



Tidal Workload Automation Oracle Database Adapter Guide

Version 6.3.3

First Published: January 2018

tidalautomation.com

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 PRODUCTS IN THIS MANUAL ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR STA GROUP REPRESENTATIVE FOR A COPY.

The 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. © 1981 Regents of the University of California. All rights reserved.

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 uncontrolled copies and the original online version should be referred to for latest version.

© 2018 STA Group LLC. All rights reserved.

Contents

Contents	3
Preface	5
Audience	5
Related Documentation	5
Obtaining Documentation and Submitting a Service Request	5
Document Change History	6
Introducing the Oracle Database Adapter	7
Overview	7
Prerequisites	7
Accessing Audit Trails for Database Events	8
Terms to Know	9
Configuring the Oracle Database Adapter	11
Overview	11
Licensing an Adapter	11
Defining Oracle Database Adapter Users	12
Defining Runtime Users	13
Defining the Oracle Database Access Account	14
Adding an Oracle Database User	14
Defining an Oracle Database Security Policy	15
Defining an Oracle Database Connection	16
General Tab	17
Oracle DB Connection Tab	17
Options Tab	18
Outages Tab	19
Outage Definition Dialog	20
Description Tab	21
Verifying Oracle Database Connection Status	21
Defining an Oracle DB Agent List	21
Using the Oracle Database Adapter	23
Overview	23
Defining an Oracle Database Job	23
Defining a PL/SQL Job	24
Defining a Database Job	26
Monitoring Oracle Database Job Activity	28
Oracle Database Job Details	28
Oracle DB Tab	28
Run Info Tab	29
Oracle Database Job Output	30

Monitoring Changes in the Oracle Database	30
Defining Oracle Database Events	30
Oracle DB Tab.....	31
Schedule Tab	33
Associated Action(s) Tab	33
Description Tab	33
Trigger History Tab.....	33
Troubleshooting Issues with Oracle Database Jobs	33
Controlling Adapter and Agent Jobs	34
Holding a Job	35
Aborting a Job	35
Rerunning a Job	35
Making One Time Changes to an Adapter or Agent Job Instance.....	35
Deleting a Job Instance before It Has Run.....	36
.....	36
Configuring service.props	37
About Configuring service.props	37
service.props Properties	37

Preface

This guide describes the installation, configuration, and usage of the Oracle Database Adapter with Tidal Workload Automation (TWA).

Audience

This guide is for administrators who install and configure the Oracle Database Adapter for use with TWA, and who troubleshoot TWA installation and requirements issues.

Related Documentation

For a list of all Tidal Workload Automation guides, see the *Tidal Workload Automation Documentation Overview* of your release on [tidalautomation.com](http://docs.tidalautomation.com) at:

<http://docs.tidalautomation.com/>

Note: We sometimes update the documentation after original publication. Therefore, you should also review the documentation on tidalautomation.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 Tidal Product Documentation* at:

<https://docs.tidalautomation.com/rss>

Subscribe to *What's New in Tidal Product Documentation*, which lists all new and revised Tidal 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 *Tidal Workload Automation Oracle Database Adapter Guide*.

Table 2-1

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>TWA Installation Guide</i> ; usage provided in online Help only.
6.2.1 SP3	May 2016	Consolidated all Oracle Database Adapter documentation into one document.
6.3	August 2016	Rebranded “Cisco Tidal Enterprise Scheduler (TES)” to “Cisco Workload Automation (CWA)”. Miscellaneous edits for the 6.3 release.
6.3.3	January 2018	Rebranded “Cisco Workload Automation (CWA)” to “Tidal Workload Automation (TWA)”.

1

Introducing the Oracle Database Adapter

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

- [Overview, page 7](#)
- [Prerequisites, page 7](#)
- [Terms to Know, page 9](#)

Overview

To run and monitor an Oracle database job from Tidal Workload Automation (TWA), you need to create a job definition in TWA for it. The database job may already exist as a defined job within the Oracle database or it can be SQL statements defined while creating the TWA job definition. The job can either run inline or it can run in the background as a DBMS job. If the job runs in TWA, TWA maintains a large degree of control but if the job runs as a DBMS job than control over the job is more limited from within Oracle. Job output is available if the job runs in TWA but no output can be retrieved from DBMS jobs.

The Oracle Database Adapter is part of the regular TWA installation but it can only be used if the appropriate license file is applied to TWA . Before beginning the configuration procedures, verify that your environment meets the following prerequisites.

Prerequisites

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

The Oracle Database Adapter provides advanced scheduling capabilities through the standard Oracle database scheduler, which is a collection of functions and procedures in the DBMS_SCHEDULER package. Jobs are executed by a job coordinator background process that is automatically started and stopped as needed. By default, this job coordinator is not up and running, but the database does monitor whether there are any jobs to be executed. If jobs are detected, the database starts the job coordinator.

The initialization parameter JOB_QUEUE_PROCESSES only applies to DBMS_JOB. When DBMS_SCHEDULER is used, the coordinator will automatically determine how many job slaves to start based on CPU load and the number of outstanding jobs. In special scenarios a DBA can still limit the maximum number of slaves to be started by the coordinator by setting the MAX_JOB_SLAVE_PROCESSES with the DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE procedure.

Your Oracle environment must have the following to successfully work with the Oracle Database Adapter:

- DBMS_SCHEDULER package.
- System Privileges:
 - Create any job
 - Execute any program
 - Execute any class
 - Manage TWA

- Object Privileges:
 - Execute
 - Alter
 - All
- Read access to the following database scheduler views:

Table 1

View	Description
DBA_SCHEDULER_SCHEDULES	These views show all schedules.
DBA_SCHEDULER_PROGRAMS	These views show all programs.
DBA_SCHEDULER_PROGRAM_ARGUMENTS	These views show all arguments registered with all programs as well as the default values if they exist.
DBA_SCHEDULER_JOBS	These views show all jobs, enabled as well as disabled.
DBA_SCHEDULER_GLOBAL_ATTRIBUTE	These views show the current values of TWA attributes.
DBA_SCHEDULER_JOB_ARGUMENTS	These views show all arguments for all jobs, assigned and unassigned.
DBA_SCHEDULER_JOB_CLASSES	These views show all job classes.
DBA_SCHEDULER_WINDOWS	These views show all windows.
DBA_SCHEDULER_JOB_RUN_DETAILS	These views show all completed (failed or successful) job runs.
DBA_SCHEDULER_WINDOW_GROUPS	These views show all window groups.
DBA_SCHEDULER_WINDOWGROUP_MEMBERS	These views show the members of all window groups, one row for each group member.
DBA_SCHEDULER_RUNNING_JOBS	These views show state information on all jobs that are currently being run.
SELECT_CATALOG_ROLE	These views show stored event information.

Accessing Audit Trails for Database Events

Event monitors can be created in TWA to watch for certain changes to the tables and indexes in the Oracle database. However, TWA cannot monitor the changes that occur within the Oracle database unless audit trails can be accessed in the database. Oracle database events/auditing is implemented through the standard Oracle Audit Trail feature. Make sure that the database initialization parameter, `AUDIT_TRAIL` is set to 'DB.' Once this parameter is initialized, you must shut down and restart your database before the new setting can take effect. Make sure that the Oracle environment has access to the following standard audit trail views.

Note: These views (except `STMT_AUDIT_OPTION_MAP`) are created by the `CATALOG.SQL` and `CATAUDIT.SQL` scripts.

Table 2

View	Description
STMT_AUDIT_OPTION_MAP	Contains information about auditing option type codes. Created by the <code>SQL.BSQ</code> script at <code>CREATE DATABASE</code> time.
AUDIT_ACTIONS	Contains descriptions for audit trail action type codes.
ALL_DEF_AUDIT_OPTS	Contains default object-auditing options that will be applied when objects are created.

Table 2

View	Description
DBA_STMT_AUDIT_OPTS	Describes current system auditing options across the system and by user.
DBA_PRIV_AUDIT_OPTS	Describes current system privileges being audited across the system and by user.
DBA_OBJ_AUDIT_OPTS USER_OBJ_AUDIT_OPTS	Describes auditing options on all objects. The USER view describes auditing options on all objects owned by the current user.
DBA_AUDIT_TRAIL USER_AUDIT_TRAIL	Lists all audit trail entries. The USER view shows audit trail entries relating to current user.
DBA_AUDIT_OBJECT USER_AUDIT_OBJECT	Contains audit trail records for all objects in the system. The USER view lists audit trail records for statements concerning objects that are accessible to the current user.
DBA_AUDIT_SESSION USER_AUDIT_SESSION	Lists all audit trail records concerning CONNECT and DISCONNECT. USER view lists all audit trail records concerning connections and disconnections for the current user.
DBA_AUDIT_STATEMENT USER_AUDIT_STATEMENT	Lists audit trail records concerning GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM statements throughout the database, or for the USER view, issued by the user.
DBA_AUDIT_EXISTS	Lists audit trail entries produced BY AUDIT NOT EXISTS.
DBA_AUDIT_POLICIES	Shows all the auditing policies on the system.
DBA_FGA_AUDIT_TRAIL	Lists audit trail records for value-based auditing.

Terms to Know

- **Background Job** – A job that runs in the background runs within Oracle rather than inline in TWA . TWA cannot retrieve job output from a job that runs in Oracle.
- **Database Job** – A type of job that can be run from the Oracle Database Adapter if the system is using the 10g version of Oracle. A database job in this case is a SQL statement or SQL block that has been predefined and saved in the database.
- **Database Schema** – A group of related database objects, like tables and indexes, owned by a user account. When defining an event in an Oracle database, a change to a specified schema must be specified as the event.
- **Delimiter** – A character inserted into unformatted job output as a means to separate output values that would otherwise run together.
- **JDBC (Java DataBase Connectivity)** – Definition of a connection between the database and the Client Manager using the Java programming language.
- **PL/SQL (Procedural Language/Structured Query Language)** – The proprietary version of SQL developed by the Oracle corporation to interact with relational databases. One of the job types that can be defined in the Oracle Database Adapter is PL/SQL consisting of a SQL block.
- **Session** – A number used in Oracle to identify the database session within Oracle that the job ran in.
- **SQL (Structured Query Language)** – The most common computer language used with RDBMS (relational database management systems) such as Oracle and Microsoft SQL Server to retrieve and manipulate data.

- **SQL block** – A series of programming instructions or statements in the SQL database language. A block of SQL statements is treated as a single unit similar to a paragraph of sentences.

2

Configuring the Oracle Database Adapter

Overview

The Oracle Database Adapter software is already installed as part of a normal installation of TWA . However, you must perform the following steps to license and configure the adapter before you can define Oracle Database jobs:

- [Licensing an Adapter, page 11](#) – License the connection(s) to the Oracle instance. You cannot define an Oracle Database connection until you have applied the Oracle Database license.
- [Defining Oracle Database Adapter Users, page 12](#) – Create one or more user definitions in TWA with access to the Oracle instance using Oracle user accounts and passwords provided by the Oracle administrator. Users in TWA are authorized to run Oracle Database jobs on behalf of these Oracle runtime users. Defining and securing Oracle Database users with TWA is covered in these sections:
 - [Defining Runtime Users, page 13](#)
 - [Defining the Oracle Database Access Account, page 14](#)
 - [Defining an Oracle Database Security Policy, page 15](#)
- [Defining an Oracle Database Connection, page 16](#) – Define a connection so the master can communicate with the Oracle instance.
- [Defining an Oracle DB Agent List, page 21](#) – Define agent lists to access useful functions not available with individual connections.

See also [Configuring service.props, page 37](#) for information about general and adapter-specific properties that can be set to control things like logging and connection properties.

Licensing an Adapter

Each TWA 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 TWA , 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>Tidal 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.

Defining Oracle Database Adapter Users

There are two types of users associated with the Oracle DB 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 Oracle DB jobs represent those users and passwords required for Oracle DB Authentication. If the Oracle DB server requires authentication based on user and password credentials, these users need to be defined as runtime users.

■ Schedulers

Schedulers are those users who will define and/or manage Oracle DB jobs. There are three aspects of a user profile that grant and/or limit access to scheduling jobs that affect Oracle DB:

- Security policy—Grants or denies add, edit, delete and view capabilities for Oracle DB jobs.
- Authorized runtime user list—Grants or denies access to specific Oracle DB authentication accounts for use with Oracle DB jobs.
- Authorized agent list—Grants or denies access to specific Oracle DB Adapter connections for use when defining Oracle DB jobs.

Defining and securing Oracle Database users with TWA is covered in these sections:

- [Defining Runtime Users, page 13](#)
- [Defining the Oracle Database Access Account, page 14](#)
- [Defining an Oracle Database Security Policy, page 15](#)

Defining Runtime Users

Before you can create Oracle Database jobs, a user with access to the Oracle database instance must be defined to TWA . TWA cannot run any Oracle Database job unless it knows the user name(s) and password(s) defined in the user account that accesses the Oracle database instance. TWA then interacts with the Oracle database as that defined user, exchanging information to monitor and control the execution of the Oracle database jobs through TWA . Other users in TWA are authorized to run the database jobs on behalf of Oracle Database runtime users.

Note: The user of the Oracle Database Adapter for TWA is limited to the same privileges available to the Oracle database user definition created by the database administrator.

The credentials specified for the runtime user are used to store the information about the simple user security context consisting of a user name and password pair and to pass this information to the adapters. This runtime user can be used for database targets when needing database authentication.

To define a runtime user:

1. From the **Navigator** pane, expand the **Administration** node and select **Runtime Users** to display the defined users.
2. Right-click **Runtime Users** and select **Add Runtime User** from the context menu (Insert mode).

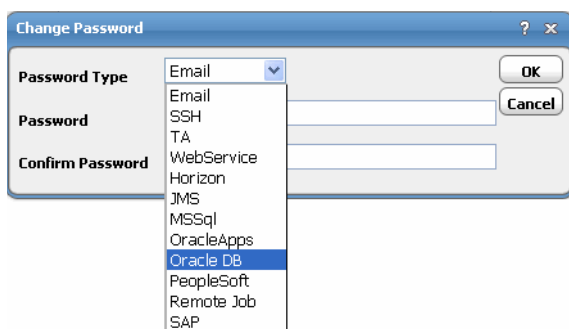
-or-

You can also right-click a user in the **Runtime Users** pane and select **Edit Runtime User** from the shortcut menu (Edit mode).

The **User Definition** dialog displays.

3. If this is a new user definition, enter the new user name in the **User/Group Name** field.
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. To define this user as a runtime user for Oracle DB jobs, click **Add** on the **Passwords** tab.

The **Change Password** dialog displays.



7. Select **Oracle DB** from the **Password Type** list.
8. Enter a password (along with confirmation) in the **Password/Confirm Password** fields.

Only those users with a password specified for Oracle DB will be available for use with Oracle DB jobs. The password might be the same as the one specified for Windows/FTP jobs.

9. Click **OK** to return to the **User Definition** dialog.

The new password record displays on the **Passwords** tab.

The image shows a 'User Definition' dialog box with the 'Passwords' tab selected. The 'User Name' field is 'Oracle DB User'. The 'Full Name' and 'Domain' fields are empty. Below the 'Passwords' tab, there is a table with columns 'Adapter' and 'Password'. The table contains one row: 'Oracle DB Password' with a password of '***'. To the right of the table are buttons for 'Add', 'Edit', and 'Delete'. At the top right of the dialog are 'OK' and 'Cancel' buttons.

Adapter	Password
Oracle DB Password	***

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

Defining the Oracle Database Access Account

The Oracle administrator must provide a user account to be used with TWA . This user account must have access to the database objects needed to run Oracle jobs from TWA . Multiple accounts with this access can be created but it is recommended that the number of user accounts be restricted to a minimum. Once created, other users can access the Oracle database jobs through this Oracle Database user account.

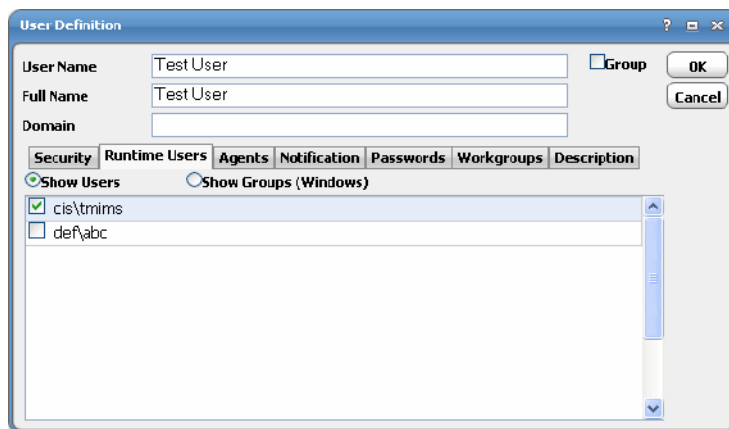
Adding an Oracle Database User

To add an Oracle database user:

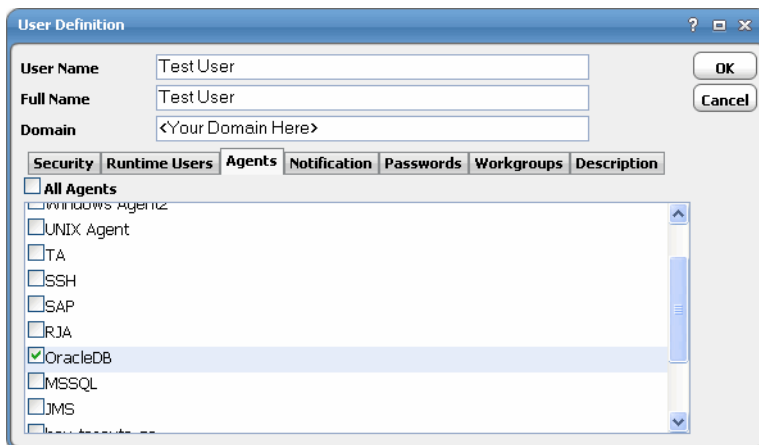
1. From the **Navigator** pane, expand the **Administration** node and select **Interactive Users** to display the defined users.
2. Right-click **Interactive Users** and select **Add Users** from the context menu (*Insert mode*). You can also right-click a user in the **Interactive Users** pane and select **Edit Interactive User** from the shortcut menu (*Edit mode*).

The **User Definition** dialog displays.

3. If this is a new user definition, enter the new user name in the **User/Group Name** field.
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 **Security** page, select the **Other** option and then select the security policy that includes authorization for Oracle DB jobs.
7. Click the **Runtime Users** tab.



8. Select the Oracle DB users that this scheduling user may use for Oracle DB authentication in Oracle DB jobs.
9. Click the **Agents** tab.



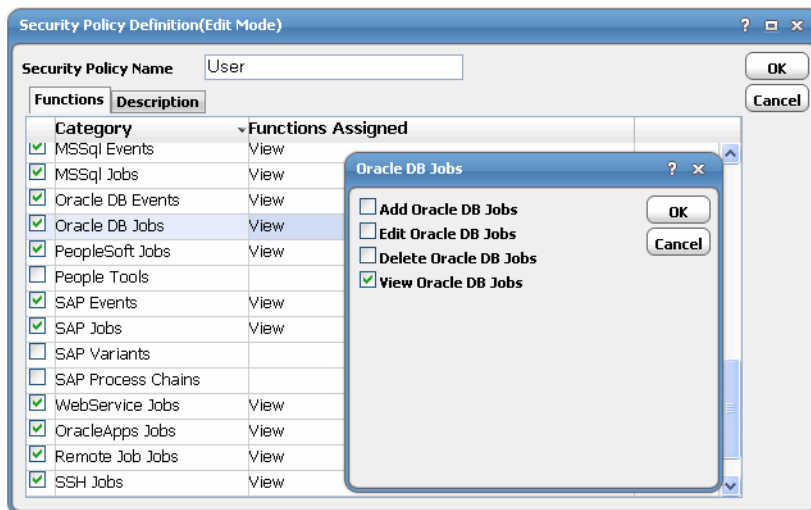
10. Select the check boxes for the Oracle DB connections that this scheduling user can access when scheduling jobs.
11. Click **OK** to save the user definition.

Defining an Oracle Database Security Policy

Access to the Oracle database is controlled by assigning an Oracle Database security policy with specified privileges to designated user accounts. The scheduling administrator should create a new security policy in TWA, that in addition to the normal user privileges includes the capability to create and/or edit SAP jobs. A user whose assigned security policy does not include Oracle Database privileges cannot create and/or run Oracle Database jobs.

To grant Oracle Database access privileges:

1. In the **Navigator** pane, select **Security Policies** to display the **Security Policies** pane.
2. Select a security policy for the Oracle Database job privileges and double-click on it to display its **Security Policy Definition** dialog.



3. Scroll down the list of function categories and double-click on the **Oracle DB Jobs** category to display the available functions.
4. Double-click the category row to select the desired job privileges then click **OK**.

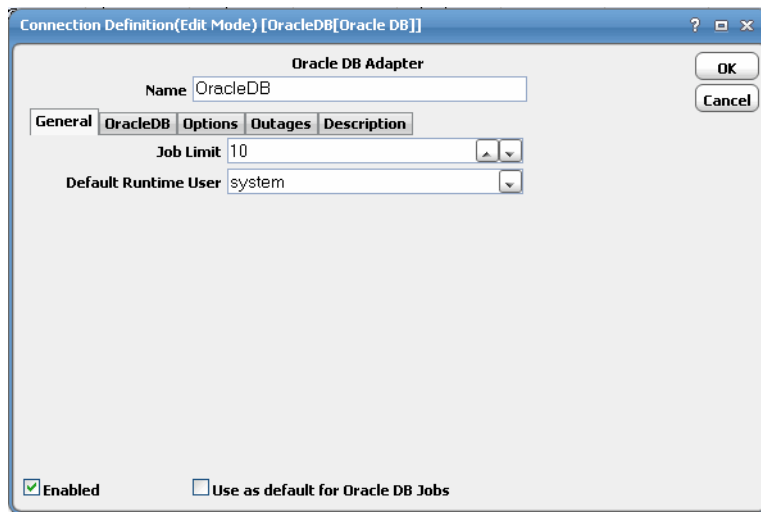
A check mark appears next to the **Oracle DB Jobs** function category indicating that one or more functions are selected within the category.

5. Repeat this process for the **Oracle DB Events** function category to authorize use of Oracle Database events for this security policy.

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

Defining an Oracle Database Connection

You must create a connection to the Oracle database instance(s) before TWA can run your Oracle database jobs. The connection for the Oracle Database Adapter is defined like other TWA connections except it requires configuration information unique to Oracle database instances. Oracle Database agent lists are also supported (see [Defining an Oracle DB Agent List, page 21](#)). A connection is defined through the **Connection Definition** dialog.



This dialog contains the following elements:

- **Name** – Enter a name for the Oracle Database Adapter.
- **Enabled** – Activates or disables 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 adapter that is disabled (or if the connection to the adapter is lost) run to completion; however, their completion status cannot be returned until the connection to the adapter is enabled.
- **Use as default for Oracle DB jobs** – If there are multiple Oracle Database connections defined, selecting this option designates that connection as the default connection to use when running Oracle Database jobs unless a different connection is specified in the job definition. If there is only one Oracle database connection than it is not necessary to select this option.

General Tab

The **General** tab specifies a limit to the number of jobs the agent can work on simultaneously and provides a default runtime user for Oracle database jobs.

- **Name** – The name of the connection being defined. This is the name that is displayed in the TWA console.
- **Job Limit** – Displays the maximum number of jobs that TWA can run on the adapter. In exceptional cases, where a job enters a launch immediately queue, this number is overridden as necessary to launch the job.

Windows agents have a concurrent job execution limitation based on the amount of memory the adapter has. It is recommended that you do not set the job limit to a value higher than this limitation. In most instances, it is recommended to limit adapters to running no more than 80 concurrent jobs. In general, adapters should have five MB of memory for each job that is running concurrently. Some jobs may require additional memory to process depending upon the needs of the job.

- **Default Runtime User** – Specifies a default runtime user for the connection being installed.

Oracle DB Connection Tab

The **Oracle DB Connection** tab defines how TWA will access an Oracle database instance.

Connection Definition(Edit Mode) [OracleDB[Oracle DB]]

Oracle DB Adapter

Name: OracleDB

General | OracleDB | Options | Outages | Description

DB Instance: ora11

Connect User: system

Host Name: 10.88.103.214

Port: 1521

DB Version: 11g

☐ Kill Session on job abort

☐ Override JDBC URL

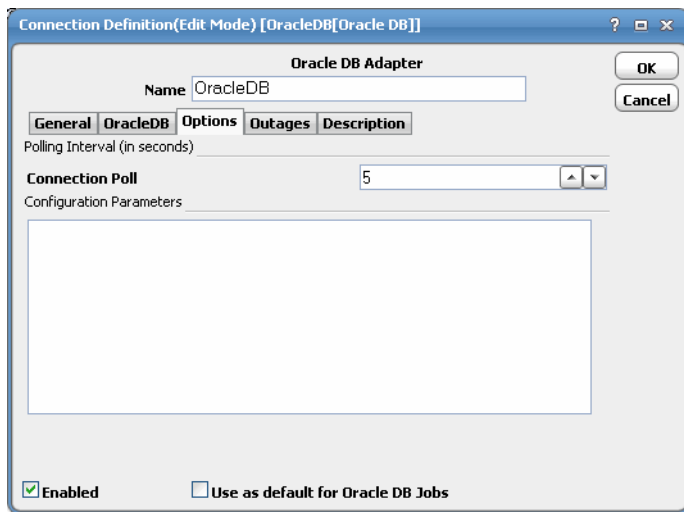
jdbc:oracle:thin:@10.88.103.214:1521:ora11

☒ Enabled ☐ Use as default for Oracle DB Jobs

- **DB Instance** – The name of the Oracle database instance. This is also known as the SID.
- **Connection User** – The domain and name for the runtime user account that accesses the database instance.
- **Host Name** – The name of the machine hosting the Oracle database instance.
- **Port** – The port number used by the master to communicate with the database instance.
- **DB Version** – The version of the Oracle database being used. Only the 10g or 11i versions of Oracle are supported by this adapter.
- **Kill Session on job abort** – Select this option to end the current session if the job aborts.
- **Override JDBC URL** – The basic connection information provided in the earlier fields is used to create the JDBC URL displayed here. If you wish to modify the Oracle connection with additional options, select this option to add custom options to the connection.
- Separate each connection option with a colon (:), e.g., *10.10.10.10:1521:vis*. In most circumstances, the connection requires no other modifications so this option would not be used.

Options Tab

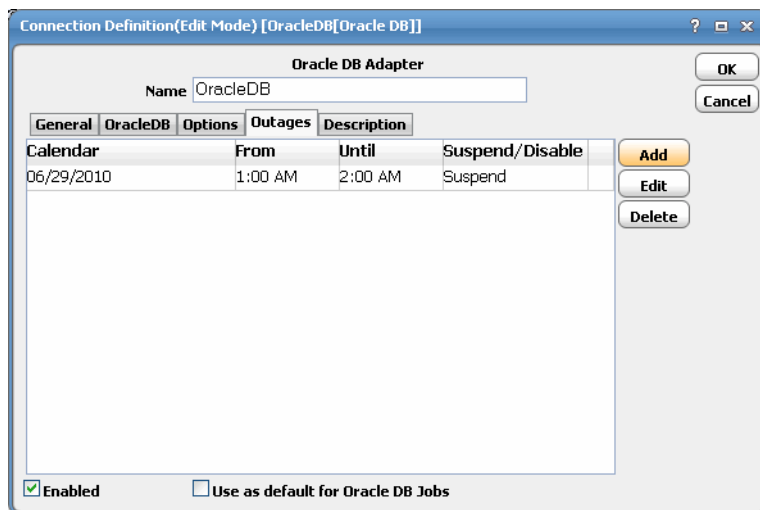
On the **Options** tab configure the polling for the connection and add any configuration parameters as needed.



- Polling Interval (in seconds)
 - **Connection poll** – The interval of time (in seconds) before the master checks that the connection is working.
 - **Configuration Parameters** – Specify any parameters to be used during polling.

Outages Tab

The **Outages** tab is part of the **Connection Definition** dialog for agents and adapters. On the **Outages** tab, create time windows keyed to planned maintenance outages to suspend jobs from being submitted that may not finish before the outage begins. These maintenance outages may be one-time occurrences or they can be assigned a calendar if they occur on a periodic basis. The actual parameters of the outage are defined from the **Outage Definition** dialog.



TWA will use the time window information entered here to manage jobs that may not complete before the outage begins. From a system configuration viewpoint, use the **Defaults** tab to configure how TWA handles jobs at risk of failing due to a planned outage. These settings can be changed as needed for individual jobs from the **Options** tab in the job definition.

- **Calendar** – Displays either the name of the calendar assigned to the outage or the date that it begins. One time outages would not have a calendar but if an outage occurs regularly, it should be assigned a calendar.

- **From** – The start time of the outage window.
- **Until** – The time that the outage ends. If the outage extends more than one day, the date that the outage ends is also displayed.
- **Suspend/Disable** – Indicates if the connection is disabled or suspended during the outage. A suspended connection will automatically enable at the end of the outage while a disabled connection must be manually enabled at the end of the outage. An agent/adaptor connection in an outage window can always be enabled to end the outage.
- **Add** – This button displays the **Outage Definition** dialog to define a new outage. See [Outage Definition Dialog, page 20](#) for how to use this dialog.
- **Edit** – This button opens the **Outage Definition** dialog to modify the parameters of the selected outage.
- **Delete** – This button deletes the selected outage.

Outage Definition Dialog

The **Outage Definition** dialog is displayed by adding a new outage period or editing an existing one from the **Outages** tab in an agent's connection definition. An outage may be a one-time event or it can be defined as a reoccurring outage by assigning the outage a calendar.

This dialog contains the following elements:

- **Specific Date Interval** – Select this option to define a specific time interval for a single outage. Click on the arrow in the **From** field to display a calendar to select a start date for the outage period. In the adjacent field, enter the time that the outage should start on the designated day. In the **Until** field, select the date and time that the outage should end. The **Until** field is unavailable if the **Enable connection at end of outage time window** option is selected.
- **Calendar** – Select this option to define a reoccurring outage on multiple dates by assigning an existing calendar. Once the outage is assigned a calendar of dates, enter the starting time of the outage in the **From** field and an ending time in the **Until** field. The **Until** field is unavailable if the **Enable connection at end of outage time window** option is selected.
- **Enable connection at end of outage time window** – This option enables the connection when the outage window ends. By default this option is selected, suspending a connection during an outage so that when the outage ends, the connection will automatically enable itself. Clearing this option ensures that the connection is disabled at the start of the outage window. The connection will not be enabled until an operator manually enables it. If this option is not selected, then the **Until** options used to define the end of the outage are disabled since the outage cannot end until the connection is enabled manually by a user.

Description Tab

Describes the connection and provides a convenient place for notes about its purpose and use.

Verifying Oracle Database Connection Status

If the TWA master cannot connect to an Oracle database instance or loses its connection, you will see a red status light next to its connection in the **Connections** pane. If the connection's status light is yellow, check to see if the connection is within a predefined outage window as listed on the **Outages** tab of the connection's definition. However, if your client is still connected to your Oracle database instance, you can still define Oracle database jobs.


Defining an Oracle DB Agent List

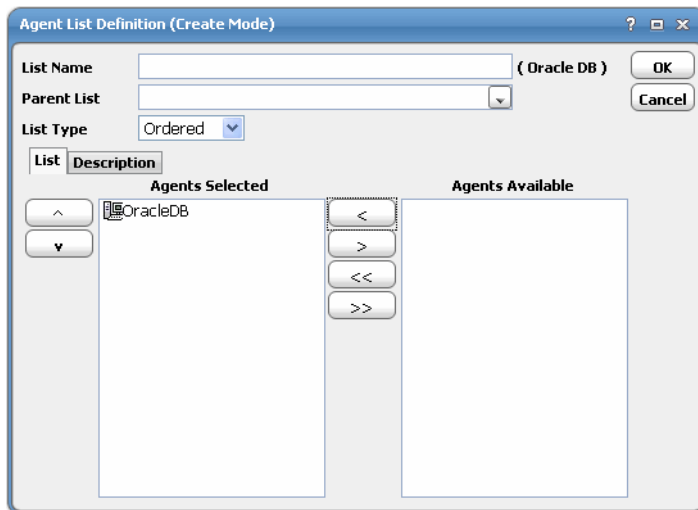
You can assign jobs to run using Oracle DB agent/adaptor lists similar to the way you assign them to individual agents. With Oracle DB lists, however, you have access to useful functions not available with individual connections. You can specify alternate connection(s) to run your jobs if the primary connection is unavailable, you can balance the workload of many jobs among the process servers in the list, or you can broadcast jobs to run on all the servers in the list at the same time.

To define an Oracle DB Agent list:

1. From the **Navigator** pane, select **Definitions>Agent Lists>Oracle DB** to display the **Oracle DB Agent Lists** pane.
2. Right-click on the **Oracle DB Agent Lists** pane and select **Add Agent List** from the context menu.

-or-

Click the **Add** button  on the TWA toolbar to display the **Agent List Definition** dialog.



3. In the **List Name** field, enter the name of the Oracle DB list (up to 60 characters).
4. In the **Parent List** field, you can select a parent list if you want to create nested lists.

If you specify a parent list, the list you are editing will be a child list of the parent and will appear below its parent in the list hierarchy.

5. From the **List Type** list, select the type of list based on how you want your jobs to be assigned to an Oracle DB connection in the list.
 - **Ordered** – In support of high-availability, TWA chooses the first available agent to run the job based on the order that the connections appear in the **Agents Selected** section. For example, if the first (primary) agent in the list is not available, TWA tries the second (alternative) agent in the list. You might want to use this list type for critical jobs – specifying substitute agents in case the primary agent becomes unavailable for any reason.
 - **Random** – Chooses connections from the **Agents Selected** field randomly. This is a crude form of workload balancing. For more advanced load balancing, you should choose the **Balanced** option.
 - **Rotation** – TWA cycles through the list of connections in the **Agents Selected** field and launches jobs assigned to that list in rotation. When the end of the list is reached, the first connection in this list is chosen. This is a form of workload balancing where you can predict where the next job will run.
 - **Balanced** – TWA chooses a connection from the **Agents Selected** section based on lightest load when the job is ready to launch. The **Balanced Agent** list uses Oracle DB connection load information to decide which connection in the list gets to run a job when its dependencies are met. You can choose this option to balance your workload across all the available connections in the list.
6. From the **Agents Available** section on the **List** tab, highlight the appropriate Oracle DB agent.
7. Click the left arrow button to add the selected connection to the list by moving it from the **Agents Available** section to the **Agents Selected** section. You can also drag and drop connections from one section to another.
8. Click **OK**.

3

Using the Oracle Database Adapter

Overview

This chapter describes how to use the Oracle Database Adapter in these topics:

- [Defining an Oracle Database Job, page 23](#)
- [Monitoring Oracle Database Job Activity, page 28](#)
- [Monitoring Changes in the Oracle Database, page 30](#)
- [Troubleshooting Issues with Oracle Database Jobs, page 33](#)
- [Controlling Adapter and Agent Jobs, page 34](#)

TWA can define two different types of Oracle database jobs but the type of job that can be defined is determined by the version of the Oracle database being used. With the release of Oracle 10g, jobs could be defined and saved for reuse. Running Oracle 10g provides the capability to create jobs by selecting predefined jobs stored in the database while retaining the ability to run SQL queries and PL/SQL blocks as before. TWA supports either Oracle 10g or 11i.

With both type of jobs, you can define parameters and specify the values to be used when the job runs. With the predefined jobs, you can run SQL statements before the job in the database runs.

You can start creating and scheduling Oracle database jobs once you have:

- added the Oracle database user to TWA .
- applied the Oracle database license file.
- defined your Oracle database connection(s).

See [Configuring the Oracle Database Adapter, page 11](#) for more information.

Defining an Oracle Database Job

The Oracle database job is defined from the **Oracle DB** tab of the **Oracle DB Job Definition** dialog. You select the type of job to be defined from the **Job Type** list on this tab.

The job type is either a PL/SQL type or a database type. If the Oracle Database Adapter selected on the **Run** tab of the job definition has a connection to an Oracle 11i database then only the **PL/SQL** option is available. If the selected adapter is connected to an Oracle 10g instance, both types of jobs can be defined.

You can add an Oracle database job using the right-click menu within the **Jobs** pane and selecting the **Add Oracle Database Job** option. You can also **Edit**, **Copy** and **Delete** an existing Oracle database job. If you add Oracle database jobs to a TWA job group, items common between the job group and the Oracle database job are inheritable. However, unless the parent group has an Oracle database agent assigned to it, you must clear the **Inherited** option and select the correct Oracle database connection.

These sections explain how to create both Oracle database job types:

- [Defining a PL/SQL Job, page 24](#)
- [Defining a Database Job, page 26](#)

Defining a PL/SQL Job

PL/SQL jobs can either run in TWA or can run in the Oracle database. The output from jobs that run inline can be retrieved as XML or as simple text that can be formatted for display or printing. Job output cannot be retrieved from those jobs that run within the Oracle database.

To define a PL/SQL job:

1. From the **Navigators** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click in either the **Navigators** or **Jobs** pane and select **Add Oracle Database Job** from the context menu to display the **Oracle DB Job Definition** dialog.

Oracle DB Job Definition [OracleDB]

Oracle DB Job Name: OracleDB

Job Class: [dropdown]

Parent Group: [dropdown]

Owner: Schedulers

OK Cancel

OracleDB Schedule Run Dependencies Resources Job Events Options Run Book Notes History Images

Agent/Adapter Information

Agent/Adapter Name: OracleDB[Oracle DB]

Agent List: [dropdown]

Runtime User: system

Tracking

Use: ☒ 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

Last Modified 06/28/2010 11:33:55

3. In the **Job Name** field, enter a name for your job. You can choose a name of up to 50 characters in length.

The **Job Name** is an identifier for TWA only. All of the other job definition information, such as **Job Class**, **Owner** and **Parent Group**, is the same as any other TWA job.

If you are putting your Oracle database job into a job group, note that unless the parent group selected has an Oracle database agent connection assigned, you **must** clear the **Inherited** option.

4. On the **Run** tab, in the **Agent/Adapter Name** list, select an Oracle database instance. If no Oracle database instances are available from the menu, verify that an Oracle database connection has been defined.

Note that the name in this list represents all the information in the Oracle database connection definition, including the version number, runtime user, database instance, database user ID, etc. To use different parameters, modify the connection definition, or define a new Oracle database connection with a new agent name and then use that agent.

5. In the **Runtime User** list, select an Oracle database user.

6. In the **Tracking** section, select a method to determine a job's completion status. While a standard TWA job has an exit code option, an Oracle database job has a **Request Status** option. If you select **Request Status** then a job has completed normally when the job status in Oracle database is **Normal** or **Warning**.

A **Warning** status is much like a **Normal** status except that warning notices were generated as the job completed successfully. To differentiate between a **Normal** status and a **Warning** status, you can select the **Translate "Warning" status to "Completed Abnormally"** option. Selecting this option means that a job that completes with a **Warning** status is converted to a **Completed Abnormally** status.

For more information on the other **Tracking** and **Duration** options, refer to the **Run** tab section in *Jobs* chapter of the TWA *User Guide*.

Note: If you wish to change the default job output settings for individual job instances, select the Options tab and modify the Job Output Option setting. Subsequent job instances will revert to the default job output setting specified in the System Configuration dialog.

7. Click the **Oracle DB Job Definition** tab.

Oracle DB Job Definition [OracleDB]

Oracle DB Job Name: OracleDB

Job Class: [Dropdown]

Parent Group: [Dropdown]

Owner: Schedulers

Job Type: PL/SQL

ID	Statement
1	select * from msglog
2	

Please type one statement at a time:

select * from tab

Buttons: Add, Delete, Edit, Clear

Run as DBMS job: ☐ Output as XML: ☐ Output format: Align columns Delimiter: [Text] Include headers: ☐

Enabled: ☒ Last Modified: 06/28/2010 11:33:55

8. Select a job type from the **Job Type** list.

Select whether the Oracle database job being defined is a PL/SQL statement or a predefined database job. If the adapter selected on the **Run** tab of the job definition uses the Oracle 9i version of the database, only the PL/SQL job option is available. Both, database jobs and PL/SQL statements are supported with the 10g database version.

9. On the **SQL** tab, enter the SQL statement that the PL/SQL job will run.

Multiple SQL statements and SQL code blocks can be entered here. The output from the multiple SQL statements (if the job is configured to save output) is displayed in the order that the statements run.

Each parameter used in the SQL statement will be listed on the **Parameters** tab where its value can be specified. A parameter will be recognized only if written in the proper format. Each parameter name must begin with a colon (:) and end with a space; for example, :URL . If a parameter used in the SQL statement is not listed on the **Parameters** tab, verify that it is correctly formatted in this field.

10. Click **OK**.

Defining a Database Job

To define a Database job:

1. From the **Navigator** pane, select **Definitions>Jobs** to display the **Jobs** pane.
2. Right-click in either the **Navigator** or **Jobs** pane and select **Add Oracle Database Job** from the context menu to display the **Oracle DB Job Definition** dialog.
3. In the **Job Name** field, enter a name for your job. You can choose a name of up to 50 characters in length.

The **Job Name** is an identifier for TWA only. All of the other job definition information, such as **Job Class**, **Owner** and **Parent Group**, is the same as any other TWA job.

If you are putting your Oracle database job into a job group, note that unless the parent group selected has an Oracle database agent connection assigned, you **must** clear the **Inherited** option.

4. On the **Run** tab, in the **Agent/Adapter Name** list, select the appropriate instance. If no Oracle database instances are available from the menu, verify that an Oracle database connection has been defined.

Note that the name in this list represents all the information in the Oracle database connection definition, including the version number, runtime user, database instance, database user ID, etc. To use different parameters, modify the connection definition, or define a new Oracle database connection with a new agent name and then use that agent.

5. In the **Runtime User** list, select an Oracle database user.
6. In the **Tracking** section, select a method to determine a job's completion status. While a standard TWA job has an exit code option, an Oracle database job has a **Request Status** option. If you select **Request Status** then a job has completed normally when the job status in Oracle database is **Normal** or **Warning**.

A **Warning** status is much like a **Normal** status except that warning notices were generated as the job completed successfully. To differentiate between a **Normal** status and a **Warning** status, you can select the **Translate "Warning" status to "Completed Abnormally"** option. Selecting this option means that a job that completes with a **Warning** status is converted to a **Completed Abnormally** status.

For more information on the other **Tracking** and **Duration** options, refer to the **Run** tab section in *Jobs* chapter of the *TWA User Guide*.

Note: If you wish to change the default job output settings for individual job instances, select the **Options** tab and modify the **Job Output Option** setting. Subsequent job instances will revert to the default job output setting specified in the **System Configuration** dialog.

7. Click the **Oracle DB Job Definition** tab.

Oracle DB Job Definition [OracleDB]

Oracle DB Job Name: OracleDB

Job Class: [dropdown]

Parent Group: [dropdown]

Owner: Schedulers

OK Cancel

OracleDB Schedule Run Dependencies Resources Job Events Options Run Book Notes History Images

Job Type: Database Job

Database Job: Database Job, PL/SQL

Preprocessing SQL Parameters

ID	Statement
1	select count(*) from dba_tables

Please type one statement at a time:

^ v

Add Delete Edit Clear

☒ Enabled

Last Modified 06/28/2010 11:33:55

8. Select **Database Job** from the **Job Type** list.

9. From the **Database Job** list, select the name of the DBMS job predefined in Oracle.

This field is only available if the **Database Job** option was selected in the **Job Type** field. The list will show all the jobs available for the schema owner. Wildcard characters cannot be used in this field.

10. On the **Preprocessing SQL** tab, enter the SQL code block that should run before the job runs.

These SQL statements will run inline before the database job runs. The SQL code specified here can be used to configure the job environment and any of the parameters needed by the database job that was selected in the **Database Job** field. If the SQL syntax is incorrect, an error message is displayed when the user attempts to save the job. This is an optional field as no preprocessing SQL is necessary for a database job to run successfully.

Multiple SQL statements and SQL code blocks can be entered here. The output from the multiple SQL statements (if the job is configured to save output) is displayed in the order that the statements run.

11. Click the **Parameters** tab to specify a value for each parameter used in the SQL statement.

Parameters displayed here are contained within the SQL block that was entered on the **SQL** tab. A parameter must be written in the designated format to be recognized. A parameter that is not formatted correctly will not be displayed here. Each parameter name must begin with a colon (:) and end with a space; for example, :URL. The parameter name cannot be modified from this tab. This tab will display the parameter as it appears in the SQL block. Select the parameter and enter a value in the **Value** column for the parameter. You can either enter a specific value or use the **Variables** button to select a predefined variable.

12. Click **OK**.

Monitoring Oracle Database Job Activity

TWA monitors the progress of an Oracle database job as it runs within the Oracle database. The current status of a job running in Oracle is displayed in real-time in the **Job Activity** pane. Details about the Oracle job instance are collected in the **Job Details** dialog. From the job's details, a user can view the original settings of the job including the SQL code and its parameters, make any needed modifications and rerun the job. The Oracle Database Adapter can monitor for designated changes within the Oracle database and once detected, trigger predefined actions.

When running a job, TWA assigns each job instance a unique ID number to identify that instance from all other job instances. This ID number is called the job number and is displayed in the **Job No.** column in the **Job Activity** pane. The number of the session that the job ran in within Oracle is used by TWA to track the job inside of Oracle. This session number is referenced in TWA as the External ID so that users can match the job record in TWA with the same job's record in Oracle. This External ID number can be found on the **Status** tab of the **Job Details** dialog and is listed in the **Ext ID** column in the **Job Activity** pane. The session number on the **Run Info** tab will match the external ID.

Oracle Database Job Details

In the **Job Activity** pane, you can study the details of any Oracle database job that is currently running or that has completed. Double-click on the job record listed in the **Job Activity** pane to display the **Job Details** dialog for an Oracle database job. The **Job Details** dialog for an Oracle database job has many of the same tabs of information as any other type of job but it does have two tabs of information unique to Oracle database jobs.

Job Details [OracleDB (1)]

Job Name: OracleDB (1) Job No. 58 OK

Status Audit Log Output Dependencies Resources Override Runbook Notes History OracleDB RunInfo Cancel

Current Status

Status Completed Normally Reruns 0

Est. Start Time 11:29 PM (6/28/2010) Disable Carryover ☐

Act. Start Time 11:29 PM (6/28/2010)

Est. Duration 0 min 15 s

Act. Duration 0 min 1 s

Job Owner qatest

Scheduled By Calendar

Exit Code 0

External ID 11291501

Print Defaults

One tab, the **Oracle DB** tab, contains the original definition of the job while the other tab, the **Run Info** tab, displays the job as it was defined for the last or current run. The options on the **Oracle DB** tab will vary to reflect the type of Oracle database job that was run.

Oracle DB Tab

The **Oracle DB** tab in the **Job Details** dialog displays the SQL code used in the job definition. This **Oracle DB** tab has the same options as the **Oracle DB** tab in the job definition but what displays varies according to the type of job that was defined. Only the text field where the SQL statements and their parameters are displayed can be modified. The job can then be rerun with the changes.

Job Details [OracleDB (1)]

Job Name: OracleDB (1) Job No.: 58

Status Audit Log Output Dependencies Resources Override Runbook Notes History OracleDB RunInfo

Job Type: PL/SQL

SQL Parameters

ID	Statement
1	select * from msglog

Please type one statement at a time:

Run as DBMS job ☐ Output as XML ☐ Output format: Align columns Delimiter: Include headers ☐

Print Defaults

Only the text fields on the **SQL** and **Parameters** tabs can be edited. The other text fields are read-only. From this tab you can change the SQL blocks and/or the parameters used in the last job run and rerun the job.

Job Details [OracleDB (1)]

Job Name: OracleDB (1) Job No.: 106

Status Audit Log Output Dependencies Resources Override Runbook Notes History OracleDB RunInfo

Job Type: Database Job

Database Job:

Preprocessing SQL Parameters

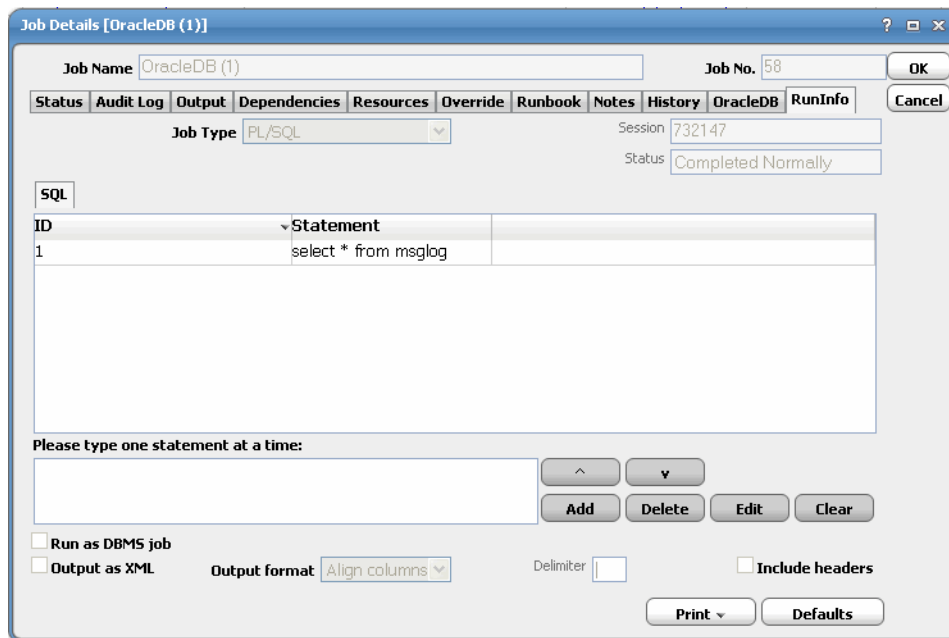
ID	Statement
1	select(*)count from dba_tables

Please type one statement at a time:

Print Defaults

Run Info Tab

The **Run Info** tab shows the SQL code associated with the job instance as defined in the job's definition. This tab serves as a reference for the SQL code used the last time the job ran. While this tab may display the same SQL code as appears on the **Oracle DB** tab, this tab is read-only. Any changes to a job can be made on the **Oracle DB** tab and compared with the "original" SQL on the **Run Info** tab.



Oracle Database Job Output

Oracle database jobs do not produce output that can be captured and stored. The Oracle Database Adapter can display the audit log generated by the database job session. This audit log is displayed on the **Output** tab of the **Job Details** dialog. However, the Oracle audit log cannot be recorded if the audit trail views needed to track the job as it runs were not enabled. A list of the audit trail views needed for tracking the changes within the Oracle database are in the [Accessing Audit Trails for Database Events, page 8](#).

Monitoring Changes in the Oracle Database

Certain changes in the Oracle database can be defined as events. When the specified condition occurs in the database, TWA recognizes it as an event and triggers an action that is associated to that condition. Actions can be created to run Oracle database jobs in response to the predefined conditions.

Defining Oracle Database Events

The **Oracle DB Event Definition** dialog is displayed whenever you add or edit an Oracle database event. An Oracle database event is the detection of a change within a monitored Oracle datatable. This change can be the addition, modification or deletion of rows, indexes and tables.

You must assign a calendar to the event from the **Schedule** tab. If needed, you can configure the monitor to operate only during certain time periods or leave the monitor in operation at all times.

To define Oracle database events:

1. In the **Navigator** pane, select **Definitions>Events** to display the **Oracle DB Events** pane.
2. Right-click and select **Add Event** from the context menu to display the **Oracle DB Event Definition** dialog.

OracleDB Event Definition (Edit Mode) [OracleDB_E]

Event Name: OracleDB_E

Owner: qatest

Calendar: OracleDB

Calendar Name: Daily

From: 06/28/2010 To: 07/05/2010

Offset: 0

Forecast

☒ 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	00:44:06	00:44:06

☒ Public ☐ Enabled

Buttons: Insert..., Edit..., Delete, Clear

This dialog contains the following elements:

- **Event Name** - The name of the event, up to 30 characters.
- **Owner** - The user or workgroup with full control over the event. By default, the owner is the creator of the event.
- **Public** - Makes the event available to all users of TWA .
- **Enabled** - Makes the event active and available for use. When this option is cleared (signifying disabled or inactive), you can still assign the event to a job, but it will not be triggered when the job runs.

Oracle DB Tab

The **Oracle DB Monitor** tab of the **Oracle DB Event Definition** dialog designates the datatable and indexes in the Oracle database to be monitored and defines the condition that will be considered an event. Once the designated change to the datatable 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 text fields are read-only to provide information about the event that occurred.

OracleDB Event Definition (Edit Mode) [OracleDB_E]

Event Name: OracleDB_E

Owner: qatest

OK Cancel

OracleDB Schedule Associated Action(s) Description Trigger History

DB Connection: OracleDB

DB Event: Row(s) added to table

DB schema: TIDAL

Table Name: MSGLOG

By User: TIDAL

☒ Public ☐ Enabled

This tab contains the following elements:

- **DB Connection** - Specifies which database connection to monitor from the list of defined connections.
- **DB Event** - Designates one of the listed database changes from the list as the event trigger.

The following events are available:

- Index modified
- Table modified
- Index created
- Table created
- Index deleted
- Table deleted
- Row(s) added to table
- Row(s) modified in table
- Row(s) deleted from table

- **DB Schema** - Specifies the database schema to be monitored. Initially this value defaults to the connection user. The value entered here must be all uppercase without any spaces.
- **Table Name** - The name of the datatable 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.

- **By User** – The name of the user making the change in the database. The value entered here must be all uppercase with without any spaces. The wildcards characters * and ? can be used in this text field.

Schedule Tab

From the **Schedule** tab, you define when the event monitor will operate. Refer to the *Schedule tab* section in the *Events* chapter of the *Tidal Workload Automation User Guide* for a description of the options on this tab of the **Oracle DB Event** dialog.

Associated Action(s) Tab

This tab associates action(s) with the Oracle database event. An event must have an associated action before it can be enabled. For information about the options on this tab, refer to *Associated Action(s) tab* in the *Events* chapter of the *Tidal Workload Automation User Guide*.

Description Tab

This is a free text field where you can provide a description and notes about this particular event, up to 255 characters.

Trigger History Tab

The **Trigger History** tab lists all of the instances when the event was triggered during the trigger history retention period. This retention period is configured on the **Default** tab of the **System Configuration** dialog. For information about the options on this tab, refer to *Trigger History tab* in the *Events* chapter of the *Tidal Workload Automation User Guide*.

Troubleshooting Issues with Oracle Database Jobs

Some of the more basic conditions to check whenever encountering any difficulty processing Oracle database jobs are provided below.

Event monitors do not work correctly.

Verify that all of the required audit trail views are enabled. The audit trail views that are required for data event monitoring are listed in the section, [Accessing Audit Trails for Database Events, page 8](#).

There are no database jobs in the Job Name field.

If you are running Oracle 10g and no jobs display in the **Job Name** field, check the status of the connection to the Oracle database. A valid connection is required to retrieve and display the list of predefined jobs from the Oracle database. Even if there is no connection though, the name of the job can be typed into the text field and the job definition will run when the connection is reestablished (as long as the job name is correctly spelled).

No Oracle jobs will run.

Verify that there is a good connection to the Oracle database.

- If there is a red status light next to the Oracle database connection in the Connections pane, verify that the connection information in the connection definition is accurate.
- If the Oracle connection's status light is yellow, check if the connection is in a planned outage. In the **Job Activity** pane, the Oracle jobs will show **Agent Outage** as their status.
- If the status light is green, verify that your user profile is on a runtime user list of a user account authorized to run Oracle database jobs.

The PL/SQL job completed successfully but there is no job output.

Verify that the **Save Output Option** on the **Options** tab of the job definition is not set to discard the job output.

Verify that the **Run as DBMS Job** option on the **Oracle DB** tab of the job definition is not selected. Jobs that run as DBMS jobs run within Oracle where TWA cannot retrieve the job output. Clear the option if selected and rerun the job to retrieve the job output.

The job output is difficult to read because all of the values run together.

On the **Oracle DB** tab of the job definition, select an output format other than “Raw.” Raw format has no format so the values run together. If you prefer not to format the job output, enter a character in the **Delimiter** field to be used to separate the job output values.

Oracle 9i has job execution errors.

Whenever a job fails, information about the failure is recorded in a trace file and in the alert log. Oracle writes message number ORA-12012 and includes the job number of the failed job.

If a job returns an error as Oracle attempts to execute it, Oracle automatically tries to execute it again. The first attempt to rerun the job occurs after one minute, the second attempt to rerun the job occurs after two minutes, the third attempt after four minutes and so on, doubling the time intervals between each attempt. If the job still fails to complete after 16 attempts, Oracle quits resubmitting the job and marks the job as “broken.” Between the subsequent attempts to run the job, the operator has the opportunity to diagnose and correct the condition causing the job to fail.

A job needs to be canceled.

A job running in Oracle 10g can be canceled by the Oracle DBA using the STOP_JOB procedure call. This procedure call will set the state to stopped in Oracle.

The syntax for the STOP_JOB procedure call is:

```
DBMS_SCHEDULER.STOP_JOB(
  job_name IN VARCHAR2
  force IN BOOLEAN DEFAULT FALSE);
```

Replace the **job_name** with the name of the job or job class. This can be a comma-delimited list. For a job class, the SYS schema should be specified.

If the name of the job class is specified, the jobs that belong to that job class are stopped. The job class is not affected by this call.

Replace **force** with either **TRUE** or **FALSE**.

- If **FALSE**, TWA tries to gracefully stop the job using an interrupt mechanism. This method gives control back to the slave process, which updates the status of the job in the job queue to **STOPPED**. If this process fails, an error is returned.
- If **TRUE**, TWA immediately terminates the job slave. Oracle recommends using the **TRUE** parameter only after an attempt to use the **FALSE** parameter fails.

Note: The force option requires the MANAGE SCHEDULER system privilege.

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**.

4

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.

Table 1

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.
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.

Table 1

Property	Applicable Adapter(s)	Default	What It Controls
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.
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.

Table 1

Property	Applicable Adapter(s)	Default	What It Controls
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.
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.

