Cisco Workload Automation Oracle Applications Adapter Guide

Version 6.3

First Published: August, 2015
Last Updated: September 7, 2016

Contents

Preface

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

Audience

This guide is for administrators who install and configure the Oracle Applications 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 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 for any updates.

Obtaining Documentation and Submitting a Service Request

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

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

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

Document Change History

The table below provides the revision history for the *Cisco Workload Automation Oracle Applications Adapter Guide*.

Version Number	Issue Date	Reason for Change
6.1.0	October 2012	New Cisco version.
6.2.1	June 2014	Available in online Help only.
6.2.1 SP2	June 2015	Configuration provided in the *TES Installation Guide*; usage provided in online Help only.
6.2.1 SP3	May 2016	Consolidated all Oracle Applications Adapter documentation into one document. Added new section for Installing the Oracle Application Bridge Using a Custom APPS User, page 15 and appendices to support this process. Added a new section for Working with Multiple Organization Access (MOAC), page 50.
6.3 BETA	June 2016	Rebranded “Cisco Tidal Enterprise Scheduler (TES)” to “Cisco Workload Automation (CWA)”. Added the Configuring SSL and HTTPS Protocol for the Oracle Applications Adapter, page 21 section.

1

Introducing the Oracle Applications Adapter

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

Overview

The CWA adapter for Oracle Applications integrates Oracle Applications into CWA using a concurrent manager bridge.

The Oracle Applications Adapter from CWA uses Net*8 (SQL*NET) to connect directly to Oracle databases when accessing Oracle Applications.

Oracle databases compile and store procedures and functions in units called packages. The Oracle Applications Adapter uses Oracle's packages and other packages customized by CWA in combination with SQL statements to integrate the CWA job scheduler with the Concurrent Manager process that monitors and controls the Oracle Applications job. The Concurrent Manager monitors and responds to the data stored within the Oracle database using the packages available to it.

The customized packages supplied by CWA must be compiled in Oracle Applications before a connection between CWA and Oracle Applications can be established. An error occurs in CWA if you try to establish a connection to an Oracle Applications instance before the proper customized packages are installed on the designated Oracle Applications instance.

Any inserting and updating to the standard tables of Oracle Applications is done using standard APIs present in the Oracle Applications database. Nothing is deleted from the standard Oracle Applications database just as no database schema objects are modified.

The Oracle Applications adapter is part of the regular Scheduler installation. It can only be used though if the appropriate license is applied to Scheduler. Though no installation is required for the adapter, you must install a Bridge component that links the adapter to the Oracle Applications program. After the Bridge is installed, you must perform some configuration procedures for the adapter.

Monitoring Oracle Applications Jobs

Scheduler uses a different identification system to track jobs than the system used by Oracle Applications. Each application has its own system of tracking a job instance by assigning an identification number. In the **Job Activity** pane, the **Job No.** column refers to the number assigned by Scheduler to the job instance. This number is how Scheduler identifies the Oracle Applications job and its associated requests. Scheduler does not track individual requests within a

request set comprising a job. It tracks the request set as a whole and thus does not have control over individual requests within a request set. The ID number in Scheduler is assigned to the first request and applies to all of the requests within the request set.

Oracle Applications, on the other hand, does not treat a request set as a whole. It assigns each request within the request set an individual request ID number. The request ID number assigned by Oracle to the first request of a request set is displayed in the **Ext ID** column (External ID) of the **Job Activity** pane. The ID numbers of each individual request are listed on the **Oracle Apps** tab of the **Job Detail** dialog of the job.

To achieve control over individual requests and to assign dependencies to a single request within a request set, break down the request set into separate requests for each job and organize the single requests into a job group in Scheduler.

Multi-Organization Access (MOAC)

Multi-Org or multiple organization access (MOAC) is the ability to access multiple operating units from a single application responsibility.

In Release 11i, you had to enter or process data for multiple operating units; you had to login to different responsibilities because each responsibility could only access one operating unit. For example, if you were managing Payables for Sweden, Norway, and Finland, you needed to define three different responsibilities. In Release 12, you can create a Security Profile and assign as many operating units as you required. You can tie that security profile to a single responsibility using a profile option called MO: Security Profile. For example, you can assign the security profile to the EMEA Payables responsibility to allow that responsibility to process invoices across all three operating units.

The CWA Oracle Applications Adapter supports multiple organization access for Oracle Application Server R12 and later. See Working with Multiple Organization Access (MOAC), page 50 in the Using the Oracle Applications Adapter, page 35 chapter for how to configure an Oracle Applications job definition for multiple organization access.

Terms to Know

- **Application** – Group of modules that perform a particular enterprise func- tion, such as Accounts Receivable, Accounts Payable, General Ledger, etc.
- **Concurrent Manager** – Concurrent processing facility that monitors and runs your requests/processes/reports without tying up the users terminal.
- **Concurrent Process** – Task that is in process in the concurrent manager. Each time you submit a task to the concurrent manager, then a concurrent process is created to run that task. Many concurrent processes can be running simultaneously without tying up the users terminal or requiring user intervention.
- **Concurrent Queue** – The list of concurrent requests that are in the concurrent manager awaiting processing. The concurrent manager can be set up to have multiple concurrent queues, each queue having its own list of concurrent requests waiting for processing. Each Concurrent Request can be setup to run in specific queues or in any queue, then the Concurrent Manager assigns the request to a Concurrent Queue accordingly.
- **Concurrent Request** – A request to run a process/report/job in the Concurrent Manager. Once a Concurrent Request is submitted to the Concurrent Manager, then the Concurrent Manager takes over the processing of the request without any intervention from the user or tying up the users terminal.
- **Internal Concurrent Manager (ICM)** – The ICM oversees all of the other worker concurrent managers and assigns requests to the appropriate concurrent manager. The ICM monitors the health of the other concurrent managers, restarting them if necessary, and follows the status of requests that run on the other worker concurrent managers.
- **List of Values** – List of valid values available for a particular parameter in Oracle Applications.
- **Module** – Programs and/or procedures that implement a particular business function within an application, such as forms/screens, reports, concurrent programs and subroutines.

- **Open Interface** – A function that lets you import or export data from other systems through open interface tables and processes provided by Oracle.
- **Parameter** – Variables used in concurrent processes to restrict information or determine the form of the report.
- **Report Group** – List of concurrent programs (reports, processes, etc.) that is grouped together based upon a particular function. Each Oracle Applications Responsibility has at least one Report Group assigned to it, though some responsibilities can have multiple Report Groups. If a responsibility has multiple Report Groups associated with it, then it will have multiple menu entries to launch the Submit Concurrent Request screen, one for each Report Group the responsibility has access to.
- **Report Options** – Options for sorting, formatting, selecting and summarizing the information on a report. Not all Oracle reports use report options. Report Set – Group of reports that are always run together.
- **Request Set** – One or more sequential stages each of which include one or more concurrent programs that may or may not run in a particular sequence. Links can be maintained between stages for warning, completion and error handling.
- **Responsibility** – Security feature that restricts Oracle Application users to a specific set of screens and reports based on their user login name.
- **Stage** – Represents the sequence for concurrent programs in a request set (report set). Each stage of a request set has a sequence number, stage name, concurrent program name with On Success, On Error and On Warning links as well as a critical flag that determines whether the stage affects the entire report set.

Minimum Software Requirements

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

The minimum software requirements for the Oracle Applications Adapter for CWA are:

- Oracle Applications software 11.5.8, 11.5.9, 11.5.10, 12 – 12.04, 12.1.1, 12.1.3, 12.2.0
- Oracle database 10g 11g, 11g R2, 12c
- MKS Toolkit 7.1 or higher installed on all of the Forms Servers
- CWA version 6.0 or later
- Cisco Workload Automation Adapters require Java 7. (Refer to *Cisco Workload Automation Compatibility Guide* for further details).

2

Configuring the Oracle Applications Adapter

Overview

The Oracle Applications Adapter software is installed as part of a standard installation of Cisco Workload Automation. However, before the Oracle Applications Adapter can be used, the following configuration procedures must be completed:

- Installing the Oracle Applications Bridge, page 11—Install the Oracle Applications Bridge (install_11i or install_R12).
- Licensing an Adapter, page 16—License the Oracle Applications Adapter to work with CWA.
- Defining a Security Policy, page 17—Define a security policy for running Oracle Applications jobs.
- Configuring Users for the Oracle Applications Adapter, page 17—Define a user account that can access the Oracle database.
- Authorizing Users, page 19—Authorize users to run Oracle Applications jobs using the database user account.
- Configuring SSL and HTTPS Protocol for the Oracle Applications Adapter, page 21—Obtain, export, and import the security certificate into the Java keystore.
- Defining an Oracle Applications Connection, page 24—Define a connection to the Oracle database.
- Configuring Oracle for Scheduler, page 31—Configure Oracle to work with CWA.

See Configuring service.props, page 57 for information about general and adapter-specific properties that can be set to control things like logging and connection properties.

Installing the Oracle Applications Bridge

There are two components to the Oracle Applications adapter. One component is the Oracle Applications adapter itself while the other part is a bridge component that provides a link between the adapter and the Oracle Applications program. The Oracle Applications adapter is part of the normal CWA installation and does not require a separate installation. However, the Bridge component does require installation and the procedure to install it is described in this section.

The Oracle Applications Bridge is comprised of various PL/SQL stored procedures and forms used to pass job parameters to the Oracle database. The Bridge component of the Oracle Applications adapter is not part of the regular CWA installation and requires a separate installation procedure.

Note: If you need to create a custom APPS user for CWA, then you also need to use a different installation process for the Oracle Applications Bridge as described in Installing the Oracle Application Bridge Using a Custom APPS User, page 15.

This section describes the prerequisites and processes for the Oracle Applications Bridge:

Prerequisites for Installing the Oracle Applications Bridge

The following prerequisites must be completed before installing the Oracle Applications Bridge:

- The user must be logged on to Windows/Unix as the application owner (usually **`applmgr`**).
- Run the application environment file (usually *Appsora.env* under **`$APPL_TOP`**) in the current shell.
- Grant the execute privilege on *sys.dbms_obfuscation_toolkit* file to the **APPLSYSPUB** user. This package is used by the Bridge to encrypt and decrypt the data. To grant this privilege, connect to the database as system (or **SYSDBA**) and from the SQL prompt, enter:

```
SQL>grant execute on sys.dbms_obfuscation_toolkit to applsyspub;
SQL>commit;
Create tablespace for the Table and Index spaces before starting installation. To configure the
tablespaces to autoextend:
SQL>CREATE TABLESPACE sabdg_data DATAFILE '/do1/oracle/testdata/sabdg_data.dbf' SIZE 100M
AUTOEXTEND ON NEXT 20M MAXSIZE UNLIMITED;
SQL>CREATE TABLESPACE sabdg_index DATAFILE '/do1/oracle/testdata/sabdg_idx.dbf' SIZE 50M AUTOEXTEND
ON NEXT 10M MAXSIZE UNLIMITED;
SQL>commit;
```

- Create the tablespace for the Table and Index spaces before starting installation. To configure the tablespaces to autoextend:

```
SQL>CREATE TABLESPACE sabdg_data DATAFILE '/do1/oracle/testdata/sabdg_data.dbf' SIZE 100M
AUTOEXTEND ON NEXT 20M MAXSIZE UNLIMITED;
SQL>CREATE TABLESPACE sabdg_index DATAFILE '/do1/oracle/testdata/sabdg_idx.dbf' SIZE 50M AUTOEXTEND
ON NEXT 10M MAXSIZE UNLIMITED;
SQL>commit;
```

- While no existing **`$APPL_TOP`** objects/files are modified when installing the Bridge, three new objects/files that start with **SABDG** are created.

  Note: The database schema names used above are only examples. You can use your own names for the database schemas.

Installing the Bridge for 11i or R12

The batch file that installs the Bridge requires the following parameters:

- **APPS user** (or equivalent) – The equivalent Apps user in the Oracle Applications program.
- **APPS password** – The password of the Apps user used to access Oracle Applications.
- **Data tablespace** – The name of the data tablespace (sabdg_data).
- **Index tablespace** – The name of the index tablespace (sabdg_index).
- **TNS name** – The TNS string to connect to the database (Windows only)

- **Temp tablespace** – The temporary tablespace for the user **SABDG**.
- **System password** – The database user system password. This is required for when the installation process creates the **SABDG** user in the database to own tables, sequences and indexes.

Use these parameters when running the batch file to install the Bridge. The installation and upgrade procedures for both Windows and Unix forms server are described next.

Initial Installation or Upgrade Windows Forms Server

To install the 11i or R12 Bridge (Windows):

1. Copy the 11i or R12 Bridge files from the **\OraAppsBridge\Windows** directory in the installation DVD-ROM to a temporary directory on the forms server. The temporary directory must be on the same drive as the **APPL_TOP**.
2. On the command line, using the listed parameters, enter:

```
install_11i <APPS User> <APPS Password> <Data Tablespace> <Index Tablespace> <TNS Alias Name> <Temp
Tablespace> <System Password>
```

-or-

```
install_R12 <APPS User> <APPS Password> <Data Tablespace> <Index Tablespace> <TNS Alias Name> <Temp
Tablespace> <System Password>
```

To upgrade the 11i or R12 Bridge (Windows):

1. On the command line, using the listed parameters, enter:

```
upgrade_11i <APPS User> <APPS Password> <Data Tablespace> <Index Tablespace> <TNS Alias Name> <Temp
Tablespace> <System Password>
```

-or-

```
upgrade_R12i <APPS User> <APPS Password> <Data Tablespace> <Index Tablespace> <TNS Alias Name>
<Temp Tablespace> <System Password>
```

Initial Installation or Upgrade Unix Forms Server

Copy the 11i or R12 Bridge *TdlOraAppsBdg.tar* file from the **/OraAppsBridge/Unix** directory in the installation DVD-ROM to a temporary directory on the forms server.

To install the 11i or R12 Bridge (Unix):

1. Extract files from *TdlOraAppsBdg.tar* file

```
tar xvf TdlOraAppsBdg.tar
```

2. From the temporary directory where you copied the Bridge files, at the cursor, enter:

```
chmod 777*
```

3. At the cursor, using the listed parameters, enter:

```
sh ./install_11i.sh <APPS User> <APPS Password> <Data Tablespace> <Index Tablespace> <Temp
Tablespace> <System Password>
```

-or-

```
sh ./install_R12.sh <APPS User> <APPS Password> <Data Tablespace> <Index Tablespace> <Temp
Tablespace> <System Password>
```

To upgrade the 11i or R12 Bridge (Unix):

1. Extract files from *TdlOraAppsBdg.tar* file

   ```
   tar xvf TdlOraAppsBdg.tar
   ```

2. From the temporary directory where you copied the Bridge files, at the cursor, enter:

   ```
   chmod 777*
   ```

3. At the cursor, using the listed parameters, enter:

   ```
   sh ./upgrade_11i.sh <APPS User> <APPS Password> <Data Tablespace> <Index Tablespace> <Temp
   Tablespace> <System Password>
   ```

   -or-

   ```
   sh ./upgrade_R12.sh <APPS User> <APPS Password> <Data Tablespace> <Index Tablespace> <Temp
   Tablespace> <System Password>
   ```

   Note: If you have a multi-tier architecture of Oracle Applications containing multiple form servers, then the Bridge for Oracle Applications must be installed on only one forms server and upgraded on the rest of the form servers to ensure distribution of the Bridge forms.

Verifying Successful Bridge Installation or Upgrade

Installation and upgrade procedures can be verified by checking a log file that is created when the Bridge is installed. This log file, called *Verify_post.log*, is created in the same directory where the Bridge was installed.

Open the *Verify_post.log* file.

There are three values displayed in the log file:

- TOT(Total Objects)
- VAL(Valid Objects)
- INV(Invalid Objects)

The TOT and VAL values should read 36.

The INV value should read 0.

If the values displayed in the log file are the proper values, then installation/upgrade was successful. Any deviation from these values indicates that the installation/upgrade was unsuccessful.

Uninstalling the Bridge

To uninstall the Bridge component of the Oracle Applications adapter, you must delete all of the Bridge objects, the Bridge owner and all of the forms on the forms server. The procedures to delete the Bridge owner and its objects are the same for both the Windows and Unix platforms but the procedures for deleting forms from the forms server differ for each platform.

To delete the Bridge objects (Windows and Unix):

1. Login as **Apps** to the apps database.

2. Run the *sabdg_drobj.sql* script that is found in the **\OraAppsBridge\Windows\sabdg_obj.sql** file in the Windows directory on the installation DVD-ROM.

To delete the Bridge owner (Windows and Unix):

1. Login as **system** to the apps database.
2. Drop user sabdg cascade.

To delete all forms on the forms server:

Windows:

```
rm %AU_TOP%\forms\US\SABDG\*.fmb
rm -r %FND_TOP%\forms\US\sabdg
rm -r %APPL_TOP%\sabdg
```

Unix:

```
rm $AU_TOP/forms/US/SABDG/*.fmb
rm -r $FND_TOP/forms/US/sabdg
rm -r $APPL_TOP/sabdg
```

Installing the Oracle Application Bridge Using a Custom APPS User

If your security policy does not allow the APPS user the authority to access Oracle Applications jobs from CWA, then you can create a custom APPS user and then use it to install the Oracle Applications Bridge. The custom APPS user can then be used to allow CWA to access the Oracle Applications database jobs.

To install the Oracle Applications Bridge using a custom APPS user

1. In the Oracle Applications server, create these table spaces:

```
CREATE SMALLFILE TABLESPACE "SABDG_DATA" DATAFILE '+DATA_ERPCRP2/sabdg_data01.dbf' SIZE 100M
AUTOEXTEND ON NEXT 20M MAXSIZE UNLIMITED LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT
AUTO
CREATE SMALLFILE TABLESPACE "SABDG_INDEX" DATAFILE '+DATA_APS/sabdg_idx01.dbf' SIZE 100M AUTOEXTEND
ON NEXT 20M MAXSIZE UNLIMITED LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO

sqlplus '/ as sysdba'
SQL> grant execute on sys.dbms_obfuscation_toolkit to applsyspub;
Grant succeeded.
SQL> commit;
Commit complete.
SQL>
```

2. Install the Bridge using the custom APPS user named TIDAL:
   a. Create the TIDAL user using the following script:

```
CREATE USER "TIDAL" PROFILE "DEFAULT" IDENTIFIED BY "tidal12345" DEFAULT TABLESPACE "SABDG_DATA"
TEMPORARY TABLESPACE "TEMP" QUOTA UNLIMITED ON "SABDG_DATA" QUOTA UNLIMITED ON "SABDG_INDEX"
ACCOUNT UNLOCK;
```

   b. Run the tidal.sql as the system user from tidal.sql Commands, page 61 in Appendix A.
   c. Run the sys_grant.sql as the sys user from sys_grant.sql Commands, page 76 in Appendix A.
   d. Run the tidal_syn.sql from tidal_syn.sql Commands, page 74 in Appendix A as the TIDAL user created in step a.
   e. Copy the TdlOraAppsBdgUnix.tar / TdlOraAppsBdgWindows.zip to any location in the OracleApps server and unzip it.
   f. Install the Bridge using this command:

```
$ ./install_R12.sh ADT_TIDAL adt123 SABDG_DATA SABDG_INDEX TEMP1 <system user (system) password >
Installation Output:
TNS Ping Utility for Linux: Version 10.1.0.5.0 - Production on 10-SEP-2013 09:38:04
Copyright (c) 1997, 2003, Oracle.  All rights reserved.
Used parameter files:
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION= (ADDRESS_LIST= (LOAD_BALANCE=YES) (FAILOVER=YES)
(ADDRESS=(PROTOCOL=tcp)(HOST=Tidalsoft.com)(PORT=1541))
(ADDRESS=(PROTOCOL=tcp)(HOST=Tidalsoft.com)(PORT=1541))) (CONNECT_DATA= (SERVICE_NAME=erpcrp2)))
OK (0 msec)
Installing Tidal OraApps Bridge
************* Installing Oracle Bridge ***************
Unable to install Tidal OraApps Bridge
INS:21 All objects did not get created
```

g. Install the custom sabdg_global package as the TIDAL user using the sabdg_global_new.sql in Appendix B–sabdg_global Package, page 77.

h. Make the OracleApps connection from CWA master using the custom Apps user.

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.

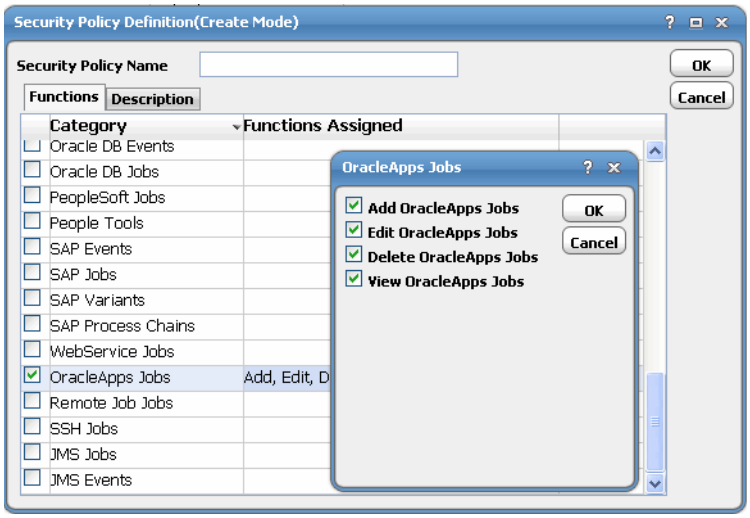
5. To validate that the license was applied, select **Registered License** from **Activities** main menu.

Defining a Security Policy

You can create a new security policy or modify an existing security policy in CWA that includes the authority to add, edit, view and delete Oracle Applications jobs. In the **Security Policy Definition** dialog, there is an **OracleApps Jobs** category.

To grant Oracle Application access privileges:

1. In the **Navigator** pane, select **Security Policies** to display the **Security Policies** pane.
2. Select a security policy for the Oracle Application 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 **OracleApps Jobs** category to display the available functions that can be assigned to the security policy:



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

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

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

For more information about the Security Policy Definition dialog, refer to the chapter on User Administration in your CWA *User Guide*.

Configuring Users for the Oracle Applications Adapter

The user of the Oracle Applications adapter for Scheduler is limited, when running Oracle jobs, to the same privileges available to the Oracle Applications user definition created by the Oracle Applications Administrator.

There are two types of users that must be defined before you can submit Oracle Application jobs:

- **Database user (APPS)** – This is the user definition created in Scheduler with the APPS authority to access the Oracle instance. This user adds other users defined in Scheduler to its runtime user list. The APPS user grants or revokes the access of every other user in CWA to Oracle Applications jobs. Only Scheduler users that belong to the APPS user's runtime list can run or create jobs in Oracle Applications.

  If your security does not allow the APPS user the authority to access Oracle Applications jobs from CWA, then you might need to create a custom APPS user. This process is described in this section.

- **Runtime user** – These users are defined in CWA without access rights to run Oracle Applications jobs on their own. These other users in CWA may have varying levels of security privileges but none of them can run Oracle Applications without being on the APPS runtime user list. Once these users are added to the APPS runtime user list they can create and/or modify existing Oracle Applications jobs.

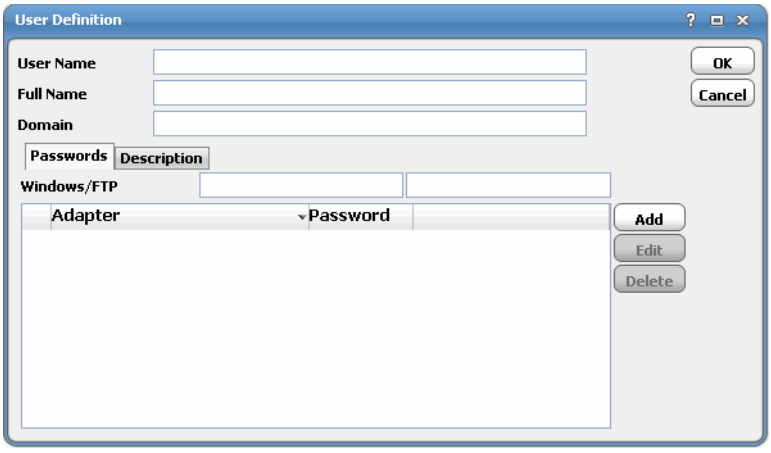
Note: If necessary, you can create multiple APPS users with different passwords. In the user definition, use the Domain field to distinguish one APPS user from another. The Full Name field can describe which APPS user is which.

This section contains these topics:

Adding the APPS database user to CWA

To add the APPS database user:

1. From the **Navigator** pane, select **Administration>Runtime Users** folder to display the **Users** pane.
2. Either click the **Add** button in the toolbar, or right-click in the **Users** pane and select **Add Runtime User** from the context menu to display the **User Definition** dialog.

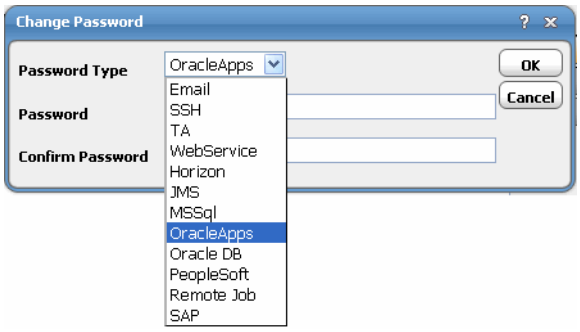


3. In the **User Name** field, enter the user name for the database user account provided by the Oracle Applications System Administrator for CWA use.

   It is recommended to use **APPS** for easy identification and security purposes. No value is necessary in the **Full Name** field or the **Domain** field.

4. On the **Security** tab, select the **Runtime User Only** option.

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



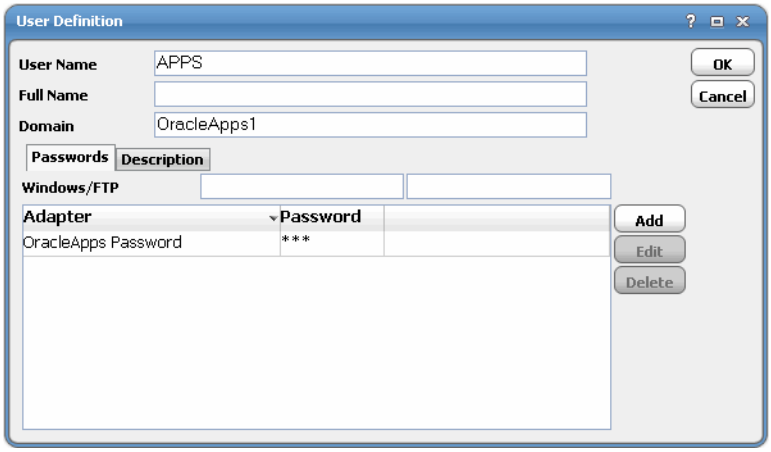
6. Select **OracleApps** from the **Password Type** list.

7. Enter a password (along with confirmation) in the **Password/Confirm Password** fields.

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

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

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



9. Click **OK**.

If the **OracleApps Password** and **Confirm Password** values do not match, you must re-enter the password in both columns. For more information about the **User Definition** dialog, refer to the chapter on Users in your CWA *User Guide*.

Authorizing Users

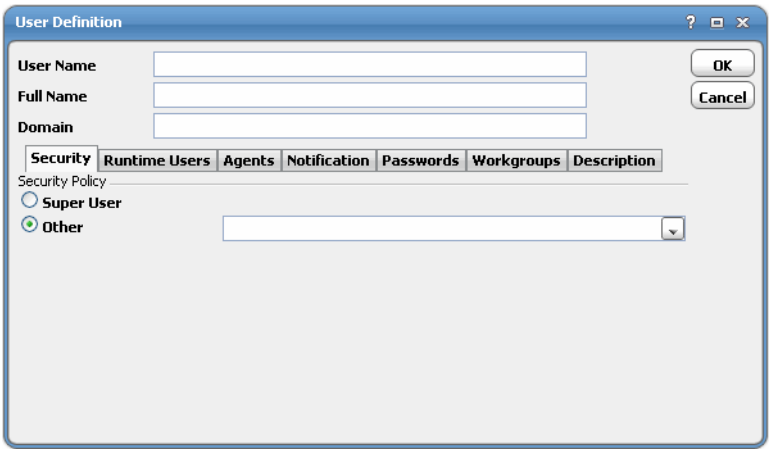
If a user is to work with Oracle Applications jobs, the user needs the defined Oracle Applications user added to their runtime users list. When the user runs an Oracle Applications job, the user is recognized and allowed to access the Concurrent Manager datatables as needed.

Note: The runtime users defined in CWA are subject to the same limitations placed on the Oracle Applications user in the Oracle Applications environment.

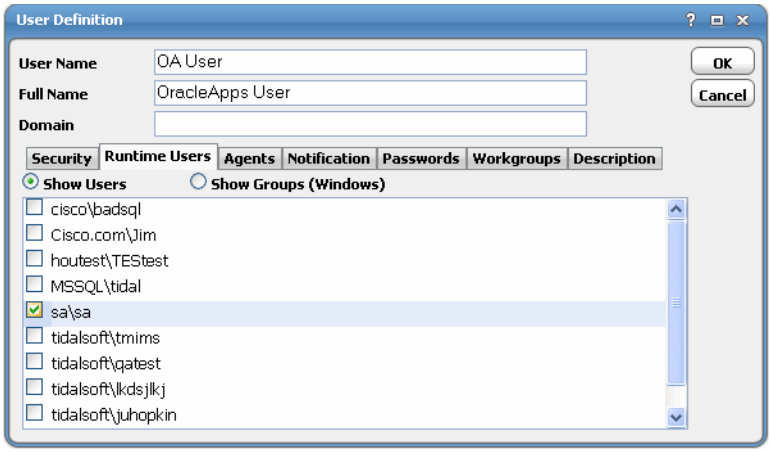
Adding an Oracle Applications User as a Runtime User

To add a user as a runtime user:

1. From the **Navigator** pane, select **Administration>Runtime Users** folder to display the **Users** pane.
2. Double-click the name of the user who will be running Oracle Applications jobs to display the **User Definition** dialog.



3. Click the **Runtime Users** tab.



4. In the **Runtime Users** list, select the check box(es) next to the name(s) of the Oracle Applications user(s) you created in Adding the APPS database user to CWA, page 18 and click **OK**.

Verifying TNS File Formatting (Oracle Users Only)

You must ensure that the TNS file is formatted so that CWA can access it.

To verify the TNS file formatting:

1. In the *master.props* file in the *config* directory on your master machine, specify the location of the *tnsnames.ora* file.
2. On a separate line in the *master.props* file, enter:

   ```
   TNSNamesFile=<directory pathname>
   ```

   For example:

```
TNSNamesFile=C:/oracle/ora90/network/ADMIN/tnsnames.ora
```

Note: If you are using Unix, then there is no drive letter at the beginning of the pathname. The directory pathname is case-sensitive.

Solution Open the TNS file and using the same name specified in the **TNS Name** field of the **Connection Definition** dialog for the Oracle Applications adapter, do a search to find that name.

Solution When you find the definition of the TNS, replace the service name or if the using the SID verify that the parameters specifying the **protocol**, **host** and **port** are provided. The parameters are case-sensitive and must be in capital letters as in the following example:

```
#SID for oraapps
oraapps2k =
 (DESCRIPTION=
   (ADDRESS_LIST=
     (ADDRESS= (PROTOCOL=TCP)(HOST=oraapps2k)(PORT=1523))
   )
   (CONNECT_DATA=
      (SID=vis)
   )
)
```

Configuring SSL and HTTPS Protocol for the Oracle Applications Adapter

It is recommended that Oracle Applications Web servers be configured to use SSL via the HTTPS protocol for Data Services/Data Integrator. If your environment is configured to use HTTP, you can skip this section.

For complete instructions on configuring Oracle Applications servers to use the HTTP or HTTPS protocol, refer to the Oracle Applications documentation that ships with the product.

There are three steps to configuring SSL via HTTPS:

- Obtaining the Security Certificates, page 21
- Exporting the Security Certificates, page 22
- Importing the Target Server Certificates into a Java Keystore, page 23

Obtaining the Security Certificates

From a Windows desktop, you can obtain a security certificates for each target Oracle Applications server using the Microsoft Internet Explorer Certificate Cache.

Note: Although other procedures are available for obtaining the required certificates, the procedure below can be performed from your Windows desktop.

Note: You need to carry out the following instructions only if your server certificate is generated in-house (that is, self-signed) or if your server certificate is signed by a Certification Authority that is not trusted by the version of Java you are using. Alternatively, if your adapter connection fails by reporting the error “unable to find valid certification path to requested target” you need to carry out the following instructions.

To obtain target Oracle Applications server security certificates

1. Open the Internet Explorer browser and navigate to the following dispatch URL (replacing the server name and port as it applies to your environment).

```
https://<adminHost:adminPort>/admin/servlet/webservices
```

where `adminHost` is where the Data Integrator Administrator is installed and `adminPort` is the port the Data Integrator Administrator is listening on.

A **Security Alert** message displays.

2. Click **View Certificate** to open the **Certificate** dialog.
3. Click **Install Certificate**.
4. On the **Certificate Import Wizard Welcome** panel, click **Next**.
5. On the **Certificate Store** panel, use the default option **Automatically select the certificate store based on the type of certificate** and click **Next**.
6. On the **Completing Certificate Import Wizard** panel, click **Finish**.

   A **Security Warning** message displays informing you that you are about to install a certificate from a certification authority.
7. Click **Yes** to continue with the certificate installation.

   A message stating *The import was successful* displays.
8. Click **OK** to close the message and return to the **Certificate** dialog.
9. Click **OK** on the **Certificate** dialog. You can close your browser now.
10. Repeat the process for each Oracle Applications server that you want to connect to with the Oracle Applications adapter.

Exporting the Security Certificates

After you have obtained the security certificates for the target servers, you must export them from the Internet Explorer cache to a local directory.

To export the cached certificates to a local directory

1. On the local computer, create the following directory for the certificates:

   ```
   C:\OA-Certs
   ```
2. In Internet Explorer, select **Tools>Internet Options**.
3. On the **Internet Options** dialog, select the **Content** tab.
4. In the Certificates area, click **Certificates**.
5. On the **Certificates** dialog, select the **Trusted Root Certification Authorities** tab to display the list of trusted certificates. This list should contain the certificates for the target servers that were obtained in the previous procedure (see Obtaining the Security Certificates, page 21).
6. Scroll through the list of certificates to find the certificates.
7. Perform the following procedure for each target server certificate:
   a. Select the certificate and click **Export** to launch the Certificate Export Wizard.
   b. On the **Welcome** panel, click **Next**.
   c. On the **Export File Format** panel, use the default option DER encoded binary X.509 (.CER) and click **Next**.
   d. On the **File To Export** panel, enter the complete path to the *OA-Certs* directory and a unique name for the certificate:

```
C:\OA-Certs\servername.cer
```

e. Click **Next**.

f. On the **Completing the Certificate Export Wizard** panel, click **Finish** to complete the export.

A message stating The export was successful displays.

g. Click **OK** to close the message box.

8. After all target server certificates have been exported, click **Close** to exit the **Certificates** dialog.
9. Click **OK** to close the **Internet Options** dialog.

Importing the Target Server Certificates into a Java Keystore

You must now import the target server certificates into a local Java keystore.

Note: These instructions assume that a JRE or JDK is in your system PATH.

To import certificates into a Java keystore

1. Open a Windows **Command Prompt** window.
2. Change to the directory where the certificates are stored by entering the following commands:

```
c:
cd \OA-Certs
```

3. Use the Java keytool utility to import a certificate. The following syntax is used:

```
keytool -import -file <certificate-filename> -alias <servername>-keystore
<your_trusted_keystore.keystore-filename>
```

For example:

```
C:\OA-Certs>keytool -import -file sdkpubs01.crt -alias sdkpubs01 -keystore BOXI.keystore
```

4. When prompted to create a password for the keystore, enter a password at the prompt. The keystore utility displays the certificate information.
5. At the **Trust this certificate? [no]** prompt, type **`yes`** and press **Enter**. The certificate is imported into the **<your_trusted_keystore>.keystore** keystore and the following message displays:

```
Certificate was added to keystore
```

6. Repeat this procedure for each target server.
7. Navigate to the following folder where the Cisco Workload Automation Oracle Applications Adapter is installed and create a new directory named **config**:

```
<install dir>\master\services\{88EBA24D-7B9A-4EAC-855B-F29D99CE37E9}\config
```

8. Create a text file named *service.props* in the \config directory located under the Adapter's GUID directoryif it doesn't already exist.
9. Open the *service.props* text file and add the following line:

```
Keystore=c:\\OA-Certs\\<your_trusted_keystore>.keystore
```

(Note the use of escaped backslashes for Windows directories).

See also Configuring service.props, page 57 for information about general and adapter-specific properties that can be set to control things like logging and connection properties.

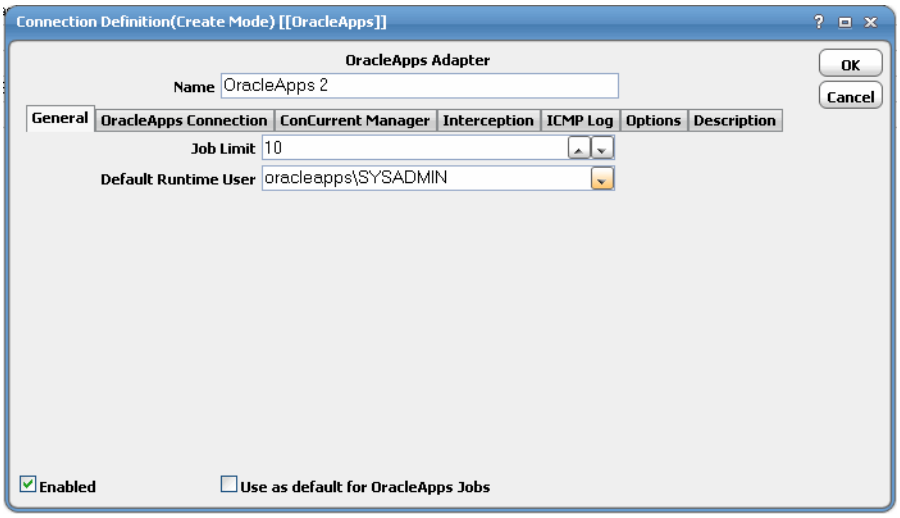
Defining an Oracle Applications Connection

You must configure a connection between CWA and the Oracle Applications Bridge. Follow the procedure below to display the **Connection Definition** dialog to define the connection for the adapter. Enter the required information on each tab of the dialog.

To define a connection:

1. From the **Navigator** pane, select **Administration>Connections** to display the **Connections** pane.
2. Click the **Add** button to display a menu that lists the various types of connections that you can define and select the **OracleApps Adapter** option, or right-click in the **Connections** pane and select **Add Connection>OracleApps Adapter** from the context menu to display the **Connection Definition** dialog.

   Note: You can add an Oracle Applications connection only if you have purchased a license and installed the bridge supplied with the CWA software.

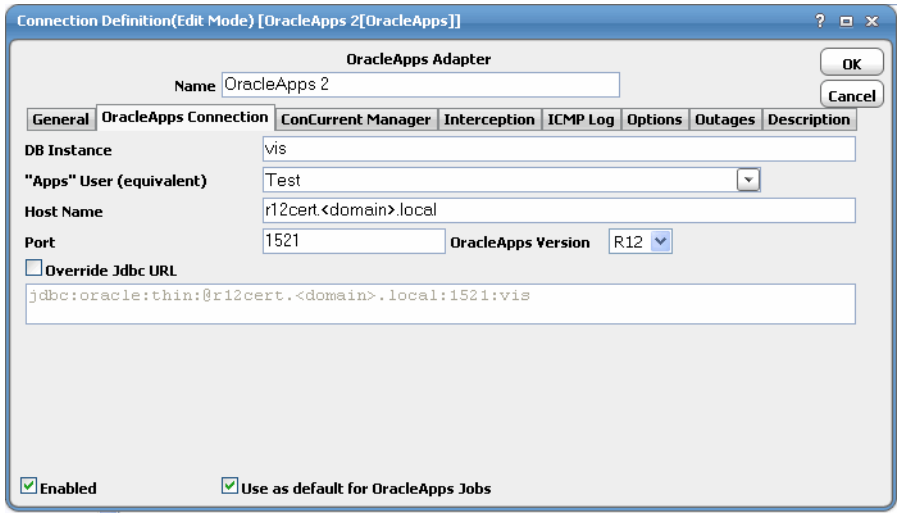


   Note: You cannot modify the connection parameters once the connection is created. You must delete the connection and create a new connection.

3. In the **Job Limit** field of the **General** tab, enter the maximum number of jobs that should run on the server concurrently. This number is checked and enforced by CWA prior to job launch.
4. From the **Default Runtime User** field, select an Oracle Applications user to be used as the default runtime user for Oracle Applications jobs.

   Do not use the Oracle database user; this list refers to an application user. The user entered here is automatically supplied as the runtime user in CWA Oracle Applications job definitions.

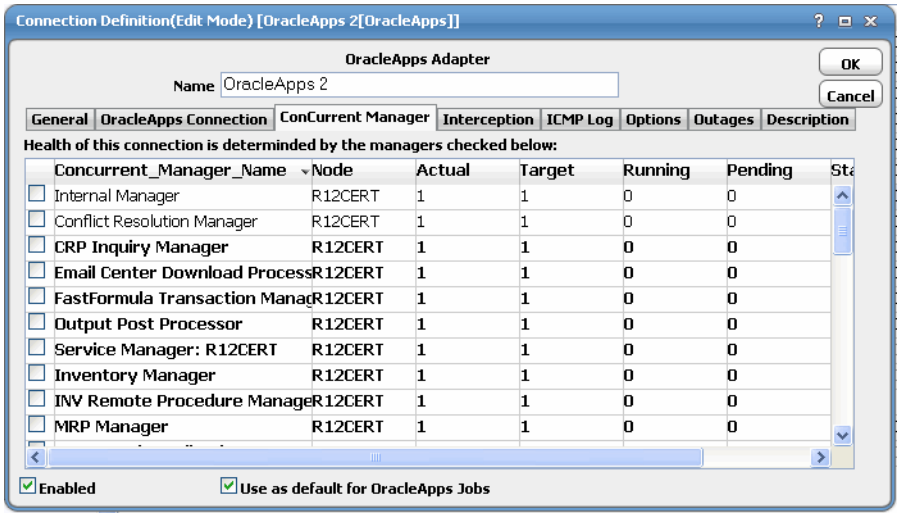
5. Click the **OracleApps Connection** tab.



The **OracleApps Connection** tab supplies CWA the information needed to logon to the Oracle database instance.

6. Jn the **DB Instance** field, enter a database instance.
7. From the **APPS User** list, select the database user to be used when the CWA master connects to the Oracle Applications database.

   If you followed the recommendation in Adding the APPS database user to CWA, page 18, **APPS** is the proper user.
8. In the **Host Name** field, enter the host IP address or host name.
9. In the **Port** field, enter the port number.
10. From the **OracleApps Version** list, select the version of the application being used.
11. Click the **Concurrent Managers** tab.



The **Concurrent Managers** tab defines how CWA determines the health of the Oracle Applications connection. The health of an Oracle Applications connection is determined by monitoring the status of specified concurrent managers. Only the concurrent managers that have been selected are monitored. The status of the concurrent managers not selected are irrelevant to the status of the defined Oracle Applications connection. You should select only the concurrent managers that control processes affecting the jobs you create.

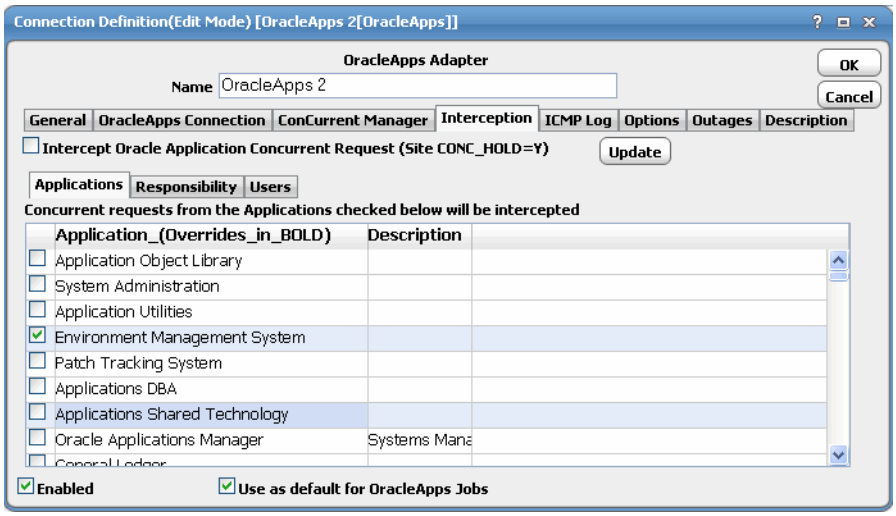
The following concurrent managers are monitored by default:

- Internal Manager
- Conflict Resolution Manager (if active)
- Standard Manager

There are several columns of information on the **Concurrent Managers** tab describing the current status of the listed concurrent managers.

- **Concurrent Manager Name** – Name of the listed concurrent manager.
- **Node** – Name of the machine that the Concurrent Manager is located on.
- **Actual** – Displays how many defined instances (or threads) of the listed concurrent manager are actually running. Any concurrent managers with a 0 value in the Actual column are considered to have an unhealthy status. However, only a 0 value for a *selected* concurrent manager is of concern.
- **Target** – Displays how many instances (or threads) of the listed concurrent manager are defined (running or not).
- **Running** – Displays how many requests are currently running.
- **Pending** – Displays how many requests are queued up and waiting to run as soon as resources are available to process the job.
- **Status** – Displays the status of the listed concurrent manager. The status column should be blank unless there is a problem. Any messages in the status column are usually error messages concerning a problem with the concurrent manager. Possible status values are:
  - Activate concurrent manager
  - Deactivate concurrent manager
  - Deactivated
  - Restart concurrent manager
  - Target node/queue unavailable
  - Terminate requests and deactivate manager
  - Terminated
  - Update concurrent manager environment information
  - Verify concurrent managers status

12. Click the **Interception** tab.



The **Interception** tab configures criteria for managing unscheduled jobs that are submitted to Scheduler. Ad-hoc jobs bypass the scheduled management of jobs that were compiled into the day's schedule. An excessive number of ad-hoc jobs impacts the running of jobs within the production schedule by consuming system resources. By imposing criteria on the concurrent requests that are submitted to the concurrent manager outside of CWA, you can create queue filters and use queue limits to control when these ad-hoc requests actually run. Ad-hoc jobs that do not match the interception criteria are not intercepted.

Using the Oracle Profile option, **CONC_HOLD**, the Applications adapter can impose an increasingly more detailed interception criteria when evaluating unscheduled jobs. The four levels of interception criteria ranging from the broadest based criteria to the most detailed criteria are: **Site**, **Application**, **Responsibility** and **User**. Through the careful selection of the options on this tab, you can create an interception profile as specific as your system requires.

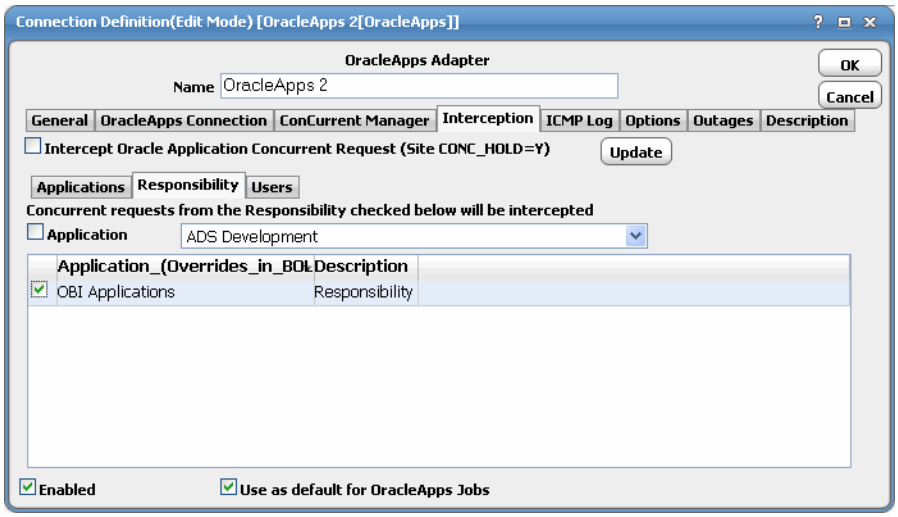
This tab contains the following elements:

- **Intercept Oracle Application Concurrent Requests (Site CONC_HOLD=Y)** – Selecting this option enables the interception feature at its broadest definition. Any concurrent request that comes from any application, from any responsibility and from any user will be held. This Site setting will enable all of the interception criteria ensuring that all unscheduled jobs are directed to a designated queue where their entry into production can be managed by Scheduler. By default, this option is selected during installation. After selecting the Site setting, you can still go to the other options on the tabs of this tab to specify that concurrent requests originating from designated applications, responsibilities or users can bypass interception.

- **Update** – Clicking this button refreshes the information on this tab to display the current interception configuration after any changes are made to this tab.

- **Applications tab** – Lists the various applications available within Oracle Applications. If the Site level option is not selected, you would select only those applications that you want to include. Any concurrent requests that originate from those applications with a check mark are intercepted. If the Site level setting option is selected then all of the listed applications are in gray text to indicate that they are selected by default as part of the Site level setting. Any applications displayed in bold text indicate that the application is overriding its default setting.

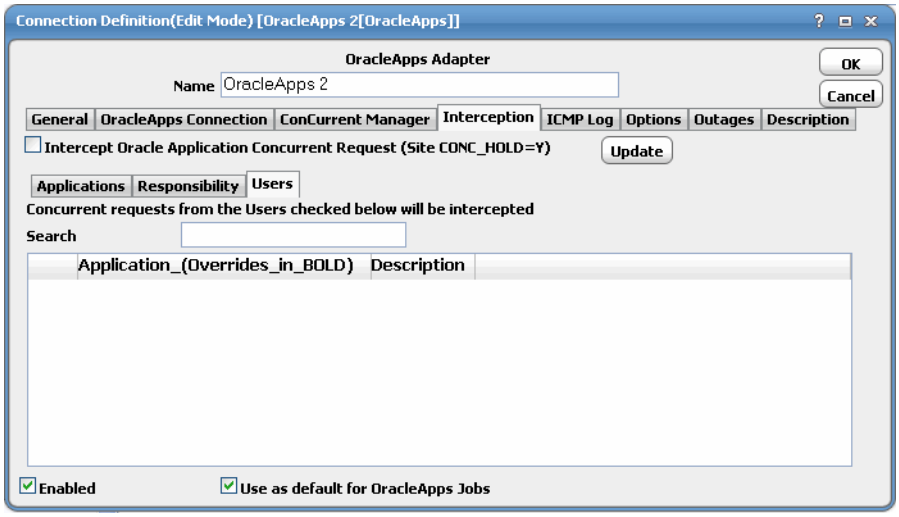
- **Responsibility tab** – Lists the various responsibilities available within a selected application. On this tab, you can select an application listed on the **Applications** tab and refine its interception criteria by adjusting its responsibilities.



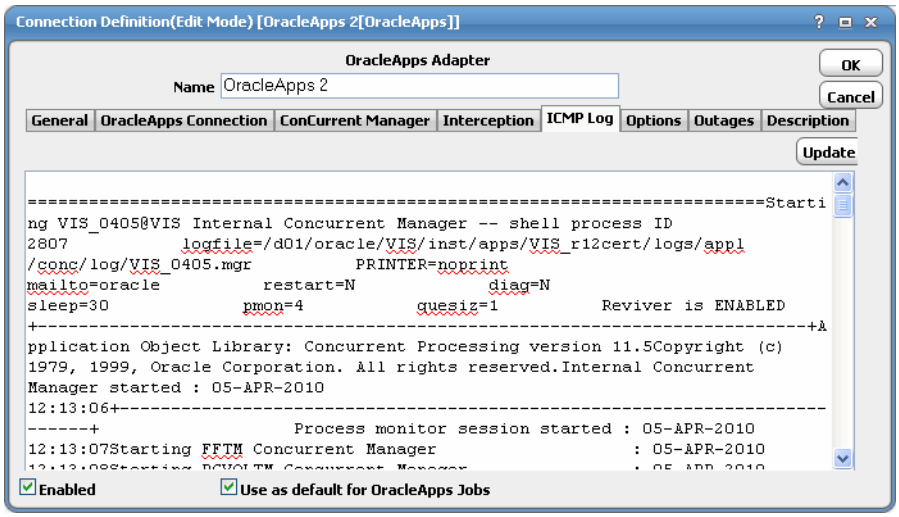
  - **Application option** – Selects all of the responsibilities for the selected application for interception. Only the concurrent requests that originate from those responsibilities with a check mark are intercepted. Any responsibility displayed in gray text indicates that it is selected by default as part of a higher level setting (**Site** or **Application**). You can override the default setting by clearing the check box beside the responsibility. Any responsibilities displayed in bold text indicate that the responsibility is overriding its default setting.

Caution: A bug in Oracle prevents responsibilities from functioning properly as interception criteria. Service Request 5099851.992 reporting this problem has been logged with Oracle. Until Oracle resolves the issue, do not use responsibilities to flag jobs for interception.

  - **Users tab** – Lists the defined users within the Oracle Applications instance. All of the users on this tab with a check mark beside them will have their ad-hoc concurrent requests intercepted. If the check box beside the user name is cleared then none of the concurrent requests submitted by that user are intercepted. Any user name displayed in gray text indicates that user is selected by default as part of the **Site** level setting. You can override the default setting by clearing the check box beside the user name.

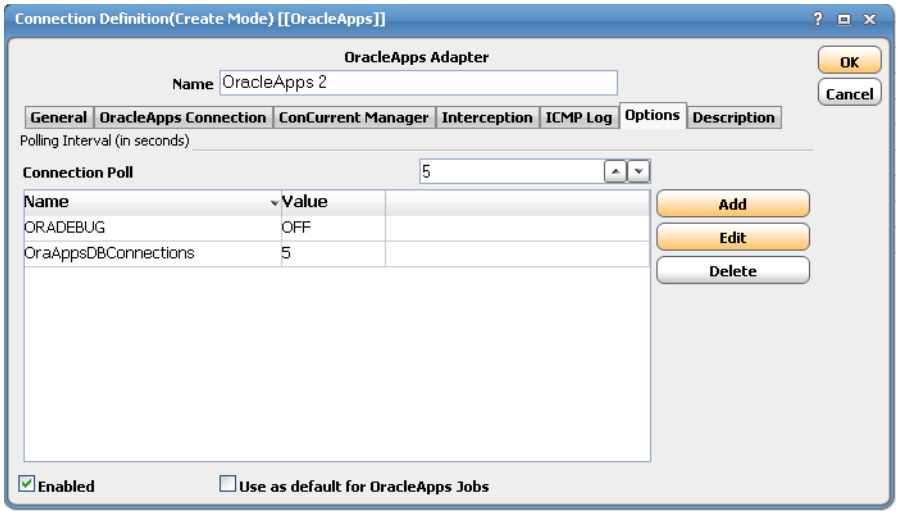


13. Click the **ICM Log** tab.



This tab provides a window to the Internal Concurrent Manager (ICM) of Oracle without having to leave CWA. The ICM oversees all of the other concurrent managers and assigns requests to the appropriate worker concurrent manager. The ICM monitors the health and status of the requests that run on the other worker concurrent managers. A user can check the log of the ICM by going to this tab. This is a convenient way to check the status of current processes that are running within Oracle Applications.

14. Click the **Options** tab.



The **Options** tab of a **Connection Definition** dialog allows for the configuration of polling for the connection and for the addition of any configuration parameters as needed.

The **Options** tab contains the following elements:

- **Polling Interval (in seconds)**
  - **Connection poll** – Enter the interval of time (in seconds) before the master checks that the connection is working.
  - **Name** – Enter the parameter name.

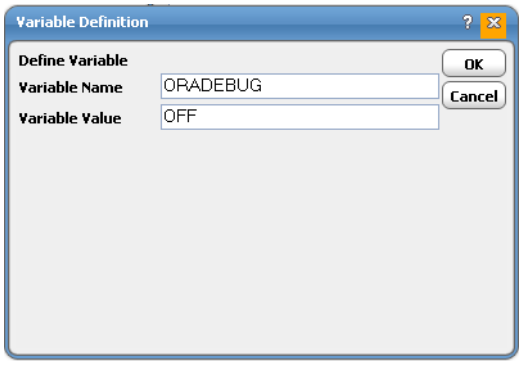
- **Value** – Enter the parameter value.

The **Name** section can contain the following parameters and default values:

- **BATCH_STATUS_CHECK** – Indicates whether the status of requests will be performed in batches (default = Y). If set, **BATCH_STATUS_CHECK_SIZE** specifies the batch size (default=100). When request statuses are checked in batches, fewer Database transactions are sent to the server. Only after the batch size limit is reached, will a single Database transaction be initiated to get the status of all requests defined in the batch. Using this parameter may result in a slight delay in retrieving request statuses, if many requests are running. If **BATCH_STATUS_CHECK = N**, then request statuses are not performed in batches. This means a Database transaction is initiated for every request, resulting in more transactions being sent to the server. With this option, there is no delay in waiting for the batch limit to be reached, however, it may result in slower performance as more Database transactions are initiated, 1 for every request.
- **DATA_CACHE_TIMEOUT** – Specifies the expiration time (in seconds) of any cached data. The default is 600 seconds or 10 minutes.
- **ORADEBUG** – Enables debugging for the Oracle Applications adapter. Default value is **OFF**.
- **OraAppsDBConnections** – Specifies the maximum number of connections opened to the Oracle Applications database. Default value is 5.
- **IGNORESTDMGR** – If **IGNORESTDMGR=Y**, the health of the Standard Manager will be ignored when verifying the connection health.
- **IGNORECONFLICTMGR** – If **IGNORECONFLICTMGR=Y**, the health of the Conflict Manager will be ignored when verifying the connection health.

To add/edit a variable:

a. Click **Add** or **Edit** to display the **Variable Definition** dialog.



b. Enter the name of the variable into the **Variable Name** field.

c. Enter the value for the variable in the **Variable Value** field.

d. Click **OK**. The variable record displays in the **Options** tab.

15. Click **OK**.

Checking the Connection Status

Once you define an Oracle Applications connection, the newly defined connection is listed in the Connections pane as an agent.

The Scheduler master continually monitors the health of the connection. If the Oracle Applications connection is not available to the master, the status light in front of the connection name 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).

You can test the Oracle Applications connection by right-clicking within the **Connections** pane and selecting the **Test** option from the context menu. This test checks the status of the connection between the database and the client but does not verify the health of the connection between the master and the client.

Configuring Oracle for Scheduler

There are some configuration procedures that Cisco recommends be performed to ensure optimal interaction between Oracle and the Scheduler adapter:

- Configuring Oracle Parameters, page 31
- Configuring User Access, page 31
- Configuring Interception Criteria Access, page 31
- Configuring Interception Parameters, page 32
- Defining an Oracle Applications Agent List, page 32

Configuring Oracle Parameters

At the time that the Oracle Applications adapter is installed, the following parameters in Oracle should be configured by the Oracle database administrator:

Parameter	Value
REL_EXT_REQ	Specifies whether external requests should be released. Valid values are Y or N.
LOG_EXT_REQ	Specifies whether to log the messages about external requests. Valid values are Y or N
MASTER_ALLOWED	Specifies which Scheduler master can release external requests.
PURGE_FREQUENCY	Specifies how often the log data recorded about the external requests should be purged. Values are numerical expressing the number of days to keep the data.
PROFILE_ACCESS	Specifies whether the Scheduler administrator can access the Oracle Applications Profile settings to modify the interception criteria. Valid values are Y or N.

Configuring User Access

In most cases, user accounts should be locked down by the DBA to prevent users from going into their accounts to modify their access. If users have access to their account, they could enhance their security access to ensure that their concurrent requests bypass the interception criteria that was created to control the flow of concurrent requests being processed.

The Oracle DBA should configure each user's access to their profile to prevent users from accessing the profile to make changes. On the **CONC_HOLD** screen of a profile, configure the profile's security by clearing the checkmark in the **Updatable** option in the **User Access** section. The user can still access their account but they cannot modify the profile.

Configuring Interception Criteria Access

The ability to set up criteria for intercepting jobs is controlled from within Oracle. The Oracle DBA can maintain total control over job interception if needed or the DBA can give the administrator in Cisco Workload Automation the capability to manage job interception. To grant the Cisco Workload Automation administrator this ability to manage job interception criteria, the Oracle DBA must select the **Updatable** option in the **Program Access** section of the **CONC_HOLD** screen.

If a user in Scheduler cannot modify the interception criteria, the **Update** button on the **Interception** tab of the Oracle connection definition is unavailable.

Configuring Interception Parameters

A parameter configuration table called **SABDG_PARAMETERS** manages job interception in Oracle Applications.

The following two parameters should be added to this table:

- **PROFILE_ACCESS** – Specifies whether a user can configure the job interception queue filter criteria. The value is either **Y** or **N**.
- **MASTER_ALLOWED** – Specifies which CWA Master is allowed to release jobs. The value is the host machine name.

The Oracle Database Administrator should use the following SQL statements to populate the **SABDG_PARAMETERS** table:

```
Insert into sabdg_parameters(PARAMETER, VAL_BOOL) values ('PROFILE_ACCESS','Y')
Insert into sabdg_parameters(PARAMETER, VAL_STR) values ('MASTER_ALLOWED', 'Host Name')
```

Defining an Oracle Applications Agent List

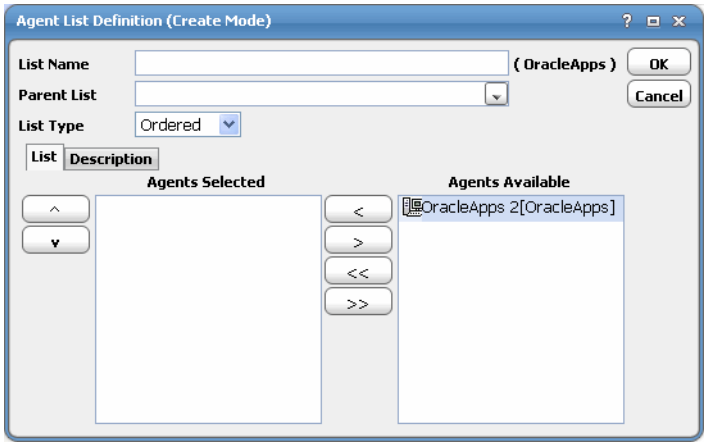
You can assign jobs to run using OracleApps agent/adapter lists similar to the way you assign them to individual agents. With OracleApps lists, however, you have access to useful functions not available with individual connections. You can specify alternate connection(s) to run your jobs if the primary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

To define an Oracle Applications Agent list:

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

   -or-

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



3. In the **List Name** field, enter the name of the OracleApps adapter 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 a PeopleSoft connection in the list.

- **Ordered** – In support of high-availability, Scheduler 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) connection in the list is not available, Scheduler tries the second (alternative) connection in the list. You might want to use this list type for critical jobs where multiple connections are capable of processing the request.
- **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** – Scheduler 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.
- **Broadcast** – Scheduler runs the job on every connection in the **Agents Selected** section. For example, you can use this option to schedule a maintenance process on each machine in the list using the same job.

6. From the **Agents Available** section on the **List** tab, highlight the appropriate OracleApps 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 Applications Adapter

This chapter describes how to use the Oracle Applications Adapter with Cisco Workload Automation which entails:

Overview

To schedule and monitor an Oracle Applications job from CWA, you must create a CWA job definition for it. An Oracle Applications job consists of one or more concurrent requests or programs. These requests may already exist in Concurrent Manager or a new request can be created using the CWA adapter when defining the job.

For greater control over all aspects of an Oracle Applications job, it is recommended that a separate CWA job be created for each concurrent program. All of the resulting CWA jobs with their respective concurrent requests can then be organized and processed as a CWA job group.

When scheduling an Oracle Applications job in CWA, dependencies (job, calendar, etc.) are defined like any other CWA job. Once the dependencies of the job are satisfied and a slot within a job queue becomes available, CWA notifies the Concurrent Manager(s) to launch the appropriate request. For request sets, as each request completes, the next appropriate request of the defined request set is launched in proper sequence.

The Oracle Applications Concurrent Manager maintains control over requests, launching and tracking the status of each request to completion. CWA monitors this activity. When the request is completed, the output and log files are available to CWA users.

Prerequisites

An Oracle Applications job created in CWA contains all the information necessary to define and launch processes in an Oracle Applications environment.

Before creating and running Oracle Applications jobs, you must:

- Install, license and verify that CWA is fully operational before installing this adapter. (Refer to the *CWA Installation Guide* for more information.)

- Verify that you can log into the Oracle instance from all clients that will modify job parameters. (Refer to your Oracle Applications documentation for more information.)
- License the Oracle Applications Adapter for CWA. (Refer to “Licensing an Adapter”.)
- Verify that the database alias entry for the Oracle Applications database in the *tnsnames.ora* is the same for the master and all client machines. (Refer to your Oracle database documentation for more information.)
- Install Oracle Net*8 database connectivity software on each master and its clients.
- Provide the CWA master or client with the alias listed in the *tnsnames.ora* file. (Refer to Verifying TNS File Formatting (Oracle Users Only), page 20.)
- Ensure that there is no other user called **SABDG**. The Bridge installation creates a user called **SABDG** and if a user called **SABDG** already exists the Bridge installation will fail.
- Ensure the user account installing the Oracle Applications adapter has Administrator or Super User capabilities. (Refer to Configuring Users for the Oracle Applications Adapter, page 17.)
- Add the Oracle Applications database user to CWA as a CWA user. (Refer to Authorizing Users, page 19.)

This integration does not place any significant resource requirements on any of the systems beyond what is expected by normal operation of the CWA and Oracle Applications products. Refer to the documentation of CWA and Oracle Applications to determine your actual system requirements.

Creating an Oracle Applications Job

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

- Installed the Bridge component.
- Applied the Oracle Applications license file.
- Added the Oracle Applications database user to CWA.
- Defined your Oracle Applications connection(s).

This section describes:

Defining an Oracle Applications Job

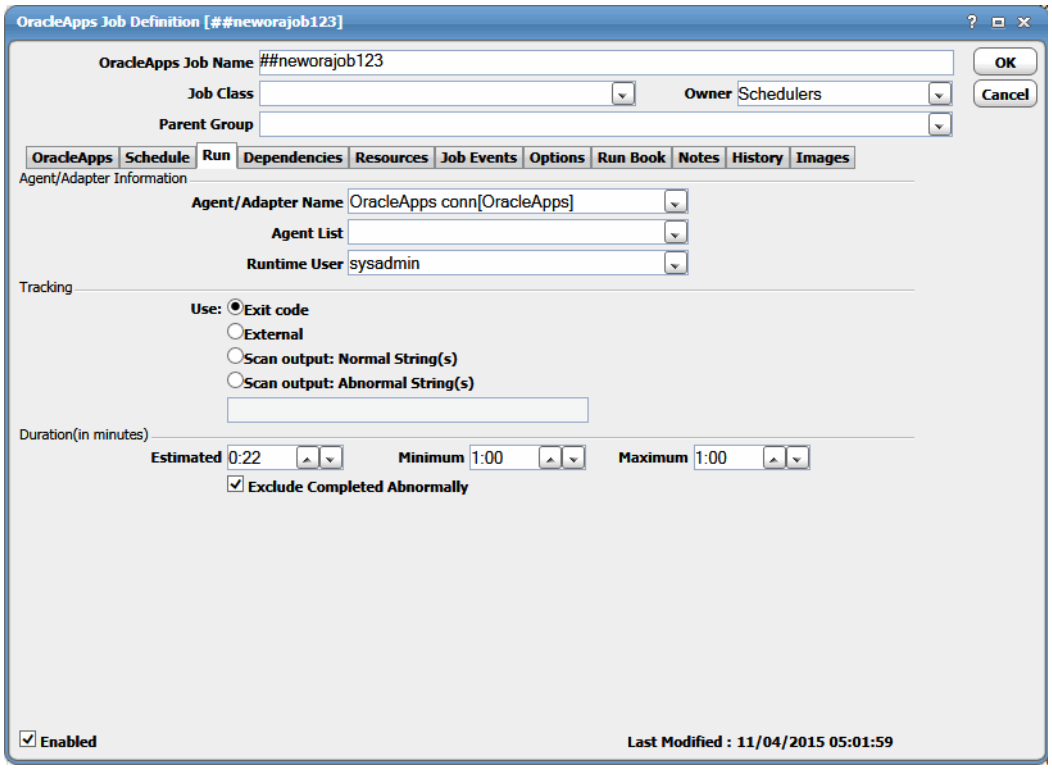
You can add an Oracle Applications job using the right-click menu within the Job Rules window and selecting the **Add OraApps Job** option. You can also **Edit**, **Copy** and **Delete** an existing Oracle Applications job. If you add Oracle Applications jobs to a CWA job group, items common between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

Note: To define a job in Oracle Applications, you need to go to Oracle’s website to download and install Oracle’s JInitiator component. Before installing the JInitiator, go to the Internet options for your browser and on the Security tab, adjust the security level for the internet to the Low setting. After downloading the JInitiator, reset the internet security setting to its original configuration.

To define a 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 OraApps Job** from the context menu to display the **OracleApps Job Definition** dialog.



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

   You can use a name of up to 50 characters in length. The OracleApps Job Name is an identifier for CWA only. All of the other job definition information, such as Job Class, Owner and Parent Group, is the same as any other CWA job.

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

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

   The **Agent/Adapter Name** refers to the connection definition to the Oracle Applications database instance on which the Concurrent Manager requests will run. This is the name used by CWA to identify the Oracle Applications database instance specified in the **OracleApps Connection Definition** dialog.

   Note that the name in this list represents all the information in the OracleApps 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 Applications connection with a new agent name and then use that agent.

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

   The default user is specified as the **Default OraApps User** in the **OracleApps Connection Definition** dialog. The user must be authorized as a valid runtime user for the user defining the job.

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

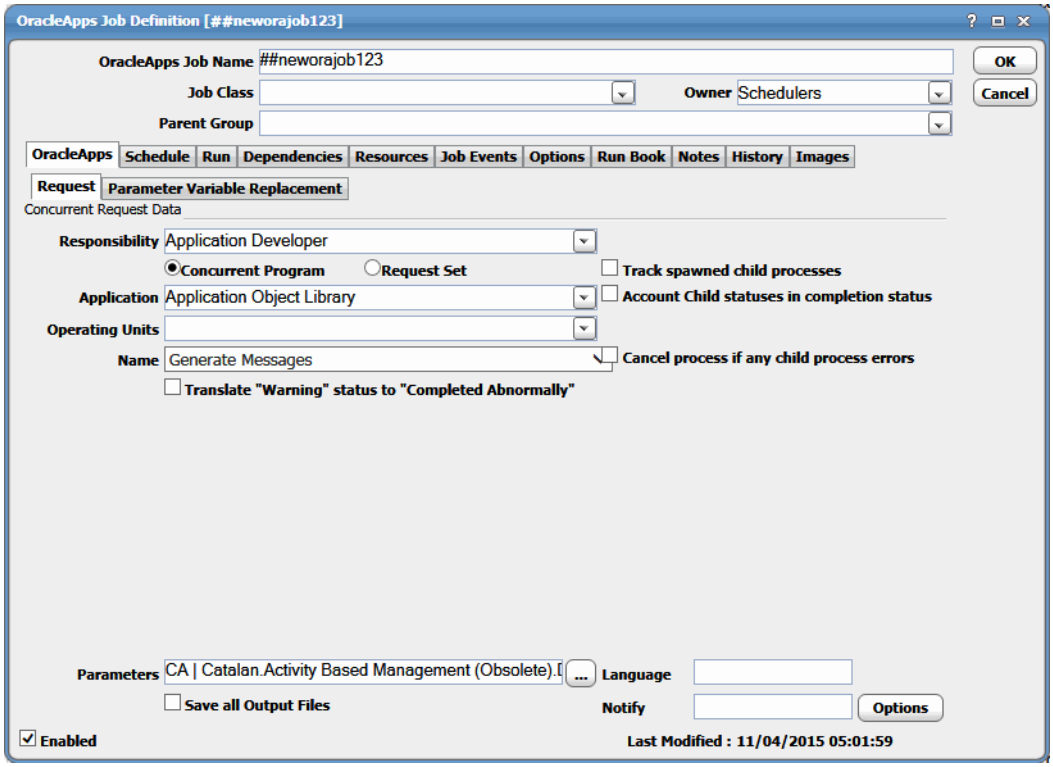
A **Warning** status is much like a **Normal** status except that warning notices were generated as the job completed successfully. To differentiate between a **Normal** status and a **Warning** status, you can select the **Translate "Warning" status to "Completed Abnormally"** option. Selecting this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

   The **Request** tab displays.



8. In the **Responsibility** list, select a responsibility option.

   The responsibility is a set of programs or job groups that a user is authorized to access. The responsibilities available vary according to the **OraApps User** selected.

9. Select the concurrent program or request set that CWA should run in Oracle Applications when the job is ready to run.

   - **Concurrent Program** – Select this option to run a specific concurrent program.
   - **Request Set** – Select this option to run a defined request set. (A request set is a group of requests similar to a job group in CWA.)

   Note: It is recommended that any Oracle Applications job consisting of multiple concurrent programs be broken down into simpler jobs with just one concurrent program per job. This practice offers greater control over the job from the CWA.

10. From the **Application** list, select an application.

This will narrow down the number of programs or displayed request sets to show only those that belong to the selected application.

11. From the **Operating Units** list, specify the concurrent program for multiple organizations.

    See Working with Multiple Organization Access (MOAC), page 50 for more information.

12. From the **Name** list, select a concurrent program or request set associated with the selected application.

    If you choose a request set, the selected request set is displayed in the **Stages** list. The meaning of the columns are:

    - **Stage** – The number of the listed stage.
    - **Stage Name** – The name of the listed stage.
    - **Seq** – The sequence number of the listed programs within the stage.
    - **Program** – The program associated with this stage.

    If you want to modify the parameters and/or runtime options of a concurrent program of the stages within the **Stages** list, select the desired program or stage and click the **Parameters** button.

    Clicking the **Parameters** button, opens a custom form to enter information pertaining to the selected program or request set stage. The parameter information entered on the form is stored in a custom CWA table in the Oracle Applications database until the request is submitted to the Concurrent Manager.

    Note: If you are defining a request set, each program needs to have its own set of parameters defined.

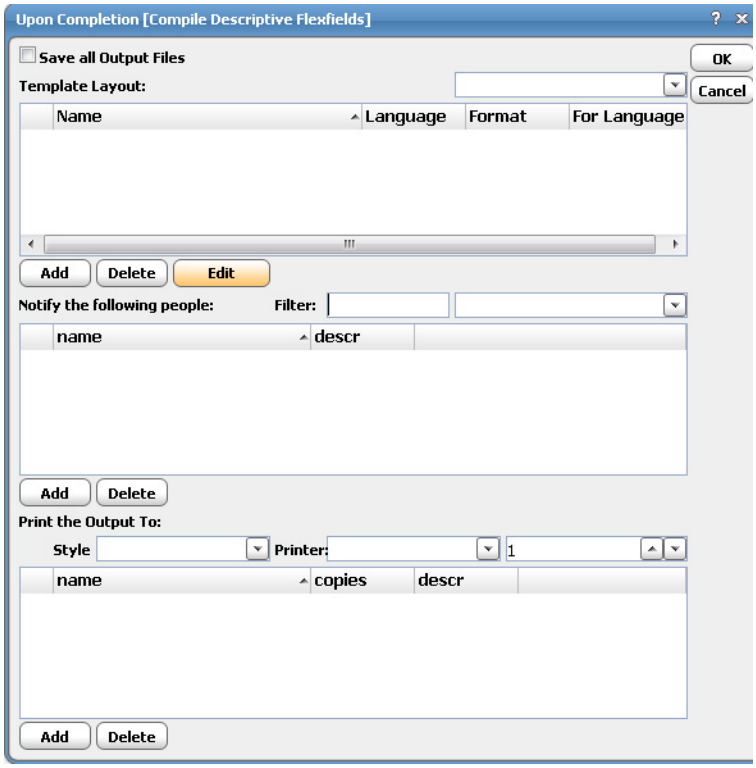
13. (Optional) Select any of these options to refine the Oracle Applications job definition:

    - **Track spawned child processes** – Select to track child processes for an OracleApps Request Set (RS) or Concurrent Program (CP) and consider the child process's status in the final completion status of the RS or CP. The final status will be the most severe status of all processes, including child processes. For example, if there is a child process which has completed with an error, then the final CWA status will be **Completed Abnormally**.
    - **Account child statuses in completion status** – Select to cancel all non-completed processes, if any child process completes in error.
    - **Cancel process if any child process errors** – Select to cancel all processes if any child process completes in error.
    - **Translate "Warning" status to "Completed Abnormally"** – Select the to differentiate between a job that completes successfully with a **Normal** status and one that ends with a **Warning** status.

    A **Warning** status is much like a **Normal** status except that warning notices were generated as the job completed successfully. Note that a job that ends with a **Warning** status still completes successfully but a job that generates warnings as it runs may indicate potential problems that you may want to be aware of.

14. (Optional) Specify the report format, notification, and printing options for this job:

a. Click **Options.**



b. Enter the desired specifications:

Template Layout: Apply the preferred layout to the report. From the drop down, select from the available layouts.

Notify the following people: Use to send notifications to people once the job is completed. You can type in few characters in the **Filter** text box and then click on the drop down to select the notification names. Use % as a wildcard character.

Print the Output To: Use to send the output to the printer. You can select the style and the printer available for the appropriate profile and the number of copies to print. By default the number of copies to print is 1.

c. Click **Add**.

15. Click **OK**.

Working with Request Sets

An Oracle Applications job consists of a series of concurrent requests executed in sequence and organized into a request set. A concurrent request is a job, report or process submitted to a Concurrent Manager for processing. You can modify the parameters of a request according to your needs.

Substituting Different Parameter Values

You can use different parameter values for different times that a job runs. You cannot substitute parameter values for one of the stages that comprise the job though unless the job already exists, so you must first create and save the job.

To replace existing parameter values:

1. Double-click on the existing Oracle Applications job to display its job definition dialog.
2. Click the **OracleApps** tab to display a list of the stages that comprise the job.

3. Select the stage that will have a different parameter value by clicking on it
4. Click the **Parameter Values** button to display the **Oracle Applications Job Parameter Details** dialog. (This button is unavailable until the newly created job is saved.)

   At the top of the dialog are listed the name of the stage, name of its associated program and the stage number and sequence number of the selected stage.

   The **Parameters** section displays a table with columns listing the names of the parameters associated with that stage, defined value of each parameter and the value that will replace the current value.
5. To substitute a different value for a parameter, click the check box in front of the parameter name to make the **Replacement Value** column available for entering different values. If desired, you can click the **Variables** button to use an existing variable as the replacement value.

The next time that the job runs, the current value of the parameter is replaced by the replacement value that you entered or the value of a variable at the time that the job runs.

The same procedure is used to replace existing parameter values for an Oracle Applications job consisting of a concurrent program.

Working with Oracle Applications Jobs

This section covers some of the tasks you might perform while working with Oracle Applications jobs in Cisco Workload Automation:

Oracle Application Procedures

Some of the typical procedures you might perform include:

Adding a Concurrent Program to an Oracle Application Job

CWA can monitor and control concurrent requests in the Oracle Applications environment that are defined as jobs in CWA. By specifying that a job should run on a CWA Oracle Applications agent, concurrent requests are sent from CWA to the Oracle Applications Concurrent Manager. These jobs can then be managed and monitored from CWA.

To add a concurrent program to an Oracle Applications job:

1. On the **OraApps** tab of the **OracleApps Job Definition** dialog, click the **Concurrent Program** option.
2. In the **Application** list, select an application to narrow down which programs are available.
3. In the **Name** list, select a program associated with the specified application.

4. You can modify the parameters of any listed program by clicking **Parameters** to display the **Job Parameters** screen.

Adding a Request Set to an Oracle Application Job

An Oracle Applications job can consist of a series of concurrent programs executed in sequence and organized into a request set. A concurrent program is a job, report or process submitted to a Concurrent Manager for processing. You can modify the parameters of a request in the request set.

To add a request set to an Oracle Applications job:

1. On the **OracleApps** tab of the **OracleApps Job Definition** dialog, click the **Request Set** option to display the **Concurrent Request Data** text field.
2. In the **Application** list, select an application to narrow down which request sets are available.
3. In the **Name** list, select a request set associated with the specified application to display its requests.
4. You can modify the parameters of any listed request by selecting the request and clicking the **Parameters** button to display the **Job Parameters** dialog.

Substituting Parameter Values

You can use different parameter values for different times that a job runs. You cannot substitute parameter values unless the job already exists, so you must first create and save the job.

To substitute different parameter values:

1. Double-click on the existing Oracle Applications job to display its job definition dialog.
2. Click on the **OracleApps** tab to display a list of the stages that comprise the job.
3. Select the stage that will have a different parameter value by clicking on it.
4. Once the stage is selected, click **Parameter Values** to display the **Oracle Applications Job Parameter Details** dialog. (This button is unavailable until the newly created job is saved.)

   The **Parameters** section displays a table with columns listing the names of the parameters associated with that stage, current value of each parameter and the value that will replace the current value.
5. To substitute a different value for a parameter, click the check box in front of the parameter name to make the **Replacement Value** column available for entering different values. If desired, you can click **Variables** to use an existing variable as the replacement value.

The next time that the job runs, the current value of the parameter is replaced by the replacement value that you entered or the value of a variable at the time that the job runs.

Viewing the Parameters of a Completed Job

Once an Oracle Applications job completes, you can view the parameters used when the job ran.

To view the parameters of a completed job:

1. In the **Job Activity** pane, double-click on the Oracle Applications job to display its **Job Details** dialog. An Oracle Applications job has an **OracleApps** tab that lists the various requests that comprised the job. (The requests are indented to show their parent-child relationship with each other.)
2. Select one of the requests listed on the tab and click **Parameters** at the bottom of the tab to display the **Oracle Applications Job Parameter Details** dialog.

   The parameter and the value used when the job was processed is displayed.

Deleting Oracle Applications Jobs from the CWA Database

Oracle Applications jobs are deleted like other CWA jobs but the jobs are deleted from the CWA database only. They must be deleted from the Oracle database as a separate task.

The default job history retention period for jobs in general is configured on the **Defaults** tab of the **System Configuration** dialog. You can set job history retention for an individual job on the **Options** tab of its job definition. CWA will automatically delete job history from its own logs when the job history retention limit is reached. The default retention period is 30 days. To delete the job history from Oracle database is a separate procedure.

Deleting Oracle Applications Jobs from the Oracle Database

You must periodically delete jobs from the Oracle database as a matter of database maintenance. You can do this maintenance as a scheduled job in CWA.

Concurrent request records are created each time an Oracle Applications job is submitted by the CWA. As these records accumulate and become obsolete, they must be deleted as a routine maintenance task.

Oracle Applications provides a concurrent program, called Purge Concurrent Request and/or Manager Data Program, to purge the datatables of these old request records. You can schedule this program in CWA to purge request records on a regular basis.

Managing User-Submitted Oracle Applications Jobs

A dynamic production environment requires a way to manage jobs that are submitted outside of the CWA. While end users may run unscheduled jobs, high volumes of these jobs will impact the timely completion of the day's production schedule. On the other hand, the production manager must ensure enough flexibility exists in the system to accommodate the inevitable unscheduled jobs that occur during a production day. The need for balancing system resources between scheduled and unscheduled jobs is heightened by the limitations on the number of available background work processes in Oracle.

To improve manageability, CWA's queue management facility supports a queue filter that can be configured to monitor an Oracle Applications instance for ad hoc jobs submitted by users or programs and assign them to a designated queue where they can be managed. By intercepting the ad hoc jobs and directing them to designated queues, the launching of unscheduled jobs can be regulated with priorities and limits of both jobs and queues, to manage resources better.

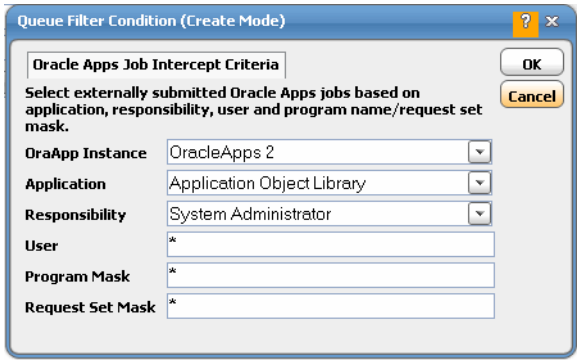
This queue filter, called **Oracle Job (Intercepted)**, can be created to designate a queue or queues that will capture any submitted Oracle jobs that do not belong to the day's production schedule.

With this interception filter monitoring for Oracle jobs, you can separate unscheduled Applications jobs generated outside of CWA from other scheduled jobs and direct those unscheduled Oracle Applications jobs to different queues where the jobs are controlled by the Queue Manager using the same limits on jobs, priorities and hierarchies typically assigned to a queue. You can add as many Oracle queue filters as needed to cover any combination of users, jobs and instances. By intercepting these ad hoc Oracle jobs before they are launched, CWA manages system resources by controlling when the jobs are launched based on queuing rules.

Caution: Do not configure two different CWA masters to intercept jobs from the same Oracle Applications system. Neither Cisco Workload Automation nor Oracle is designed to accommodate simultaneous interception with multiple schedulers and will produce erratic results.

Intercepting Oracle Applications Jobs

The filter to intercept Oracle jobs can be created by clicking the **Add** button on the **Filters** tab of the **Queue Definition** dialog and selecting the **Oracle Job (Intercepted)** option. Job interception will not work unless **SABDG_PARAMETERS** table has been properly configured as described on Configuring Interception Parameters, page 32.



This **Queue Filter Condition** dialog contains the following options that can be used as needed to fine-tune filter criteria:

- **OraApps Instance** – Specifies the Oracle instance that should be monitored for jobs scheduled outside of CWA.
- **Application** – Specifies that jobs that originate from the specified application will be intercepted.
- **Responsibility** – Specifies that jobs that originate from the specified responsibility will be intercepted.
- **User** – Designates that the queue filter monitor for jobs submitted by the specified user (or creator) of the Oracle jobs. A mask using the asterisk (*) wildcard character can be used here.
- **Program Mask** – Creates a mask with the asterisk (*) wildcard character (though wildcards do not need to be used) to intercept particular jobs that originating from programs that meet the specified criteria.
- **Request Set Mask** – Creates a mask using the asterisk (*) wildcard character (though wildcards do not need to be used) to intercept jobs originating from requests that meet the specified criteria.

You can use as much of the criteria as needed to create a broad based or specific filter. Only the Oracle jobs that match all of the specified criteria will be filtered out for controlled release. These intercepted jobs are placed on hold and can be viewed from the **Queue Definition** dialog.

The **Current Jobs** tab of the **Queue Definition** dialog for the queue defined to accept the intercepted Oracle jobs displays all jobs that are active or waiting to run. You can double-click on any of the jobs listed on the **Current Jobs** tab to see basic information about the Oracle job in the **OraApps Request Details** dialog.

The **Oracle Job (Intercepted)** queue filter manages the launching of ad hoc Oracle jobs. Here are a few scenarios to demonstrate how the queue filter works.

For the scenarios, let us assume that an Oracle Applications parent queue contains two child queues–one for externally submitted jobs and one for production jobs created by CWA. The externally scheduled jobs queue (called EB Queue for External Jobs in this example) also has an Oracle interception filter defined, with a limit of one, and the production queue (called EB Queue for Cisco Workload Automation Jobs) has a Production Oracle Connection filter and a queue limit of 3.

Release a scheduled job from CWA. It will go to the EB Queue for Cisco Workload Automation jobs, launch and go into an Active status since an execution slot is available.

From Oracle, release a job (using the same user defined in the Oracle Applications interception filter). Since there is a queue with a defined Oracle Applications job interception filter, the unscheduled job goes to the EB Queue for External Jobs. There is an execution slot available so the job is launched and the EB queue for externally scheduled jobs shows the job is active.

Again from within Oracle Applications, release another job (again as the same user defined in the Oracle Applications interception filter). This job also is directed to the EB Queue for External Jobs. However, there is no execution slot available since the queue has a queue limit of one. You can see that one job is active and the other is waiting. Instead of unexpectedly interfering with the production workload, the external Oracle Applications job is captured by a queue and held up based on the criteria defined for this queue.

Normally, the second job cannot run until the first external job completes because of the queue limit. However, an authorized operator can override the defined queue criteria and release an Oracle Applications job that is waiting.

You can view the active and waiting jobs for a queue by double-clicking on the queue and selecting the **Current Jobs** tab. The ID, job name, status, agent, external ID and priority of a queue's active and/or waiting jobs can be viewed. If a large number of jobs are in the queue, you may only want to see either just the active or just the waiting jobs. In any case, you must select at least one of the options or no jobs will display on the tab. The numbers next to the check boxes indicate how many of that type of job are in the queue. Releasing a job that is waiting results in the following queue situation where the active jobs in a queue exceeds its defined limit.

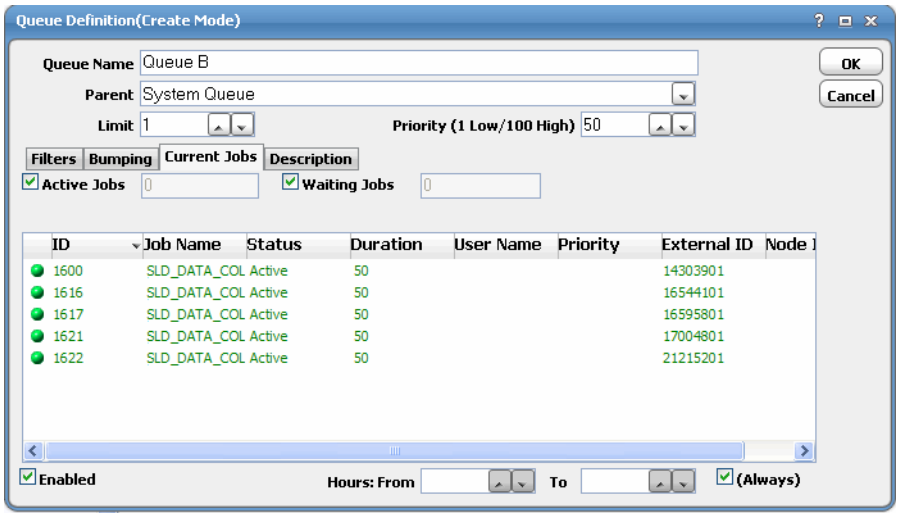
You can release unscheduled jobs that are waiting (or cancel a job if it is active) from the **Current Jobs** tab of the queue's definition dialog.

Select one of the jobs on the **Current Jobs** tab and right-click to display a context menu of options. Select the **Release** option to launch a waiting job immediately. (The user must be authorized with the appropriate security functions to be able to use the job control menu options.)

Warning: The job that is in Waiting on Resource status can be released even if this would cause a queue to exceed its limit.

The context menu also allows you to manually refresh the queue status or to configure the tab to update the job status automatically every 10 seconds, so you can watch the jobs in real time.

Each job displayed in the text field has a color-coded status light to denote a job's status at a glance. The color green indicates a job that is in active or launched status, the color yellow indicates a job in a waiting status and the color red indicates a job unable to run due to an unavailable connection.



The listing of current jobs in the queue has the following columns of information about the jobs:

- **ID** – The identification number assigned to the job by CWA.
- **Job Name** – The name of the job.
- **Status** – The current status of the job in CWA.

- **Duration** – The time in minutes that a job has remained in its present status within the queue.
- User Name – The user name of the user running the job.
- **Priority** – The priority number assigned to the job.
- **External ID** – The identification number assigned by jobs created in applications other than CWA. For Oracle Applications, this is the ID.
- **Node ID** – The identification number assigned to nodes in applications other than CWA.

Studying Details of the Oracle Applications Concurrent Request

The **OraApps Request Details** dialog displays by double-clicking on one of the concurrent requests that have been intercepted on the **Current Jobs** tab of the **Queue Definition** dialog. You can view details about the concurrent request from this dialog. These are the same details displayed in the **OracleApps** tab of the **Job Details** dialog for the Oracle job.

This dialog contains the following columns of information about the request:

- **ID** – The identification number that Oracle Applications assigns to the request instance. This number corresponds to the external identification number displayed for the job in the **Ext ID** column of the **Job Activity** pane.
- **Request** – The name of the concurrent request.
- **Start** – The time that the concurrent request started processing.
- **Finish** – The time that the concurrent request completed processing.
- **Phase** – The current phase that the concurrent request is in.
- **Status** – The current status of the concurrent request.

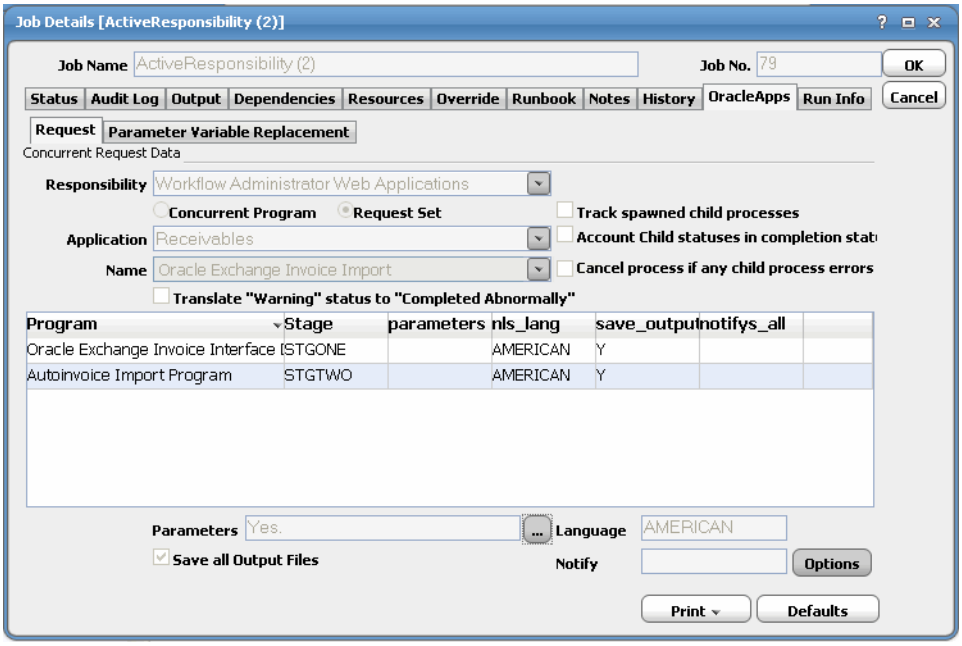
Oracle Applications Job Details Dialog

The **Job Details** dialog of an Oracle Applications job is similar to the **Job Detail** dialog of any other CWA job with a few exceptions. An Oracle Applications job has its own tab in the **Job Detail** dialog. This tab shows the concurrent requests that comprise the job.

To display the Job Details dialog for an Oracle Applications job

1. In the **Navigator** tree, select **Job Activity**.
2. Double-click the **OracleApps** job.

3. Click the **OracleApps** tab.



This tab contains the following elements:

- **Responsibility** – Contains a responsibility option.

  The responsibility is a set of programs or job groups that a user is authorized to access. The responsibilities available vary according to the OraApps User selected.

- **Concurrent Program** – Selected if a specific concurrent program.

  It is recommended that any Oracle Applications job consisting of multiple concurrent programs be broken down into simpler jobs with just one concurrent program per job. This practice offers greater control over the job from the CWA.

- **Application** – Contains the selected application.

- **Operating Units** – Contains the concurrent program for multiple organizations.

- **Name** – Contains the selected concurrent program or request set associated with the selected application.

- **Request Set** – Selected to run a defined request set. (A request set is a group of requests similar to a job group in CWA.)

  If you choose a request set, the selected request set is displayed in the Stages list. The meaning of the columns are:

  – **Stage** – The number of the listed stage.

  – **Stage Name** – The name of the listed stage.

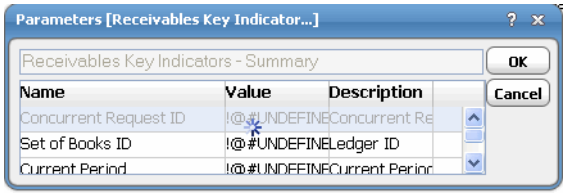
  – **Seq** – The sequence number of the listed programs within the stage.

  – **Program** – The program associated with this stage.

- **Track spawned child processes** – Select to consider the child process's status in the final completion status of the RS or CP. The final status will be the most severe status of all processes, including child processes. For example, if there is a child process which has completed with an error, then the final CWA status will be **Completed Abnormally**.

- **Account child statuses in completion status** – Select to cancel all non-completed processes, if any child process completes in error.
- **Translate "Warning" status to "Completed Abnormally"** – Selected to differentiate between a job that completes successfully with a **Normal** status and one that ends with a **Warning** status.

  A **Warning** status is much like a **Normal** status except that warning notices were generated as the job completed successfully. Note that a job that ends with a **Warning** status still completes successfully but a job that generates warnings as it runs may indicate potential problems that you may want to be aware of.
- **Parameter Values** – At the bottom of the tab is the **Parameters Values** button. You can view the parameters of any request in a job that completed by selecting the request and clicking this button to display the **Oracle Applications Job Parameter Details** dialog containing the parameter information.

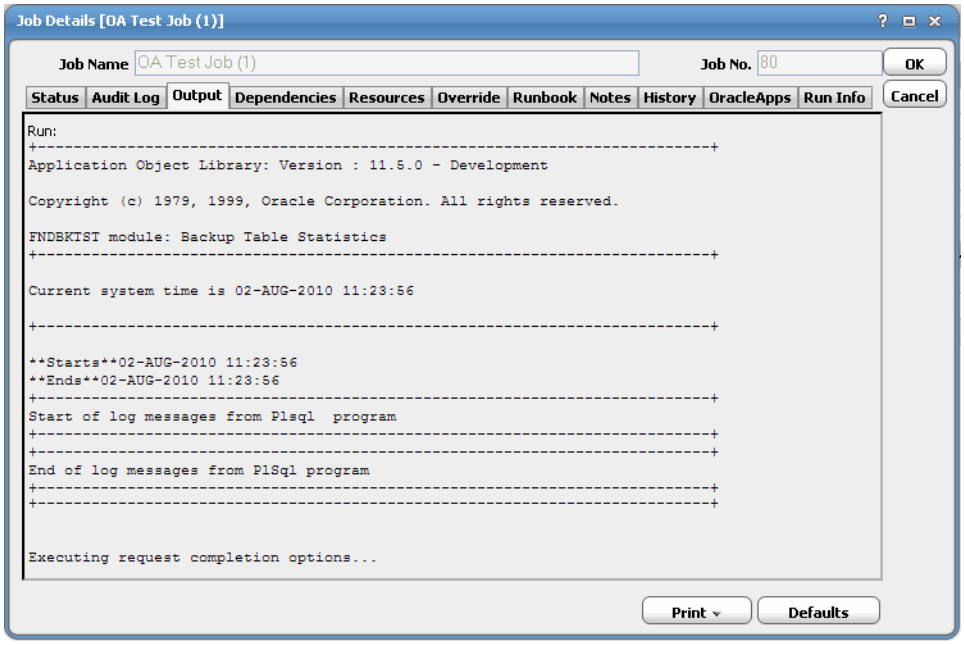


Viewing Oracle Applications Job Output

As with any other CWA job, you can view the output from an Oracle Applications job on the **Output** tab of the **Job Details** dialog. However, the output from an Oracle Applications job can be quite extensive. Rather than examine the entire output of a job, a user may prefer to include only a summary and log of the job activity. By selecting the **Summary Only** option on the **Options** tab of the **OracleApps Job Definition** dialog, only a summary and log of the job is stored and available for display in the CWA client.

To view job output:

1. On the **Job Activity** pane and double-click the Oracle Applications job to display the **Job Details** dialog.
2. Click the **Output** tab.



The **Output** tab displays the Oracle Applications output generated for the job.

Performing Routine Maintenance

Concurrent request records are created each time an Oracle Applications job is submitted by the CWA. As these records accumulate and become obsolete, they must be deleted as a routine maintenance task.

Oracle Applications provides a concurrent program, called **Purge Concurrent Request** and/or **Manager Data Program**, to purge the datatables of these old request records. You can schedule this program in CWA to purge request records on a regular basis.

Monitoring Oracle Applications Jobs

The parameters of Oracle Applications jobs are defined within the Oracle Applications environment but the scheduling of these jobs is controlled by CWA. Jobs are launched on a concurrent request by concurrent request basis. CWA watches for the completion of each concurrent request and any other change in the status of the concurrent request. Once the process completes, CWA determines the completion status and can return the output and log of the process. The status of a CWA Oracle Applications job with multiple concurrent requests is determined by the concurrent request that completed last.

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**.

Working with Multiple Organization Access (MOAC)

Multiple organization access is supported in CWA by defining the operating units for each Oracle Applications job definition. This feature is supported for the Oracle Application Server R12 and later.

Concurrent Programs Behavior with MOAC

The **Operating Units** field is included to the OracleApps job definition window to allows user to specify the concurrent programs for multiple organizations. The concurrent programs can be categorized into Single, Multiple or Null. The default value is Null. The concurrent program is used to execute the multiple organizations initialization and also determine when to display Operating Unit field in the Submit Requests window and Schedule Requests window.

There are two categories of Concurrent programs:

Single Organization Concurrent Programs

Single Organization concurrent programs are non-report programs that report or process data for one Operating Unit only. These programs show data for the Operating Unit specified by MO: Operating Unit profile option. These programs are flagged as Single for Operating Unit mode in the Define Concurrent Programs window.

Multiple Organization Concurrent Programs

The Multiple Organization Concurrent Programs process or report date for multiple operating units specified by the profile option MO: Security Profile. These programs display the operating unit as an optional parameter. The user selects an operating unit and submits the program or leaves it blank. If the parameter is left blank, the concurrent program processes or reports data for the operating units specified in the MO: Security Profile.

Users may choose to enter a value for the Operating Unit or leave it blank and submit the request. If you specify a value, the operating units invoices are processed for the respective operating units else invoices are processed for all the operating units in the security profile.

CWA Support for Multiple Organizations

The Oracle Applications Adapter supports the MOAC feature of Oracle Application Server R12 version.

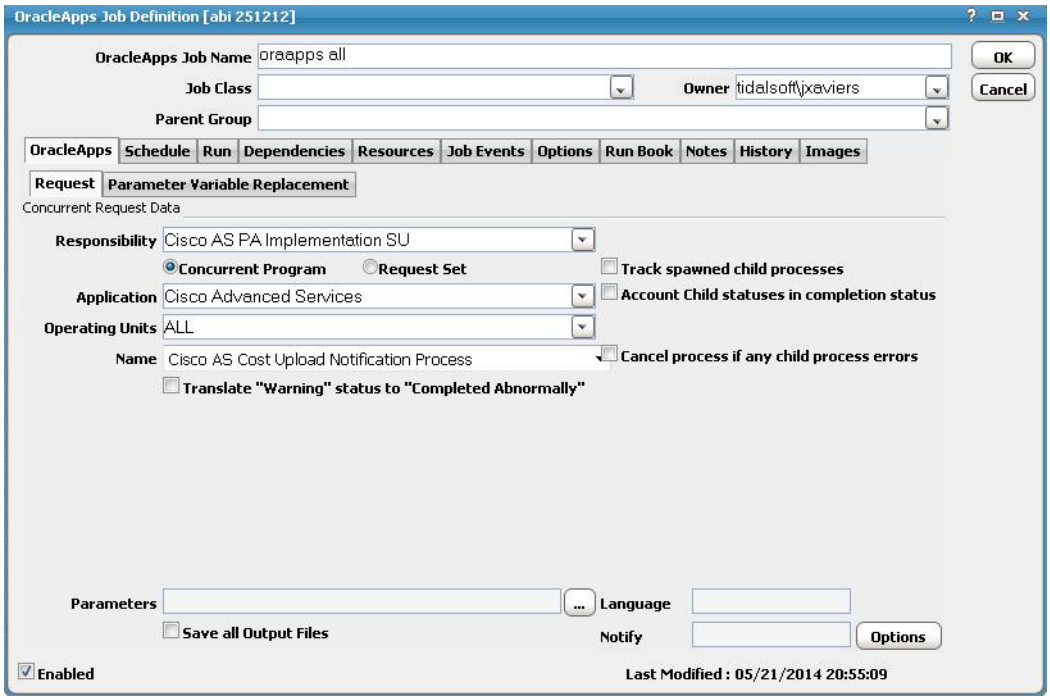
Note: MOAC is supported for R12 versions. MOAC is not supported for request set jobs.

We support the following three use cases:

- Run the Request for Multiple Operating Units, page 52
- Select a Particular Operating Unit, page 53
- MOAC Feature Disabled, page 53

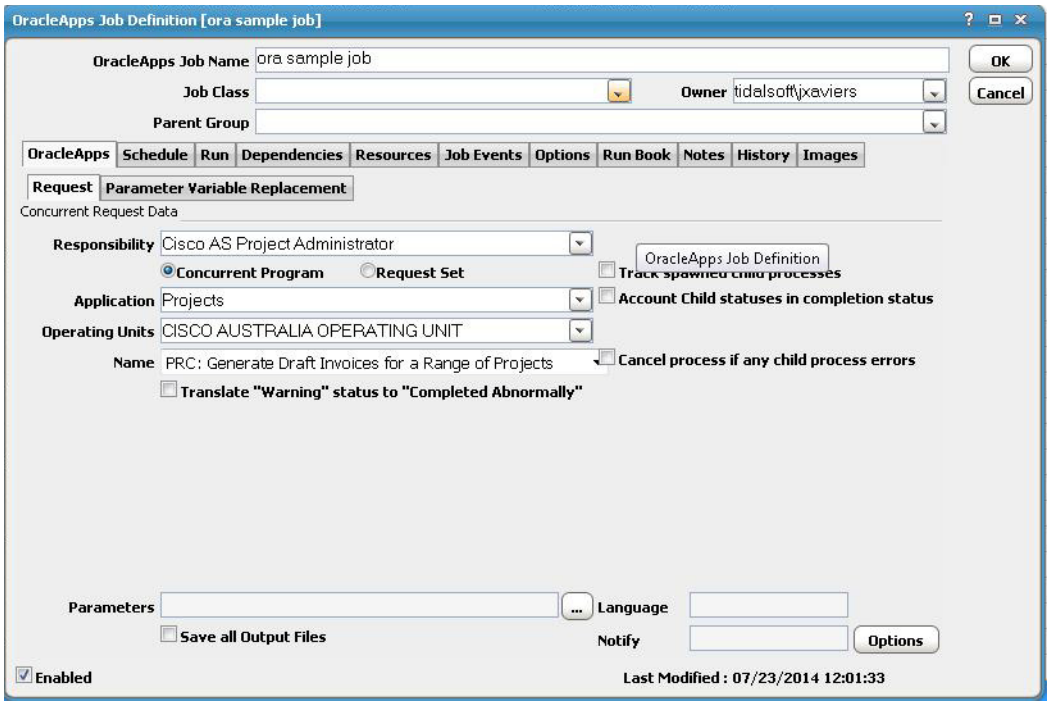
Run the Request for Multiple Operating Units

You can run the request for all operating units defined in the MO: Security Profile. In the **OracleApps Job Definition** panel **Request** tab, use the default value of **ALL** as shown below.



Select a Particular Operating Unit

The **Operating Units** field on the **Requests** tab of the **OracleApps Job Definition** panel allows you to select the desired operating unit for a responsibility. Once selected, the operating units are used in job processing. Here's an example for the Web Client:



MOAC Feature Disabled

If the MOAC feature is not enabled in the Oracle Application Server, clicking on the **Operating Units** drop down list generates an error message.

Using an Alternative Database User

As an alternative to the standard APPS user, Cisco Workload Automation supports the use of a custom database user to access Oracle EBS. This user's security profile would be limited to the minimal set of objects required to perform Concurrent Manager scheduling functions. This database account should be granted privileges to access the objects below. Cisco Workload Automation can provide assistance in assigning the correct permissions.

Packages
dbms_obfuscation_toolkit
fnd_amp_private.
fnd_application_pkg.
fnd_global
fnd_message
Fnd_profile
fnd_request
fnd_submit

Packages
fnd_web_sec.
fnd_webfile
Htp
Utl raw

Forms and Associated Libraries
FNDRSRUN
FNDRSRUN
FNDSCSGN

Sequences
fnd application s

Tables and Views
fnd_application
fnd_application_tl
Fnd_application_vl
fnd_conc_request_arguments
fnd_concurrent_processes
fnd_concurrent_programs
FND_CONCURRENT_PROGRAMS_TL
fnd_concurrent_programs_vl
FND_CONCURRENT_REQUESTS
fnd descr flex col usage vl
fnd_descr_flex_column_usages
fnd_languages_vl
fnd_lookups
fnd_menu_entries_vl
fnd_menus_vl
FND_PRINTER
FND_PRINTER_INFORMATION
fnd_printer_styles
fnd_product_groups
fnd_product_installations
fnd_profile_option_values
fnd_profile_options
fnd_req_set_run_form_v
fnd_request_group_units
fnd_request_set_programs
fnd_request_set_stages

Tables and Views
fnd_request_sets
fnd_request_sets_tl
fnd_request_sets_vl
fnd_resp_functions
FND_RESPONSIBILITY_TL
fnd_responsibility_vl
FND_USER
fnd_user_resp_groups
wf_roles

5

Configuring service.props

About Configuring service.props

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

service.props Properties

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

Property	Applicable Adapter(s)	Default	What It Controls
BYPASS_SEC_VALIDATION	Oracle Apps	N	If set to Y, the secondary user validation is bypassed. If not, secondary user validation is performed.
CLASSPATH	All	<none>	(Optional) – The path to the JDBC driver. If the default CLASSPATH used when the Adapter process is started does not include an appropriate JDBC driver jar required to connect to the PowerCenter Repository Database, you will need to specify this *service.props* configuration
CONN_SYNC	Informatica, Oracle Apps, SAP	N	Setting this flag to Y allows synchronous connections without overloading the RDOnly Thread. If set to N, the adapter might stop trying to reconnect after an outage or downtime.
DISCONN_ON_LOSTCONN	Informatica	N	Setting this flag to Y avoids an unnecessary logout call to the Informatica server when the connection is lost. This logout call usually hangs.
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 *Informatica Adapter Guide* for how to set this for Windows and Unix environments.
TDLINFA_REQUESTTIMEOUT	Informatica	<none>	(Optional) – The number of seconds before an API request times out. The default is 120 seconds, if not specified.
TDLJDBC_LIBPATH	JDBC	<none>	(Windows only, optional) An alternate path to the JDBC library files. The library file path should have been configured given system environment variables. This option is available in case you wish to use an alternate set of libraries and may be helpful for trouble-shooting purposes.

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.

A

Appendix A–Custom APPS User SQL Commands

This appendix contains the SQL commands used to install the Oracle Applications Adapter Bridge as the custom APPS user as described in Installing the Oracle Application Bridge Using a Custom APPS User, page 15.

tidal.sql Commands

```
GRANT ALTER SYSTEM TO "TIDAL";
GRANT CREATE PROCEDURE TO "TIDAL";
GRANT CREATE PUBLIC SYNONYM TO "TIDAL";
GRANT CREATE SYNONYM TO "TIDAL";
GRANT CREATE TABLE TO "TIDAL";
GRANT CREATE VIEW TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_APPLICATION" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_APPLICATION" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_APPLICATION_S" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_APPLICATION_TL" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_APPLICATION_TL" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_APPL_TOPS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_APPL_TOPS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CACHE_VERSIONS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CACHE_VERSIONS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONCURRENT_PROCESSES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONCURRENT_PROCESSES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONCURRENT_PROGRAMS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONCURRENT_PROGRAMS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONCURRENT_PROGRAMS_TL" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONCURRENT_PROGRAMS_TL" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_CONCURRENT_QUEUES" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
```

```
GRANT DELETE ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_CONCURRENT_REQUESTS" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_CONCURRENT_REQUESTS_S" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONCURRENT_REQUESTS_S" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_CONC_DEFERRED_ARGUMENTS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONC_PP_ACTIONS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONC_PP_ACTIONS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONC_PROG_ONSITE_INFO" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONC_PROG_ONSITE_INFO" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_CONC_RELEASE_CLASSES" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_S" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_S" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_CONC_RELEASE_CLASSES_TL" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONC_REQUEST_ARGUMENTS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONC_REQUEST_ARGUMENTS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CONC_REQ_OUTPUTS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CONC_REQ_OUTPUTS" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_CP_IPP_PRINTERS" TO "TIDAL";
```

```
GRANT ALTER ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_CP_SERVICES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DATA_GROUPS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DATA_GROUPS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DATA_GROUP_UNITS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DATA_GROUP_UNITS" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_DEBUG_RULES" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_DEBUG_RULES_S" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DEBUG_RULES_S" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_DEBUG_RULE_OPTIONS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DESCRIPTIVE_FLEXS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DESCRIPTIVE_FLEXS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DESCRIPTIVE_FLEXS_TL" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DESCRIPTIVE_FLEXS_TL" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DESCR_FLEX_COLUMN_USAGES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DESCR_FLEX_COLUMN_USAGES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DESCR_FLEX_COL_USAGE_TL" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DESCR_FLEX_COL_USAGE_TL" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DESCR_FLEX_CONTEXTS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DESCR_FLEX_CONTEXTS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_DESCR_FLEX_CONTEXTS_TL" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_DESCR_FLEX_CONTEXTS_TL" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_FILE_TEMP" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_FLEX_VALIDATION_TABLES" TO "TIDAL";
```

```
GRANT SELECT ON "APPLSYS"."FND_FLEX_VALIDATION_TABLES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_FLEX_VALUES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_FLEX_VALUES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_FLEX_VALUE_SETS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_FLEX_VALUE_SETS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_LANGUAGES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_LANGUAGES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_LOGINS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_LOGINS" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_LOGINS_S" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_LOGINS_S" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_LOG_MESSAGES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_LOOKUP_TYPES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_LOOKUP_TYPES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_MO_PRODUCT_INIT" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_MO_PRODUCT_INIT" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_NEW_MESSAGES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_NEW_MESSAGES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_NODES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_NODES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_OAM_CONTEXT_FILES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_OAM_CONTEXT_FILES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_OAM_FORMS_RTI" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_OAM_FORMS_RTI" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_OAM_FRD_LOG" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_OAM_FRD_LOG" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_ORACLE_USERID" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_ORACLE_USERID" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PRINTER" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PRINTER" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PRINTER_INFORMATION" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PRINTER_INFORMATION" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PRINTER_STYLES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PRINTER_STYLES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PRODUCT_GROUPS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PRODUCT_GROUPS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PRODUCT_INITIALIZATION" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PRODUCT_INITIALIZATION" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PRODUCT_INIT_CONDITION" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PRODUCT_INIT_CONDITION" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PRODUCT_INIT_DEPENDENCY" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PRODUCT_INIT_DEPENDENCY" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PRODUCT_INSTALLATIONS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PRODUCT_INSTALLATIONS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PROFILE_OPTIONS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PROFILE_OPTIONS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_PROFILE_OPTION_VALUES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_PROFILE_OPTION_VALUES" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_RAND_S" TO "TIDAL" WITH GRANT OPTION;
GRANT SELECT ON "APPLSYS"."FND_RAND_S" TO "TIDAL" WITH GRANT OPTION;
GRANT ALTER ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT DEBUG ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT DELETE ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT FLASHBACK ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT INDEX ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
```

```
GRANT INSERT ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT QUERY REWRITE ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT REFERENCES ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT SELECT ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT UPDATE ON "APPLSYS"."FND_RAND_STATES" TO "TIDAL" WITH GRANT OPTION;
GRANT INSERT ON "APPLSYS"."FND_REQUEST_GROUPS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_REQUEST_GROUPS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_REQUEST_GROUP_UNITS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_REQUEST_GROUP_UNITS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_REQUEST_SETS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_REQUEST_SETS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_REQUEST_SETS_TL" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_REQUEST_SETS_TL" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_REQUEST_SET_PROGRAMS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_REQUEST_SET_PROGRAMS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_REQUEST_SET_PROGRAM_ARGS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_REQUEST_SET_PROGRAM_ARGS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_REQUEST_SET_STAGES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_REQUEST_SET_STAGES" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_RESPONSIBILITY" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_RESPONSIBILITY_TL" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_RESPONSIBILITY_TL" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_RESP_FUNCTIONS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_RESP_FUNCTIONS" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_RUN_REQUESTS" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_RUN_REQ_LANGUAGES" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
```

```
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_RUN_REQ_PP_ACTIONS" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_SESSIONS" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_S_FILE_TEMP_ID" TO "TIDAL";
GRANT ALTER ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT DEBUG ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT DELETE ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT FLASHBACK ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT INDEX ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT QUERY REWRITE ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT REFERENCES ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT UPDATE ON "APPLSYS"."FND_TERRITORIES" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_USER" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_USER" TO "TIDAL";
GRANT INSERT ON "APPLSYS"."FND_USER_PREFERENCES" TO "TIDAL";
GRANT SELECT ON "APPLSYS"."FND_USER_PREFERENCES" TO "TIDAL";
GRANT EXECUTE ON "APPS"."APP_EXCEPTION" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FNDCP_CRM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FNDCP_SCH" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FNDCP_TMSRV" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ACCESS_CONTROL_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ADPATCH" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_AMP_PRIVATE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_AOLJ_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_APD" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_API" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_APPFLDR" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_APPLET_LAUNCHER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_APPLICATION_PKG" TO "TIDAL";
GRANT DEBUG ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT DELETE ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT FLASHBACK ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT MERGE VIEW ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT QUERY REWRITE ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT REFERENCES ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT UPDATE ON "APPS"."FND_APPLICATION_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_APPS_MODE_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_APPTREE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_APP_SERVER_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_APP_SYSTEM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ATTACHED_DOCUMENTS2_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ATTACHED_DOCUMENTS3_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ATTACHED_DOCUMENTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ATTACHMENT_UTIL_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_AUDIT_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_AUDIT_REPORT" TO "TIDAL";
```

```
GRANT EXECUTE ON "APPS"."FND_AUDIT_SEQ_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_AUDIT_UPDATE_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_BC4J_CLEANUP_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_BES_PROC" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_BUSINESS_PURPOSES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CACHE_VERSIONS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CLIENT_INFO" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CODE_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_COL_PRIV_ATTRIBUTES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC" TO "TIDAL";
GRANT DEBUG ON "APPS"."FND_CONCURRENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONCURRENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONCURRENT_ARGUMENTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONCURRENT_BUSINESS_EVENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONCURRENT_FILE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONCURRENT_PROGRAMS_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_CONCURRENT_PROGRAMS_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_CONCURRENT_PROGRAMS_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONCURRENT_PROG_ANNO_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONCURRENT_QUEUES_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_CONCURRENT_QUEUES_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_CONCURRENT_QUEUES_VL" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_CONCURRENT_WORKER_REQUESTS" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_CONCURRENT_WORKER_REQUESTS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_CLONE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_CONNECT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_DATABASE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_DATE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_GLOBAL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_MAINTAIN" TO "TIDAL";
GRANT DEBUG ON "APPS"."FND_CONC_PP" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_PP" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_PRIVATE_UTILS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_QUEUES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_RAC_UTILS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_RELEASE_CLASSES_PKG" TO "TIDAL";
GRANT DEBUG ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT DELETE ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT FLASHBACK ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT MERGE VIEW ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT QUERY REWRITE ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT REFERENCES ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT UPDATE ON "APPS"."FND_CONC_RELEASE_CLASSES_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_RELEASE_CLASS_UTILS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_RELEASE_DISJS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_RELEASE_PERIODS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_RELEASE_STATES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_REQUEST_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_SSWA" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_STAT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_STATE_LOOKUPS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_STATE_LOOKUP_TYPES_PG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_STAT_ROLLUP" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_SUMMARIZER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_TEMPLATES" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONC_WEB_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONST" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONTEXT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CONTEXT_UTIL" TO "TIDAL";
GRANT DEBUG ON "APPS"."FND_CORE_LOG" TO "TIDAL";
```

```
GRANT EXECUTE ON "APPS"."FND_CORE_LOG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_FNDSM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_GSM_IPC" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_OPP_CMD" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_OPP_IPC" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_OPP_REQ" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_OPS_MAINT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_RT_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_SERVICES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_TMSRV_PIPE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CP_TMSRV_QUEUE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CRYPTO" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CRYPTO_CONSTANTS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CRYPT_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CSS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CTL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CURRENCIES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CURRENCY" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_CURRENCY_CACHE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DATADICT_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DATADICT_SEQ_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DATA_SECURITY" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DATE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DATE_TZ" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DCP" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DEBUG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DEBUG_OPTIONS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DEBUG_OPTION_VALUES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DEBUG_REP_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DEBUG_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DESCRIPTIVE_FLEXS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DESCR_FLEX_COL_USAGE_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_DESCR_FLEX_COL_USAGE_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_DESCR_FLEX_COL_USAGE_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DESCR_FLEX_CONTEXTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DFLEX" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DIAG_CM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DICTIONARY_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DISCONNECTED" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DM_PRODUCTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DM_REPOSITORY_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DOCUMENTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DOCUMENT_DATATYPES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DOCUMENT_ENTITIES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DOCUMENT_MANAGEMENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DOC_CATEGORIES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_DSQL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_EVENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_EXECUTABLES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_EXEC_MIG_CMDS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FILE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FILE_PRIVATE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FILE_UPLOAD" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEXBUILDER_PARAMS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_APIS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_DATE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_DESCVAL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_DIAGNOSE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_DSC_API" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_EXT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_HIERARCHIES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_HIERARCHY_COMPILER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_KEYVAL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_KEY_API" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_LOADER_APIS" TO "TIDAL";
```

```
GRANT EXECUTE ON "APPS"."FND_FLEX_ORACLE_REPORTS_APIS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_PLSQL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_PLSQL_CACHE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_SERVER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_SERVER1" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_SERVER2" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_SERVER4" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_TRIGGER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_TYPES" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_UPD_FMT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_UPGRADE_UTILITIES" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_VALUES_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_FLEX_VALUES_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_FLEX_VALUES_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_VALUE_RULES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_VAL_API" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_VAL_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_VDATION_RULES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_WF_PROCESSES" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_WORKFLOW" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_WORKFLOW_APIS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FLEX_XML_PUBLISHER_APIS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FORM_CUSTOM_RULES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FORM_FUNCTIONS_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_FORM_FUNCTIONS_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_FORM_FUNCTIONS_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FORM_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_FORM_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_FORM_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FUNCTION" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FUNCTION_SECURITY" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FUNCTION_SECURITY_CACHE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_FUNCTION_W" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_GENERIC_POLICY" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_GEN_MIG_CMDS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_GFM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_GLOBAL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_GRANTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_GSM_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_HASH_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_HELP" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_HELP_BUILDER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_HTTP" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_HTTP_TICKET" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ICX_SEC" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ID_FLEX_SEGMENTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ID_FLEX_STRUCTURES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_IMP_CONV_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_IMP_DEPENDENCY_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_IMP_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_IMUTL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_INDUSTRIES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_INDUSTRY_ACTIVATOR" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_INSTALLATION" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_INSTALLATION2" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_IREP_CP_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_IREP_DEFERRED_LOADER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_IREP_LOADER_PRIVATE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ISO_LANGUAGES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_JOBS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LANGUAGES_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_LANGUAGES_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_LANGUAGES_VL" TO "TIDAL";
```

```
GRANT EXECUTE ON "APPS"."FND_LDAP_ERRM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LDAP_MAPPER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LDAP_USER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LDAP_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LDAP_WRAPPER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOADER_OPEN_INTERFACE_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOAD_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOGVIEW" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOG_ADMIN" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOG_ATTACHMENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOG_ATTACHMENT_FRM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOG_REPOSITORY" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOG_SUMMARY" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOG_UTIL" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_LOOKUPS" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_LOOKUPS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOOKUP_ASSIGNMENTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOOKUP_TYPES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_LOOKUP_VALUES_PKG" TO "TIDAL";
GRANT DEBUG ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT DELETE ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT FLASHBACK ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT MERGE VIEW ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT ON COMMIT REFRESH ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT QUERY REWRITE ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT REFERENCES ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT UPDATE ON "APPS"."FND_LOOKUP_VALUES_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MANAGER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MENUS_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_MENUS_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_MENUS_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MENU_ENTRIES_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_MENU_ENTRIES_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_MENU_ENTRIES_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MESSAGE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MESSAGE_CACHE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MIME_TYPES_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_MIME_TYPES_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_MIME_TYPES_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MLS_REQUEST" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MLS_SUBMIT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MO_PRODUCT_INIT_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MO_REPORTING_API" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_MSG_PUB" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_NATURAL_LANGUAGES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_NET_SERVICES" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_NEW_MESSAGES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_NUMBER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_BF_ASSOC_CUST_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_BF_ASSOC_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_BF_COMP_CUST_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_BF_COMP_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_BF_CUST_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_BF_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_BF_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_BF_WIT_CUST_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_BF_WIT_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_CHARTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_CHART_GRPS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_CHART_ITEMS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_COLLECTION" TO "TIDAL";
```

```
GRANT EXECUTE ON "APPS"."FND_OAM_CPCHARTS_COL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DASHBOARD_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DEBUG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DOC_CATEGORY_ASSOC_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DOC_CATEGORY_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DOC_CATEGORY_TYPE_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCFG_API_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCFG_COMPILER_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCFG_INSTANCES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCFG_MAPPED_KEYS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCFG_OBJECTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCFG_PROCS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCFG_PROC_LIBRARY_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCFG_PROPERTIES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCFG_UTILS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_ARGS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_BUNDLES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_DIAG_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_DMLS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_PLSQLS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_RUNS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_STATS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_TASKS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_UNITS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DSCRAM_UTILS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DS_ALGOS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DS_POLICIES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_DS_PSETS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_EM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_KBF" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_KBF_SUBS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_METALINK_CREDS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_METALINK_DOCS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_METALINK_DOC_ASSOC_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_METVAL_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_MET_GRPS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_TESTER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_UMS_LOADER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_USER_INFO" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OAM_WS_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OBJECTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OBJECT_DEFINITIONS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OBJECT_INSTANCE_SETS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OBJECT_TABLESPACES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OBJECT_TABLESPACES_PUB" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ODF_GEN" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ODF_UPD" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OID_BULKLOAD" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OID_DIAG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OID_PLUG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OID_SUBSCRIPTIONS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OID_USERS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_OID_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ORACLE_SCHEMA" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_ORACLE_USER_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PERFMON" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PII_UTILITY_PVT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PLSQL_CACHE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PORTLET_DEPENDENCY_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PREFERENCE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PRINT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PRINTER_PKG" TO "TIDAL";
```

```
GRANT EXECUTE ON "APPS"."FND_PRINTER_STYLES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PRIVACY_ATTRIBUTES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PRIV_ATTR_CHILDREN_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PRODUCT_INITIALIZATION_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PROFILE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PROFILE_CATS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PROFILE_CAT_OPTIONS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PROFILE_HIERARCHY_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PROFILE_OPTIONS_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_PROFILE_OPTIONS_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_PROFILE_OPTIONS_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PROFILE_OPTION_VALUES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PROFILE_SERVER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PROGRAM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PROXY_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PUB_MESSAGE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PURGE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_PURPOSE_ATTRIBUTES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_RANDOM_NUMBER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_RANDOM_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_REGISTRATION_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_REGISTRATION_UTILS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_RELEASE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_REQUEST" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_REQUEST_COPY" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_REQUEST_INFO" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_REQUEST_SET" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_REQUEST_SETS_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_REQUEST_SETS_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_REQUEST_SETS_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_REQUEST_SET_STAGES_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_REQUEST_SET_STAGES_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_REQUEST_SET_STAGES_VL" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_REQ_SET_RUN_FORM_V" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_REQ_SET_RUN_FORM_V" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_RESPONSIBILITY_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_RESPONSIBILITY_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_RESPONSIBILITY_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_RESP_FUNCTIONS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_RESUB" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_RESUB_PRIVATE" TO "TIDAL";
GRANT DEBUG ON "APPS"."FND_RT_REQUEST" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_RT_REQUEST" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_RUN_FUNCTION" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SEARCH_EVENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SECURITY_GROUPS_API" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SECURITY_GROUPS_PKG" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_SECURITY_GROUPS_VL" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_SECURITY_GROUPS_VL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SECURITY_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SEC_BULK" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SEED_STAGE_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SEQNUM" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SEQ_CATEGORIES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SESSION_MANAGEMENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SESSION_UTILITIES" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SET" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SET_COPY" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SIGNON" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SOA_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SSO" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SSO_MANAGER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SSO_REGISTRATION" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_STAGE_FN_PARAMETERS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_STATS" TO "TIDAL";
```

```
GRANT EXECUTE ON "APPS"."FND_SUBMIT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SVC_COMPONENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SVC_COMPONENTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SVC_COMP_PARAMS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SVC_COMP_PARAM_VALS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SVC_COMP_REQUESTS_H_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SVC_COMP_REQUESTS_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SVC_COMP_TYPES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SYSTEM_ALERT_INTG_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_SYSTEM_EXCEPTION" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TABLESPACES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TABLESPACES_PUB" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TERRITORIES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TIMEZONES" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TIMEZONES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TIMEZONES_PVT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TIMEZONE_PUB" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TM_TESTER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TRACE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TRACE_UTILS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TRANSACTION" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TRANSACTION_PIPE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TRANSACTION_QUEUE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TS_MIG_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_TS_SIZE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_UMS_ANALYSIS_ENGINE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_UMS_LOADER" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_UPDATE_USER_PREF_PUB" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_UPDATE_USER_PREF_PUB_W" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_USER_AP_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_USER_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_USER_PKG_WRP" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_USER_PVT" TO "TIDAL";
GRANT INSERT ON "APPS"."FND_USER_RESP_GROUPS" TO "TIDAL";
GRANT SELECT ON "APPS"."FND_USER_RESP_GROUPS" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_USER_RESP_GROUPS_API" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_USER_VALIDATION" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_VAL_ATTRIBUTE_TYPES_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_VAULT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_VSET" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WEB" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WEBATTCH" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WEBFILE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WEBFILEPUB" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WEB_CONFIG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WEB_SEC" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WF_ENGINE" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WF_EVENT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WF_NOTIFICATION" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WF_STANDARD" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_WF_SYNCH" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_XDFAWINTERFACE_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_XDFDICTIONARY_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_XDF_APP_UTIL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."FND_XDF_UTIL_PKG" TO "TIDAL";
GRANT EXECUTE ON "APPS"."HR_SIGNON" TO "TIDAL";
GRANT EXECUTE ON "APPS"."ICX_CALL" TO "TIDAL";
GRANT EXECUTE ON "APPS"."JG_CONTEXT" TO "TIDAL";
GRANT EXECUTE ON "APPS"."MO_GLOBAL" TO "TIDAL";
GRANT INSERT ON "APPS"."WF_ROLES" TO "TIDAL";
GRANT SELECT ON "APPS"."WF_ROLES" TO "TIDAL";
GRANT INSERT ON "ICX"."ICX_PARAMETERS" TO "TIDAL";
GRANT SELECT ON "ICX"."ICX_PARAMETERS" TO "TIDAL";
```

```
GRANT "CONNECT" TO "TIDAL";
GRANT SELECT ON "APPS.FND_USER" TO "TIDAL";
GRANT SELECT ON "APPS.FND_APPLICATION" TO "TIDAL"";
```

tidal_syn.sql Commands

```
CREATE OR REPLACE SYNONYM APP_EXCEPTION FOR APPS.APP_EXCEPTION;
CREATE OR REPLACE SYNONYM DBMS_OBFUSCATION_TOOLKIT FOR SYS.DBMS_OBFUSCATION_TOOLKIT;
CREATE OR REPLACE SYNONYM FNDCP_CRM FOR APPS.FNDCP_CRM;
CREATE OR REPLACE SYNONYM FND_AMP_PRIVATE FOR APPS.FND_AMP_PRIVATE;
CREATE OR REPLACE SYNONYM FND_APPLICATION FOR APPS.FND_APPLICATION;
CREATE OR REPLACE SYNONYM FND_APPLICATION_PKG FOR APPS.FND_APPLICATION_PKG;
CREATE OR REPLACE SYNONYM FND_APPLICATION_S FOR APPS.FND_APPLICATION_S;
CREATE OR REPLACE SYNONYM FND_APPLICATION_TL FOR APPS.FND_APPLICATION_TL;
CREATE OR REPLACE SYNONYM FND_APPLICATION_VL FOR APPS.FND_APPLICATION_VL;
CREATE OR REPLACE SYNONYM FND_APPL_TOPS FOR APPS.FND_APPL_TOPS;
CREATE OR REPLACE SYNONYM FND_CACHE_VERSIONS FOR APPS.FND_CACHE_VERSIONS;
CREATE OR REPLACE SYNONYM FND_CLIENT_INFO FOR APPS.FND_CLIENT_INFO;
CREATE OR REPLACE SYNONYM FND_CODE_PKG FOR APPS.FND_CODE_PKG;
CREATE OR REPLACE SYNONYM FND_CONCURRENT FOR APPS.FND_CONCURRENT;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_BUSINESS_EVENT FOR APPS.FND_CONCURRENT_BUSINESS_EVENT;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_FILE FOR APPS.FND_CONCURRENT_FILE;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_PROCESSES FOR APPS.FND_CONCURRENT_PROCESSES;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_PROGRAMS FOR APPS.FND_CONCURRENT_PROGRAMS;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_PROGRAMS_TL FOR APPS.FND_CONCURRENT_PROGRAMS_TL;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_PROGRAMS_VL FOR APPS.FND_CONCURRENT_PROGRAMS_VL;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_QUEUES FOR APPS.FND_CONCURRENT_QUEUES;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_QUEUES_VL FOR APPS.FND_CONCURRENT_QUEUES_VL;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_REQUESTS FOR APPS.FND_CONCURRENT_REQUESTS;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_REQUESTS_S FOR APPS.FND_CONCURRENT_REQUESTS_S;
CREATE OR REPLACE SYNONYM FND_CONCURRENT_WORKER_REQUESTS FOR APPS.FND_CONCURRENT_WORKER_REQUESTS;
CREATE OR REPLACE SYNONYM FND_CONC_DATE FOR APPS.FND_CONC_DATE;
CREATE OR REPLACE SYNONYM FND_CONC_DEFERRED_ARGUMENTS FOR APPS.FND_CONC_DEFERRED_ARGUMENTS;
CREATE OR REPLACE SYNONYM FND_CONC_GLOBAL FOR APPS.FND_CONC_GLOBAL;
CREATE OR REPLACE SYNONYM FND_CONC_PP FOR APPS.FND_CONC_PP;
CREATE OR REPLACE SYNONYM FND_CONC_PP_ACTIONS FOR APPS.FND_CONC_PP_ACTIONS;
CREATE OR REPLACE SYNONYM FND_CONC_PROG_ONSITE_INFO FOR APPS.FND_CONC_PROG_ONSITE_INFO;
CREATE OR REPLACE SYNONYM FND_CONC_RELEASE_CLASSES FOR APPS.FND_CONC_RELEASE_CLASSES;
CREATE OR REPLACE SYNONYM FND_CONC_RELEASE_CLASSES_S FOR APPS.FND_CONC_RELEASE_CLASSES_S;
CREATE OR REPLACE SYNONYM FND_CONC_RELEASE_CLASSES_TL FOR APPS.FND_CONC_RELEASE_CLASSES_TL;
CREATE OR REPLACE SYNONYM FND_CONC_RELEASE_CLASSES_VL FOR APPS.FND_CONC_RELEASE_CLASSES_VL;
CREATE OR REPLACE SYNONYM FND_CONC_RELEASE_CLASS_UTILS FOR APPS.FND_CONC_RELEASE_CLASS_UTILS;
CREATE OR REPLACE SYNONYM FND_CONC_REQUEST_ARGUMENTS FOR APPS.FND_CONC_REQUEST_ARGUMENTS;
CREATE OR REPLACE SYNONYM FND_CONC_REQ_OUTPUTS FOR APPS.FND_CONC_REQ_OUTPUTS;
CREATE OR REPLACE SYNONYM FND_CONC_SSWA FOR APPS.FND_CONC_SSWA;
CREATE OR REPLACE SYNONYM FND_CP_IPP_PRINTERS FOR APPS.FND_CP_IPP_PRINTERS;
CREATE OR REPLACE SYNONYM FND_CP_SERVICES FOR APPS.FND_CP_SERVICES;
CREATE OR REPLACE SYNONYM FND_CRYPT_PKG FOR APPS.FND_CRYPT_PKG;
CREATE OR REPLACE SYNONYM FND_DATA_GROUPS FOR APPS.FND_DATA_GROUPS;
CREATE OR REPLACE SYNONYM FND_DATA_GROUP_UNITS FOR APPS.FND_DATA_GROUP_UNITS;
CREATE OR REPLACE SYNONYM FND_DATE FOR APPS.FND_DATE;
CREATE OR REPLACE SYNONYM FND_DEBUG_RULES FOR APPS.FND_DEBUG_RULES;
CREATE OR REPLACE SYNONYM FND_DEBUG_RULES_S FOR APPS.FND_DEBUG_RULES_S;
CREATE OR REPLACE SYNONYM FND_DEBUG_RULE_OPTIONS FOR APPS.FND_DEBUG_RULE_OPTIONS;
CREATE OR REPLACE SYNONYM FND_DESCRIPTIVE_FLEXS FOR APPS.FND_DESCRIPTIVE_FLEXS;
CREATE OR REPLACE SYNONYM FND_DESCRIPTIVE_FLEXS_TL FOR APPS.FND_DESCRIPTIVE_FLEXS_TL;
CREATE OR REPLACE SYNONYM FND_DESCR_FLEX_COLUMN_USAGES FOR APPS.FND_DESCR_FLEX_COLUMN_USAGES;
CREATE OR REPLACE SYNONYM FND_DESCR_FLEX_COL_USAGE_TL FOR APPS.FND_DESCR_FLEX_COL_USAGE_TL;
CREATE OR REPLACE SYNONYM FND_DESCR_FLEX_COL_USAGE_VL FOR APPS.FND_DESCR_FLEX_COL_USAGE_VL;
CREATE OR REPLACE SYNONYM FND_DESCR_FLEX_CONTEXTS FOR APPS.FND_DESCR_FLEX_CONTEXTS;
CREATE OR REPLACE SYNONYM FND_DESCR_FLEX_CONTEXTS_TL FOR APPS.FND_DESCR_FLEX_CONTEXTS_TL;
CREATE OR REPLACE SYNONYM FND_FILE_TEMP FOR APPS.FND_FILE_TEMP;
CREATE OR REPLACE SYNONYM FND_FLEX_VALIDATION_TABLES FOR APPS.FND_FLEX_VALIDATION_TABLES;
```

```
CREATE OR REPLACE SYNONYM FND_FLEX_VALUES FOR APPS.FND_FLEX_VALUES;
CREATE OR REPLACE SYNONYM FND_FLEX_VALUES_VL FOR APPS.FND_FLEX_VALUES_VL;
CREATE OR REPLACE SYNONYM FND_FLEX_VALUE_SETS FOR APPS.FND_FLEX_VALUE_SETS;
CREATE OR REPLACE SYNONYM FND_FORM_FUNCTIONS_VL FOR APPS.FND_FORM_FUNCTIONS_VL;
CREATE OR REPLACE SYNONYM FND_FORM_VL FOR APPS.FND_FORM_VL;
CREATE OR REPLACE SYNONYM FND_GLOBAL FOR APPS.FND_GLOBAL;
CREATE OR REPLACE SYNONYM FND_HASH_PKG FOR APPS.FND_HASH_PKG;
CREATE OR REPLACE SYNONYM FND_LANGUAGES FOR APPS.FND_LANGUAGES;
CREATE OR REPLACE SYNONYM FND_LANGUAGES_VL FOR APPS.FND_LANGUAGES_VL;
CREATE OR REPLACE SYNONYM FND_LOG FOR APPS.FND_LOG;
CREATE OR REPLACE SYNONYM FND_LOGINS FOR APPS.FND_LOGINS;
CREATE OR REPLACE SYNONYM FND_LOGINS_S FOR APPS.FND_LOGINS_S;
CREATE OR REPLACE SYNONYM FND_LOG_MESSAGES FOR APPS.FND_LOG_MESSAGES;
CREATE OR REPLACE SYNONYM FND_LOG_REPOSITORY FOR APPS.FND_LOG_REPOSITORY;
CREATE OR REPLACE SYNONYM FND_LOOKUPS FOR APPS.FND_LOOKUPS;
CREATE OR REPLACE SYNONYM FND_LOOKUP_TYPES FOR APPS.FND_LOOKUP_TYPES;
CREATE OR REPLACE SYNONYM FND_LOOKUP_VALUES_VL FOR APPS.FND_LOOKUP_VALUES_VL;
CREATE OR REPLACE SYNONYM FND_MENUS_VL FOR APPS.FND_MENUS_VL;
CREATE OR REPLACE SYNONYM FND_MENU_ENTRIES_VL FOR APPS.FND_MENU_ENTRIES_VL;
CREATE OR REPLACE SYNONYM FND_MESSAGE FOR APPS.FND_MESSAGE;
CREATE OR REPLACE SYNONYM FND_MIME_TYPES_VL FOR APPS.FND_MIME_TYPES_VL;
CREATE OR REPLACE SYNONYM FND_MO_PRODUCT_INIT FOR APPS.FND_MO_PRODUCT_INIT;
CREATE OR REPLACE SYNONYM FND_NEW_MESSAGES FOR APPS.FND_NEW_MESSAGES;
CREATE OR REPLACE SYNONYM FND_NODES FOR APPS.FND_NODES;
CREATE OR REPLACE SYNONYM FND_OAM_CONTEXT_FILES FOR APPS.FND_OAM_CONTEXT_FILES;
CREATE OR REPLACE SYNONYM FND_OAM_FORMS_RTI FOR APPS.FND_OAM_FORMS_RTI;
CREATE OR REPLACE SYNONYM FND_OAM_FRD_LOG FOR APPS.FND_OAM_FRD_LOG;
CREATE OR REPLACE SYNONYM FND_ORACLE_USERID FOR APPS.FND_ORACLE_USERID;
CREATE OR REPLACE SYNONYM FND_PRINT FOR APPS.FND_PRINT;
CREATE OR REPLACE SYNONYM FND_PRINTER FOR APPS.FND_PRINTER;
CREATE OR REPLACE SYNONYM FND_PRINTER_INFORMATION FOR APPS.FND_PRINTER_INFORMATION;
CREATE OR REPLACE SYNONYM FND_PRINTER_STYLES FOR APPS.FND_PRINTER_STYLES;
CREATE OR REPLACE SYNONYM FND_PRODUCT_GROUPS FOR APPS.FND_PRODUCT_GROUPS;
CREATE OR REPLACE SYNONYM FND_PRODUCT_INITIALIZATION FOR APPS.FND_PRODUCT_INITIALIZATION;
CREATE OR REPLACE SYNONYM FND_PRODUCT_INIT_CONDITION FOR APPS.FND_PRODUCT_INIT_CONDITION;
CREATE OR REPLACE SYNONYM FND_PRODUCT_INIT_DEPENDENCY FOR APPS.FND_PRODUCT_INIT_DEPENDENCY;
CREATE OR REPLACE SYNONYM FND_PRODUCT_INSTALLATIONS FOR APPS.FND_PRODUCT_INSTALLATIONS;
CREATE OR REPLACE SYNONYM FND_PROFILE FOR APPS.FND_PROFILE;
CREATE OR REPLACE SYNONYM FND_PROFILE_OPTIONS FOR APPS.FND_PROFILE_OPTIONS;
CREATE OR REPLACE SYNONYM FND_PROFILE_OPTIONS_VL FOR APPS.FND_PROFILE_OPTIONS_VL;
CREATE OR REPLACE SYNONYM FND_PROFILE_OPTION_VALUES FOR APPS.FND_PROFILE_OPTION_VALUES;
CREATE OR REPLACE SYNONYM FND_RANDOM_PKG FOR APPS.FND_RANDOM_PKG;
CREATE OR REPLACE SYNONYM FND_RAND_S FOR APPLSYS.FND_RAND_S;
CREATE OR REPLACE SYNONYM FND_RAND_STATES FOR APPLSYS.FND_RAND_STATES;
CREATE OR REPLACE SYNONYM FND_REQUEST FOR APPS.FND_REQUEST;
CREATE OR REPLACE SYNONYM FND_REQUEST_GROUPS FOR APPS.FND_REQUEST_GROUPS;
CREATE OR REPLACE SYNONYM FND_REQUEST_GROUP_UNITS FOR APPS.FND_REQUEST_GROUP_UNITS;
CREATE OR REPLACE SYNONYM FND_REQUEST_SETS FOR APPS.FND_REQUEST_SETS;
CREATE OR REPLACE SYNONYM FND_REQUEST_SETS_TL FOR APPS.FND_REQUEST_SETS_TL;
CREATE OR REPLACE SYNONYM FND_REQUEST_SETS_VL FOR APPS.FND_REQUEST_SETS_VL;
CREATE OR REPLACE SYNONYM FND_REQUEST_SET_PROGRAMS FOR APPS.FND_REQUEST_SET_PROGRAMS;
CREATE OR REPLACE SYNONYM FND_REQUEST_SET_PROGRAM_ARGS FOR APPS.FND_REQUEST_SET_PROGRAM_ARGS;
CREATE OR REPLACE SYNONYM FND_REQUEST_SET_STAGES FOR APPS.FND_REQUEST_SET_STAGES;
CREATE OR REPLACE SYNONYM FND_REQUEST_SET_STAGES_VL FOR APPS.FND_REQUEST_SET_STAGES_VL;
CREATE OR REPLACE SYNONYM FND_REQUEST_TESDB FOR APPS.FND_REQUEST_TESDB;
CREATE OR REPLACE SYNONYM FND_REQ_SET_RUN_FORM_V FOR APPS.FND_REQ_SET_RUN_FORM_V;
CREATE OR REPLACE SYNONYM FND_RESPONSIBILITY FOR APPS.FND_RESPONSIBILITY;
CREATE OR REPLACE SYNONYM FND_RESPONSIBILITY_VL FOR APPS.FND_RESPONSIBILITY_VL;
CREATE OR REPLACE SYNONYM FND_RESP_FUNCTIONS FOR APPS.FND_RESP_FUNCTIONS;
CREATE OR REPLACE SYNONYM FND_RT_REQUEST FOR APPS.FND_RT_REQUEST;
CREATE OR REPLACE SYNONYM FND_RUN_REQUESTS FOR APPS.FND_RUN_REQUESTS;
CREATE OR REPLACE SYNONYM FND_RUN_REQ_LANGUAGES FOR APPS.FND_RUN_REQ_LANGUAGES;
CREATE OR REPLACE SYNONYM FND_RUN_REQ_PP_ACTIONS FOR APPS.FND_RUN_REQ_PP_ACTIONS;
```

```
CREATE OR REPLACE SYNONYM FND_SECURITY_GROUPS_VL FOR APPS.FND_SECURITY_GROUPS_VL;
CREATE OR REPLACE SYNONYM FND_SESSIONS FOR APPS.FND_SESSIONS;
CREATE OR REPLACE SYNONYM FND_SUBMIT FOR APPS.FND_SUBMIT;
CREATE OR REPLACE SYNONYM FND_S_FILE_TEMP_ID FOR APPLSYS.FND_S_FILE_TEMP_ID;
CREATE OR REPLACE SYNONYM FND_TERRITORIES FOR APPS.FND_TERRITORIES;
CREATE OR REPLACE SYNONYM FND_USER FOR APPS.FND_USER;
CREATE OR REPLACE SYNONYM FND_USER_PREFERENCE FOR APPLSYS.FND_USER_PREFERENCES;
CREATE OR REPLACE SYNONYM FND_USER_PREFERENCES FOR APPS.FND_USER_PREFERENCES;
CREATE OR REPLACE SYNONYM FND_USER_RESP_GROUPS FOR APPS.FND_USER_RESP_GROUPS;
CREATE OR REPLACE SYNONYM FND_WEBFILE FOR APPS.FND_WEBFILE;
CREATE OR REPLACE SYNONYM FND_WEB_SEC FOR APPS.FND_WEB_SEC;
CREATE OR REPLACE SYNONYM GET_REQUEST_SET_PRIORITY FOR APPS.GET_REQUEST_SET_PRIORITY;
CREATE OR REPLACE SYNONYM HR_SIGNON FOR APPS.HR_SIGNON;
CREATE OR REPLACE SYNONYM HTP FOR APPS.HTP;
CREATE OR REPLACE SYNONYM ICX_CALL FOR APPS.ICX_CALL;
CREATE OR REPLACE SYNONYM ICX_PARAMETERS FOR APPS.ICX_PARAMETERS;
CREATE OR REPLACE SYNONYM JG_CONTEXT FOR APPS.JG_CONTEXT;
CREATE OR REPLACE SYNONYM MO_GLOBAL FOR APPS.MO_GLOBAL;
CREATE OR REPLACE SYNONYM SABDG_JOBS FOR SABDG.SABDG_JOBS;
CREATE OR REPLACE SYNONYM SABDG_JOBS_S FOR SABDG.SABDG_JOBS_S;
CREATE OR REPLACE SYNONYM SABDG_MENU_TEMP FOR SABDG.SABDG_MENU_TEMP;
CREATE OR REPLACE SYNONYM SABDG_MENU_TEMP_S FOR SABDG.SABDG_MENU_TEMP_S;
CREATE OR REPLACE SYNONYM SABDG_PARAMETERS FOR SABDG.SABDG_PARAMETERS;
CREATE OR REPLACE SYNONYM SABDG_RELEASED_REQS FOR SABDG.SABDG_RELEASED_REQS;
CREATE OR REPLACE SYNONYM UTL_RAW FOR SYS.UTL_RAW;
CREATE OR REPLACE SYNONYM WF_ROLES FOR APPS.WF_ROLES;
```

sys_grant.sql Commands

```
GRANT EXECUTE ON "SYS"."DBMS_APPLICATION_INFO" TO "TIDAL";
GRANT DEBUG ON "SYS"."DBMS_OBFUSCATION_TOOLKIT" TO "TIDAL";
GRANT EXECUTE ON "SYS"."DBMS_OBFUSCATION_TOOLKIT" TO "TIDAL";
GRANT EXECUTE ON "SYS"."DBMS_SYSTEM" TO "TIDAL";
GRANT SELECT ON "SYS"."ERROR$" TO "TIDAL";
GRANT EXECUTE ON "SYS"."HTP" TO "TIDAL";
GRANT SELECT ON "SYS"."TRIGGER$" TO "TIDAL";
GRANT DEBUG ON "SYS"."UTL_RAW" TO "TIDAL";
GRANT EXECUTE ON "SYS"."UTL_RAW" TO "TIDAL";
```

B

Appendix B—sabdg_global Package

The sabdg_global package contained in this appendix is used when you need to install the Oracle Applications Bridge using a custom APPS user as described in Installing the Oracle Application Bridge Using a Custom APPS User, page 15.

```
CREATE OR REPLACE PACKAGE BODY sabdg_global AS
PROCEDURE decrypt(pc_err_code OUT VARCHAR2,
                  pn_job_trx_id IN NUMBER,
                  pc_input IN VARCHAR2,
                  pc_output OUT VARCHAR2,
                  pc_debug IN VARCHAR2 DEFAULT NULL) IS

BEGIN
    DECLARE

      vc_err_code         VARCHAR2(100);
      vc_dbg_err_code     VARCHAR2(100);
      vc_debug             VARCHAR2(100) := substr(pc_debug,1,32);
      vc_module            VARCHAR2(100) := 'GLOBAL.DECRYPT';
      vc_message        VARCHAR2(100);
        vh_input              VARCHAR2(2048) := pc_input;
        vc_input              VARCHAR2(2048) ;
        vt_job_trx_id            NUMBER := pn_job_trx_id;
        vc_decrypt               VARCHAR2(2048);
        vc_key                VARCHAR2(56);

    BEGIN
    BEGIN
      SELECT to_char(creation_date,'YYYYHH24MMMIDDSSMM')
      INTO vc_key
      FROM TIDAL.sabdg_jobs
      WHERE job_trx_id = vt_job_trx_id;
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        vc_err_code := 'BDG-10092 '||'Job does not exists';
        IF vc_debug IS NOT NULL THEN
         vc_message := vc_err_code;
          END IF;
    END;
    IF vc_err_code IS NULL THEN
        vc_input :=   UTL_RAW.CAST_TO_VARCHAR2(HEXTORAW(vh_input));
        dbms_obfuscation_toolkit.DESDecrypt(input_string => vc_input,key_string =>
vc_key,decrypted_string => vc_decrypt);
        pc_output := ltrim(rtrim(vc_decrypt));
       pc_err_code := vc_err_code;
      ELSE
        pc_output := NULL;
        pc_err_code := vc_err_code;
      END IF;
```

```
    EXCEPTION
    WHEN OTHERS THEN
       pc_err_code := SUBSTR(SQLERRM,1,100);
       RAISE;
    END;
END decrypt;
END sabdg_global;
/
```