

**Oracle® Product Data Quality**  
PIM Connector Implementation Guide  
Version 5.6.1  
**E23406-01**

April 2011

Copyright © 2001, 2011 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Contents

<b>Preface</b> .....	v
Audience .....	v
Related Documents .....	v
Conventions .....	vi
<b>1 Introduction</b>	
<b>2 Installing the Schema and Template</b>	
<b>Installation Prerequisite</b> .....	2-1
<b>Installation Overview</b> .....	2-1
Preparing for Installation .....	2-1
Installing the Client Template Files .....	2-2
Services for Excel Templates .....	2-2
Governance Studio Project Files .....	2-2
Installing the DSA and Data Lens Template Files .....	2-2
DSA Packages .....	2-3
Data Lenses .....	2-3
<b>Creating the d1s User Schema</b> .....	2-3
<b>3 Installing the Database Schema</b>	
<b>Installing the Database Schema</b> .....	3-1
<b>Creating a New Batch Assignee and Batch User</b> .....	3-2
Creating the Roles .....	3-2
Creating the Users .....	3-3
Updating the Profile Options .....	3-3
DLS Profile Option Information .....	3-4
<b>4 Installing the Application Files</b>	
<b>Installation Prerequisite</b> .....	4-1
<b>Installing the AU PIM Connector Application Template Files</b> .....	4-1
<b>Oracle Product Data Quality DSA Overview</b> .....	4-2
Default Entries in DSAs .....	4-2
Creating and Updating Semantic Models from PIM Metadata Import .....	4-3
Importing Alternate Catalog Metadata .....	4-3

Creating and Updating the Semantic Cache .....	4-3
Creating Production Batches .....	4-3
Processing a Batch of Data Thorough the Cleansing and Matching Process .....	4-3
Secondary Data Service Applications for Reprocess and Apply .....	4-3
Pre-Configured Governance Studio Projects .....	4-4
Pre-Configured AutoBuild Wizard Excel Worksheets .....	4-4

---

---

# Preface

This manual highlights the installation and configuration of Oracle Product Data Quality Connector.

By extending Oracle Product Data Quality capabilities, Oracle enables customers to improve the accuracy, completeness, and integrity of their master product data.

To understand the concepts presented, you must use this reference guide in conjunction with the Oracle Product Data Quality documents listed in "[Related Documents](#)" on page -v.

You must have Oracle Product Data Quality client software installed on your computer.

The guide uses the Vision Demo Database as a means to provide access to a common set of data, which is available to use as examples and validate the setup and configuration of the server. The guide will use the Item Class Category Name of Capacitors under the Item Class of Passives.

---

---

**Note:** Oracle Product Data Quality is only certified with the Oracle Site Hub and Oracle Product Hub for Retail Media Pack.

---

---

## Audience

You should have a basic understanding of the DataLens Technology.

This document is intended for IT administrators.

## Related Documents

For more information, see the following documents in the documentation set:

- The *Oracle Product Data Quality PIM Connector User's Guide* provides highlights of the core process steps and features of Oracle Product Data Quality Connector.
- The *Oracle Product Data Quality Oracle DataLens Server Installation Guide* provides detailed Oracle Product Data Quality Oracle DataLens Server installation instructions.
- The *Oracle Product Data Quality Oracle DataLens Server Administration Guide* provides information about installing and managing an Oracle DataLens Server.
- The *Oracle Product Data Quality PIM Connector API Interface Guide* provides information about using the Oracle PIM Connector API.

- The *Oracle Product Data Quality COM Interface Guide* provides information about installing and using the Oracle DataLens Server COM APIs.
- The *Oracle Product Data Quality Java Interface Guide* provides information about installing and using the Oracle DataLens Server Java APIs.
- The *Oracle Product Data Quality Application Studio Reference Guide* provides information about creating and maintaining Data Service Applications (DSAs).
- The *Oracle Product Data Quality AutoBuild Reference Guide* provides information about creating initial data lens based on existing product information and data lens knowledge.
- The *Oracle Product Data Quality Knowledge Studio Reference Guide* provides information about creating and maintaining data lenses.
- The *Oracle Product Data Quality Governance Studio Reference Guide* provides information about creating and maintaining Data Service Applications (DSAs).
- The *Oracle Product Data Quality Glossary* provides definitions to commonly used Oracle Product Data Quality technology terms.
- The *Oracle Product Data Quality Services for Excel Reference Guide* provides information about creating a DSA based on data contained in a Microsoft Excel spreadsheet.
- The *Oracle Product Data Quality Task Manager Reference Guide* provides information about managing tasks created with the Task Manager or Governance Studio applications.

See the latest version of this and all documents listed at the Oracle Product Data Quality Documentation Web site at:

[http://download.oracle.com/docs/cd/E20593\\_01/index.htm](http://download.oracle.com/docs/cd/E20593_01/index.htm)

## Conventions

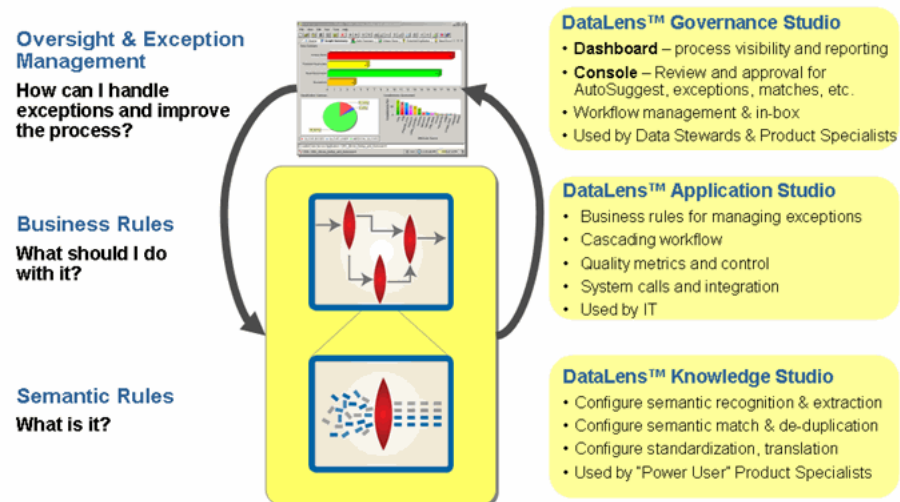
The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, text that you enter, or a file, directory, or path name.
<b>monospace</b>	Boldface, monospace type indicates commands or text that you enter.

## Introduction

Oracle DataLens Server is built on industry-leading DataLens™ Technology to standardize, match, enrich, and correct product data from different sources and systems. The core DataLens Technology uses patented semantic technology designed from the ground up to tackle the extreme variability typical of product data.

Oracle Product Data Quality uses three core DataLens Technology modules: Governance Studio, Knowledge Studio, and Application Studio. The following figure illustrates the process flow of these modules.



The Oracle DataLens Server can be configured to run with multiple servers:

- Oracle DataLens Administration Server
- Oracle DataLens Transform Server

The administration of all servers in a multi-server configuration is controlled with the Oracle DataLens Administration Server. The purpose of the Administration Server is to manage the various administrative tasks of the servers for the server groups (referred to as Transform Servers) and can itself serve as its own Transform Server when installed alone in a single node configuration. By spreading the data processing load across multiple servers the Oracle DataLens Server system provides scalability and configuration control over the various functional areas involved in developing, testing, and ultimately executing Oracle DataLens jobs.

---

The type of Oracle DataLens Server Group that a Transform Server belongs to controls the individual server functionality. A server group may contain one or many physical servers. There are three server groups:

- Development Server Group
- Production Server Group
- Quality Assurance Server Group

The Server Groups contain individual Oracle DataLens Servers on physical machines that can **load balance** among servers within the group. The data lenses and DataLens Service Applications (DSAs) are deployed from one group to the next beginning with the development group, then migrating to the Quality Assurance Group for testing before arriving in the Production Group for deployment to production. This multiple group migration function facilitates an enterprise business process where multiple functional areas work on data lens objects in stages before releasing them to production.

The integration of Oracle Apps Unlimited (AU) Product Information Management (PIM) and Oracle Product Data Quality adds a package called the "AU PIM Connector" to the Oracle Product Data Quality solution upload package.



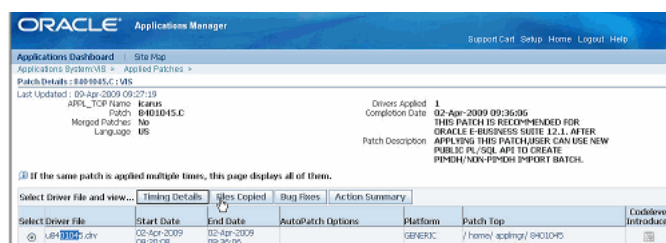
## Installing the Schema and Template

### Installation Prerequisite

First, install the Oracle E-Business Suite (EBS) server including the latest patch sets.

For 12.1 you must install the Oracle Batch Creation API patch (patch number 8401045:R12.EGO.C).

You can check the patch version from the Applications Manager as follows:



### Installation Overview

The AU PIM Connector consists of the following:

- `oracle_product_hub_connector.zip` - This is the main installation file for the AU PIM Connector.
- `dls_pim_connector_install.zip` - AU PIM Connector E-Business Suite integration files.
- `opdq_client_templates.zip` - Governance Studio and Services for Excel client template files.
- `opdq_server_templates.zip` - AU PIM Connector Data Lens and DSA template files that can be automatically deployed to your Oracle DataLens Server.

### Preparing for Installation

You must obtain the current AU PIM Connector product, which is delivered as a patch, from the Oracle Support Web site.

Locate and download the AU PIM Connector version 5.6 patch:

1. Browse to the My Oracle Support Web site:

<https://support.oracle.com>

2. Log in or register.
3. Click the **Patches & Updates** tab.
4. Click **Product or Family (Advanced Search)**.
5. Search for Oracle Product Data Quality by beginning to enter the name in the **Product** field and then select it when it is displayed in the list.
6. Select the **Release** list, click the arrow adjacent to the Oracle Product Data Quality folder.
7. Select the **OPDQ 5.6** release and then click **Close**.
8. Click **Search**.
9. Select the AU PIM Connector version 5.6 patch from the list of patches, and then click **Download**.
10. Click the patch file name, and then save it to a temporary directory.
11. Change directories to the temporary directory.
12. Unzip the patch file in the temporary directory.
13. Unzip the `oracle_product_hub_connector.zip` file.

## Installing the Client Template Files

The `opdq_client_templates.zip` file contains the following directory structure:

```
opdq_client_templates
    services_for_excel_templates
    gov_studio_projects
```

The client template files are installed on the client computer where they are used by the Oracle Product Data Quality client applications to interface to the AU PIM Connector. For installation information, see "[Installing the Application Files](#)" on page 4-1.

### Services for Excel Templates

The `services_for_excel_templates` directory contains two Excel files that contain VBA applications that are used to pull the semantic model data from PIM and to add an alternate catalog to an existing data lens. For information about how the AutoBuild templates are used to create a data lens and add an alternate catalog to the data lens, see the *Oracle Product Data Quality PIM Connector User Guide*.

### Governance Studio Project Files

The `gov_studio_projects` directory contains Governance Studio project files (\*.ams) used to create a semantic cache for matching applications, create a batch of data for AU PIM to match against, and run the cleansing and matching operations.

## Installing the DSA and Data Lens Template Files

The `opdq_server_templates.zip` file contains the following directory structure:

```

opdq_server_templates
    autodeploy
        dsa
        lens

```

The `autodeploy` directory containing the DSA and data lens template files are copied to the Oracle DataLens Server computer for automatic deployment (known as `autodeploy`) for use by the Oracle Product Data Quality client applications to interface to the AU PIM Connector interface.

### DSA Packages

The `dsa` directory contains the DSA files (\*.xml) that perform the backend data cleansing and matching transformations. These files are imported to the Oracle DataLens server using the Application Studio. For information about checking in DSAs to an Oracle Product Data Quality, see the *Oracle Product Data Quality Application Studio Reference Guide*.

### Data Lenses

The `lens` directory contains the default data lens project files (\*.project) for all DSAs that process items in Oracle Product Data Quality.

## Creating the d1s User Schema

The AU PIM Connector API components are installed in a Oracle Database user account that must be named, `d1s`.

1. Copy the `d1s_pim_connector_install.zip` file that you extracted from the patch set to your Oracle E-Business Server.
2. Login as user, `applmgr` (or whichever user installed the original Oracle E-Business instance).
3. Ensure that `applmgr` environment is sourced.

Following is an example of how to run the `applmgr` environment file:

```
. /d01/oracle/VIS/apps/apps_st/appl/APPSVIS_hercules.env
```

4. Check that the environment is set by running the following command and getting a value:

```
echo $TWO_TASK
```

5. Unzip the `d1s_pim_connector_install.zip` file into a temporary directory owned by the `applmgr` user.
6. Change the file permissions of the AU PIM Connector installation script:

```
chmod +x *.sh
```

7. Create the AU PIM Connector Database Schema (though if the system user is being used, `sys` can also be used):

```
sqlplus system/manager@VIS < d1s_user_creation.sql
```



---

---

## Installing the Database Schema

The main installation script, `dls_pim_connector_install.sh`, installs tables, views, packages, etc. into a `dls/dls` schema that you create. The necessary grants and privileges to the tables accessed by the Oracle Product Data Quality are done in the `apps/apps` schema.

- The installation script uses the existing *Transaction* tablespace and for the installation of the DLS tables in indexes such as the semantic key cache. This is consistent with current Oracle practice (see MetaLink note 248857.1 under the Product tablespaces link).
- The temporary tables used by the Oracle Product Data Quality will be written to the Interface tablespace (see MetaLink note 248857.1 under the Product tablespaces link).
- The installation script creates a new schema called `dls` that is granted quota on the Transaction and Interface table spaces.

### Installing the Database Schema

1. Ensure that you are logged into the E-Business Server as `applmgr` and have sourced its environment file.
2. Change directories to the temporary directory that contains the `dls_pim_connector_install`.
3. Install the database schema:  

```
./dls_pim_connector_install.sh
```

Please enter Apps Schema Name for VIS:
  4. Enter:  

```
apps
```

Please enter Apps Password for VIS:
    5. Enter:  

```
apps
```

Please enter DLS Schema Name for VIS:
      6. Enter:  

```
dls
```

Please enter DLS Password for VIS:
        7. Enter:

### **dls**

The `dls` database schema has now been populated with the tables, views, packages, synonyms and triggers. The `apps` schema has also been populated with the grants and synonyms needed to run the Oracle Product Data Quality AU PIM Connector.

A log file is written to the `dls_pim_connector_install` directory.

8. Review the installation log file and ensure that there were no errors.

## **Creating a New Batch Assignee and Batch User**

The overview for setting up the above users is as follows.

- Create two new roles, one role to be used by `DLSUSER` and one role to be used by `DLSMANAGER`. These two new roles could be the same name as the user names.
- Add the required privileges to each of the two new roles.
- Create the two new users of `DLSUSER` and `DLSMANAGER`, assigning the users the above roles as appropriate.

## **Creating the Roles**

The following list contains the privileges that are needed to run Oracle Product Data Quality AU PIM Connector.

Create news role named, `dlsuser` and `dlsmanager`, with the following privileges:

- Add/Delete Approved Manufacturer Parts Item
- Add/Delete Customer Item Cross References
- Add/Delete Item Cross References
- Add/Delete Related Items
- Add/Update Item Revision
- Change Item Status
- Create Item Supplier Assignment
- Create/Edit Item Organization Assignments
- Edit Item
- Edit Item Catalog Assignments
- Edit Key Metrics
- Edit Source System Item Cross Reference
- View Approved Manufacturer Parts Item
- View Customer Item Cross References
- View Item
- View Item/Item revision lifecycle
- View Item Catalog Assignments
- View Item Cross References
- View Item Organization Assignments

- View Item Revision List
- View Item Supplier Assignment
- View Key Metrics
- View Related Items

## Creating the Users

Create the two users as follows:

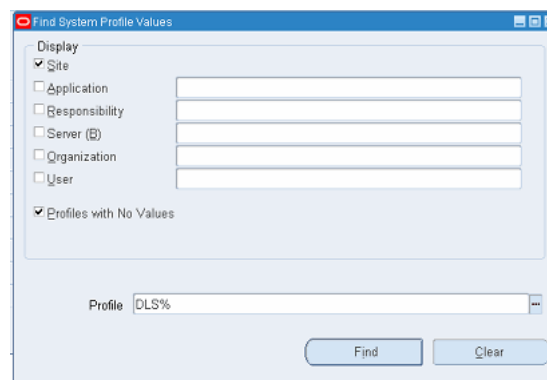
- `dlsuser` - The user assigned to the batches created by the Oracle Product Data Quality AU PIM Connector.
  - `dlsmanager` - The user who creates the batches with the Oracle Product Data Quality AU PIM Connector.
1. Assign each of these new users the responsibility that uses the new roles (`dlsuser` and `dlsmanager` roles) that were just created.
  2. Be sure to assign the user to an organization (such as, Vision Operations) that will be used.
  3. Add Cross Reference Privileges to the organization by adding a new role called "Item Author" to the organization for all users.
  4. By default, the profile options for `DLS_PIMMGR_USER` and `DLS_EGO_IMPORT_BATCH_ASSIGNEE` are assigned to the user "operations". Change these profile options to use `dlsuser` and `dlsmanager` respectively.

## Updating the Profile Options

There are three profile options that need to be updated before running the PIM Connector as follows:

- DLS Default Import Source System
- DLS Import Batch Assignee
- DLS PIM Manager User

Go to **System Administrator/profile - System/** and search for the "DLS%" profile options.



These values can be edited:

Profile Option Name	Site
DLS Catalog Display Type	SemanticModel
DLS Default Import Source System	Product Information Managen
DLS Import Batch Assignee	DLSUSER
DLS Import Batch Size	10000
DLS PIM Manager Responsibility	Product Information Managen
DLS PIM Manager User	DLSMANAGER

### DLS Profile Option Information

Profile Option	Description
DLS Catalog Display Type	Matches the Display Type that you setup for the attributes at the catalog level. This is used to define the subset of item attributes that the DLS system will process.
DLS Default Import Source System	The name of the source system where the new batch is created and this is set to the "PIM Data Hub". For example, in the case of a production pull, where you are going to clean the production data, you pull all the information and put it in the I/F tables, and then create the batch against this source.
DLS Import Batch Assignee	Once the batch is created, you assign the batch to this user. If this is left blank, then the Production API will assign this to the PIM Manager User. This parameter is a fallback if the API is called without specifying the Import Batch Assignee.
DLS_Import Batch Size	This is the "chunk size" for creating batches where there are a large number of records. For example, if you are creating a batch for a large set of data such as PLM High Tech. These separate batches are tied together with the DLS Job Id. This parameter is a fallback if it is not passed with the DLS API call.
DLS PIM Manager Responsibility	The user and the responsibility together set the context for the execution of the API. This information is used when the DGS needs to do a production pull. This information is passed to the Oracle APIs to pull the attribute information.
DLS PIM Manager User	The user that has the privilege to create the batch. This parameter is a fallback if the API is called without specifying the PIM Manager User.



---

---

## Installing the Application Files

Oracle Product Data Quality AU PIM Connector includes a set of application components that the users will use to process data. These include a set of DSAs, Governance Studio Projects, and AutoBuild Import templates. Each of these is described in this chapter.

### Installation Prerequisite

The following two components need to be installed before installing the Oracle Product Data Quality template files:

- Oracle Product Data Quality must be installed and configured. For more information, see *Oracle Product Data Quality Oracle DataLens Server Installation Guide* and *Oracle Product Data Quality Oracle DataLens Server Administration Guide*.
- Oracle Product Data Quality Services for Excel Add-in must be installed on the client machine. For more information, see *Oracle Product Data Quality Services for Excel Reference Guide*.

### Installing the AU PIM Connector Application Template Files

Install the AU PIM Connector application template files:

1. Ensure that you are logged into the E-Business Server as `applmgr` and have sourced its environment file.
2. Change directories to the temporary directory that contains the `opdq_client_templates.zip` file.
3. Unzip the `opdq_client_templates.zip` file and extract the subdirectories.
4. Copy the `services_for_excel_templates` and `gov_studio_projects` directories onto the desktop of the client machine on which you will be running Governance Studio and Services for Excel.
5. Log into your Oracle DataLens Server.
6. Automatically deploy (autodeploy) the DSA and data lens template files by copying the `autodeploy` directory to the `local` directory on your Oracle DataLens Server as follows:

**On Linux:** `/opt/Oracle/Middleware/opdq/local`

**On Windows:** `C:\Oracle\Middleware\opdq\local`

The Oracle DataLens Server polls the `local` directory and autodeploys the DSA and data lens template files in the subdirectories to the server, which are then available for check out by client applications.

7. On your client machine, start the Knowledge Studio and check out the `DLS_Import_Template` data lens.

Before using the AutoBuild utility within Services for Excel, the `DLS_Import_Template` data lens *must* be checked out from the Oracle DataLens Server. For information about checking data lenses in and out of the Oracle Product Data Quality, see the *Oracle Product Data Quality Knowledge Studio Reference Guide*.

8. Start the Governance Studio and import each project file (`*.ams`) in the `desktop/opdq_clients_templates/gov_studio_projects` directory.

For more information, see the *Oracle Product Data Quality Governance Studio Reference Guide*.

## Oracle Product Data Quality DSA Overview

There are four main categories of DSAs that define the architecture of Oracle Product Data Quality.

- Metadata Import
  - Semantic Model build from PIM metadata extract.
  - Alternate Catalog build from PIM metadata extract.
- Creating and updated the semantic cache information that is used by the integration services to perform semantic matches.
- Creating a new batch from a query of Production data.
- Processing interface table batches thorough the Integration Services DSA and updating the results back in the interface tables.

## Default Entries in DSAs

All DSAs shipped as part of the Oracle Product Data Quality AU PIM Connector have default entries that allow for a quick configuration. The following default entries will need to be reviewed and configuration changes may be required. These changes are outlined in the *Oracle Product Data Quality PIM Connector Users Guide*.

### Default Database Connection

The default database connection name for all DSAs is, `PIM_Connector`. This database connection will be pointing to a fictitious server name with a fictitious user and password that will need to be configured as part of the initial server configuration.

### Default Data Lens

The default data lens for all DSAs that process items in Oracle Product Data Quality is `001_Capacitors_Demo_V2`. This data lens contains semantic models that work with the Capacitors Item Class category in the Vision Database installed with most AU PIM Connector systems. A process of testing the system using this sample lens and then updating this data lens with actual customer specific data lenses is documented in Oracle Product Data Quality.

## Creating and Updating Semantic Models from PIM Metadata Import

### **AUPIM\_CREATE\_SEMANTIC\_MODEL**

This DSA is used by AutoBuild to generate data lenses from the PIM Metadata. This DSA is typically run using the Oracle Product Data Quality Services for Excel to create the data needed by the AutoBuild process. This data includes attributes and sample descriptions.

## Importing Alternate Catalog Metadata

### **AUPIM\_IMPORT\_ALT\_CATALOGS**

This DSA is used by the AutoBuild to import one or more alternate catalogs and create classification schemas inside a selected data lens. This DSA is typically run using the Services for Excel to create the data needed by the AutoBuild process. This data includes the metadata for the Alternate Catalog entered during the run including the id and parent information.

## Creating and Updating the Semantic Cache

### **AUPIM\_CREATE\_SEMANTIC\_CACHE**

This DSA is run as a nightly job to update the Semantic Index from Oracle PIM Production data; this semantic index is used for matching and checking for duplicates.

## Creating Production Batches

Oracle PIM System users will be able to process data from the interface tables where the source system is an external source system or where the source system is the PIM Data Hub. In order to process data from external source systems, the PIM users will continue to use the existing methods of loading data into the interface tables, usually the Excel Import Templates or a direct database load. The PIM users will also have a new feature to load data directly from PIM Production tables into the interface tables using a batch creation process that is included in this application.

### **AUPIM\_CREATE\_EXTERNAL\_BATCH**

This DSA is run to insert your external data into interface tables by the Oracle DataLens Server.

### **AUPIM\_CREATE\_PRODUCTION\_BATCH**

This DSA is run to create a new batch of data from the PIM Production tables. This batch will then be processed by the Oracle DataLens Server and updated back into interface tables.

## Processing a Batch of Data Thorough the Cleansing and Matching Process

### **AUPIM\_MAIN\_PROCESS**

This DSA is run as the main DSA from the DGS. This will take an Oracle PIM batch and `clean/standardize/classify/extract attrs/etc` from the description and load it into the DGS for further processing. The batches are populated with data from external systems or as pulls from the PIM Production data.

### **Secondary Data Service Applications for Reprocess and Apply**

Secondary DSAs called by the main process as previously described that Reprocess or Apply the changes to the interface table.

DSA Name	Notes
AUPIM_REPROCESS	<p>Reprocesses rows from the following outputs of the Main Process DSA:</p> <ul style="list-style-type: none"> <li>▪ Dups within Batch</li> <li>▪ Match on Mfg Part</li> <li>▪ Match Against PIM</li> <li>▪ Items for Enrichment</li> <li>▪ Exceptions</li> </ul> <p>Based on specific process flags, this DSA will reroute the items back to the Governance Studio Main project, or will update the rows back in the interface tables with the correct match_status.</p>
AUPIM_APPLY_RESULTS	Applies rows from the Ready for Load step and updates all pertinent fields in the interface tables.
AUPIM_EXTRACT_ALT_CATALOGS	Applies rows from the Alternate Catalog Review and updates all pertinent fields related to Alternate Catalogs
AUPIM_SET_BATCH_TO_ACTIVE	Updated the batch status from Pending to Active so that the PIM user can go into the Import Workbench UI and invoke the import process.

## Pre-Configured Governance Studio Projects

Oracle Product Data Quality AU PIM Connector includes the following Governance Studio project. This project is used by users to create batch specific Governance Studio projects to process interface batch data. The process is documented in the *Oracle Product Data Quality PIM Connector User Guide*.

### **DataLens\_Cleansing\_and\_Matching**

The main process that runs the main Data Service Applications that cleanse and match the items in the interface tables.

### **Create\_PIMDH\_Production\_Batch**

The process that automatically creates a new Import Workbench batch from a set of production items.

### **Update\_Semantic\_Cache**

This administrative process can be run thorough the Governance Studio or can be scheduled to run as a nightly or weekly process. This should be run by System Administrators when not batches are being processed.

## Pre-Configured AutoBuild Wizard Excel Worksheets

The Oracle Product Data Quality AU PIM Connector includes the following AutoBuild Excel worksheets. These preconfigured worksheets will be used by the users to export PIM metadata and use it to create a set of semantic models with one or more associated alternate catalogs that will be used to cleanse and match customer specific product data.

### **CreateSemanticModels.xls**

This is the worksheet you use to export PIM metadata and run the AutoBuild wizard to create or refresh the semantic models for customer specific product data. The process will harvest the Item Class Category specific hierarchy, attributes, and valid values from the metadata in order to bootstrap the semantic model creation.

**ImportAlternateCatalogs.xls**

This is the worksheet you use to export PIM metadata related to the Alternate Catalogs in the PIM that user would like to associate with existing Semantic Models (data lenses). Once a data lens has been created using the Create Semantic Model AutoBuild process. You can then import one or more PIM Alternate Catalogs to allow for Auto Classification of items during the standardization and matching process.