<div align="right">

APPENDIX **A**

</div>

Setting Up Oracle for Prime Fulfillment

This appendix describes how to set up an Oracle Database 11g, Enterprise Edition Release 11.2.0.1.0 - 64 bit Production server that works with Cisco Prime Fulfillment. This appendix is written for database administrators who are familiar with Oracle.

> **Note** Prime Fulfillment 6.2 was tested with Oracle Database 11g, Enterprise Edition Release 11.2.0.1.0 - 64 bit Production. If you would like to use another version of Oracle, see Oracle's compatibility information.

This chapter does not cover all the details about installing and setting up this Oracle server. For the complete information, see the Oracle Installation Guide. Prime Fulfillment provides schema files to be loaded on an Oracle server. The Prime Fulfillment customer must decide on the Oracle server configuration.

This appendix contains the following sections that should be addressed in order:

This appendix also contains a "Troubleshooting" section on page A-8.

Prerequisites

Prime Fulfillment support for an Oracle database is for Oracle Database 11g, Enterprise Edition Release 11.2.0.1.0 - 64 bit Production. This is the version of Oracle with which Prime Fulfillment 6.2 was tested. If you would like to use another version, see Oracle's compatibility information.

The remaining prerequisites are as specified in the following steps:

Step 1   When the Oracle server is set up, the following initialization parameters should be in the database **init** file:

- db_block_size = 8192 or larger

- compatible = "11.2.0.1.0"

- open_cursors = 512 or larger

- processes = 150 or larger

Step 2   Record the following information about the server setup. This information is needed during the Prime Fulfillment installation:

- Oracle server name

- Oracle server instance identifier (SID)

✎
Note   This is specified in Step 21.

- database port number for client connections (default: 1521)

- Oracle user ID and password created for Prime Fulfillment

✎
Note   Create an Oracle database userid and password. This is needed during Prime Fulfillment installation. Do not use the **system** or **sys** account for Prime Fulfillment data. Use a separate table space other than the system table space.

Step 3   Before loading the Prime Fulfillment database schema, make sure the Oracle database has been successfully started and the database user has proper privileges. See the Oracle Administration Guide for detailed instructions about how to set up the database and manage user accounts.

Step 4   Proceed to the section "Installing Oracle."

Installing Oracle

The following information about an Oracle installation is just one example.

You must install Oracle before you install the Cisco PRIME Fulfillment (Prime Fulfillment) software (or at least know your Oracle home directory, host machine, and Oracle Server ID), and your database and its listener must be running when you launch the Prime Fulfillment servers.

If you intend to use the same Oracle installation with more than one installation of the Prime Fulfillment servers, you must create a unique Oracle SID and Oracle tablespace for each Prime Fulfillment installation.

init*ORACLE_SID*.ora

This file should already exist in the `/dbs` subdirectory of your Oracle installation. (The filename contains your database's SID in place of *ORACLE_SID*. For example, if you named your database Prime Fulfillment, this file is named `initISC.ora`.)

oratab

The `oratab` file should be located in the `/var/opt/oracle` directory on the machine on which the database is installed. It is used by Oracle's **dbstart** utility to identify your database.

The `oratab` file must contain the following line:

database_name:*location_of_your_Oracle_executables*:Y

If your Oracle home directory is `/oracle/10.2.0` and your database SID is Prime Fulfillment, the oratab entry would be as follows:

Prime Fulfillment:/oracle/10.2.0:Y

This file identifies the name and location of your database for the Oracle utility **dbstart** (and its companion **dbshut**). The **dbstart** utility starts Oracle; the "Y" at the end of the `oratab` entry tells the **dbstart** utility to open the database named Prime Fulfillment. (Substitute your database name for Prime Fulfillment in the sample. List the path to your Oracle installation as an absolute path, not a relative path.)

To make this happen automatically following a reboot (after a power interruption, for example), execute the **dbstart** utility from a script in the `/etc/init.d` directory on the Oracle host machine.

Verifying and Launching Oracle

Your Oracle database must be open before you can install or use the Prime Fulfillment software.

First, verify the Oracle processes, as described in the following section. If the processes are running, you can skip the succeeding section.

Verifying Oracle Processes

Log into the Oracle host machine and enter the following on the command line to see if the Oracle processes are running:

ps -ef | grep ora_

ps -ef | grep tnslsnr

If there is no output displayed from the **ps** command, Oracle is not running.

If Oracle is running and the listener process is running, you should see something similar to the following:

```
oracle   328   1   0   14:25:18      0:00 ora_pmon_ISC
oracle   328   1   0   14:25:18      0:00 ora_dbwr_ISC
oracle   328   1   0   14:25:18      0:00 ora_lgwr_ISC
oracle   328   1   0   14:25:18      0:00 ora_ckpt_ISC
oracle   328   1   0   14:25:18      0:00 ora_smon_ISC
```

```
oracle   328   1   0   14:25:18      0:00 ora_reco_ISC
oracle   328   1   0   14:25:18      0:00 ora_wmon_ISC
oracle   328   1   0   14:25:18      0:00 tnslsnr LISTENER -inherit
```

These are the Oracle processes currently running (your output might not match this list exactly, depending on which Oracle components are installed).

Launching Oracle and Opening Your Database

Your Oracle database must be open before you can install or use the Prime Fulfillment software.

If Oracle is not currently running, you must use the startup utilities located in the `/bin` subdirectory of your Oracle installation.

To open your database, you must be logged into the Oracle host workstation under the Oracle administrator (DBA) user ID; you then locate your `$ORACLE_HOME/bin` subdirectory.

On the command line, enter the following:

dbstart

The `dbstart` script starts the database identified in the `oratab` file. If the database starts successfully, you should see several lines of output, including the following:

```
SQL>   Connected to an idle instance.
SQL>   ORACLE instance started.
```

...and ending with the following:

```
Server Manager Complete.
Database "Prime Fulfillment" warm started.
```

If the listener process is not running, you must also start that process. On the command line, enter the following:

lsnrctl start

You should see several lines of output as the process is invoked, then you should see output similar to the following:

```
Services Summary...
   Prime Fulfillment    has 1 Service handler(s)
```

The command completed successfully.

Setting Up Your Oracle Files

To configure your database to work with the Prime Fulfillment software, you must create a tablespace and configure several files.

You must be logged into the Oracle host using the user ID (such as `oracle`) created during the Oracle installation procedure.

Oracle Tablespace Requirements

You must create an Oracle tablespace for your Prime Fulfillment tables.

To create the tablespace, Oracle must be running and your database must be open.

Log into the Oracle host using the `oracle` user ID. Identify (or create) the directory where your Prime Fulfillment data should be stored, and grant write permission to the `oracle` user ID. Be sure your `ORACLE_SID` and `ORACLE_HOME` environment variables are set correctly, then launch the Oracle utility `sqlplus`, which is located in the `$ORACLE_HOME/bin` directory.

At the SQL prompt, enter the following on the command line:

connect / as sysdba;

CREATE TABLESPACE ISC_DAT

DATAFILE '/*your_data_directory*/ISC_DAT_01.dbf' size 500M

autoextend on

next 50M

maxsize unlimited;

The data directory you specify must already exist. The `TABLESPACE` and `DATAFILE` names are arbitrary. You can use any names that help you keep track of which files are associated with which database. The only requirement is that the name given to the tablespace at the time of its creation (`ISC_DAT` in the example) must be the same as the default tablespace listed when you create the `prime` user account.

The autoextend option allows ORACLE to automatically extend your data file. The maximum size of the data file is limited only by the available space on the file's disk.

prime Oracle User Account

While `sqlplus` is still running, create an `prime` user account using your `ISC_DAT` tablespace as follows:

CREATE USER prime IDENTIFIED BY cisco

DEFAULT TABLESPACE ISC_DAT;

GRANT CONNECT TO prime;

GRANT RESOURCE TO prime;

You should use this user and password when entering Oracle information in the script `prime.configure`.

Testing Your Oracle Database Connection for Oracle User prime

When you have configured your database and listener file, enter the following (for the Oracle user `prime` and for the database named Prime Fulfillment) on the command line:

sqlplus <*username*>/<*password*>

<*username*> is a database username (in our previous example, we used **prime**).

<*password*> is a database password (in our previous example, we used **cisco**).

If your system is set up properly (and your Oracle database is running), you should see a message advising you that you are connected to Oracle. Enter `quit` on the command line to exit the database.

Prime Fulfillment Software Installation

> **Note** The Prime Fulfillment database schema files are loaded during the installation.

Perform the following:

Step 1 Follow the **custom** install instructions in Chapter 2, "Installing and Logging Into Cisco Prime Fulfillment," section Installing Prime Fulfillment, page 2-2, and log in, as explained in the section Logging In for the First Time, page 2-21.

Step 2 Proceed to the section "Verify Prime Fulfillment Installation with Oracle".

Verify Prime Fulfillment Installation with Oracle

To verify the Prime Fulfillment installation with Oracle, do the following:

Step 1 Run **sqlplus** *<oracle_id>*/*<oracle_password>* on the Oracle server.

Step 2 From the **SQL>** prompt, run **select host_name** from **vpnsc_host;**

This command returns the installed Prime Fulfillment hostname.

Step 3 Log into the Prime Fulfillment server.

Step 4 Check the file /opt/PrimeFulfillment/etc/vpnsc.properties and make sure that the **<oracle server>** and *<ORACLE_SID>* are correct in the following entry in the file:

repository.persistence.url=jdbc:oracle:thin:@*<oracle server>*:*<ORACLE_SID>*

Step 5 Execute the schema verification script to verify the repository schema version, as follows:

cd $PRIMEF_HOME

cd /bin

./checkSchemaVer.sh *<oracle_id>*/*<oracle_password>*

where: *<oracle_id>* is the Prime Fulfillment userid in the Oracle database and *<oracle_password>* is its password.

Step 6 The output from the script should be "Current schema version = 6.0". If that is not the output from the script, Prime Fulfillment might not have been installed properly or the Prime Fulfillment repository might not have been upgraded successfully.

Importing an Oracle Repository Dump

To import the Oracle repository dump in Prime Fulfillment, do the following:

Step 1 Log into the Oracle Webapp.

Step 2 Create the User and Tablespace.

Step 3 Log into the Oracle Server and source the enviornment:

rlogin <Oracle Server Name>

where:

<Oracle Server Name> Specify the Oracle Server Name that is being used.

Step 4 Enter: **su - <user name>**

Step 5 Enter: **cd $ORACLE_HOME/bin**

Step 6 Enter: **source coraenv**

Step 7 Enter: **setenv ORACLE_HOME to $ORACLE_HOME/bin**

Step 8 Enter: **setenv ORACLE_SID to orcl**

Step 9 Copy the .dmp file to a directory on the Oracle Server.

Step 10 Enter: **cd $ORACLE_HOME/bin**

Step 11 Run the command: **imp**

When you run this script, you are asked to enter the values for the following prompts:

a. Import file: <specify the full path to .dmp file>

b. Enter insert buffer size: **30720** (accept the default value)

c. List contents of import file only: **no** (accept the default value)

d. Ignore create error due to object existence: **no** (accept the default value)

e. Import grants: **no** (accept the default value)

f. Improt table data: **yes**

g. Import entire export file: **yes**

Configuring Oracle RAC

In addition to having already installed Prime Fulfillment and followed the steps required to configure an Oracle server, you must follow these steps when using Oracle Real Application Clusters (RAC). Prime Fulfillment does not support client load balancing with Oracle RAC.

Note A limitation of Oracle RAC is that any uncommitted transactions made during an instance or node failure and recovery period are lost. The recovery of these transactions is not supported. For this reason, the behavior of tasks that are running at the time as an instance or node fail over is undetermined. These tasks should be redeployed.

In case of a failure, for more information see the Oracle RAC documentation for database instance recovery time details.

Step 1 Verify that the new Oracle RAC servers are available and have an Prime Fulfillment tablespace with user configured. If you need help setting this up, see the "Verify Prime Fulfillment Installation with Oracle" section on page 6.

Step 2 Modify $PRIMEF_HOME/runtime.properties to have the correct values for the following parameters:

- **db_server**
- **db_url**—A sample URL is jdbc:oracle:thin:@//Virtual IP:*<port>*/globalSID, where *<port>* is the port number, which defaults to **1521**.
- **db_driver**
- **db_usr**
- **db_pwd**

Step 3 Prepopulate the database user name and password into the database
./pime.sh execjava.sh com.cisco.vpnsc.common.BootStrapHelper put repository *<oracle username> <oracle password>*

Step 4 If running, use the **./prime.sh stop** command to stop Prime Fulfillment.

Step 5 Verify that the value for the DCPL property watchdog/server/dbpoller/connectionextend is still set to the default: 5. See Appendix C, "DCPL Properties," in the *Cisco Prime Fulfillment User Guide 6.2*.

Step 6 To update the database with the changes, enter:

./prime.sh startdb

./prime.sh initdb.sh

Step 7 Use **./prime.sh stop** to stop the database.

Step 8 Then enter **./prime.sh start** to start Prime Fulfillment.

Backup of Oracle Database

See Appendix D, "Backup and Restore of Prime Fulfillment Repository and Standby System."

Troubleshooting

This section lists Oracle database-related trouble shooting tips based on the following error messages:

- **ORA-01631: max # extents (4096) reached in table xyz**

  If you receive this message, it is typically an Oracle server storage configuration issue. This problem occurs when the tablespace for Prime Fulfillment exceeds the limit set by the database configuration. To prevent this, plan proper storage before Prime Fulfillment is set up. If this problem occurs, increase the initial or next extent, increase the growth percentage (such as, PCT_INCREASE), or reset the number of max extents (can be unlimited). The Prime Fulfillment data must be exported and imported to the tablespace with the new tablespace parameters.

- **Unable to contact Rbac Manager**

  If you receive this message on Prime Fulfillment and are unable to log in, this might be because Prime Fulfillment cannot connect to the Oracle database. To avoid this situation, increase the number of Oracle server processes.

- **Cannot log into Inventory Manager or Topology Manager**

  If you cannot log into the Inventory Manager or Topology Manager, verify that the Oracle hostname is accessible from a client machine, either by DNS or a host file.

- **Resynchronize Prime Fulfillment with new or updated Oracle ID and password**

    If the Oracle ID and password change after the Prime Fulfillment installation, you must execute the following:

    a. execjava.sh com.cisco.vpnsc.common.BootStrapHelper put repository *<oracle_id>* *<oracle_password>*

    b. update etc/spe/cns.properties and modify these two properties:

    DataAccess.principal.1 *<oracle_id>*

    DataAccess.credentials.1 *<oracle_password>*

Troubleshooting