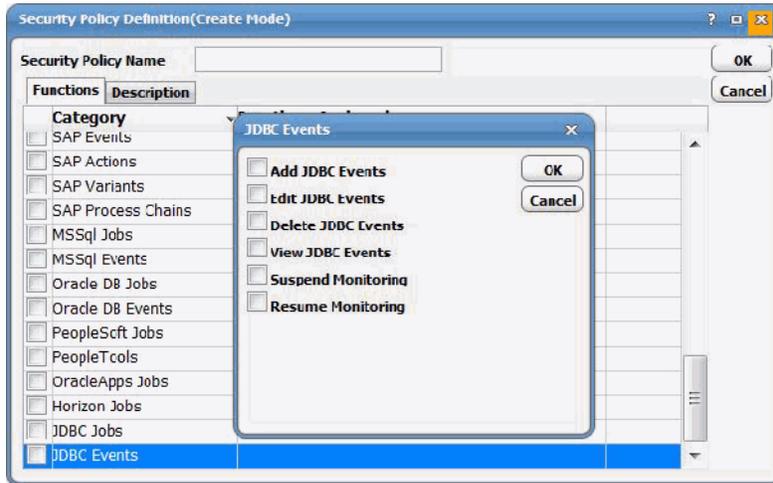
Solution Scroll down the list of function categories and double-click the **JDBC Jobs** category to display the available job functions.



- Select the desired job privileges check boxes (**Add JDBC Jobs**, **Edit JDBC Jobs**, **Delete JDBC Jobs**, **View JDBC Jobs**) and click **OK**.

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

4. Double-click the **JDBC Events** category to display the available event functions.



5. Select the functions to be authorized under this policy (**Add JDBC Events, Edit JDBC Events, Delete JDBC Events, View JDBC Events, Suspend Monitoring, Resume Monitoring**) and click **Close**.

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

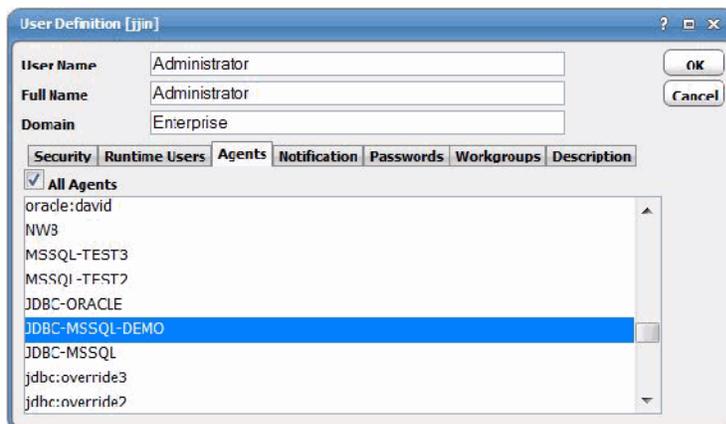
6. Click **OK** to save the policy.

Defining a CWA User of JDBC Jobs and Events

To define a CWA user to work with JDBC jobs and events:

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

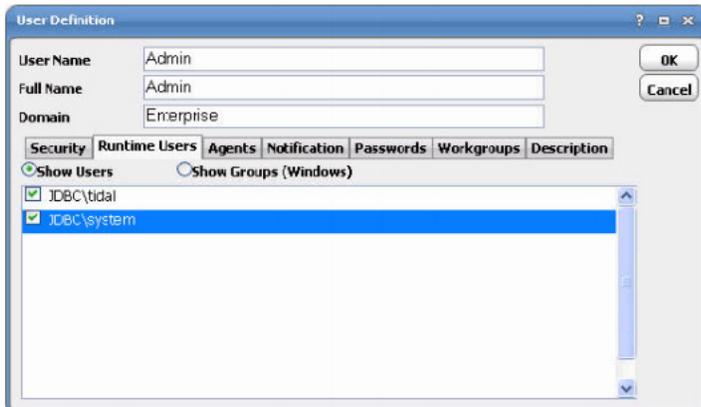
Note: Refer to the *Cisco Workload Automation User Guide* for a general discussion on setting up a user to work with CWA.



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

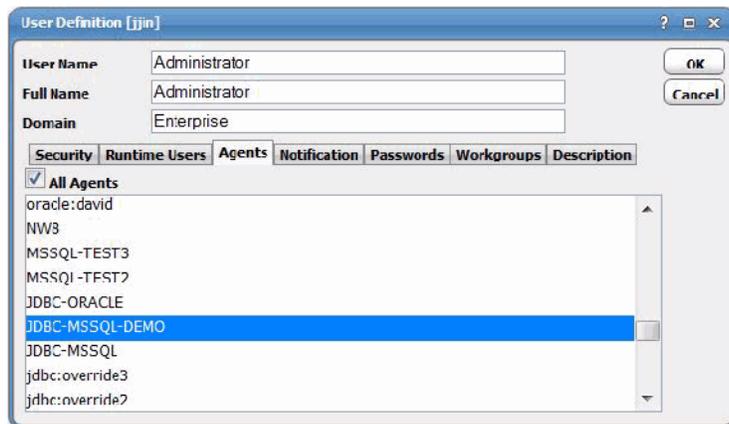
Defining a Connection Using JDBC

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



5. Select the database users that this scheduling user can use when defining JDBC jobs and/or connections for database event monitoring.

6. Click the **Agents** tab.



7. Select which JDBC connections that this scheduling user can access when scheduling jobs or events.

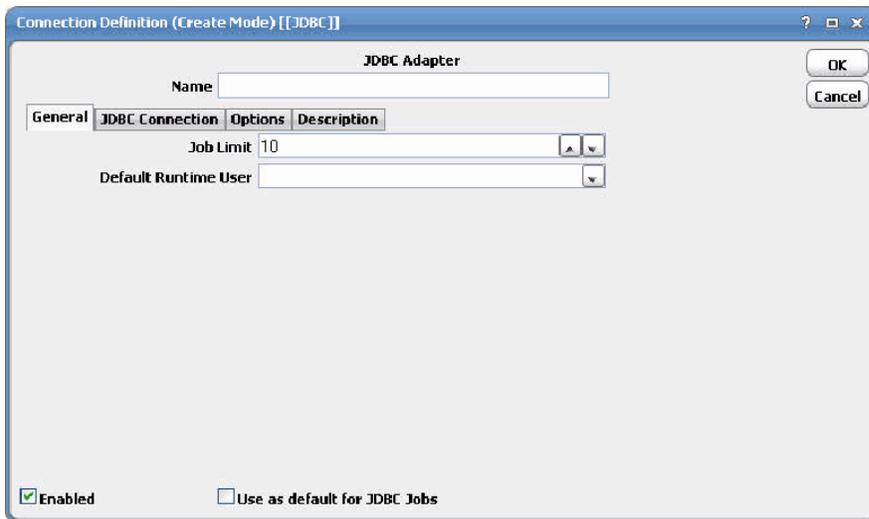
8. Click **OK** to save the user definition.

Defining a Connection Using JDBC

You must create connections to one or more databases using JDBC before CWA can run your JDBC jobs or monitor database events. These connections also must be licensed before CWA can use them.

The connection for the JDBC adapter is defined like other CWA connections except it requires configuration information unique to JDBC.

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



This dialog contains the following elements:

- **Name** - Enter a name for this JDBC adapter connection.
- **Enabled** - Activates or deactivates the connection, shutting down communication with the database.
A job cannot run on an adapter if its connection is disabled.
- **Use as default for JDBC Jobs** - Select to use the connection as your default connection when setting up future JDBC jobs.
- **General** - This tab designates a job limit and default runtime user.
- **JDBC Connection** - This tab provides information to connect to the designated database.
- **Options** - This tab provides configuration options for accessing the database.
- **Description** - This tab describes the connection and provides a convenient place for notes about its purpose and use.

Adding a Connection Using JDBC

To add a connection:

1. From the **Navigator** pane, select **Administration>Connections** to display the **Connections** pane.
2. Click the **Add** button on the toolbar or right-click and select **Add Connection>JDBC Adapter** from the context menu to display the **JDBC Connection Definition** dialog.
3. Enter a name for the new connection in the **Name** field.
4. In the **Job Limit** field of the **General** tab, select the maximum number of concurrent active processes that CWA should submit to this database at one time.
5. (Optional) In the **Default Runtime User** list, select the name of a default user for JDBC jobs.

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

6. Click the **JDBC Connection** tab.

This tab defines how to access a database using JDBC. There are several different ways to connect to a database using JDBC. If you wish to connect to a specific database, you can designate that database for the connection.

7. Use automatic population or manual entry to enter data:

Automatic population: Use predefined templates for the most commonly used database connections:

- a. In the **JDBC Template** field, select a template from the drop-down menu.

The **JDBC Driver** field and the **JDBC Connection String (URL)** field are automatically populated.

Note: You must select the JDBC driver before overriding the JDBC connection string, otherwise, the connection will not be saved.

- b. In the **JDBC Host** field, enter the host machine.

The <Host> placeholder in the **JDBC Connection String (URL)** field is replaced.

- c. In the **Connect User** field, optionally enter a user that is used to connect to the database from the drop-down menu.

You must enter a user to obtain a drop-down list of databases in the **Database/DSN** field.

- d. (Optional) In the **Database/DSN** field, select from the drop-down menu or enter the name of the database for this connection.

Note: If the JDBC connection was created using an Oracle/DB2 database, the “Database Name” is the Schema Name and not the actual Database name used while creating JDBC jobs and events.

The drop-down menu only lists databases for connections where a database name is optional; otherwise, if a database name is required to make a JDBC connection, you must type in the name.

This value is added to the JDBC connection string in the **JDBC Connection String (URL)** field.

Manual entry: If the driver for your RDBMS is not listed in the **JDBC Template** drop-down menu, enter the information manually:

Defining a Connection Using JDBC

- a. In the **JDBC Driver** field, enter your JDBC driver’s fully qualified class name.

The driver class entered here must be found in the CLASSPATH on the master machine. For information, see [Requirements, page 7](#).

- b. In the **Connect User** field, enter the user that is used to connect to the database from the drop-down menu.

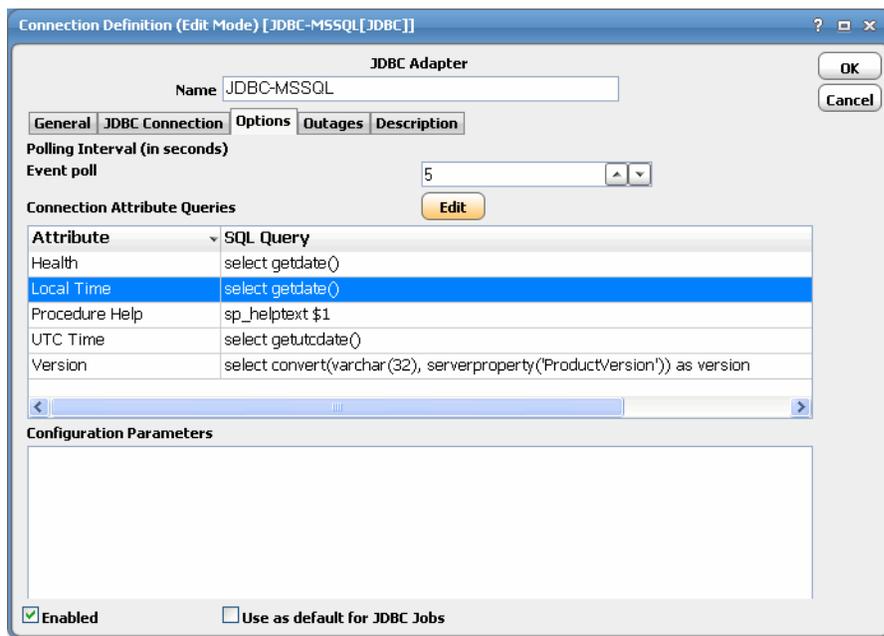
You must enter a user to specify a database in the **Database/DSN** field.

- c. (Optional) In the **Database/DSN** field, select from the drop-down menu or enter the name of the database for this connection.

The drop-down menu only lists databases for connections where a database name is optional; otherwise, if a database name is required to make a JDBC connection, you must type in the name.

- d. In the **JDBC Connection String (URL)** field, enter the JDBC driver connection URL as specified in your driver’s documentation.

8. Click the **Options** tab.



Note: If the **DB Server** is Azure SQL database, add `CONN_TYPE=Azure` as the **Configuration Parameter**.

- 9. In the **Event poll** field, enter the number of seconds between successive poll checks of an event monitor that the adapter uses to check whether it should raise an event.
- 10. Enter queries specific to your database platform in the **Connection Attribute Queries** pane.

If you had selected a JDBC template previously, queries for the selected database will populate automatically in many cases. Optionally, you can override these queries or provide your own custom queries.

- **Version** - Queries the version number of the database.
- **Local Time** - Queries the time of the database’s server.
- **UTC Time** - Queries the UTC time for the database server.

Defining a Connection Using JDBC

The local time and the UTC is used to get the time zone and calculate the time difference between the database server and the scheduling master.

- **Health** – Queries the health of the database based on any custom SQL select statement.

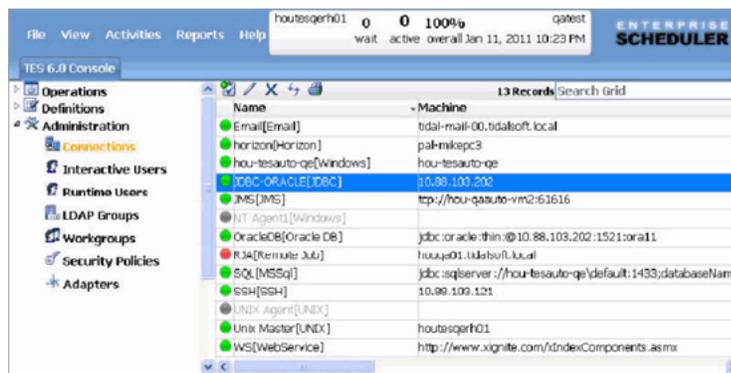
If the query returns a result set, the connection is considered healthy (green). If an empty result set is returned, the connection is considered unavailable (red).

- **Procedure Help** – Queries the content of a stored procedure that is displayed in the **SQL** tab in the **JDBC Job Definition** dialog.

Use \$1 as a placeholder for the stored procedure name, which is replaced by an actual procedure name when the adapter needs to get the help or text associated with the stored procedure.

11. Click **OK**.

JDBC Connection Status Light



If the connection using JDBC is not available to the master, the connection status light in the first column of the display is red, and the **Connections** status light at the bottom of the console is red (or yellow if other agents are connected to the master).

Note: This light is relative to the status of the connection to the CWA master, not the connection to the client.

The status light indicates the health of the adapter connection to the database. If a health query has been provided on the **Options** tab described previously in [Adding a Connection Using JDBC, page 17](#), it will also be used to determine the connection status.

If the CWA master cannot connect or loses its connection to a database, you will see a red status light next to your connection using JDBC in the **Connections** pane.

Note: If the adapter connection using JDBC is not available, you will not be able to define JDBC jobs.



3

Defining JDBC Jobs

Overview

You can start creating and scheduling JDBC jobs once you have completed the following:

- Added one or more database users to CWA
- Defined your connections using JDBC

You can create a JDBC job using the context menu within the **Jobs** pane. You can also edit, copy, and delete an existing JDBC job. If you add a JDBC job to a CWA job group, items common between the job group and the JDBC job are inheritable. However, unless the parent group has a JDBC agent assigned to it, you must clear the **Inherited** option and choose an appropriate JDBC connection.

Selecting the **Add JDBC Job** option from the CWA **Jobs** pane displays the **JDBC Job Definition** dialog.

There are two types of JDBC adapter jobs that you can define in CWA:

- **SQL** - Direct inclusion of SQL statements, with or without dynamic parameter replacement.
- **Stored Procedure** - Reference to a stored procedure, including dynamic or static parameter replacement.

This chapter describes:

- [Adding a JDBC SQL Job, page 21](#)
- [Adding a JDBC Stored Procedure Job, page 25](#)

Adding a JDBC SQL Job

To add a JDBC SQL job:

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

- Click the **Add** button from the CWA toolbar or right-click in either the **Navigator** or **Jobs** pane and select **Add JDBC Job** from the context menu.

The screenshot shows the 'JDBC Job Definition' dialog box with the following configuration:

- JDBC Job Name:** Sybase JDBC
- Job Class:** (empty dropdown)
- Parent Group:** (empty dropdown)
- Owner:** Schedulers
- Agent/Adapter Name:** JDBC-Sybase[JDBC]
- Agent List:** (empty dropdown)
- Runtime User:** JDBC\Tidal
- Tracking:**
 - Exit code
 - External
 - Scan output: Normal String(s)
 - Scan output: Abnormal String(s)
- Duration (in minutes):**
 - Estimated: 1:00
 - Minimum: 1:00
 - Maximum: 1:00
- Exclude Completed Abnormally
- Enabled

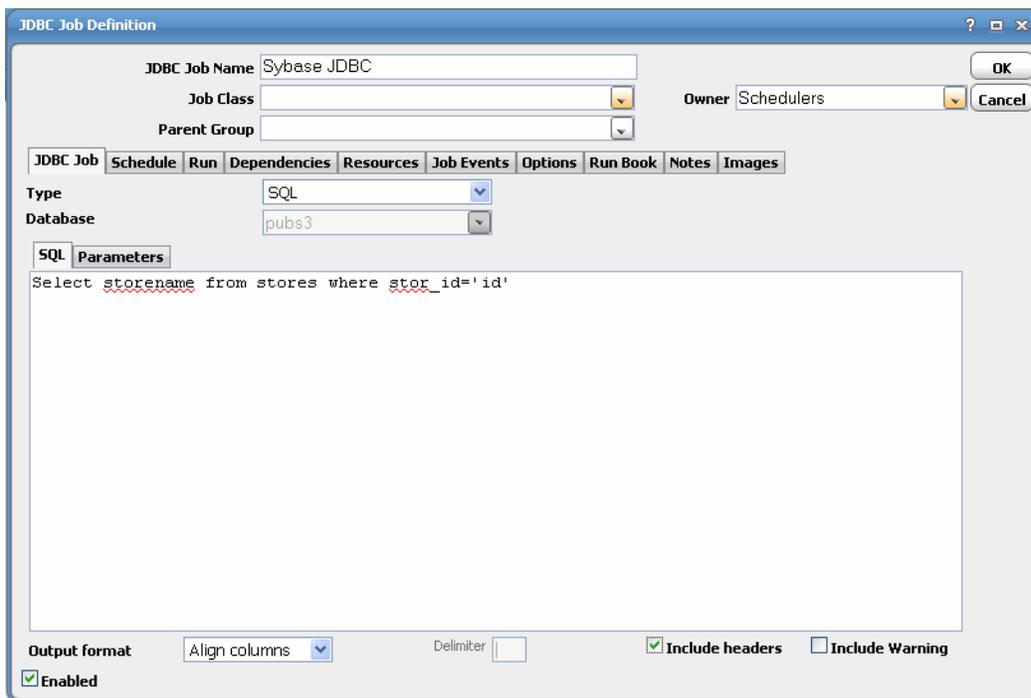
- In the **JDBC Job Name** field, enter a name up to 50 characters in length for your job.

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

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

- On the **Run** tab, select an agent or adapter name from the **Agent/Adapter Name** list.
- Select a runtime user from the **Runtime User** list.

For information on authorizing a runtime user, see [Defining Runtime Users \(Database Users\)](#), page 12.

6. Click the **JDBC Job** tab.7. Select the **SQL** job type from the **Type** list.8. If you did not select a database in the **Connection Definition** dialog, you must select one from the **Database** drop-down menu.

If you did select a database in the **Connection Definition** dialog, it appears here on the **JDBC Job** tab as read-only in the **Database** field.

9. In the **SQL** tab field, type in one or more SQL statements with multiple statements separated by semicolons to execute here.

To include parameters that are replaced at runtime, use unique parameter names of your choice preceded by a colon, such as `:id`.

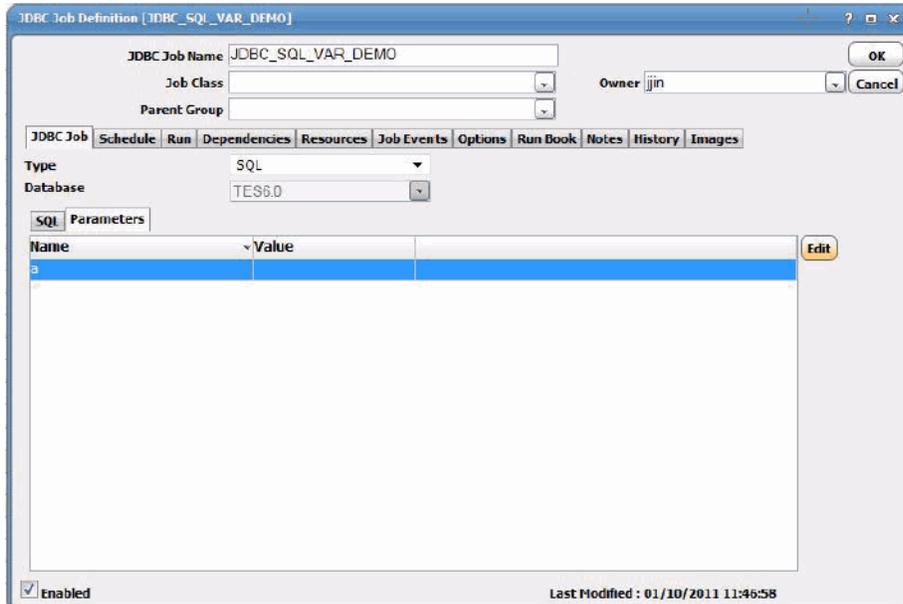
10. In the **Output format** drop-down menu, select how the query results are formatted.

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

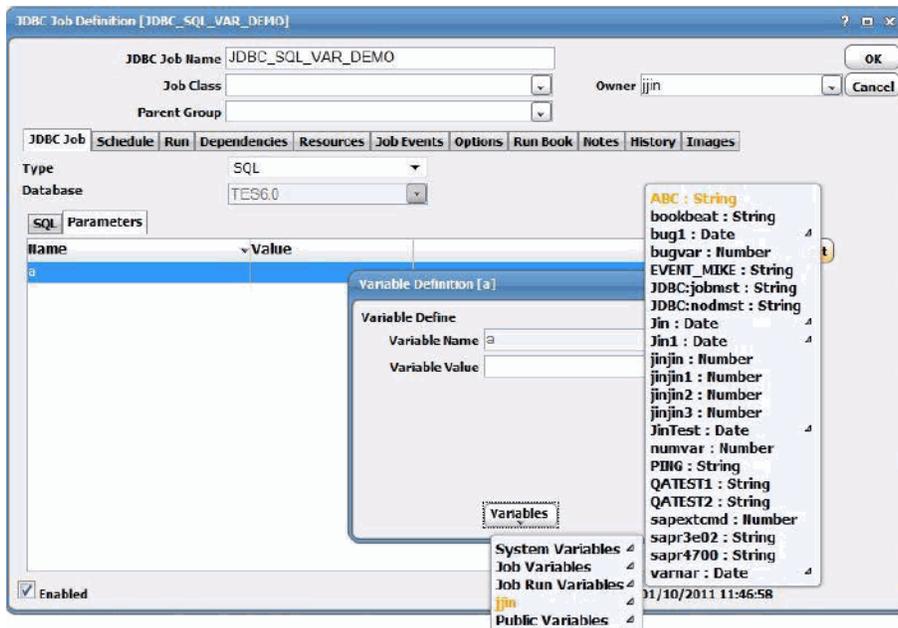
11. In the **Delimiter** field, specify the custom character to use for delimiting the column data from the query results if **Raw** or **CSV Format** is selected for **Output Format**.12. Select the **Include Headers** option to write out the column headers of the results as well.13. Select the **Include Warning** option to include notices and warnings generated during SQL execution in the output.14. If you are using parameters to be replaced at runtime, click the **Parameters** subtab.

Adding a JDBC SQL Job

This view provides a list of parameters that have been preceded by a colon in the SQL statement, as described in Step 9, where you can provide values for the parameters. If there are no parameters defined in the SQL statement, you cannot enter anything on this subtab.



15. Place the cursor in the **Value** field and enter a value or click **Variables** and select one to insert a dynamically replaced variable value.



16. Select the **Enabled** option to enable or disable the job definition.

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

Adding a JDBC Stored Procedure Job

To add a stored procedure job:

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Click the **Add** button from the CWA toolbar or right-click in either the **Navigator** or **Jobs** pane and select **Add JDBC Job** from the context menu to display the **JDBC Job Definition** dialog.
3. In the **JDBC Job Name** field, enter a name up to 50 characters in length for your job.

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

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

4. On the **Run** tab, select an agent or adapter name from the **Agent/Adapter Name** list.
5. Select a runtime user from the **Runtime User** list.

For information on authorizing a runtime user, see [Defining Runtime Users \(Database Users\)](#), page 12.

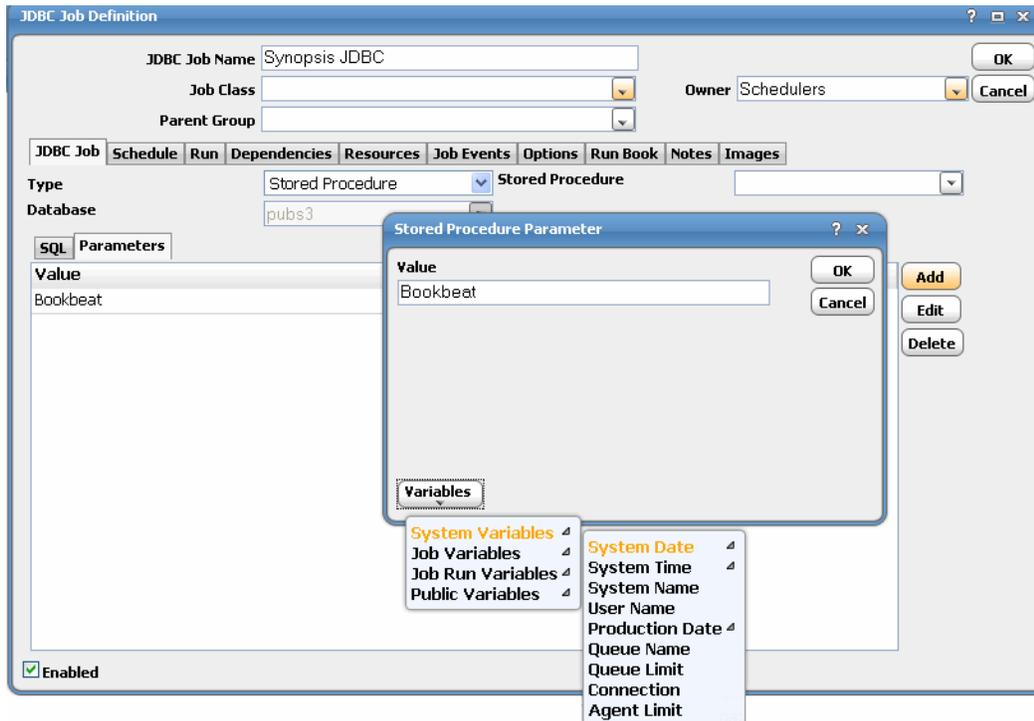
6. Click the **JDBC Job** tab, then select the **Stored Procedure** job type from the **Type** list.
7. If you did not select a database in the **Connection Definition** dialog, you must select one from the **Database** drop-down menu.

If you did select a database in the **Connection Definition** dialog, it appears here on the **JDBC Job** tab as read-only in the **Database** field.

8. From the **Stored Procedure** list, select the procedure to execute when the job is run.

You can use an asterisk (*) to limit the number of stored procedures displayed in the drop-down menu. For example, if you enter store*, only procedures that begin with store are displayed in the drop-down menu.

9. For each parameter of the stored procedure, click **Add** to display the **Stored Procedure Parameter** dialog.



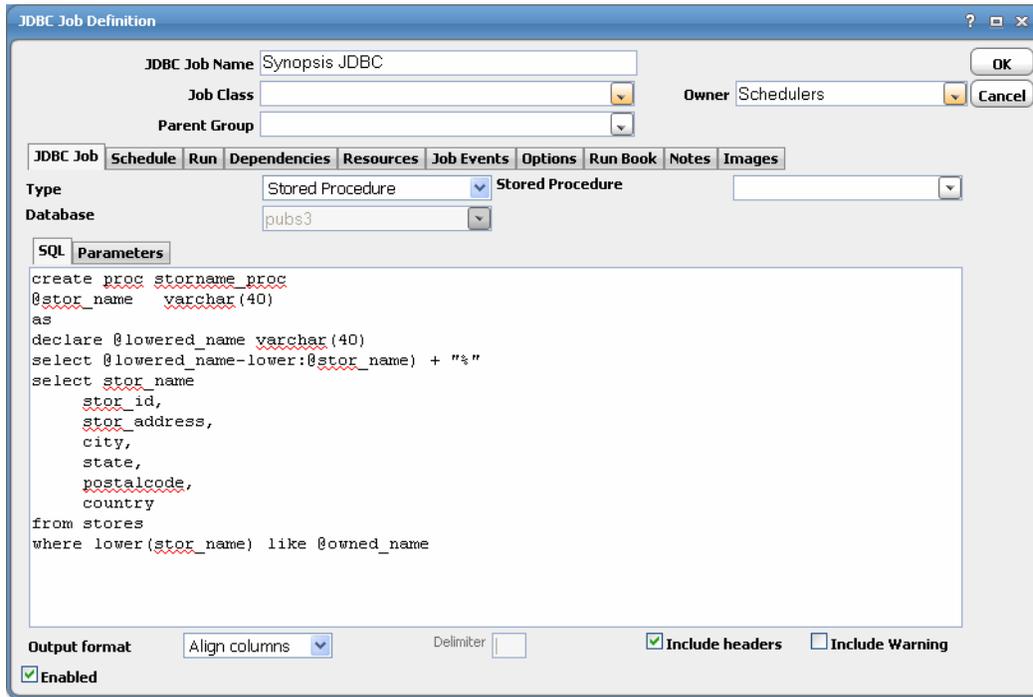
10. Enter the parameter value into the **Value** field, or select a predefined variable from the Variables button list, then click **OK**.

A new row is created each time you add a value so you can add any number of required parameters for the stored procedure.

11. Select the **Enabled** option to enable or disable the job definition.
12. Click the **SQL** subtab to view the stored procedure.

Note: The SQL subtab is populated only if a query has been defined to retrieve stored procedure help or text on the Options tab of the Connection Definition dialog as described in [Adding a Connection Using JDBC, page 17](#).

This subtab is read-only.



13. In the **Output format** drop-down menu, select how the query results are formatted.
 - XML – Writes the query results in XML format.
 - **Align Columns** - Displays the values in the most readable format.
 - **CSV Format** - Separates values with commas.
 - **Raw** - Separates values with a user-defined character.
14. In the **Delimiter** field, specify the custom character to use for delimiting the column data from the query results if **Raw** or **CSV Format** is selected for **Output Format**.
15. Select the **Include Headers** option to write out the column headers of the results as well.
16. Select the **Include Warning** option to include notices and warnings generated during SQL execution in the output.
17. Click **OK**.



4

Working with JDBC Jobs

Overview

This chapter covers these topics related to working with JDBC jobs:

- [Understanding JDBC Job Runtime Details, page 29](#)
- [Controlling Adapter and Agent Jobs, page 32](#)

Understanding JDBC Job Runtime Details

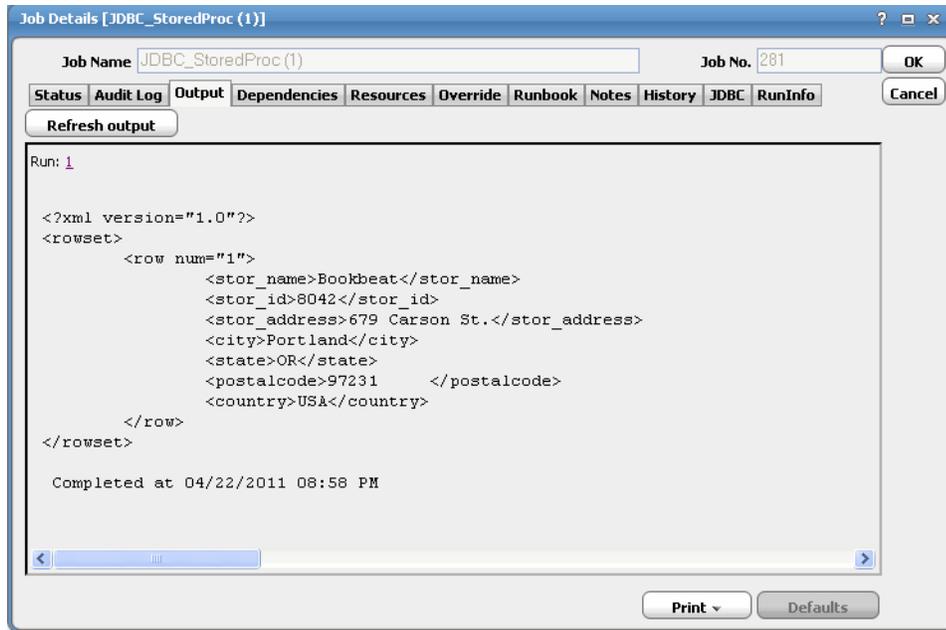
The **Job Details** dialog provides information on the job after it has completed or as it is still running. To access the **Job Details** dialog, double-click on a job instance record in the **Operations>Job Activity** pane or right-click and select the **Details** option from the context menu. The following tabs in the **Job Details** dialog are specific to the JDBC adapter:

- [Output Tab, page 29](#)
- [JDBC Tab, page 30](#)
- [Run Info Tab, page 31](#)

Output Tab

If a job is configured to save output, the **Output** tab in the **Job Details** dialog shows the result set from the SQL or stored procedure job and is formatted as specified in the job definition. CWA 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 (#).

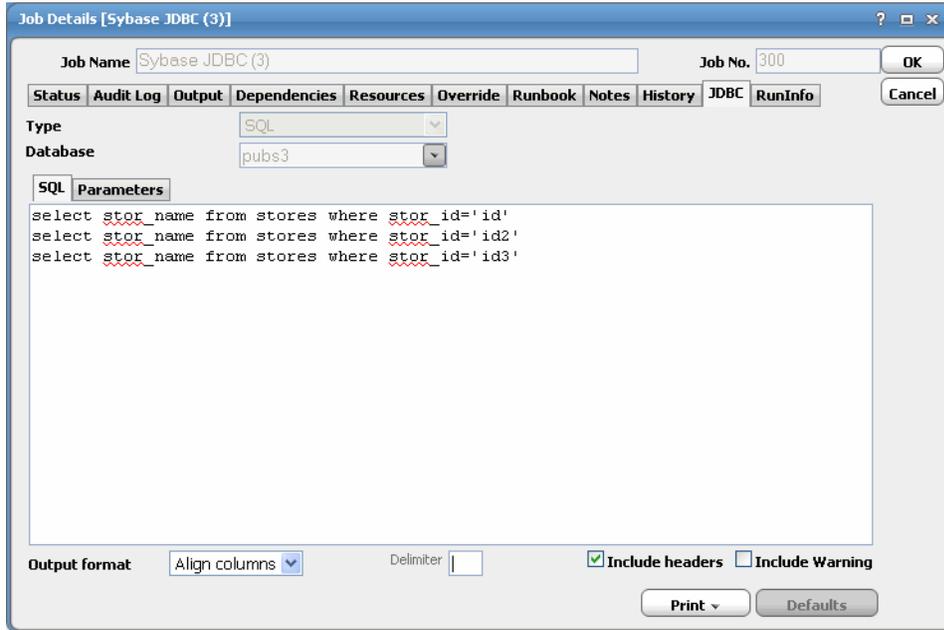
Note: CWA's default is to discard job output. If you want to view job output, you must select the Save Output option on the Options tab in the Job Definition dialog or change the system default on the Defaults tab in the System Configuration dialog.



JDBC Tab

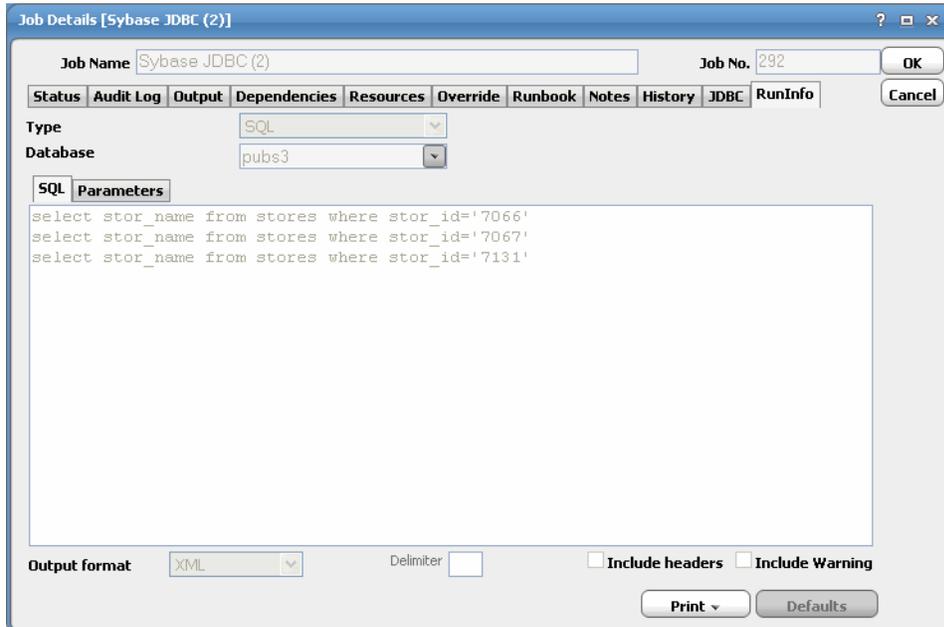
The **JDBC** tab in the **Job Details** dialog contains the original request along with variables used when this job was submitted. This tab allows you to override the SQL statements, Stored Procedures, output format, and parameter values prior to run or rerun. Overrides are not permitted when the job is running.

For information on how to make overrides, see [Defining JDBC Jobs](#), page 21.



Run Info Tab

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



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:

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

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

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

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

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

1. From the **Job Activity** pane, right-click the Adapter/Agent job to be deleted.
2. Select **Remove Job(s) From Schedule**.



5

Working with JDBC Events

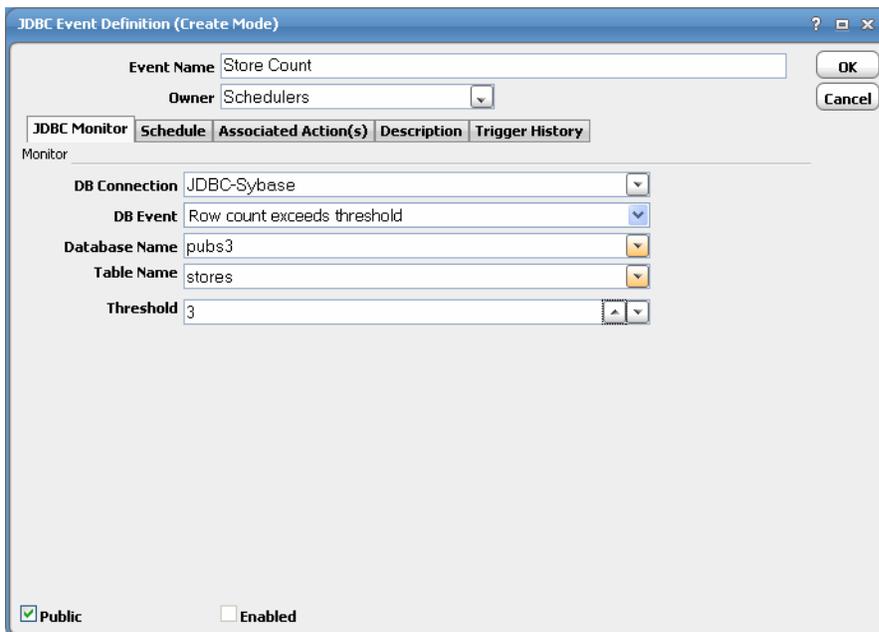
Overview

Using the JDBC Adapter, you can define events that can be used for alerting and invoking an automated response through e-mail and/or inserting additional jobs into the schedule. The **Event Definition** dialog is displayed when you add or edit a JDBC event. CWA can monitor events and then take one or more actions when the event trigger occurs. You must configure a calendar for the event from the **Schedule** tab to schedule when the event is enabled (that is, when monitoring will occur). If needed, you can configure the monitor to operate only during certain time periods or leave the monitor in operation at all times.

Adding a JDBC Event

To add an event:

1. From the **Navigator** pane, select **Events>JDBC Events** to display the **JDBC Events** pane.
2. Click the **Add** button on the **CWA** toolbar or right-click on the **JDBC Events** pane and select **Add Event** from the context menu.



Adding a JDBC Event

The **JDBC Monitor** tab designates the data table and indexes in the JDBC database to be monitored and defines the condition that will be considered an event. Once the designated change to the data table is detected, the event can be linked to an action to trigger an automatic response to the change. This same tab is displayed in the **Event Details** dialog for the defined event except the options are read-only to provide information about the event that occurred.

On the **JDBC Monitor** tab, you can do the following:

- Specify which database connection to monitor from the list of defined connections from the DB Connection list
- Define the condition that will raise the event

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

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

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

6. From the **DB Event** list, select one of the following to designate as the event trigger:

- Row count exceeds threshold
- Column MAX value exceeds threshold

If selected, a **Column Name** field displays and is described in Step 9.

The screenshot shows the 'JDBC Event Definition (Edit Mode) [Store Count]' dialog box. The 'Event Name' field contains 'Store Count'. The 'Owner' dropdown is set to 'Schedulers'. The 'JDBC Monitor' tab is active, showing a 'Monitor' section with several dropdown menus: 'DB Connection' (JDBC-Sybase), 'DB Event' (Column MAX value exceeds threshold), 'Database Name' (pubs3), 'Table Name' (stores), and 'Column Name' (payterms). A list of column names is displayed below the 'Column Name' dropdown, with 'payterms' highlighted. At the bottom, the 'Public' and 'Enabled' checkboxes are checked. 'OK' and 'Cancel' buttons are visible in the top right corner.

- Column MIN value below threshold

If selected, a **Column Name** field displays as shown above.

- Custom query returns results

Adding a JDBC Event

If selected, the **Custom Query** field displays as shown below.

The screenshot shows the 'JDBC Event Definition (Edit Mode) [Store Count]' dialog box. The 'Event Name' is 'Store Count' and the 'Owner' is 'Schedulers'. The 'JDBC Monitor' tab is selected, showing a 'Monitor' section with three dropdown menus: 'DB Connection' set to 'JDBC-Sybase', 'DB Event' set to 'Custom query returns results', and 'Database Name' set to 'pubs3'. Below this is a 'Custom Query' field with the text: 'select stor_id from stores where stor_id > '8000''. At the bottom, there are checkboxes for 'Public' (checked) and 'Enabled' (checked). 'OK' and 'Cancel' buttons are in the top right corner.

7. If there is no database displayed in the **Database Name** field, select one from the drop-down menu.

If the database was entered on the **Connection Definition** dialog, it appears here on the **JDBC Monitor** tab as read-only in the **Database Name** field.

8. From the **Table Name** list, select the name of the data table to be monitored.
9. If the **Column Name** field is displayed, select a column from the drop-down menu.
10. If the **Custom Query** field is displayed, enter a query.

A custom event triggers when a nonempty result set is returned from a custom query.

11. From the **Threshold** drop-down menu, select a threshold for the selected event.
 - For the **Row count exceeds threshold** event, once the threshold is exceeded and the event is triggered, it will trigger again only if the row count drops below the threshold and then goes above the threshold again.
 - For the MIN event, if the smallest monitored column value falls below the threshold, the event will trigger. The event will only trigger again if the column minimum value goes above the threshold and drops below it again.
 - For the MAX event, if the largest monitored column value goes above the threshold, the event will trigger. The event will only trigger again if the column maximum value drops below the threshold and goes above it again.

12. Click the **Schedule** tab to assign a schedule to the event.

JDBC Event Definition (Edit Mode) [Store Count]

Event Name: Store Count
Owner: Schedulers

JDBC Monitor | **Schedule** | Associated Action(s) | Description | Trigger History

Calendar

Calendar Name: Daily
Offset: 0
From: [calendar icon] To: [calendar icon] Forecast

Trigger Constraints
Trigger maximum of 5 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	16:23:54	16:23:54

Time Window [2]

Row: 2
From Time: 4:24:06 PM (hh:mm:ss)
Until Time: 4:24:06 PM (hh:mm:ss)

13. 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, then the event monitor will run every day.

To clear a calendar, right-click on the **Calendar** field and select **Clear** from the context menu or press the **Delete** key.

14. 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.

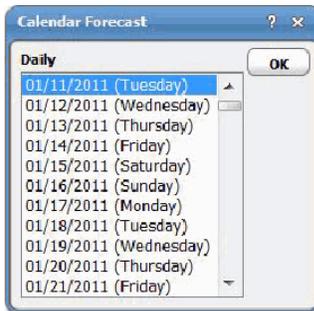
15. In the **From** field, select the earliest date that the event monitor should be active.

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.

16. 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. To choose a date using the built-in calendar dialog, click the adjacent drop-down arrow.

17. 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.



18. Select the **Trigger Constraints** option.

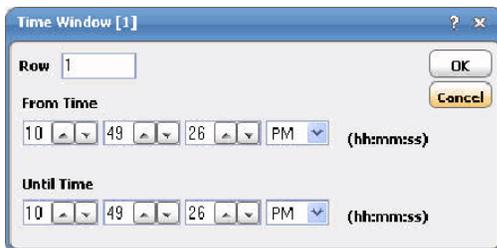
The **Trigger Constraints** section defines constraints on actions associated with the event monitor. The constraints allow you to throttle the execution of actions. At times, the event condition that you are searching for may trigger in rapid succession. Rather than act on each and every event in the 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.”

19. 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.
20. In the **occurrences in __ minute (s) windows** field, designate the time window within which the event monitor will react to the defined conditions.
21. Select the **Ignore occurrences that exceed the limit** option if you want to ignore any occurrences that exceed configured value.
22. Select the **Time Window** check box to activate the **Time Window**.

The **Time Window** limits the times that the event monitor is active to the defined times that are listed. 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.

23. Click **Insert** to display the **Time Window** dialog.



- **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.

Assigning an Action to an Event

24. In the **From Time (hh:mm:ss)** field, click the down arrow to enter a time in the clock dialog or type the time directly in the field.
25. In the **Until Time (hh:mm:ss)** field, click the down arrow to enter a time in the clock dialog or type the time directly in the field.
26. To delete a selected time window, click **Delete**.
27. To delete all of the defined time windows listed in this text field, click **Clear**.
28. Click **OK** to add the new event.

Assigning an Action to an Event

You can assign one or more actions to an event so that you know when the event has triggered or configured an automated response. You can add any action for a JDBC event that is available in CWA.

For example, to define an e-mail action for a JDBC event:

1. In the **Navigator** pane, select **Definitions>Actions >Email**.
2. Click the **Add** button on the CWA toolbar or right-click the action on the **Actions** pane and select **Add Action** from the context menu to display the **Action Definition** dialog.

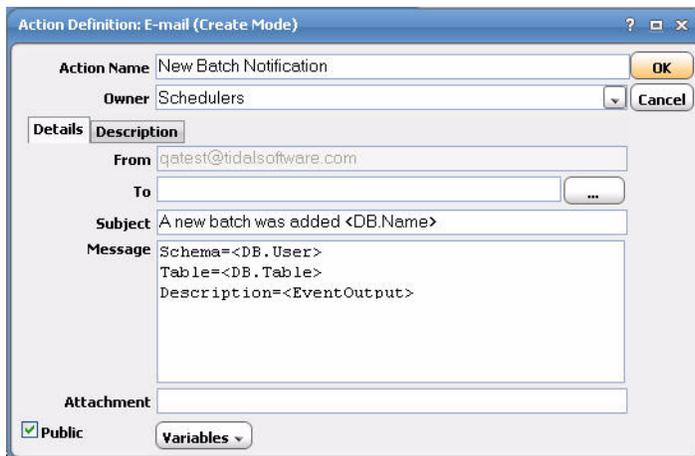


Figure 1 Action Definition Dialog

The JDBC event variables that are available with the JDBC adapter are described as follows and only apply when the action is associated with a JDBC event:

- **DB Name** – Name of database where the event was triggered.
- **DB.Table/Index** – The table or index that triggered the event.
- **DB.User** – The database user who triggered the event.
- **Row Count** – The number of rows in the table that triggered the event. Applies to all events.

For the custom monitor, the row count is the number of rows in the result set from the custom query.

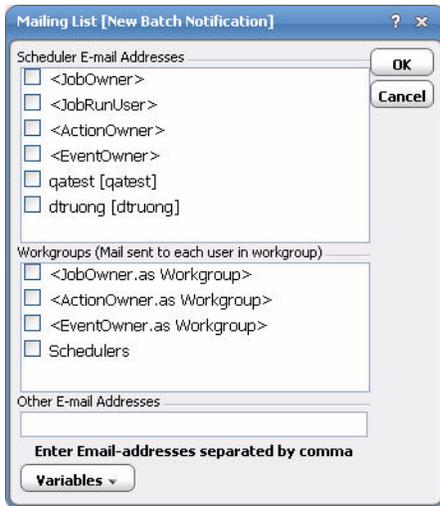
- **Column Value** – The MAX column value and the MIN column value respectively.

For the custom query monitor, the column value is the value of the first row's first column from the result set.

- **EventOutput** – The output of the event selected.

Assigning an Action to an Event

3. In the **Action Name** field, enter a name for the action.
4. From the **Owner** list, select the owner name.
5. In the **To** field, click the ellipsis button to display the **Mailing List** dialog.



6. Select the e-mail recipients.
7. Click **OK**.
8. In the **Subject** field, enter the subject of the e-mail.

You can enter a combination of text or variables in your subject. Use the **Variables** button to choose variables from the **Variables** context menu.

9. In the **Message** field, enter the message.

You can enter a combination of text or variables in your message. Click **Variables** to choose variables from the variables context menu to be dynamically replaced at runtime.

Note: When creating an e-mail message for a system event about losing your database connection, do not use variables that will not work without a connection to the database, such as <JobName> or <JobStatus>.

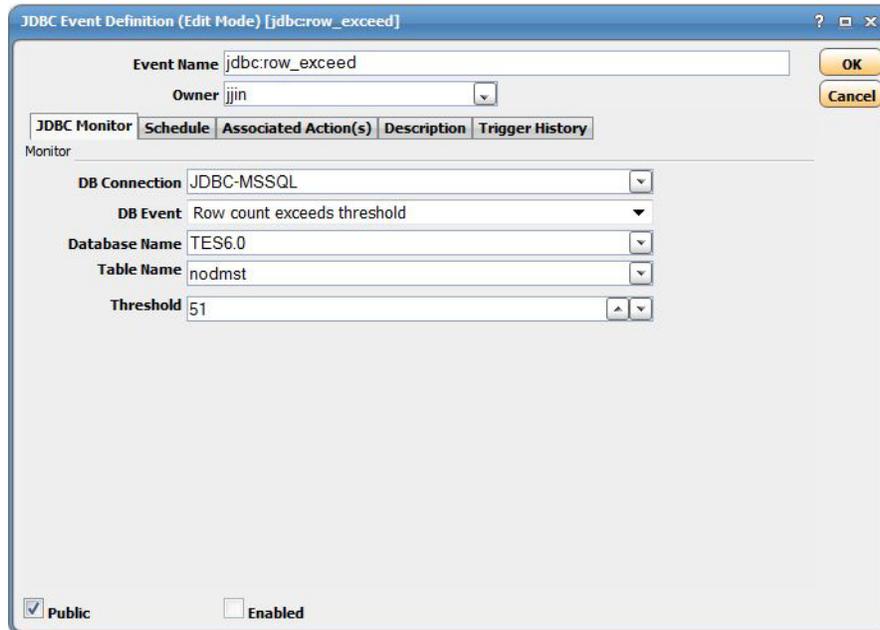
10. In the **Attachment** field, enter the location of the file to attach to your e-mail.
11. Select the **Public** option.

When selected, the action is available to all CWA 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.

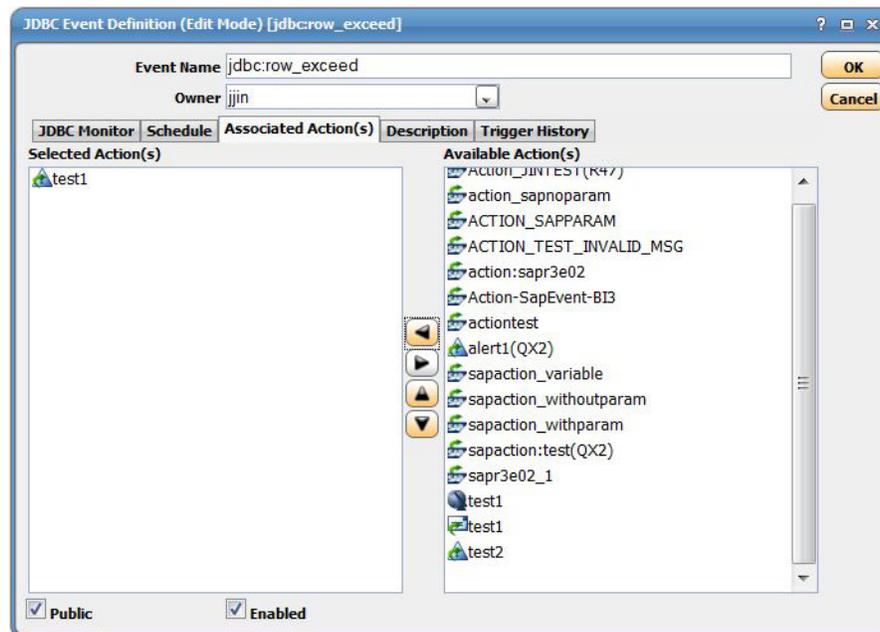
12. Click **OK**.
13. From the **Navigator** pane, select **Events>JDBC Events** to display the **JDBC Events** pane.

Assigning an Action to an Event

14. Double-click on the event that you would like to assign the action to display the **JDBC Event Definition** dialog.



15. Click the **Associated Action(s)** tab.



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

Note: You can also add a new action directly from this field by right-clicking and selecting an option from the context menu.

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

18. Click **OK**.



6

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 <i>service.props</i> configuration
CONN_SYNC	Informatica, Oracle Apps, SAP	N	Setting this flag to Y allows synchronous connections without overloading the RDOOnly 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.
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.

Property	Applicable Adapter(s)	Default	What It Controls
HADOOP_JAVA_HOME	Sqoop	<none>	If the Java version used in the Hadoop environment is lower than Java 8, then install the same lower JDK version in the in the Master and include the path of the JDK in this property.
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.
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
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.

Property	Applicable Adapter(s)	Default	What It Controls
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, PeopleSoft	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.
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 <i>Informatica Adapter Guide</i> 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.

service.props Properties

Property	Applicable Adapter(s)	Default	What It Controls
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.