

**Oracle® Application Integration Architecture**

Product Master Data Management Integration Pack  
Implementation Guide

Release 12.2

**E63489-01**

January 2017

Oracle Application Integration Architecture Product Master Data Management Integration Pack  
Implementation Guide, Release 12.2

E63489-01

Copyright © 2001, 2017, 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 is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

---

---

# Contents

<b>Preface</b> .....	ix
Audience .....	ix
Accessing Oracle Documentation .....	ix
Documentation Accessibility .....	ix
Document Revision History .....	ix

## Part I Understanding the Delivered Integrations

### 1 Understanding the Product Master Data Management Integration

About Oracle AIA and Oracle Communications Solutions Integrations.....	1-1
<b>Overview of the Product Master Data Management Integration</b> .....	1-1
Terminology.....	1-2
<b>Overview of the Business Process Flow</b> .....	1-3
<b>Core Oracle AIA Components</b> .....	1-5
<b>Integration Services</b> .....	1-6
ItemEBSV2.....	1-7
ItemResponseEBSV2.....	1-7
BillOfMaterialsEBS.....	1-7
BillOfMaterialsReponseEBS.....	1-7
PriceListEBSV2 .....	1-8
PriceListResponseEBSV2.....	1-8
SpecificationValueSetEBS .....	1-8
SpecificationValueSetResponseEBS.....	1-8
ClassificationSchemeEBS .....	1-8
ClassificationSchemeResponseEBS .....	1-9
CommunicationsPriceListEBSV2.....	1-9
CommunicationsPriceListResponseEBSV2 .....	1-9
CommunicationsItemCompositionEBSV1 .....	1-9
CommunicationsItemCompositionResponseEBSV1 .....	1-9

### 2 Product Master Data Management Integration Base Pack

About the Integration Base Pack .....	2-1
<b>About the Process Flow for Synchronizing of Items and BOMs</b> .....	2-1
About Chunking.....	2-3
Synchronizing Items Sub-Flow .....	2-3

Synchronizing Price Lists Sub-Flow .....	2-7
Synchronizing BOMs Sub-Flow .....	2-10
<b>Common Features Supported by the Base Pack</b> .....	<b>2-13</b>
Specifying Target and Destination Systems .....	2-13
Multi-Language Support for Item Synchronization .....	2-14
Communications Seeded Library Attributes .....	2-15
Updating Publishing Status in Oracle Product Hub.....	2-19
<b>Oracle Product Hub Interfaces</b> .....	<b>2-20</b>
<b>Oracle Product Hub Integration Services</b> .....	<b>2-20</b>
SyncItemListPIMEventConsumer .....	2-20
SyncItemListPIMReqABCImpl .....	2-21
QueryItemListPIMAdapter .....	2-21
QueryBillOfMaterialsListPIMAdapter .....	2-21
SyncSpecificationValueSetListPIMEventConsumer .....	2-21
SyncSpecificationValueSetListPIMReqABCImpl .....	2-21
SyncItemCatalogCategoryPIMEventConsumer .....	2-21
SyncItemCatalogCategoryPIMReqABCImpl .....	2-22
<b>Assumptions and Constraints for Oracle Product Hub</b> .....	<b>2-22</b>

### 3 Product Master Data Management Integration Option for Siebel CRM

<b>About the Siebel CRM Option</b> .....	<b>3-1</b>
<b>About the Process Flow for Synchronizing Item Catalog Categories</b> .....	<b>3-1</b>
About Multi-Language Support for Item Catalog Categories.....	3-2
About Automatically Releasing Entities.....	3-2
About Synchronizing of Relationships and Structures .....	3-2
About Synchronizing Attribute Groups .....	3-4
About Synchronizing Customer UDAs .....	3-5
About Synchronizing Transaction Attributes .....	3-6
About Synchronizing Transaction Attributes without Valuesets.....	3-7
About Synchronizing Item Catalog Category Hierarchies .....	3-7
About Associating Item Catalog Categories with Items .....	3-8
Synchronizing Item Catalog Category Implementation Flow.....	3-8
<b>About the Process Flow for Synchronizing Valuesets</b> .....	<b>3-13</b>
Synchronization of Valuesets as Part of Item Catalog Categories .....	3-13
Synchronization of Valuesets Independently .....	3-14
<b>About Other Supported Features</b> .....	<b>3-17</b>
Multi-Event Product Synchronization from Oracle Product Hub to Siebel CRM.....	3-18
Support for Class Type Relationship in Product Synchronization for Siebel CRM.....	3-18
Support for Controlling Auto-Release of Entities Published in the Same Batch .....	3-20
Synchronization of Promotions from Oracle Product Hub to Siebel CRM .....	3-21
<b>Siebel CRM Interfaces</b> .....	<b>3-26</b>
<b>Siebel CRM Integration Services</b> .....	<b>3-27</b>
SyncProductSiebelProvABCImpl .....	3-27
SyncBillOfMaterialsListSiebelProvABCImpl.....	3-28
SyncClassificationSchemeListSiebelProvABCImpl .....	3-28
SyncSpecificationValueSetListSiebelProvABCImpl .....	3-28
ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl .....	3-28

SyncItemCompositionListSiebelCommsProvABCImpl.....	3-29
<b>Assumptions and Constraints for the Siebel CRM Option .....</b>	<b>3-29</b>
<b>Multiple Price List Methodology .....</b>	<b>3-31</b>
About Price Lists and Rate Plans .....	3-31
Creating or Updating Rate Plans in Oracle Product Hub/BRM.....	3-32
Associating Rate Plans in Oracle Product Hub/BRM with Siebel CRM Price Lists .....	3-32
Offering a Product in Multiple Currencies.....	3-33
Managing Pricing in Rate Plans and Price Lists .....	3-34
Changing a Product from Multiple Price Lists to a Single Price List .....	3-34
Support for Effectivity.....	3-36
<b>Synchronizing Sponsorship Items from Oracle Product Hub to Siebel CRM .....</b>	<b>3-36</b>
Sharing Group Components.....	3-36
Sponsorship .....	3-37
Discounts.....	3-37
Membership.....	3-37
Promotion Groups .....	3-37
Defining a Charge Sharing Group and Publishing to Siebel CRM.....	3-38
Defining a Charge Discount Group and Publishing to Siebel CRM.....	3-39
Synchronizing Multiple Product Lines from Oracle Product Hub to Siebel CRM .....	3-40
Synchronizing Promotion Enhancements from Oracle Product Hub to Siebel CRM.....	3-41
Integration Services.....	3-41
Cross References.....	3-42
Assumptions and Constraints.....	3-42

#### **4 Product Master Data Management Integration Option for BRM**

<b>Synchronization of Billing Products from Oracle Product Hub to BRM.....</b>	<b>4-1</b>
Multi-Event Product Synchronization from Oracle Product Hub to Siebel CRM.....	4-2
Synchronizing Discounts and Discount Models to Siebel CRM and BRM .....	4-6
Viewing Publishing Status of Items .....	4-8
<b>BRM Interfaces .....</b>	<b>4-8</b>
<b>BRM Integration Services.....</b>	<b>4-8</b>
SyncItemListBRMProvABCImpl .....	4-9
SyncPriceListListBRMProvABCImpl.....	4-9
<b>Assumptions and Constraints for Oracle Communications BRM Option .....</b>	<b>4-9</b>

#### **5 Product Master Data Management Integration Option for Design Studio**

<b>Overview.....</b>	<b>5-1</b>
<b>Assumptions and Constraints .....</b>	<b>5-2</b>
<b>Oracle Communications Design Studio Setup.....</b>	<b>5-2</b>
<b>Oracle Product Hub Web Services .....</b>	<b>5-2</b>
<b>Integration Services .....</b>	<b>5-2</b>
QueryClassificationListPIMCommsProvABCImpl .....	5-3
QuerySpecificationValueSetListPIMCommsProvABCImpl.....	5-3
QueryProdClassAndAttributesSCEReqCommsABCImpl.....	5-3
<b>Understanding the Query Item Catalog Categories Business Flow .....</b>	<b>5-3</b>
<b>Query Item Catalog Categories: Implementation .....</b>	<b>5-5</b>

Query Item Catalog Categories and Transaction Attributes .....	5-7
<b>Query Item Catalog Categories: Integration Flow .....</b>	<b>5-7</b>
<b>Support for Effectivity during Design-Time - Methodology .....</b>	<b>5-9</b>
Deciding on Effectivity and Cartridge Deployment .....	5-10

## 6 Oracle Product Master Data Management Integration Option for PDC

<b>Overview .....</b>	<b>6-1</b>
<b>Integration Process Flow .....</b>	<b>6-1</b>
<b>Synchronization of Oracle Product Hub and Pricing Design .....</b>	<b>6-2</b>
<b>Integration Services .....</b>	<b>6-4</b>
SyncItemListPDCProvABCImpl .....	6-4
SyncPriceListPDCProvABCImpl .....	6-4
CreateProductXrefBRMAQ .....	6-4
CreateSponsorXrefBRMAQ .....	6-4
CreateDiscountXrefBRMAQ .....	6-4
<b>Terminology Mapping Between Pricing Design Center and BRM .....</b>	<b>6-5</b>
<b>Pre-requisites for Multi-BRM Setup .....</b>	<b>6-5</b>
<b>Assumptions and Constraints .....</b>	<b>6-6</b>

## Part II Implementing the Delivered Process Integrations

### 7 Configuring the Oracle Product Hub Integration

<b>Adding a New Spoke Instance .....</b>	<b>7-1</b>
<b>Post Installation Configuration Set Up .....</b>	<b>7-4</b>
Extract Inventory Location and Operating Unit IDs in Oracle Product Hub .....	7-4
Identify Item Validation Organization for Oracle Product Hub .....	7-5
Mapping Siebel CRM .....	7-5
Extract Business Unit and Inventory Location IDs in Siebel CRM .....	7-6
<b>Setting Up BRM .....</b>	<b>7-6</b>
<b>Setting Up Cross-References .....</b>	<b>7-6</b>
Populating Seed Values in Cross-Reference Tables .....	7-7
Validating Cross-References .....	7-9
Creating a Pricelist in Siebel CRM .....	7-9
Configuring the Integration to Support 100 Characters .....	7-10
Assumptions .....	7-10
Setup in Oracle AIA .....	7-10
<b>Working with Cross-References .....</b>	<b>7-10</b>
<b>Working with Domain Value Maps .....</b>	<b>7-11</b>
<b>Handling Errors .....</b>	<b>7-16</b>
<b>Setting Up Configuration Properties .....</b>	<b>7-16</b>
<b>Setting Up the Data for Seeded Attributes of the Telecommunications Library .....</b>	<b>7-26</b>
<b>Setting up List of Values in Siebel CRM .....</b>	<b>7-27</b>

## **A Frequently Asked Questions**

## **B Support for Rate Tiers and Effectivity**

## **C User Defined Attributes Framework**

<b>Extending for Custom UDAs</b> .....	C-1
Handling UDAs.....	C-1
Mapping Artifacts .....	C-2
Mapping Operational Attributes to ItemEBM.....	C-2
Mapping Seeded Set of UDAs for Communications library to EBM.....	C-2
Mapping Pricelist related UDAs for Communications Library to EBM.....	C-2
<b>Implementation of UDA Transformation</b> .....	C-3
Mapping Custom UDAs to EBM .....	C-5
Artifacts (Requester ABCS).....	C-5
Artifacts (Siebel CRM Provider ABCS):.....	C-5
<b>Performance Optimization Using Chunking</b> .....	C-6
Specifying the Batch Size .....	C-6

## **D Scenarios**

<b>ASCII Code of the Item Version is Higher in Oracle Product Hub</b> .....	D-1
Workaround.....	D-2
<b>ASCII Code of the Default Item Version is Higher in Oracle E-Business</b> .....	D-3





---

---

# Preface

This document describes how to implement and use the Oracle Application Integration Architecture (Oracle AIA) Product Master Data Management integration pack.

## Audience

This document is intended for customer service representatives, billing and pricing administrators, and other individuals who are responsible for configuring, managing and maintaining the Product Master Data Management integration pack.

## Accessing Oracle Documentation

Product documentation is located on Oracle Help Center:

<http://docs.oracle.com>

Additional documentation is available from the Oracle software delivery Web site:

<http://edelivery.oracle.com>

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Document Revision History

The following table lists the revision history for this guide:

Version	Date	Description
E63489-01	January 2017	Initial release.



# Part I

---

## Understanding the Delivered Integrations

Part I contains the following chapters:

- [Understanding the Product Master Data Management Integration](#)
- [Product Master Data Management Integration Base Pack](#)
- [Product Master Data Management Integration Option for Siebel CRM](#)
- [Product Master Data Management Integration Option for BRM](#)
- [Product Master Data Management Integration Option for Design Studio](#)
- [Oracle Product Master Data Management Integration Option for PDC](#)



---

---

# Understanding the Product Master Data Management Integration

This chapter provides an overview of the Oracle Application Integration Architecture (Oracle AIA) Product Master Data Management Integration (the integration).

## About Oracle AIA and Oracle Communications Solutions Integrations

Oracle AIA is an Oracle Communications Solutions Integrations framework that provides pre-built integrations using standard integration patterns, business processes, orchestration logic, and common objects and services to connect Oracle applications.

Oracle Communications Solutions Integrations is a set of integration frameworks, technologies, and tools that lets you design and build integrations that connect Oracle applications to support end-to-end business processes for communications service providers across operations support systems and business support systems.

## Overview of the Product Master Data Management Integration

The integration provides an industry-neutral and Oracle AIA-based enterprise data management solution that lets you centralize all product information from heterogeneous systems to create a single view.

In the integration, Oracle Product Hub acts as the product master. It provides a flexible user-defined attribute framework that lets you establish a single definition of an entity and synchronize it across multiple participating applications.

The integration supports concept-to-launch processes in which you define the metadata, items, and bills of materials (BOMs) in Oracle Product Hub and then published the information to the participating applications. The integration also supports component exclusions and overrides in the context of items, BOMs, and the Oracle Product Hub telecommunications library (a set of seeded attributes that support cross-industry applications). You use the telecommunications library to model communications-related entities such as billable products, discounts, pricing, and rules for as compatibility, upgrade and downgrade, promotions, promotion commitments and charges, and promotion based discounting.

You select participating applications when installing Oracle Product Hub and configure the integration to route the messages only to the applications affected by the data change.

The following applications and middleware components support the integration:

- Oracle Fusion Middleware
- Oracle Product Hub

- Siebel CRM
- Oracle E-Business Suite
- Oracle Communications Billing and Revenue Management (BRM)
- Oracle Communications Design Studio

For more information about compatible software and versions, search for Oracle Product Master Data Management on the **Certifications** tab of My Oracle Support.

## Terminology

This section contains the common terminology used in this guide and some key differences in the terminology across the participating applications.

[Table 1–1](#) lists common terminology used in this guide with a definition for each term.

**Table 1–1 Terminology Used in this Guide**

Term	Definition
Item catalog category	Item catalog categories provide a logical classification of items that have similar characteristics. Item catalog categories support hierarchies that provide a way to inherit attributes and relationships to subcategories and products belonging to those categories.
Attributes	Attributes are associated with the item catalog category and defined, along with their associated valuesets, at the same time as the item catalog category. Attributes are then inherited by the product. There are two kinds of attribute: <ul style="list-style-type: none"> <li>■ <b>Transaction attributes</b> capture characteristics and properties of a transaction involving a specific instance of a product. The value is captured at run time during a business transaction such as a sales order or a quote.</li> <li>■ <b>Static attributes</b> capture characteristics, properties, or features of a product. The value is captured at design time when the product is defined.</li> </ul>
Structure (BOM)	A structure or a BOM is a listing of all the subassemblies, intermediates, parts, and raw materials that make up a parent assembly, with the quantities of each subassembly required to make the assembly. A structure (BOM) is used to compose complex products out of simpler products.
Relationships	Relationships capture the relationship of one entity to another for business purposes, for example, substitute or up-sell items. You add items and BOMs as relationships to the item catalog category.
UDA	UDAs capture all the detailed information (such as cost information) about objects (items, change requests, change orders). The item catalog lets you create UDAs with validation logic by creating valuesets and associating the valuesets with UDA. UDAs can have a static or dynamic list of values or a range of values.
Valuesets	Valuesets define the possible set of values that an attribute can have. For the static attributes, you assign these values at design time. For transaction attributes, you assign these values at run time.

[Table 1–2](#) lists some key differences in the terminology used by each application while referring to the same entity.

**Table 1–2 Terminology Differences Across Applications**

Entity	Oracle Product Hub	Siebel CRM	E-Business Suite	BRM	Design Studio
Bundle	Item with BOM	Customizable product	Item with BOM	N/A	N/A
Promotion	Item	Promotion	Item	N/A	N/A
Inventory location or inventory organization	Inventory organization	Inventory location	Inventory organization	N/A	N/A

**Table 1–2 (Cont.) Terminology Differences Across Applications**

Entity	Oracle Product Hub	Siebel CRM	E-Business Suite	BRM	Design Studio
Product structure	BOM	Customizable product	BOM	N/A	N/A
Business unit or operating unit	Operating unit	Business unit	Operating unit	N/A	N/A
Product	Item	Product	Item	Product	N/A
Discount	Item	Product	N/A	Discount	N/A
Product class	Item catalog category	Product class	N/A	N/A	Product class
Attribute group	Attribute group	N/A	N/A	N/A	N/A
Static attribute	Static attribute (seeded)	Field or FCA of product	Operational attribute	Field or FCA of product	N/A
Transaction attribute	Transaction attribute	Product class attribute	N/A	N/A	Product class attribute
Valueset of static attribute	Valuesets of static attribute	List of values	N/A	N/A	N/A
Dynamic or versioned valueset	Dynamic or versioned value set	Attribute definition	N/A	N/A	Data Schema element
Customer added static UDA	Customer added static UDA	Flexible attribute	N/A	Customer added field	N/A
Structure element of item catalog category	Structure element of item catalog category	Relationship/structure of the product class	N/A	N/A	N/A

## Overview of the Business Process Flow

The integration maintains data between the applications throughout the business process flow of the product lifecycle.

At the beginning of the lifecycle, the product administrator defines a common product data model using item catalog categories in Oracle Product Hub with a comprehensive definition of the products. After defining item catalog categories, the product administrator creates or updates products.

Creating or updating a product includes the following:

- Defining Item catalog categories and their hierarchy
- Defining attributes and their valuesets and associating them with item catalog categories. You can publish valuesets independently or at the same time as item catalog categories.
- Defining basic item information such as name, product line, and description.  
The information is captured in primary and user-defined attributes in Oracle Product Hub. Oracle Product Hub also supports seeded operational attributes that control the behavior of the item in various functional E-Business Suite modules, including order management and procurement.
- Defining item BOM information, including the hierarchical structure of the product and its components

This structure also includes the component-level attributes and attributes that define the relationship between parent and child items in a hierarchical

relationship. You can synchronize BOMs with item definitions separately from BOMs without item definitions.

After creating or updating the common set of data, the product administrator publishes the information to the participating applications through the Oracle Product Hub publication framework. The publication framework provides a user interface to publish item catalog categories to registered participating applications. After publishing the information to participating applications, the integration sends the publishing status back to Oracle Product Hub, which maintains the status for each participating application at each entity level.

---

---

**Note:** When the integration publishes product classes to Design Studio, an email notification is sent to the product-to-service mapping specialist and the status is set to 'success'. The integration does not return the status back to Oracle Product Hub because the product classes are imported using a Query Product Class service from Design Studio.

---

---

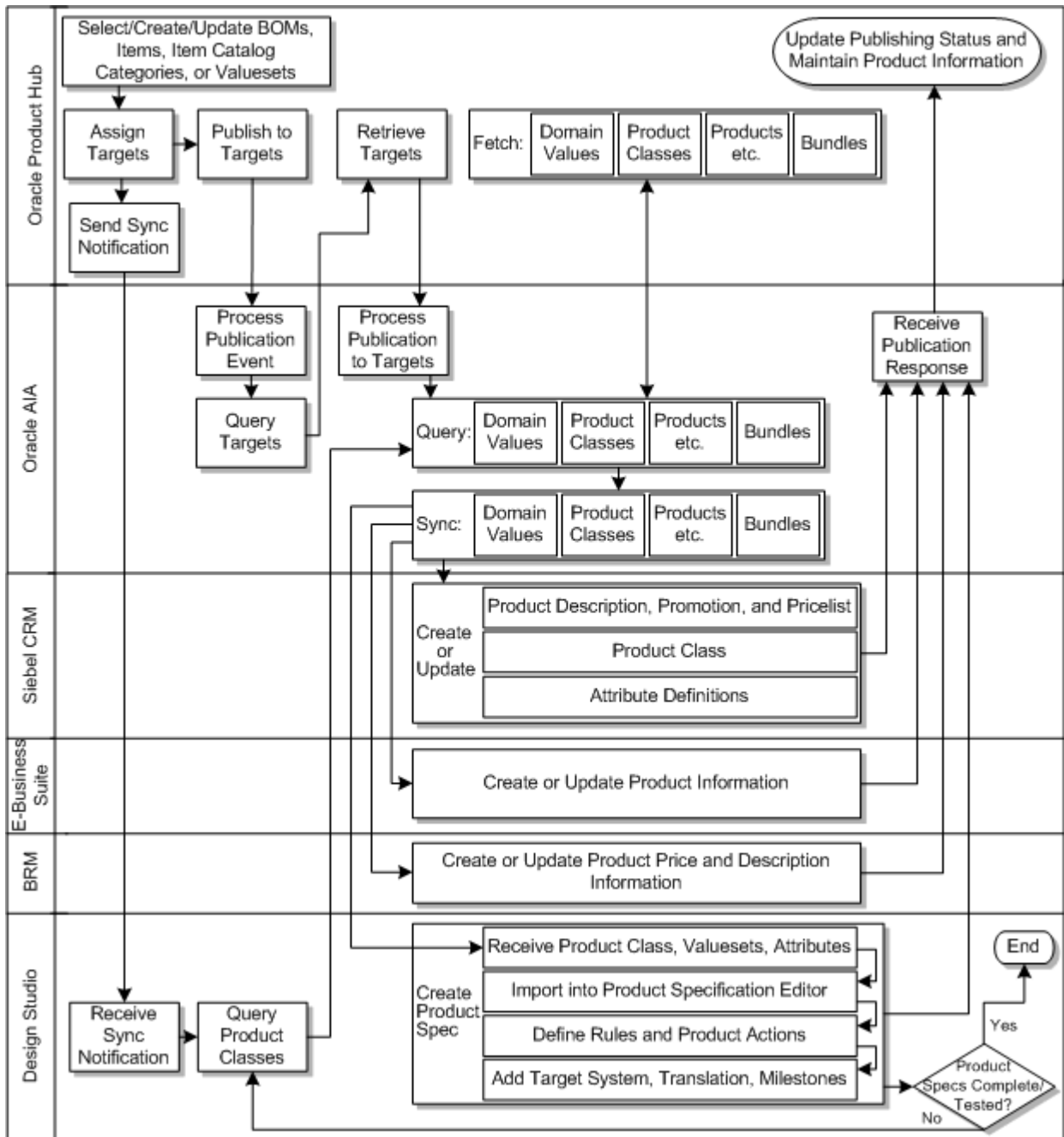
Figure 1–1 illustrates the following business process flow:

1. An Oracle Product Hub user selects, creates, or updates BOMs, Items, Item Catalog Categories, or valuesets.
2. Oracle Product Hub assigns the target systems, publishes the information to the target systems, and alerts Design Studio that the item catalog categories are ready to synchronize.
3. Design Studio receives the notification and a user queries the product classes from Oracle Product Hub.
4. Oracle AIA processes the publication event and queries the target systems from Oracle Product Hub.
5. Oracle Product Hub provides the target systems to Oracle AIA.
6. Oracle AIA processes the publication message and queries domain values, product classes, products, discounts, promotions, resources, bundles, and structures.
7. Oracle Product Hub retrieves the requested information and sends it to Oracle AIA.
8. Oracle AIA synchronizes the relevant information as follows:
  - Siebel CRM: Oracle AIA creates or updates attribute definitions from domain values, product classes from product classes, and product description, promotion, and pricelist information from products, discounts, promotions, resources, prices, bundles, and structures.
  - E-Business Suite: Oracle AIA creates or updates product information from products, discounts, promotions, resources, bundles, and structures.
  - BRM: Oracle AIA creates or updates product price and description information from products, discounts, promotions, resources, and prices.
  - Design Studio: Oracle AIA provides product classes, valuesets, and attributes. The Design Studio user uses these to create product specifications, rules, actions, translations, and milestones.
9. The participating applications send status updates to Oracle AIA.
10. Oracle AIA receives the publication response and sends it to Oracle Product Hub.



11. Oracle Product Hub updates the publishing status and maintains the product information for all participating applications.

Figure 1-1 Business Process Flow



## Core Oracle AIA Components

The integration uses the following core Oracle AIA components:

- ItemEBO
- SyncItemListEBM
- SyncItemListResponseEBM

- BillOfMaterialsEBO
- SyncBillOfMaterialsListEBM
- SyncBillOfMaterialsListResponseEBM
- PriceListEBO
- SyncPriceListEBM
- SyncPriceListResponseEBM
- SpecificationValueSetEBO
- SyncSpecificationValueSetListEBM
- SyncSpecificationValueSetListResponseEBM
- ClassificationSchemeEBO
- SyncClassificationSchemeListEBM
- SyncClassificationSchemeListResponseEBM
- Communications/PriceListEBO
- Communications/ItemCompositionEBO
- SyncItemCompositionListEBM
- SyncItemCompositionListResponseEBM
- QueryClassificationListPIMCommsProvABCImpl
- QuerySpecificationValueSetListPIMCommsProvABCImpl

The following directories contain the core component files:

- Enterprise business object (EBO) and enterprise business message (EBM) XML schema files:  
*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/*
- Enterprise business service WSDL files:  
*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/*

You can extend EBOs to add new data elements. These extensions are protected and will remain intact after a patch or an upgrade.

For more information, see the discussion of understanding extensibility in *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*.

## Integration Services

The integration includes the following integration services:

- [ItemEBSV2](#)
- [ItemResponseEBSV2](#)
- [BillOfMaterialsEBS](#)
- [BillOfMaterialsReponseEBS](#)
- [PriceListEBSV2](#)

- [PriceListResponseEBSV2](#)
- [SpecificationValueSetEBS](#)
- [SpecificationValueSetResponseEBS](#)
- [ClassificationSchemeEBS](#)
- [ClassificationSchemeResponseEBS](#)
- [CommunicationsPriceListEBSV2](#)
- [CommunicationsPriceListResponseEBSV2](#)
- [CommunicationsItemCompositionEBSV1](#)
- [CommunicationsItemCompositionResponseEBSV1](#)

## ItemEBSV2

ItemEBSV2 is a mediator routing service. It is used to route the SyncItemEBM received from Product Hub to the participating applications. This service exposes all of the enterprise operations that can be performed with an item EBO. In this integration, it implements the SyncItemList operation.

The SyncItemList operation invokes the SyncItemListEbizProvABCImpl and SyncProductSiebelProvABCImpl services.

## ItemResponseEBSV2

ItemResponseEBSV2 is a routing service. It exposes all of the enterprise operations that can be performed with an item EBO. In this integration, it implements the SyncItemListResponse operation.

The SyncItemListResponse operation sends the response message from the SyncProductListSiebelProvABCImpl and SyncItemListEbizProvABCImpl services to the SyncItemListPIMReqABCImpl service.

## BillOfMaterialsEBS

BillOfMaterialsEBS is a mediator routing service. It exposes all of the enterprise operations that can be performed with a BillOfMaterials EBO. In this integration, it implements the SyncBillOfMaterialsList operation.

The SyncItemListPIMReqABCImpl service invokes this operation. BillOfMaterialsEBS calls the SyncBillOfMaterialsListSiebelProvABCImpl and SyncBillOfMaterialsListEbizProvABCImpl services, along with other participating applications provider services.

This service synchronizes the product structure into Siebel CRM and E-Business Suite. It assumes that the components that are part of the product structure have already been synchronized to Siebel CRM and E-Business Suite as part of the item synchronization flow.

## BillOfMaterialsResponseEBS

BillOfMaterialsResponseEBS is a mediator routing service. It exposes all of the enterprise operations that can be performed with a BillOfMaterials EBO. In this integration, it implements the SyncBillOfMaterialsListResponse operation.

The SyncBillOfMaterialsListResponse operation sends the response message from the SyncBillOfMaterialsListSiebelProvABCImpl and

SyncBillOfMaterialsListEbizProvABCServiceImpl services, along with other participating applications provider services, to the SyncItemListPIMReqABCServiceImpl service.

## PriceListEBSV2

PriceListEBSV2 is a mediator routing service. It exposes all of the enterprise operations that can be performed with a price list EBO. In this integration, it implements the SyncPriceListList operation.

The SyncItemListPIMReqABCServiceImpl service invokes this operation. PriceListEBSV2 has routing rules for the participating applications. For Siebel CRM, PriceListEBSV2 sends the message to the CommunicationsPricelistEBSV2 and SyncPriceListList services to synchronize billing price lists to Siebel CRM.

You must set the RECEIVE\_PRICELIST\_FLAG in the ENTITY\_TO\_TARGET\_APPLICATION domain value map (DVM) for all applications that receive price lists. See "[Working with Domain Value Maps](#)" for more information about this DVM.

## PriceListResponseEBSV2

PriceListResponseEBSV2 is a mediator routing service. It exposes all of the enterprise operations that can be performed with a price list EBO. In this integration, it implements the SyncPriceListListResponse operation.

The SyncPriceListListResponse operation sends the response message from the CommunicationsPricelistEBSV2 service to the SyncItemListPIMReqABCServiceImpl service.

## SpecificationValueSetEBS

SpecificationValueSetEBS is a lightweight mediator routing service. It exposes all of the enterprise operations that can be performed with a Specification ValueSet EBO. In this integration, it implements the SyncSpecificationValueSetList operation.

The SyncItemCatalogCategoryPIMReqABCServiceImpl and SyncSpecificationValueSetListPIMReqABCServiceImpl services invoke this operation to synchronize valuesets. The routing rule in the operation sends the valueset message to the SyncSpecificationValueSetListSiebelProvABCServiceImpl service for Siebel CRM.

## SpecificationValueSetResponseEBS

SpecificationValueSetResponseEBS is a lightweight Mediator routing service. It exposes all of the enterprise operations that can be performed with a SpecificationValueSet EBO. In this integration, it implements the SyncSpecificationValueSetListResponse operation.

The SyncSpecificationValueSetListResponse operation sends the response message to the SyncSpecificationValueSetListPIMReqABCServiceImpl or SyncItemCatalogCategoryPIMReqABCServiceImpl service, depending on which invoked the message.

## ClassificationSchemeEBS

ClassificationSchemeEBS implements the SyncClassificationSchemeList operation.

The SyncItemCatalogCategoryPIMReqABCServiceImpl service invokes this operation to synchronize item catalog categories from Oracle Product Hub. The routing rule in the operation sends the valueset message to the SyncClassificationSchemeListSiebelProvABCServiceImpl service for Siebel CRM.

## ClassificationSchemeResponseEBS

ClassificationSchemeResponseEBS exposes all of the enterprise operations that can be performed with a ClassificationResponse EBO. In this integration, it implements the SyncClassificationListResponse operation.

The SyncClassificationListResponse operation sends the response message to the SyncItemCatalogCategoryPIMReqABCServiceImpl service.

## CommunicationsPriceListEBSV2

CommunicationsPriceListEBSV2 is a mediator routing service. It exposes all of the enterprise operations that can be performed with a communications price list EBO. In this integration, it implements the SyncPriceListList operation.

The PricelistEBSV2 service invokes this operation, which sends the request to the ProductOptimizedSyncPriceListListSiebelCommsProvABCServiceImpl service for price lists that need to be synchronized to Siebel CRM.

For more information about Communications-specific services or flows, see the *Oracle AIA Oracle Communications Order to Cash Integration Pack Implementation Guide* and *Oracle AIA Agent Assisted Billing Care Integration Pack Implementation Guide*.

## CommunicationsPriceListResponseEBSV2

CommunicationsPriceListResponseEBSV2 is a mediator routing service. It exposes all of the enterprise operations that can be performed with a communications price list EBO. In this integration, it implements the SyncPriceListListResponse operation, which sends the response to the PricelistEBSResponseEBSV2 service

For more information about Communications-specific services or flows, see the *Oracle AIA Oracle Communications Order to Cash Integration Pack Implementation Guide* and *Oracle AIA Agent Assisted Billing Care Integration Pack Implementation Guide*.

## CommunicationsItemCompositionEBSV1

CommunicationsItemCompositionEBSV1 performs item-related actions such as creating, updating, and synchronizing items. It invokes a provider service based on routing rules. In this integration, this service implements the SyncItemCompositionList operation.

The ProductOptimizedSyncPriceListListSiebelCommsProvABCServiceImpl invokes this operation, which sends the request to the SyncItemCompositionListSiebelCommsProvABCServiceImpl service for Siebel CRM.

For more information about Communications-specific services or flows, see the *Oracle AIA Oracle Communications Order to Cash Integration Pack Implementation Guide* and *Oracle AIA Agent Assisted Billing Care Integration Pack Implementation Guide*.

## CommunicationsItemCompositionResponseEBSV1

CommunicationsItemCompositionResponseEBSV1 is a mediator routing service. It exposes all of the enterprise operations that can be performed with a communications price list EBO. In this integration, it implements the SyncItemCompositionListResponse operation, which sends the response to the ProductOptimizedSyncPriceListListSiebelCommsProvABCServiceImpl service.

For more information about Communications-specific services or flows, see the *Oracle AIA Oracle Communications Order to Cash Integration Pack Implementation Guide* and *Oracle AIA Agent Assisted Billing Care Integration Pack Implementation Guide*.

---

---

# Product Master Data Management Integration Base Pack

This chapter describes the Oracle Application Integration Architecture (Oracle AIA) Product Master Data Management Integration base pack.

## About the Integration Base Pack

The integration Base Pack provides end-to-end process flows for synchronizing items, price lists, and bills of materials (BOMs).

It also supports flows for synchronizing metadata, including item catalog categories and versioned valuesets. These flows are described in the chapters for the integration options to which they pertain.

## About the Process Flow for Synchronizing of Items and BOMs

All of the integration options use this process flow.

This process flow lets you define products, discounts, and promotions as items in Oracle Product Hub and publish them to participating applications, including multiple Siebel CRM instances. You can organize simple products into structures and publish structures along with products and promotions.

The process flow reduces manual effort in keeping definitions of product, discounts, promotions, and other entities aligned across applications and ensures that a common set of products is accurately represented and available.

Oracle Product Hub messages can be generated from any number of organizations. The integration assumes that you have manually synchronized organizations between Oracle Product Hub, Siebel CRM, and E-Business Suite and created cross-references before attempting to use the process flows. The organization used for publication in Oracle Product Hub is submitted and mapped to the corresponding organization in the target application. For more information about organization cross references, see "[Setting Up Cross-References](#)".

The process flow synchronizes Oracle Product Hub items, pricing information, and BOMs as follows:

1. Oracle Product Hub publishes an item publication event containing the following information:
  - BATCH\_ID
  - STRUCTURE\_FLAG: Set to **N** if only publishing items. Set to **Y** if also publishing structures.

- Workspace Name
  - Auto-Release
2. Oracle AIA receives the item publication event and queries the list of target systems to which the batch must be published from Oracle Product Hub.
  3. Oracle AIA uses the batch ID in the item publication event to query the items to be published from Oracle Product Hub.
  4. Oracle Product Hub returns a payload containing the items published in that batch. This payload always contains all attributes and entities in the batch, whether the items are being created or updated.

Complete information is always published for each item. For each entity, the fields that are sent to the participating application with a value are updated with the provided value. Entities sent with NULL or empty values are nullified. Fields that are not sent are not updated.

Within the published payload, some of the items may not apply to some of the target systems. For example, promotions do not apply to BRM. The payload contains a multivalue attribute for each item that identifies the systems to which Oracle AIA will send the item. Oracle Product Hub implements this attribute as a multi-row user-defined attribute (UDA).

Oracle Product Hub uses the UDA Entity type to capture item type. Possible values are:

- Product
  - Discount
  - Discount Model (not published)
  - Promotion
  - Option Group
  - Bundle
5. Oracle AIA synchronizes the items to Siebel CRM, BRM, and E-Business Suite. If any item synchronization step fails, Oracle AIA sends an item status update to Oracle Product Hub and proceeds with the next item synchronization step.
  6. If there is pricing information in the item payload, Oracle AIA prepares a price list payload and synchronizes the pricing information to Siebel CRM and BRM.

If item synchronization failed for a given target system, the pricing information is not synchronized to that target system.

If any price list synchronization step fails, Oracle AIA sends an item status update to Oracle Product Hub and proceeds with the next price list synchronization step.

7. If the structure flag is set to Y, Oracle AIA uses the batch ID to query the BOMs to be published.
8. Oracle Product Hub returns a payload containing the BOMs published in that batch. This payload always contains all attributes and entities in the batch, whether the BOMs are being created or updated.

The payload contains the structures associated with the root items. Because the type of structure (for example, sales BOM, manufacturing BOM, marketing BOM, support BOM) is defined by the structure name header attribute, it is the same for all the BOMs in the payload.



Target systems are not associated directly with BOMs. All BOMs go to all providers that consume BOMs. An Oracle AIA domain value map (DVM) defines which providers consume BOMs.

9. Oracle AIA synchronizes the BOMs to the Siebel CRM and E-Business Suite.

If item synchronization failed for a given target system, the BOMs are not synchronized to that target system.

If any BOM synchronization step fails, Oracle AIA sends an item status update to Oracle Product Hub and proceeds with the next BOM synchronization step.

10. After completing synchronization, Oracle AIA sends an item status update to Oracle Product Hub.

## About Chunking

You can enable chunking to help improve performance of item synchronization. Chunking lets Oracle AIA split the total entities in a batch to multiple chunks or sub-batches based on the BATCH\_SIZE configuration property. The integration processes each of these chunks or sub-batches as a separate batch.

You enable chunking at the following levels:

- BOM level: Extracts as many BOMs as defined in the chunk size. If a BOM has an option class, the option class BOMs are also extracted into the same sub-batch.
- Item level: Extracts as many items as defined in the chunk size. If a sub-batch fails, all the items continue to be processed in that batch, but the pricelist or BOM is not processed.

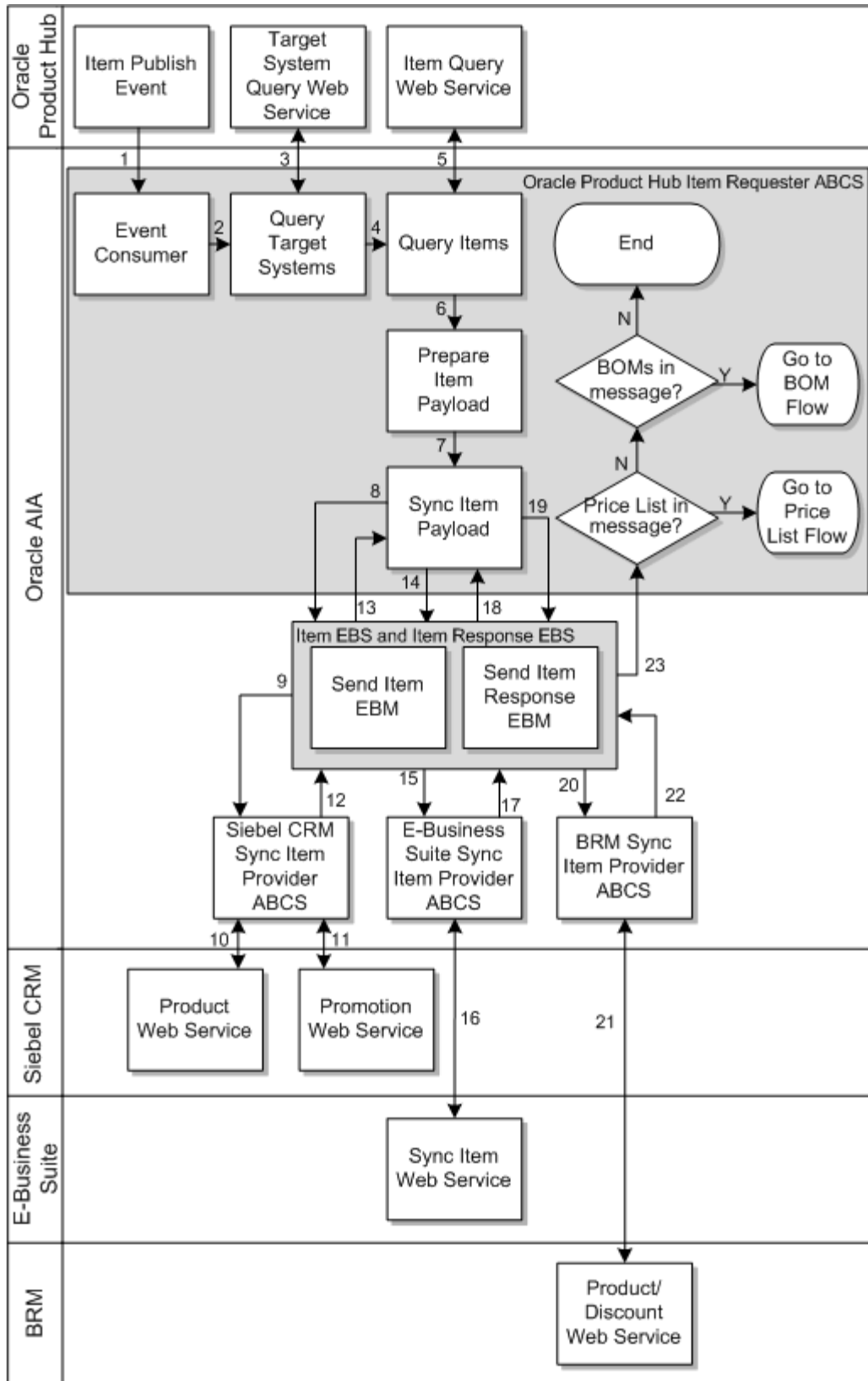
If batch synchronization fails, Oracle AIA sends a batch-level update to Oracle Product Hub.

For more information about setting batch size, see "[Setting Up Configuration Properties](#)". For more information on optimizing performance by enabling item chunking or BOM chunking, see "[Performance Optimization Using Chunking](#)".

## Synchronizing Items Sub-Flow

[Figure 2-1](#) illustrates the process flow for synchronizing items from Oracle Product Hub to Siebel CRM, BRM, and E-Business Suite.

Figure 2-1 Synchronizing Items

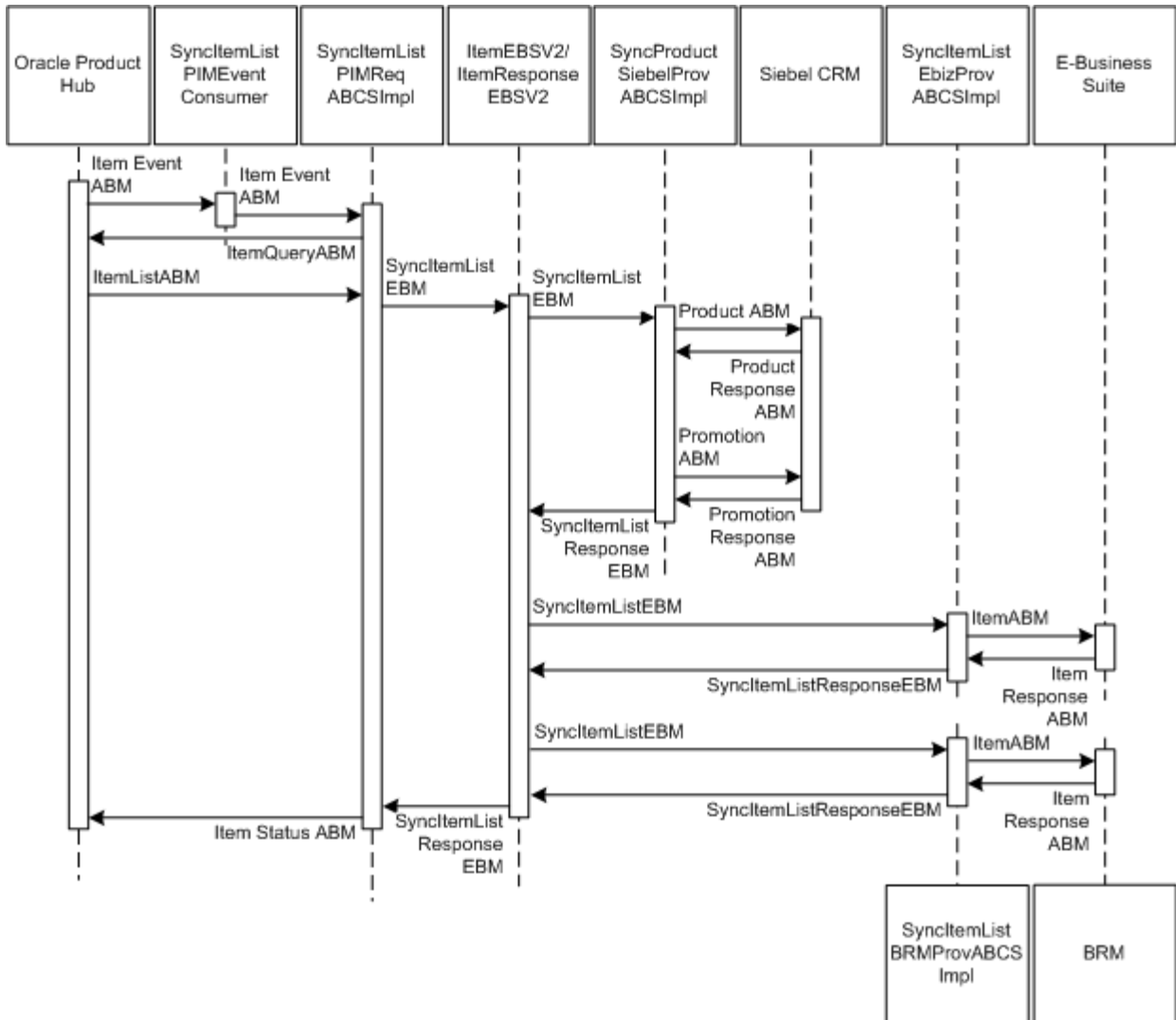


The integration synchronizes items as follows:

1. An Oracle Product Hub administrator publishes items and Oracle Product Hub raises a publish event containing a batch ID.
2. The SyncItemListPIMEventConsumer service receives the item event from Oracle Product Hub and sends it to the SyncItemListPIMReqABCServiceImpl service.
3. SyncItemListPIMReqABCServiceImpl receives the item event message and does the following:
  - a. Uses the batch ID to query the list of target systems from Oracle Product Hub.  
Oracle Product Hub specifies target systems at the batch level, returning all the target systems to which the batch should be published. If the target system is not listed in the query response, Oracle AIA does not synchronize items to that system, even if the system name is defined at the entity level.
  - b. Transforms the item event into the ItemQueryABM application business message (ABM) and uses it to query the list of items in the batch from Oracle Product Hub by using the QueryItemListPIMAdapter.  
  
If an error occurs in the item adapter while the adapter is getting data from Oracle Product Hub, the synchronization process is stopped for the entire batch.
  - c. Transforms the ItemListABM message received from Oracle Product Hub into the SyncItemListEBM message and the SyncPriceListEBM message.  
  
If there is no price list or BOM on the EBM, SyncItemListPIMReqABCServiceImpl sets the workspace auto-release flag on the SyncItemEBM, which allows the downstream services to release the application workspaces after the synchronizing the items.
  - d. Sends SyncItemListEBM to the ItemEBSV2 service in a separate call for each target system.
  - e. Sends SyncPriceListEBM to the PriceListEBSV2 service, which starts the sub-flow for synchronizing price lists.
4. ItemEBSV2 filters the items in SyncItemListEBM based on the target system and sends the message to the following services sequentially:
  - The SyncProductSiebelProvABCServiceImpl service, which does the following:
    - a. Transforms SyncItemListEBM it into a Siebel CRM product ABM containing the items with the product or discount entity type and a Siebel CRM promotion ABM containing the items with the promotion entity type.
    - b. Sends the product ABM to the SWIPProductImport web service and the promotion ABM to the SWIPromotionImport web service.
    - c. Receives responses from Siebel CRM and updates the ITEM\_ITEMID and PROMOTION\_PROMOTIONID cross reference tables with the Siebel CRM values for product and promotion ID.
    - d. Transforms the responses into a SyncItemListResponseEBM message and sends it to the ItemResponseEBSV2 service.
  - The SyncItemListEbizProvABCServiceImpl service, which does the following:
    - a. Transforms SyncItemListEBM into an E-Business Suite item ABM.
    - b. Sends the ABM to the item synchronization API.



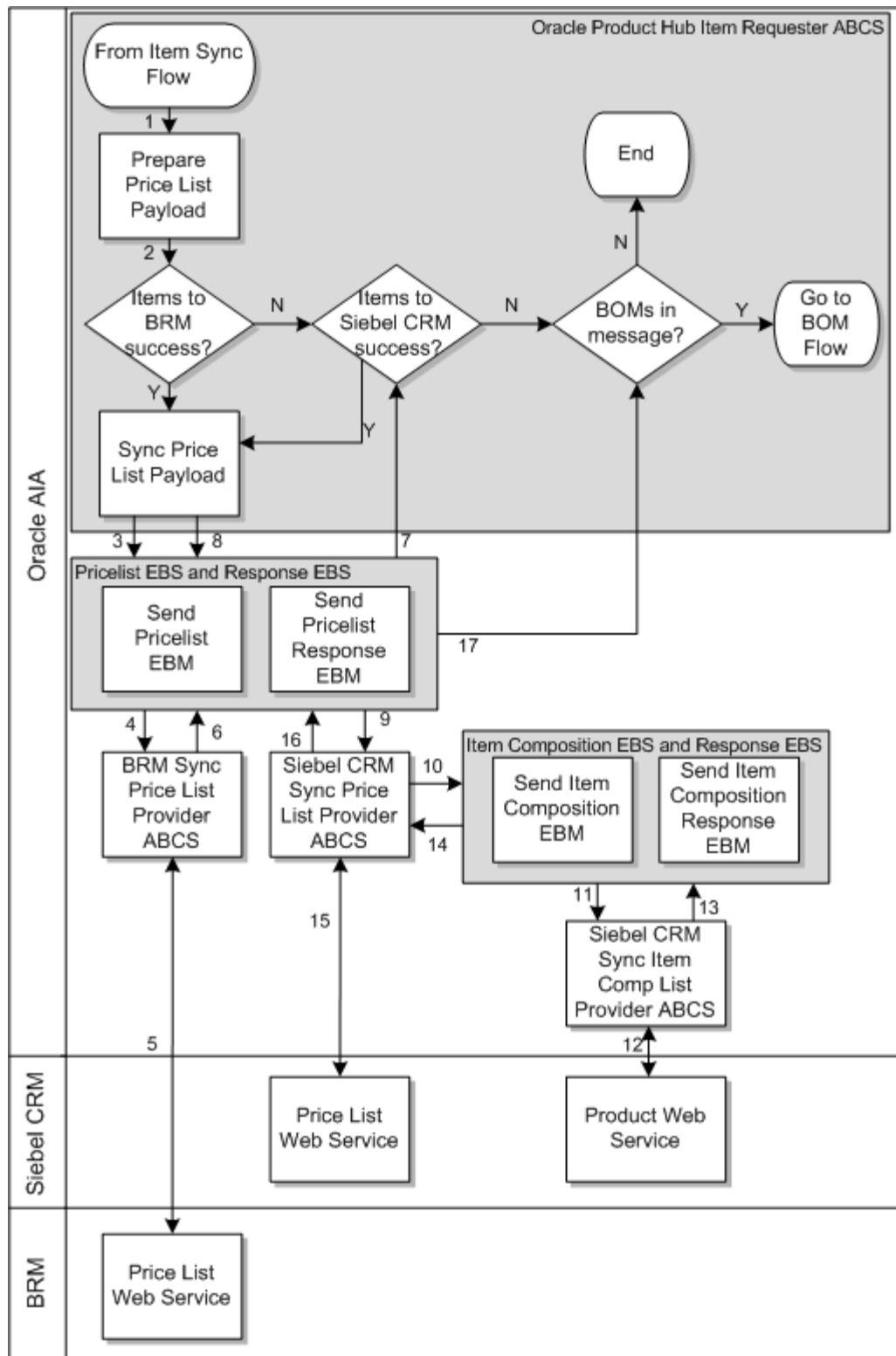
**Figure 2-2 Item Synchronization Message Sequence**



### Synchronizing Price Lists Sub-Flow

After synchronizing items from Product Hub to the participating applications, Oracle AIA synchronizes any price lists that appear in the item payload as illustrated in [Figure 2-3](#).

Figure 2-3 Synchronizing Price Lists



The integration synchronizes price lists from Oracle Product Hub as follows:

1. After the item synchronization flow, SyncItemListPIMReqABCImpl transforms ItemListABM into a SyncPriceListEBM message that includes pricing information, and sends it to the PriceListEBSV2 service, targeting BRM.

If item synchronization failed for any participating application, the price list is not synchronized to that application.

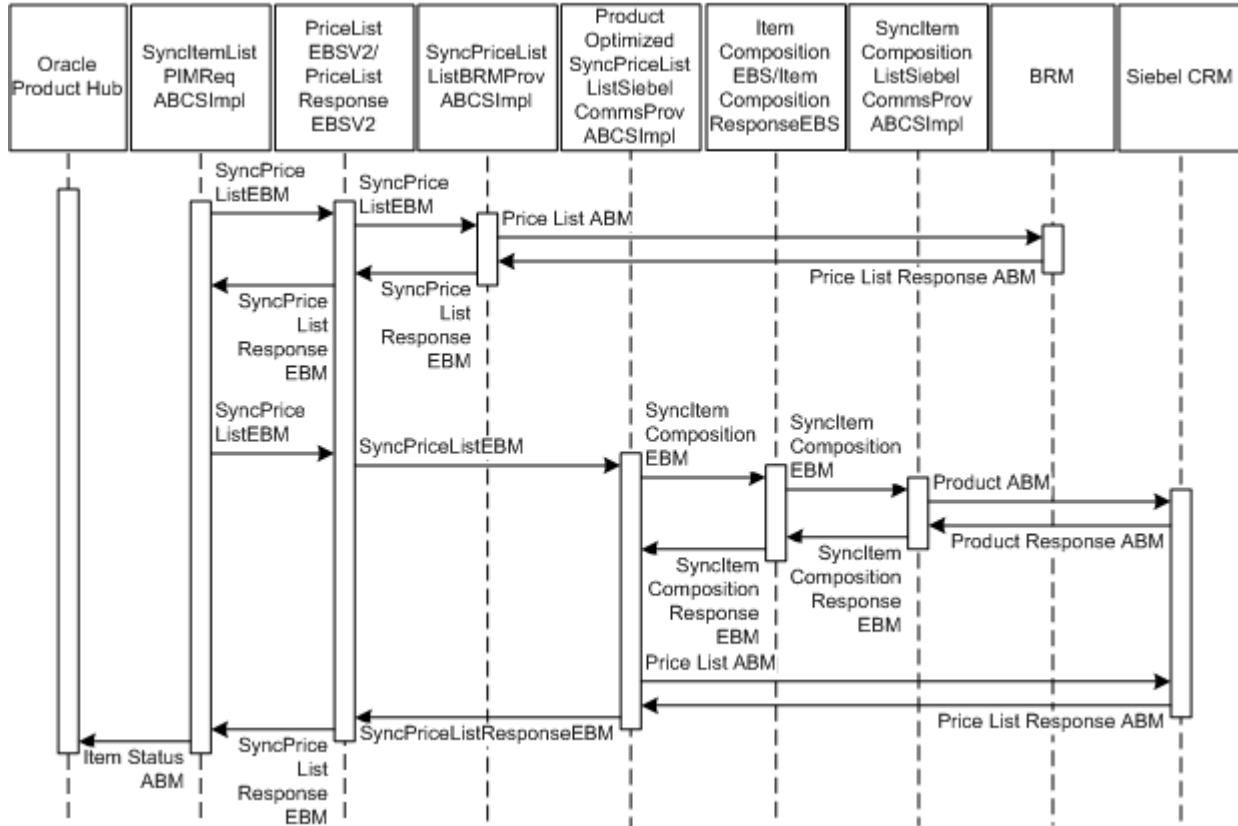
If there is no BOM on the EBM, SyncItemListPIMReqABCImpl sets the workspace auto-release flag on the SyncPriceListEBM, which allows the downstream services to release the application workspaces after the synchronizing the price lists.

2. PriceListEBSV2 sends SyncPriceListEBM to the SyncPriceListListBRMProvABCImpl service.
3. SyncPriceListListBRMProvABCImpl transforms SyncPriceListEBM into an ABM and sends it to the BRM opcode that associates the pricing information with products in BRM.
4. SyncPriceListListBRMProvABCImpl receives the response from BRM, transforms it into a SyncPriceListResponseEBM message, and sends it to the SyncPriceListResponseEBSV2 service.
5. SyncPriceListResponseEBSV2 sends SyncPriceListResponseEBM to SyncItemListPIMReqABCImpl.  
If there is a failure, SyncItemListPIMReqABCImpl sends an updated item status to Oracle Product Hub and continues with the synchronization process.
6. SyncItemListPIMReqABCImpl sends SyncPriceListEBM to PriceListEBSV2, targeting Siebel CRM
7. PriceListEBSV2 sends SyncPriceListEBM to the ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl service.
8. ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl transforms SyncPriceListEBM into a SyncItemCompositionEBM message and sends it to the ItemCompositionEBSV2 service.
9. ItemCompositionEBSV2 sends the EBM to the SyncItemCompositionListSiebelCommsProvABCImpl service.
10. SyncItemCompositionListSiebelCommsProvABCImpl transforms the EBM into a product ABM and sends it to the Siebel CRM product web service. The web service creates event products in Siebel CRM for products with multiple charges.
11. SyncItemCompositionListSiebelCommsProvABCImpl receives the response from Siebel CRM, transforms it into a SyncItemCompositionResponseEBM message, and sends it to the ItemCompositionResponseEBSV2 service.
12. ItemCompositionResponseEBSV2 sends SyncItemCompositionResponseEBM to ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl.
13. ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl the EBM into a price list ABM and sends it to the Siebel CRM price list web service, which synchronizes the pricing information on the product and the additional product created for billing events.
14. ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl receives the response from Siebel CRM, transforms it SyncPriceListResponseEBM, and sends it to SyncPriceListResponseEBSV2.
15. SyncPriceListResponseEBSV2 sends SyncPriceListResponseEBM to SyncItemListPIMReqABCImpl.

16. SyncItemListPIMReqABCImpl sends a publishing status update to Oracle Product Hub.
17. If the structure flag is set to Y in the event message from Oracle Product Hub, the process flow continues with the Synchronizing BOMs sub-flow. See ["Synchronizing BOMs Sub-Flow"](#). If not, the process flow is complete.

Figure 2-4 shows the sequence of integration services through which the price list message passes for the synchronization flow between Oracle Product Hub, Siebel CRM, and BRM.

**Figure 2-4 Price List Synchronization Message Sequence**

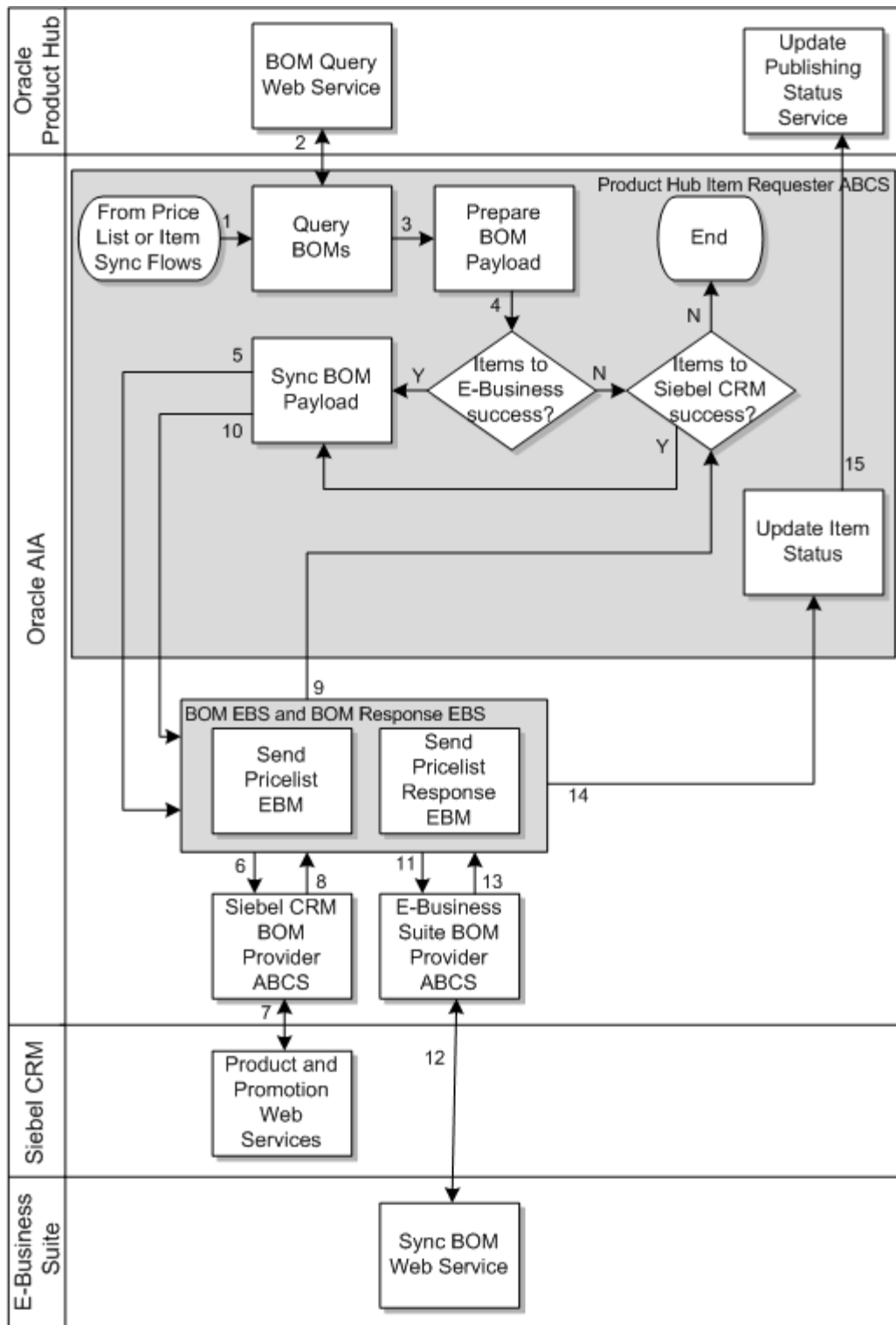


### Synchronizing BOMs Sub-Flow

After synchronizing items and price lists from Product Hub to the participating applications, Oracle AIA synchronizes any BOMs that appear in the item payload as illustrated in [Figure 2-5](#).



Figure 2-5 Synchronizing BOMs



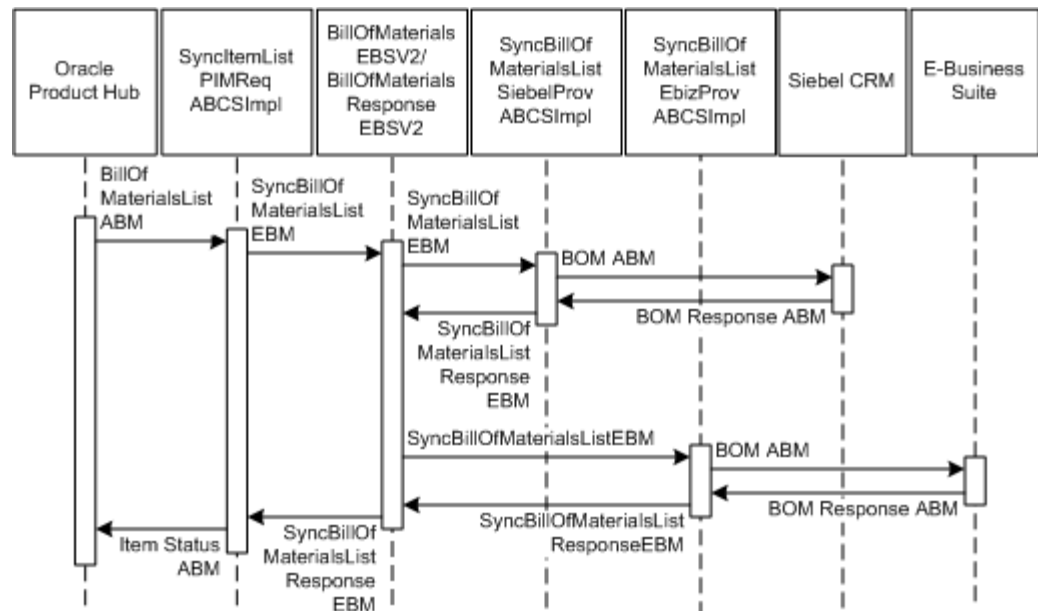
The integration synchronizes BOMs from Oracle Product Hub as follows:

1. After the item and price list synchronization flows, SyncItemListPIMReqABCImpl queries the BOMs associated with the root item in the batch from Oracle Product Hub by using the QueryBillOfMaterialsListPIMAdapter.

2. SyncItemListPIMReqABCImpl receives the BillOfMaterialsListABM from Oracle Product Hub, transforms it into the SyncBillOfMaterialsListEBM, and sends it to the BillOfMaterialsEBSV2 service.  
  
SyncItemListPIMReqABCImpl sets the workspace auto-release flag on the SyncBillOfMaterialsListEBM, which allows the downstream services to release the application workspaces after the synchronizing the BOMs.
3. BillOfMaterialsEBSV2 sends SyncBillOfMaterialsListEBM to the following services:
  - SyncBillOfMaterialsListSiebelProvABCImpl, which does the following:
    - a. Transforms SyncBillOfMaterialsListEBM into a BOM ABM.
    - b. Sends the BOM ABM to the Siebel CRM product and promotion web services, which creates the product and promotion structure in Siebel CRM.
    - c. Transforms the response from Siebel CRM into the SyncBillOfMaterialsListResponseEBM message.
    - d. Sends SyncBillOfMaterialsListResponseEBM to the BillOfMaterialsResponseEBSV2 service.
  - SyncBillOfMaterialsListEbizProvABCImpl, which does the following:
    - a. Transforms SyncBillOfMaterialsListEBM into a BOM ABM.
    - b. Sends the BOM ABM to the E-Business Suite API that synchronizes BOMs.
    - c. Transforms the response from E-Business Suite into SyncBillOfMaterialsListResponseEBM.
    - d. Sends SyncBillOfMaterialsListResponseEBM to BillOfMaterialsResponseEBSV2.
4. BillOfMaterialsResponseEBSV2 sends SyncBillOfMaterialsListResponseEBM to SyncItemListPIMReqABCImpl.
5. SyncItemListPIMReqABCImpl sends a publishing status update to Oracle Product Hub and the process flow is complete.

Figure 2–6 shows the sequence of integration services through which the BOM message passes for the synchronization flow between Oracle Product Hub, Siebel CRM, and E-Business Suite.

Figure 2–6 BOM Synchronization Message Sequence



## Common Features Supported by the Base Pack

The base pack supports the following features across all process flows:

- [Specifying Target and Destination Systems](#)
- [Multi-Language Support for Item Synchronization](#)
- [Communications Seeded Library Attributes](#)
- [Updating Publishing Status in Oracle Product Hub](#)

### Specifying Target and Destination Systems

You can specify the systems to which Oracle AIA publishes information at the batch level and the item level.

At the batch level, you specify **target systems**. Oracle AIA queries the list of target systems from Oracle Product Hub when you publish a batch of items.

At the item level, you specify **destination systems**. You use the `Destination_Sys_Specification` attribute group with the `Sync_Item` attribute to specify destination systems. This attribute group lets you synchronize items to only those systems to which they are relevant. When `Sync_Item` is set to **Y** for a system, Oracle AIA publishes the item to the corresponding system. When `Sync_Item` is set to **N**, Oracle AIA does not publish the item to the corresponding system.

If a system does not appear in the list of destination systems for an item, but the system does appear in the list of target systems for the batch, the item is published to that system. Because of this behavior, you do not need to explicitly specify each destination system to which each item should be synchronized, only those destination systems to which each item should NOT be synchronized.

For example, when a promotion has bundles and other component items, the promotion and bundles only to Siebel CRM, not to BRM, but the leaf level items apply to both.

Table 2–1 shows a promotion in a batch that Oracle Product Hub will publish with BRM and Siebel CRM in the list of target systems. The destination system attribute specifies only those systems to which items should NOT be published.

**Table 2–1 Promotion Example**

Item	Item Type	Destination System	Sync Item
Standard VoIP	Promotion	BRM	N
VoIP Bundle	Commercial Bundle	BRM	N
VoIP Service	Service Bundle	BRM	N
Basic VoIP Access	Product	N/A	N/A
VoIP Voicemail	Product	N/A	N/A
VoIP \$10 Monthly Discount	Product	N/A	N/A
VoIP Adaptor	Product	N/A	N/A

The first three items are synchronized to Siebel CRM because Siebel CRM appears in the target systems list but does not appear in the destination systems list for each item. They are not synchronized to BRM because although BRM appears in the target systems list, it also appears in the destination systems list for each item with Sync\_Item set to N.

The final four items are synchronized to both Siebel CRM and BRM because both appear in the target systems list but neither appear in the destination systems lists for each item.

## Multi-Language Support for Item Synchronization

Multi-language support lets Oracle Product Hub publish attribute values in multiple languages.

Multi-language support provides language codes and all the values in the corresponding languages for attributes that support language translation as follows:

- For free-text fields that are translatable in the target systems, such as Item Name and Description, Oracle Product Hub provides all the values in the supported languages and the corresponding language code for each value. The target systems only consume values for the language that they support; some target systems may not support all of the same languages as Oracle Product Hub.
- For free-text fields that are not translatable in the target systems, Oracle Product Hub provides all the values and corresponding language codes, but the target system consumes the values in a single configured default language.
- For fields with valuesets with translatable values, Oracle Product Hub provides a language-independent code and all the values in all the languages, but Oracle AIA passes only the language-independent code to the target system.
- For fields with valuesets with untranslatable values, Oracle Product Hub provides the code for the value that must be consumed by the target system.
- For customer user-defined attributes, Oracle AIA synchronizes the values with all values and corresponding language codes. The target applications consume these attributes based on the support offered for extended attributes.

## Communications Seeded Library Attributes

Oracle Product Hub includes seeded attribute groups for communications that have attributes that define the characteristics of an item or the characteristics of the entity that the item represents. The integration sets the values of these attributes in the target applications.

Oracle Product Hub provides the following seeded attribute libraries:

- Horizontal sellable product information library
- Horizontal product management library
- Vertical communications services billing library
- Vertical communications product details library

For more information about communications seeded library attributes and their corresponding valuesets, see the discussion of seeded item metadata libraries in the *Product Hub Implementation Guide*.

The horizontal sellable product information library contains the attribute groups listed in [Table 2-2](#).

**Table 2-2 Attribute Groups in the Horizontal Sellable Product Information Library**

Attribute Group	Definition
Product Details: Definitions	Defines the additional characteristics of an item that represents the product in Siebel CRM. Mapped to the first class attributes of the product in Siebel CRM.
Product Details: Marketing	Defines the marketing characteristics of an item that represents the product in Siebel CRM. Mapped to the first class attributes that represent the marketing details of the product in Siebel CRM.
Product Details: Service	Defines the service characteristics (customer service information) of an item. Mapped to the first class attributes that represent the customer service of the product in Siebel CRM.
Product Details: Logistics	Stores the logistics information of an item. Usually associated with the physical goods. Mapped to the first class attributes that represent the logistics information of the product in Siebel CRM.
Product Details: Additional Information	Defines any additional information associated with the item. Mapped to the first class attributes that represent the additional information of a product in Siebel CRM.
Pricing: Simple Price List	For the standard pricing model, which lets you define a simple pricing that consists of list price, price type, and discount for the products used in non-communications industries. This model is not used for product and services that are billed in BRM.  BRM uses the billing pricing model, which allows you to define pricing of products and services as supported in the BRM rate plan model. See <a href="#">Table 2-4</a> for details about attribute groups used in the billing pricing model.  You define a product's pricing model using the Pricing Code user-defined attribute.
Promotion: More Information	Defines additional information associated with items that represent promotions. Stored as a specification group in the ItemEBO of the canonical layer and mapped to the first class attribute of the promotion entity in Siebel CRM.
Charge Plan: Nonrecurring Charge Details	Defines a charge plan for nonrecurring charges associated with a promotion. Stored as specification groups in the ItemEBO of the canonical layer and mapped to the first class attribute of the promotion entity in Siebel CRM.

**Table 2–2 (Cont.) Attribute Groups in the Horizontal Sellable Product Information Library**

Attribute Group	Definition
Charge Plan: Recurring Charge Details	Defines a charge plan for recurring charges associated with a promotion. Stored as specification groups in the ItemEBO of the canonical layer and mapped to the first class attribute of the promotion entity in Siebel CRM.
Charge Plan: Charges, Adjustment, Usage Plan Details	Defines a charge plan for more advanced information associated with a promotion, such as adjustments and usage plan details. Stored as specification groups in the ItemEBO of the canonical layer and mapped to the first class attribute of the promotion entity in Siebel CRM.
Commitment: Charges Credits	Defines the commitment charges associated with a promotion. Stored as specification groups in the ItemEBO of the canonical layer and mapped to the first class attribute of the promotion entity in Siebel CRM.
Commitment: Terms	Defines the commitment terms associated with a promotion. Stored as specification groups in the ItemEBO of the canonical layer and mapped to the first class attribute of the promotion entity in Siebel CRM.
Product Promotions: Upgrade	Identifies all the promotions to which the current promotion can be upgraded. Stored as specification groups in the ItemEBO of the canonical layer and mapped to the first class attribute of the promotion entity in Siebel CRM.
Subject Compatibility Rules	Defines compatibility rules associated with the item. This is a multi-row attribute group in Oracle Product Hub where multiple rules can be defined. Stored as specification groups in the ItemEBO of the canonical layer and mapped to the first class attribute of the product entity in Siebel CRM.

The horizontal product management library contains the attribute group listed in [Table 2–3](#).

**Table 2–3 Attribute Groups in the Horizontal Product Management Library**

Attribute Group	Definition
Destination System Specification	Specifies the target system to which the item is published. The attributes in this attribute group are not mapped to any field in the target application, but are used in decision-making and routing.

The vertical communications services billing library contains the attribute group listed in [Table 2–4](#).

**Table 2–4 Attribute Groups in the Vertical Communications Services Billing Library**

Attribute Group	Definition
Billing Products Attributes General	Defines billing-related information of the item for communications. Primarily used in the integration to billing applications to create billing products. These have to be mapped to the first class attributes of the products in billing.
Billing Products Event Map	Associates charges (billable events) with billing products.
Rate Plan	For the billing pricing model. Represents rate plan attributes associated with billing products.
Tier Group	For the billing pricing model. Defines multi-tier rating for billing products.
Day Time Range	For the billing pricing model. Defines the times when the associated rate data is applied. This is a timezone-independent field in which the time range should not be translated across the systems in different time zones.

**Table 2–4 (Cont.) Attribute Groups in the Vertical Communications Services Billing Library**

Attribute Group	Definition
Days of the Week Range	For the billing pricing model. Defines the calendar dates when the associated rate data is applied. This is a timezone-independent field in which the time range should not be translated across the systems in different time zones.
Rate Data	For the billing pricing model. Defines the discount brackets, and proration details that apply to actual charges.
Balance Impacts	For the billing pricing model. Defines the actual amount or charges associated with billing products. Can be based on the type of resources, effective dates, and categories defined in the billing application.
Billing Discount Attributes	Defines the billing discount related information.
Billing Discount Event Map	Associates the actual discount model with the discount.

See the BRM documentation for more information about the billing pricing model.

The vertical communications product details library contains the attribute group listed in [Table 2–5](#).

**Table 2–5 Attribute Groups in the Vertical Communications Product Details Library**

Attribute Group	Definition
Communications: Product Info	Defines communications-specific and fulfillment-related information of the item. Mainly used during order fulfillment processing.

The horizontal sellable product information library includes the following component attribute groups, which the integration includes when synchronizing BOMs.

**Table 2–6 Component Attribute Groups**

Attribute Group	Definition
Product Promotions: Components	Associated with the BOM that represents a promotion. Defined for the immediate components of the promotion only, not the children of the components. Oracle AIA sets the corresponding values in the context of the parent-child relationship of the promotion in Siebel CRM.

**Table 2–6 (Cont.) Component Attribute Groups**

Attribute Group	Definition
Product Promotions: Pricing: Components: Adjustments	Represents the adjustments that are applied to the components of the promotion. Applied at the leaf-level component and valid only in the context of the promotion. Oracle AIA sets the values in the context of a promotion in Siebel CRM.
Component Pricing	Pricing-related information of the component and subcomponents of the promotion. Can be updated in context of the promotion. Oracle Product Hub publishes the hierarchy of items, which includes the components and the associated pricing-related attributes. Oracle AIA sets these context-specific values for the corresponding subcomponents of promotions in Siebel CRM
Version: Structure	<p>Provides information about the structure associated with the BOM. Represents the contextual information of the BOM components. Oracle AIA creates these as relationships in Siebel CRM.</p> <p>The integration supports product and class relationships only. See Option Class Support in Item Synchronization for Siebel CRM.</p> <p>During the BOM synchronization, this contextual information is set in the target application for each of the components of the BOM. For example, in Siebel CRM, this information is set for the relationship attributes of the product. The structure can also be associated with the item catalog categories in Oracle Product Hub.</p>

Multi-row attribute groups in Oracle Product Hub provide the flexibility to associate multiple sets of attribute values with the same object instance.

The communications library includes the following multi-row attribute groups:

- Pricelist
- Commitment Charge Credits
- Commitment Terms
- Product Promotions Upgrade
- Subject Compatibility Rules
- Destination System Specification
- Billing Products Discount Map
- Rate Plan
- Rate Data
- Balance Impact
- Tier Group
- Day Time Range
- Days of the Week Range
- Billing Discount Event Map

Oracle Product Hub publishes the multiple sets of attribute values associated with each of these attribute groups. The integration includes the sets of attribute values within the canonical model. Supporting multi-row attribute groups in the target application is handled in the corresponding connector services.



---



---

**Note:** Some of the target systems can perform **delete all** and **insert all** operations. The interfaces on the target systems must be carefully designed and implemented based on the usage of this set of attribute values in the corresponding applications. For example, in Siebel CRM, **delete all** and **insert all** cannot be performed on subject compatibility rules because the rules have references in other components.

---



---

You can define new multi-row attribute groups and the integration includes them in the canonical model.

---



---

**Note:** The integration to Siebel CRM does not support customer-defined multi-row attribute groups.

---



---

## Updating Publishing Status in Oracle Product Hub

Oracle Product Hub items have a publishing status which indicates if they are successfully published to a given system. After publishing all entities in a batch to target systems, Oracle AIA updates the status in Oracle Product Hub so that a product administrator can track publication results and republish in case of failure. Oracle AIA sends a separate status update after publishing items, price lists, and BOMs.

The publishing status can have the following values:

- **Batch Submitted:** Set by Oracle Product Hub for items before publication.
- **In Process:** Set by Oracle AIA after querying the data from Oracle Product Hub.
- **Succeeded:** Set by Oracle AIA when the item publication succeeds.
- **Failed:** Set by Oracle AIA when the item publication fails.
- **Ignored:** Set as a default value in Oracle Product Hub to indicate that an item is ignored by a specific system because it does not apply to it, for example, promotions for BRM.

When the target system processes the entire item or BOM EBM, such as when an Oracle AIA error prevents the entire EBM from being processed, Oracle AIA updates the status at the batch level. When the target system processes the EBM in a more granular way, such as when synchronizing promotions to Siebel CRM, Oracle AIA updates the status at the item level. When the target system processes the EBM at a level more granular than the entire payload but less granular than individual items, Oracle AIA updates the status at the item level.

The publishing status message contains the following parts:

- An **initial string**, is an English-language message prepared by Oracle AIA that indicates the service where the error occurred. Because there can be multiple interactions for a single item (such as synchronizing the item, its price list, and its structure), this message can help the Oracle Product Hub administrator understand what interaction failed.
- The **message returned** by the target system. This may be an error message or an informational message. For Siebel CRM, the message also contains the workspace ID for the batch.

## Oracle Product Hub Interfaces

The integration uses the Oracle Product Hub interfaces shown in [Table 2–7](#).

**Table 2–7 Oracle Product Hub Interfaces**

Interface	Service Name	Operation Name	Request and Response
Item Synchronization: Inbound Web Services	ItemService	GetItemInformation	ItemService_GetItemInformation, ItemService_GetItemInformationResponse
BOM Synchronization: Inbound Web Services	ItemService	getStructureDetails	ItemService_getStructureDetails, ItemService_getStructureDetailsResponse
Publication Service: Outbound Web Services	PublicationService_Service	PublicationService_UpdatePublicationStatus	PublicationService_UpdatePublicationStatus_InputMessage, PublicationService_UpdatePublicationStatus_OutputMessage
Publication Service: Outbound Web Services	PublicationService_Service	PublicationService_GetBatchSystems	PublicationService_GetBatchSystems_InputMessage, PublicationService_GetBatchSystems_OutputMessage
Item Catalog Synchronization: Inbound Web Services	ItemCatalogService	getICCDetails	ItemCatalogService_getICCDetails, ItemCatalogService_getICCDetailsResponse
Item Catalog Synchronization: Inbound Web Services	ItemCatalogService	PublicationService_GetBatchSystems	PublicationService_GetBatchSystems_InputMessage, PublicationService_GetBatchSystems_OutputMessage
Valueset Synchronization: Inbound Web Services	ValuesetService	getValueSetDetails	ValuesetService_getValueSetDetails, ValuesetService_getValueSetDetailsResponse
Product Class: Inbound Web Services	SWIProductClassIntegrationIO	SWIClassImportUpsert	SWIClassImportUpsert_Input, SWIClassImportUpsert_Output
Attribute Definition: Inbound Services	SWIAttributeIntegrationIO	SWIAttributeImportUpsert	SWIAttributeImportUpsert_Input, SWIAttributeImportUpsert_Output

## Oracle Product Hub Integration Services

The integration includes the following services for Oracle Product Hub:

- [SyncItemListPIMEventConsumer](#)
- [SyncItemListPIMReqABCImpl](#)
- [QueryItemListPIMAdapter](#)
- [QueryBillOfMaterialsListPIMAdapter](#)
- [SyncSpecificationValueSetListPIMEventConsumer](#)
- [SyncSpecificationValueSetListPIMReqABCImpl](#)
- [SyncItemCatalogCategoryPIMEventConsumer](#)
- [SyncItemCatalogCategoryPIMReqABCImpl](#)

### SyncItemListPIMEventConsumer

This consumer service listens for Oracle Product Hub item publish business events. It picks up the WF\_EVENT\_T\_msg item event message and sends it to SyncItemListPIMReqABCImpl.

## SyncItemListPIMReqABCImpl

This service is a BPEL process that coordinates the item synchronization from Oracle Product Hub to the target systems. It uses the event message to query the target systems and the full item details from Oracle Product Hub, then sends these in EBMs to the EBSs for item, price list, and BOM synchronization. It receives responses from the EBSs and updates Oracle Product Hub with the publishing status for each item.

## QueryItemListPIMAdapter

This BPEL adapter service receives the session ID, batch ID, mode, and system ID from the SyncItemListPIMReqABCImpl and retrieves the item information from Oracle Product Hub.

## QueryBillOfMaterialsListPIMAdapter

This BPEL adapter service receives the session ID, batch ID, mode, and system ID from the SyncItemListPIMReqABCImpl and retrieves the BOM information from Oracle Product Hub.

## SyncSpecificationValueSetListPIMEventConsumer

The lightweight mediator service listens for Oracle Product Hub valueset publish business events. It picks up the event message and sends it to the SyncSpecificationValueSetListPIMReqABCImpl service.

This service is used only with the Siebel CRM pre-built integration option. See ["Synchronization of Valuesets Independently"](#) for more information about the flow in which this service is used.

## SyncSpecificationValueSetListPIMReqABCImpl

This service is a BPEL process that coordinates the valueset synchronization from Oracle Product Hub to Siebel CRM.

It uses the Oracle Product Hub valueset business event message from SyncSpecificationValueSetListPIMEventConsumer to query the valuesets from Oracle Product Hub, then sends them in EBMs to the SpecificationValueSetEBS service. It receives responses from the EBSs and updates Oracle Product Hub with the publishing status for each valueset.

This service is used only with the Siebel CRM pre-built integration option. See ["Synchronization of Valuesets Independently"](#) for more information about the flow in which this service is used.

## SyncItemCatalogCategoryPIMEventConsumer

This consumer service listens to Oracle Product Hub business events. It picks up the WF\_EVENT\_T\_msg item event message from Oracle Product Hub and sends it to the SyncItemCatalogCategoryPIMReqABCImpl service.

This service is used only with the Siebel CRM pre-built integration option. See ["Synchronizing Item Catalog Category Implementation Flow"](#) for more information about the flow in which this service is used.

## SyncItemCatalogCategoryPIMReqABCImpl

This service is a BPEL process that coordinates the item catalog category synchronization from Oracle Product Hub to Siebel CRM.

It uses the Oracle Product Hub item catalog category business event message from SyncItemCatalogCategoryPIMEventConsumer to query the item catalog categories from Oracle Product Hub, then sends them in EBMs to the SpecificationValueSetEBSV2 and SyncClassificationSchemeEBSV2 services. It receives responses from the EBSs and updates Oracle Product Hub with the publishing status for each item catalog category.

This service is used only with the Siebel CRM pre-built integration option. See ["Synchronizing Item Catalog Category Implementation Flow"](#) for more information about the flow in which this service is used.

## Assumptions and Constraints for Oracle Product Hub

The assumptions and constraints for Oracle Product Hub are as follows:

- Oracle Product Hub is the product master for defining items, BOMs, and metadata. Any changes or enrichment in the target systems are not synchronized back to Oracle Product Hub.
- The multi-language support offered by Oracle Product Hub for the valuesets depends on the validation type associated with the valueset. The **independent** and **translatable** validation types are supported.
- The integration does not support the following:
  - Synchronizing products from an inventory organization that is shared among multiple operating units.
  - Publishing alternate catalogs from Oracle Product Hub is not supported by the integration.
  - The Oracle Product Hub variant type attribute group.
  - Publishing item catalog categories as a batch. You must publish each item catalog category to the target systems, except for item catalog categories that are part of hierarchies, which can be published together.
- Oracle Product Hub does not support the following:
  - Changing the parent item catalog category associated with an item catalog category after it is released, even to create a new version.
  - Adding the same item more than once as a component of a BOM, although this is allowed in Siebel CRM. If required, this can be achieved by using the operation sequence support provided by Oracle Product Hub
  - Using Boolean and integer data types. To support Boolean and integer data types, append **\_BOOL** to the end of valueset name and set predefined values, such as **True** or **False**. You can define any number of Boolean valuesets using the technique, but the values must always be same, and seeded.

---

---

# Product Master Data Management Integration Option for Siebel CRM

This chapter describes the Oracle Application Integration Architecture (Oracle AIA) Product Master Data Management integration option for Siebel CRM.

## About the Siebel CRM Option

The Siebel CRM option provides the following end-to-end process flows:

- Synchronizing item catalog categories, which includes:
  - Relationships and structures
  - Attribute groups
  - Customer user-defined attributes (UDAs)
  - Transaction attributes
  - Hierarchies

See "[About the Process Flow for Synchronizing Item Catalog Categories](#)" for more information.

- Synchronizing valuesets independently or as part of item catalog categories. See "[About the Process Flow for Synchronizing Valuesets](#)" for more information.

The Siebel CRM option also supports synchronizing items, price lists, and BOMs as described in "[About the Process Flow for Synchronizing of Items and BOMs](#)". This process flow also supports the following:

- [Multi-Event Product Synchronization from Oracle Product Hub to Siebel CRM](#)
- [Support for Class Type Relationship in Product Synchronization for Siebel CRM](#)
- [Support for Controlling Auto-Release of Entities Published in the Same Batch](#)
- [Synchronization of Promotions from Oracle Product Hub to Siebel CRM](#)

## About the Process Flow for Synchronizing Item Catalog Categories

This process flow lets you use Oracle Product Hub to publish item catalog categories to participating applications.

You create or update item catalog categories as drafts, then release them to create working or active versions. At any given point, only one active released version of the item catalog category exists.

When you publish item catalog categories from Oracle Product Hub, the active released versions or the future-dated versions of the item catalog categories are published one at a time to the participating applications.

Publishing item catalog categories sends basic information, such as item catalog category ID, to the integration. The integration queries Oracle Product Hub for the complete details of the item catalog category and all the following associated entities:

- Item catalog categories
- Attribute groups
- Seeded attributes and custom added attributes
- Transaction attributes
- Valuesets
- Structures or relationships associated with item catalog category

When you publish a new item catalog category from Oracle Product Hub, the integration creates a workspace version of a product class in Siebel CRM. When you publish an updated item catalog category from Oracle Product Hub, the integration updates the existing product class. Updates can include updated associated entities.

Republishing a version of an item catalog category without any changes does not create a new version in Siebel CRM.

After the integration creates or updates the product class in Siebel CRM, it sends the status to Oracle Product Hub. Oracle Product Hub maintains a publication history for all the versions of item catalog category from each participating application.

The versions of item catalog category in Oracle Product Hub are associated with the effective start and end dates. You can only publish the active released version or future-dated versions from Oracle Product Hub, not draft versions.

You can revert to an earlier version of an item catalog category by creating a new version with effective dates. No impact occurs to the integration with the creation of this new version because the integration process handles this as new versions.

## About Multi-Language Support for Item Catalog Categories

Item catalog category names are not translatable, but descriptions are. You can use up to 250 characters in the description to capture a language specific name. To capture language specific values, use the **translatable** validation type.

## About Automatically Releasing Entities

By default, Oracle Product Hub activates the auto-release option. Siebel CRM releases the entity if the value is set or it must create the workspace version of the entity.

For more information about auto-release, see ["Support for Controlling Auto-Release of Entities Published in the Same Batch"](#).

## About Synchronizing of Relationships and Structures

An item catalog category can be defined with relationship and structure that contains items and BOMs which have already been synchronized to the participating applications.

When the integration queries Oracle Product Hub for item catalog categories, Oracle Product Hub also provides the structures that are defined as relationships with the

item catalog category. The item catalog category references only the latest version of the structure. The integration creates relationships for the product classes in Siebel CRM.

Oracle Product Hub also returns the component UDA associated with the relationship that stores context-specific values for the relationship within the item catalog category such as minimum, maximum, and default value. The integration uses this information to create relationships, add corresponding products, and set the context-specific values specified by the component UDA for the product class in Siebel CRM.

All types of BOMs can be added as the relationship for the item catalog categories.

The structure definition in Oracle Product Hub for item catalog categories supports two types of parent-child relationships:

- Relationship of domain type as product: The product relationship represents another item that is related to the root item.
- Relationship of domain type as class: The class relationship provides a list of items that can be treated as options. In Oracle Product Hub, these items are defined as BOM with BOM item type as option class, and the options are added as components of the BOM.

---



---

**Note:** Domain type is a UDA associated with the item that identifies the type of relationship.

---



---

For more information about telecommunications seeded library attributes and their corresponding valuesets, see the discussion of seeded item metadata libraries in *Oracle Product Hub Implementation Guide*.

This is an example of how various relationships under item catalog category are handled by the process integration. This is similar to the relationship handling of items in Option Class Support in Item Synchronization for Siebel CRM.

[Table 3–1](#) is an example of how product and class type relationships are supported in Oracle Product Hub and how the process integration creates them in Siebel CRM:

Wireless Service item catalog category

Wireless Router (Item)

Bluetooth devices (Item with an option class type BOM and entity type of option group)

Samba Bluetooth Headset

B-Micro Bluetooth Headset

**Table 3–1 Product and Class Types**

Wireless Router Class	Wireless Router
Wireless Device Accessory	Bluetooth devices
Wireless Device Accessory	Samba Bluetooth Headset
Wireless Device Accessory	B-Micro Bluetooth Headset

[Table 3–2](#) depicts the Oracle Product Hub definition of the relationship in the seeded attribute group Version Structure associated with the Wireless Service item catalog category. These attributes are component attributes values the values of which are set in context of the Wireless Service item catalog category.

**Table 3–2 Attributes**

Relationship Name	Domain Type	Product	Class	Default Cardinality
Relationship1	Product	Wireless Router	1	NA
Relationship2	Class	Bluetooth devices	Wireless device accessory	1

For product relationships, the process integration creates a relationship of type product in Siebel CRM for the Wireless Service item catalog category. This relationship has an empty relationship domain in Siebel CRM.

For class relationships, the process integration creates a relationship of type class and adds all the components of the Bluetooth devices as the relationship domain. The product (Bluetooth devices) is not included as a part of the relationship in Siebel CRM.

This is an example of the support for relationship for product class in Siebel CRM:

Wireless Service item catalog category

Relationship1 Wireless Router

Relationship2 Wireless device accessory

Samba Bluetooth Headset

B-Micro Bluetooth Headset

Table 3–3 shows the structure attributes of the Wireless Service item catalog category in Siebel CRM.

**Table 3–3 Structure Attributes**

Relationship Name	Domain Type	Product	Class	Default Cardinality
Relationship1	Product	Wireless Router	NA	1
Relationship2	Class	NA	Wireless device accessory	1

Table 3–4 shows the relationship domain and quantity.

**Table 3–4 Relationship Domain**

Relationship Domain	Quantity
Samba Bluetooth Headset	X
B-Micro Bluetooth Headset	X

To update the relationship domain, more product components can be added to the product Bluetooth devices, and all the item catalog categories that have class relationship with Bluetooth devices must be synchronized to Siebel CRM for the relationship domain to be updated. Updating the Bluetooth devices and the synchronization is a prerequisite.

Any change in the relationship creates a new version of Wireless Service item catalog category in Oracle Product Hub. The process integration updates the corresponding Wireless Service item catalog category and the relationship in Siebel CRM.

## About Synchronizing Attribute Groups

Every item catalog category is associated with zero or more attribute groups. Attribute groups are not defined within a context of the item catalog category and are reused across item catalog categories. Attribute groups are used to categorize user-defined



attributes and operational attributes. The child item catalog category inherits the attribute groups of the parent item catalog category. The attribute groups are published when the associated item catalog categories are published. In the current release, Oracle Product Hub does not maintain the publication status of the attribute group in the publication history.

Whenever the integration queries Oracle Product Hub for the item catalog category, the attribute groups associated with the item catalog category are also returned. The attribute groups are represented at the canonical layer (specification group) as a part of classification scheme EBM. The attribute groups have reference attributes (specification).

None of the entities corresponds to attribute groups in Siebel CRM; therefore, they are ignored at the Siebel CRM connector service in the integration.

---

---

**Note:** Oracle Product Hub provides a seeded telecommunications library that consists of a set of predefined attribute groups consisting of attributes that have predefined mapping to the fields of the item in the downstream application (Siebel CRM). Updates to these seeded attribute groups and the associated attributes are not supported by the Oracle Product Hub. In addition, the integration does not support addition or deletion of attributes within seeded attribute groups; however, you can define new attribute groups and attributes.

---

---

New attribute groups with attributes can be defined. The integration differentiates between the customer-defined UDA and the seeded UDA and creates these as flex field attributes in Siebel CRM. Whenever customer-defined UDA is updated or new attributes are added, all the item catalog categories that are associated with the attribute groups have to be republished to the downstream applications.

Attribute groups can be of type multi-row. A multi-row attribute group is used to define characteristics that are record-based such as rules, pricelist lines, and so on. See the telecommunications seeded library attributes for definitions of multi-row seeded attribute groups.

When multi-row attribute groups are published, Oracle Product Hub provides the complete definition of these attribute groups with the records. The integration supports these multi-row attribute groups in the canonical model, but ignores them at the Siebel CRM connector. (The values for the multi-row attribute groups are defined during item definition and are set in the downstream applications during the item synchronization process.)

## About Synchronizing Customer UDAs

The UDAs are defined within the context of the attribute group. These are not versioned entities. The UDAs are associated with valuesets. Context-specific values are set for every UDA within the attribute group. The child item catalog category inherits these UDAs from the parent item catalog category.

The static UDA, component UDA, and operational attributes are published with the attribute groups when the corresponding item catalog category is published.

Whenever integration queries Oracle Product Hub for item catalog category, the UDAs that are associated with the item catalog category through attribute groups are returned along with the item catalog category definition and other entities. The attributes that exist across the attribute groups are extracted and grouped under canonical models of the attribute.

Siebel CRM does not have attributes as separate entity. The static UDA and operational attributes that are defined within attribute groups are ignored in the Siebel CRM connector. These attributes have fixed mapping with the product fields in Siebel CRM and Oracle E-Business Suite respectively. The values for these fields are set during item synchronization.

---

---

**Note:** UDAs are defined within the context of the attribute group. Any updates to the UDA results in the update of the attribute group. The item catalog category has to be republished for the updates on the UDA.

---

---

All customer-defined UDAs are stored in the flexfield support provided by CRM; therefore, you must ensure that you define unique attributes across custom attribute groups. The recommended approach is to prefix the corresponding attribute group name with XX for every customer-defined attribute. All the customer-defined attribute groups that are defined without a prefix are not mapped to any target application, but are available in the Oracle Application Integration Architecture (Oracle AIA) layer. You have to extend the connector services to map these attributes to the target applications.

The Oracle Product Hub does not track the publication status for attributes in the publication history for UDA.

## About Synchronizing Transaction Attributes

Note the following:

- The transaction attributes are defined within the context of the item catalog category in Oracle Product Hub. These are not versioned entities. The transaction attributes are associated with versioned valuesets in Oracle Product Hub. Context-specific values are set for every transaction attribute within the item catalog category. The child item catalog category inherits the transaction attributes and the valueset from the parent item catalog category.
- Whenever the integration queries the Oracle Product Hub for item catalog category, the transaction attributes that are associated with the item catalog category are also returned. The integration process extracts all the transaction attributes and includes them in the specification section of the classification composition EBM. The process integration creates these as attributes of the product class in Siebel CRM and sets the corresponding context-specific values.
- Any changes in the transaction attributes associated with the item catalog category (add, delete, or update the context specific values) updates the item catalog category and a new version is released in Oracle Product Hub. When an updated item catalog category is published, the integration updates the corresponding product class with the transaction attributes in Siebel CRM.
- The static attributes and the transaction attributes both are grouped within the canonical model. The static attributes in the specification is ignored by the Siebel CRM connector and the transaction attributes are created as product class attributes.
- Duplicate transaction attributes cannot be defined within an item catalog category. If a similar transaction attribute is defined in a parent item catalog category and the one of its child item catalog category in the hierarchy, the definition in the child hierarchy overrides the definition of the inherited transaction attribute.

- If the inherited transaction attribute is updated in the child item catalog category, the transaction attribute does not remain an inherited attribute. It becomes a native attribute of the child item catalog category.

## About Synchronizing Transaction Attributes without Valuesets

Transaction attributes are associated with valuesets and the values for the transaction attributes are selected from the list during order capture. For some transaction attributes, the values can be freeform text. For such transaction attributes, a valueset may not be associated in certain applications. For example, Oracle Product Hub.

In Oracle Product Hub, a transaction attribute is created in context of an item catalog category. The transaction attributes may or may not be associated with versionable valuesets. The following are the data types supported for the transaction attributes:

- Char
- Number
- Datetime
- Standard Datetime
- Translatable Text
- BOOL (Oracle Product Hub does not support the Boolean type. If the transaction attribute name has a BOOL suffix, it is treated as a Boolean data type.
- INT (Oracle Product Hub does not support the Integer data type. If the transaction attribute name has an INT suffix, it is treated as an Integer data type)

For the transaction attributes of the first four data types associating the valueset is not mandatory but a valueset can be associated in Oracle Product Hub. However, for a transaction attribute of data type 'Translatable Text' a valueset cannot be associated. This is restricted with the validations in Oracle Product Hub.

During the Item Catalog Synchronization from Oracle Product Hub to Siebel CRM, if a transaction attribute has no valueset associated with it, then the process synchronization creates an attribute definition of Domain Type freeform by using the transaction attribute name and the datatype provided by Oracle Product Hub.

Subsequent updates to Item Catalog categories, for instance, the addition or deletion of transaction attributes or items in the structure will not affect the transaction attributes in Siebel CRM.

## About Synchronizing Item Catalog Category Hierarchies

Note the following:

- The publication framework for item catalog category provides flags that provide these publishing options for the product administrator in Oracle Product Hub:
  - Publish only the selected item catalog category.
  - Publish selected item catalog category and all its parents.
  - Publish selected item catalog category and all its children.
  - Publish selected item catalog category and all its parents and children.
- In cases in which a single item catalog category is selected for publishing, if that item catalog category has a parent item catalog category that is not published, the parent item catalog category must be published to the downstream applications. If not, the process integration fails. You must publish the parent item catalog

category and republish the child item catalog category. In this case, the child item catalog category references the parent item catalog category. The integration process associates the child item catalog category and its parent in Siebel CRM. Alternately, whenever a parent item catalog category is updated, the product administrator must publish the complete hierarchy associated with the item catalog category.

- In cases in which the flag to publish the entire hierarchy (parents or children; or parents and children) is set, the integration process queries Oracle Product Hub for all the item catalog categories in the hierarchy. Oracle Product Hub returns the complete definition of all the item catalog categories in the hierarchy. The standard methodology to publish the entire hierarchy (parents and children) is recommended. The integration process creates the complete hierarchy of all the product classes in Siebel CRM.
- Whenever an item catalog category is updated in Oracle Product Hub and published to downstream applications, the integration updates the existing product classes. In addition, the corresponding hierarchy is updated in Siebel CRM using the references to children or parent classes. The product classes are released to reflect the hierarchy in the active version.
- The structure, transaction attributes, and the attribute groups of the parent item catalog category are passed to the child item catalog category. The inherited structure in the child item catalog category can be updated, for example, cardinality. Whenever an inherited structure is updated in the context of the child, the inheritance breaks and the structure becomes a native relationship of the child item catalog category. The behavior is the same for transaction attributes. The process integration updates the corresponding entities in the downstream applications.

## About Associating Item Catalog Categories with Items

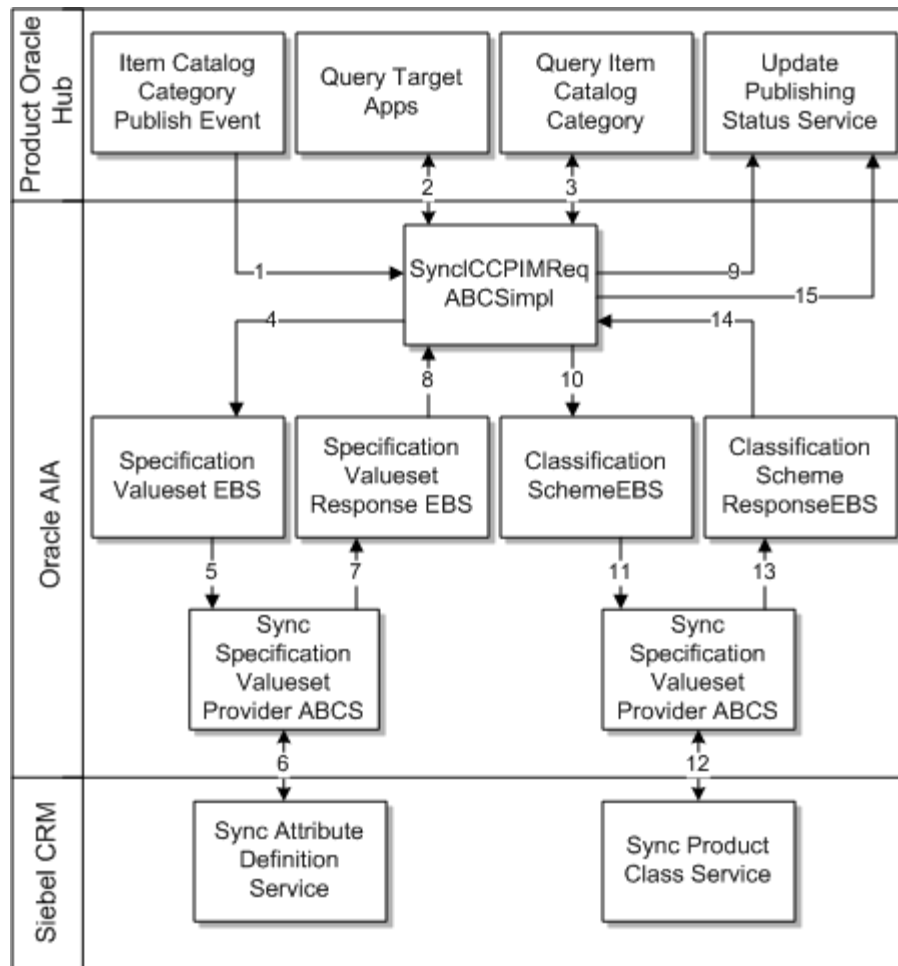
An item catalog category can be associated with the items. The items inherit the static attributes, transaction attributes, and structure associated with the item catalog category. An item can be associated with only one item catalog category.

- When items are published in the publication framework, a reference to the item catalog category is published with the item.
- The process integration creates the association of item and the item catalog category during the item synchronization process.

## Synchronizing Item Catalog Category Implementation Flow

Figure 3–1 shows the high-level integration flow for the item catalog category synchronization.

Figure 3–1 Synchronizing Item Catalog Categories



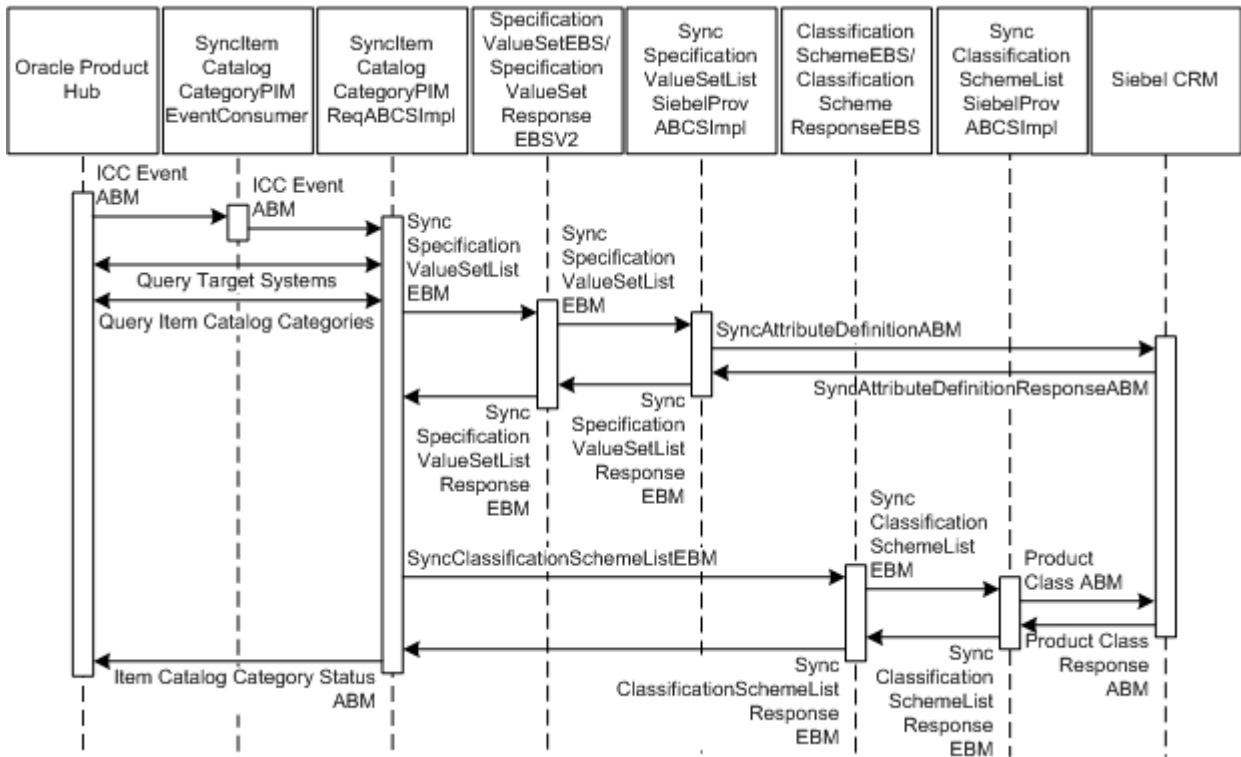
These are the synchronization steps:

1. The product administrator in Oracle Product Hub publishes the item catalog category from the publication-framework user interface. The process publishes an event and provides basic information about item catalog category. The event consumer process invokes the requester ABCS for Oracle Product Hub.
2. The requester ABCS queries the target applications from Oracle Product Hub by calling the get target systems service.
3. Once the target systems are retrieved, the requester queries the item catalog category definition from Oracle Product Hub by calling the Oracle Product Hub Query item catalog category interface. The Oracle Product Hub Query item catalog category interface returns the complete information of the item catalog category together with the complete definitions of the associated entities (attribute groups, attributes, and valuesets). Valuesets are independent entities and can be associated with different attributes within the same or different attribute group in an item catalog category.
4. The Oracle Product Hub requester connector service transforms the response from the Oracle Product Hub query item catalog category interface into the SpecificationValuesetListEBM and ClassificationSchemeListEBM. The ClassificationSchemeListEBM has complete definitions of item catalog categories, attribute groups, attributes, and their associated valuesets. The connector service

- invokes an enterprise business service and provides the SpecificationValuesetListEBM as the input.
5. The specification valueset enterprise business service routes to the appropriate synchronize attribute definition Siebel CRM provider ABCS.
  6. The attribute definition class Siebel CRM provider ABCS invokes the API provided by Siebel CRM to create the corresponding attribute definitions in Siebel CRM.
  7. The Siebel CRM synchronize attribute definition returns to the caller with indication of success or failure.
  8. Once the response is returned by the synchronize attribute definition API, the Siebel CRM connector service invokes a response EBS.
  9. The specification valueset response EBS returns the response to the requester ABCS. The Oracle Product Hub requester ABCS transforms the valueset response EBM and invokes the Oracle Product Hub service to update the status. The Oracle Product Hub service updates the publication history for the valueset.
  10. The Oracle Product Hub requester ABCS invokes the ClassificationSchemeEBS and provides ClassificationSchemeListEBM as input.
  11. The ClassificationSchemeEBS routes to the appropriate synchronize product class Siebel CRM provider ABCS. It invokes the API provided by Siebel CRM to create product classes and their attributes. The product class also has structure or relationships associated. The relationships are created based on the relationship type. Siebel CRM API also synchronizes the product class names in multiple languages.
  12. The synchronize product class API returns the status to the connector service.
  13. The Siebel CRM connector service invokes a response EBS.
  14. The ClassificationSchemeResponseEBS sends the status back to the requester ABCS.
  15. The Oracle Product Hub requester ABCS transforms the classification scheme response EBM and invokes the Oracle Product Hub service to update the status. The Oracle Product Hub service updates the publication history for the item catalog category.

Figure 3–2 illustrates the services and their interaction.

Figure 3–2 Sequence for Synchronizing Item Catalog Categories



The following are the sequence diagram steps and descriptions:

1. Oracle Product Hub item catalog category publish events are raised when the product administrator publishes the item catalog category from the Oracle Product Hub system.
2. SyncItemCatalogCategoryPIMEventConsumer listens to the business event and receives the WF\_EVENT\_T\_msg event payload. SyncItemCatalogCategoryPIMEventConsumer routes to SyncItemCatalogCategoryPIMReqABCSImpl with complete event payload.
3. The Oracle Product Hub requester, SyncItemCatalogCategoryPIMReqABCSImpl, receives the event payload and calls the Oracle Product Hub publication service to get the list of target applications by passing the batch ID. It then constructs the query item catalog category Oracle Product Hub ABM, calls the Oracle Product Hub query item catalog category web service, and gets the complete item catalog category details.

It transforms the Query item catalog category response message to SyncSpecificationValueSetListEBM (only if valuesets are associated with T-Attr in Oracle Product Hub and the SyncValueSet flag is set to True in **AIAConfigurationProperties.xml**) and SyncClassificationSchemeListEBM.

If valuesets are associated with T-Attr and the valueset synchronize property is set to True in the **AIAConfigurationProperties.xml**, SyncItemCatalogCategoryPIMReqABCSImpl invokes SpecificationValueSetEBS, passing SyncSpecificationValueSetListEBM to synchronize valuesets in Siebel CRM and other participating applications.

Once the valueset synchronization is done successfully, it calls the classification scheme EBS passing SyncClassificationSchemeListEBM to synchronize the item catalog category.

It gets the item catalog category synchronization response, constructs the Oracle Product Hub publication service ABM, and calls the Oracle Product Hub publication web service.

This service takes the batch ID and gives the list of the target systems to which the synchronization should be done.

4. Oracle Product Hub Query item catalog category web service receives the batch ID, publishes parent hierarchy and child hierarchy flags, and returns the item catalog category details associated with that batch.
5. SpecificationValueSetEBS receives SyncSpecificationValueSetListEBM as an input and routes it to the SyncSpecificationValueSetListSiebelProvABCServiceImpl service to synchronize valuesets.
6. SyncSpecificationValueSetListSiebelProvABCServiceImpl receives SyncSpecificationValueSetListEBM. It transforms the EBM into the SyncAttributeDefinition ABM and calls the Siebel CRM valueset-synchronization Web service. Once synchronization is done, it gets the response from the Siebel CRM system, transforms SyncSpecificationValueSetEBM to SyncSpecificationValueSetResponseEBM, and invokes SpecificationValueSetResponseEBS.
7. Synchronize attribute definition Siebel CRM Web service synchronizes the attribute definition in Siebel CRM.
8. SpecificationValueSetResponseEBS receives SyncSpecificationValueSetListRespEBM as input and routes it to the SyncItemCatalogCategoryPIMReqABCServiceImpl service.
9. Once the valueset synchronization is done successfully in Siebel CRM, SyncItemCatalogCategoryPIMReqABCServiceImpl calls the ClassificationSchemeEBS passing SyncClassificationSchemeListEBM as its input and routes it to the SyncClassificationSchemeListSiebelProvABCServiceImpl service.
10. SyncClassificationListSiebelProvABCServiceImpl receives SyncClassificationSchemeListEBM. It transforms the EBM into a SWIClassImportUpsert\_Input message and calls the Siebel CRM product-class synchronization web service to synchronize simple product class along with attributes or structures if associated.

It then populates the cross-reference with the corresponding Siebel CRM IDs. Once this is done, it checks for any associated hierarchy. If it exists, it makes another call to the same Siebel CRM Web service passing hierarchy information. Once synchronization is done, based on the workspace auto-release flag (Y), it calls the Siebel CRM product-class synchronization service again and passes the workspace auto-release flag. It then constructs the SyncClassificationSchemeListResponseEBM and passes it to ClassificationSchemeResponseEBS.

11. Synchronize product class Siebel CRM Web service synchronizes the product class in Siebel CRM and returns the response message.  
Once the product class synchronization is done in Siebel CRM, SyncClassificationSchemeListSiebelProvABCServiceImpl calls the Siebel CRM product-class synchronization Web service again, passing workspace name, workspace refuse flag, and workspace auto-release flag to release the workspace.
12. ClassificationSchemeResponseEBS gets the SyncClassificationSchemeListResponse message and routes it to SyncItemCatalogCategoryPIMReqABCServiceImpl.



13. Once the Oracle Product Hub requester gets the item catalog category synchronization response message, it calls the Oracle Product Hub item catalog category publish status Web service to update the publication status in Oracle Product Hub. These are the possible status values:
  - Failed
  - Succeeded
  - Partial Fail (in case valueset fails)

## About the Process Flow for Synchronizing Valuesets

The valuesets can be published independently or during the item catalog category publish from the Oracle Product Hub publication framework. The valuesets can be associated with the static UDA, customer UDA, and transaction attributes. For all the seeded valuesets of the telecommunications library, the integration provides DVMs. For all the valuesets that are associated with the transaction attributes, the integration with Siebel CRM creates the attribute definitions.

Valuesets associated with the customer-defined UDAs are handled as freeform values with the corresponding language codes. The MLS support for the valuesets depend on the validation type associated with the valueset. See "[Multi-Language Support for Item Synchronization](#)" for more information.

## Synchronization of Valuesets as Part of Item Catalog Categories

Note the following:

- Whenever the integration queries Oracle Product Hub for the item catalog category that is published, the valuesets that are associated with the attributes of the item catalog category are returned.

---

---

**Note:** This includes the nonversionable valuesets that are associated with the UDA within the attribute groups and the versionable valuesets associated with the transaction attributes. In addition, if the same valueset is associated with more than one attribute, Oracle Product Hub returns a unique set (union) of valuesets associated with the item catalog category.

---

---

- The integration supports only the synchronize operation.
- In cases in which a versioned valueset is republished to Siebel CRM, the updates on the attribute definition do not create a new version in Siebel CRM. Once the attribute definitions are created, they are associated with the corresponding attributes of the product classes in Siebel CRM.
- Even though the canonical model consists of valuesets associated with the UDA and the transaction attributes, the Siebel CRM connector ignores all the valuesets that were associated with the UDA and creates attribute definitions for all the valuesets that were associated with the transaction attributes. The integration provides DVMs for all the valuesets that are associated with the seeded attributes of the telecommunications library.
- Whenever an item catalog category is published, the publication framework extracts the version of all the valuesets that were associated when the item catalog category was released, implying that the version of the valueset that was associated with the item catalog category when the item catalog category was

created might differ from the active released version of the valueset; therefore, publishing always picks up the version of the valueset that was associated when the item catalog category was released. The version in the downstream application is driven by the versions that are published. No one-one mapping exists between the versions of entities in Oracle Product Hub and the downstream applications. Oracle Product Hub provides validation to ensure that only the released version of the valueset can be associated with the attribute.

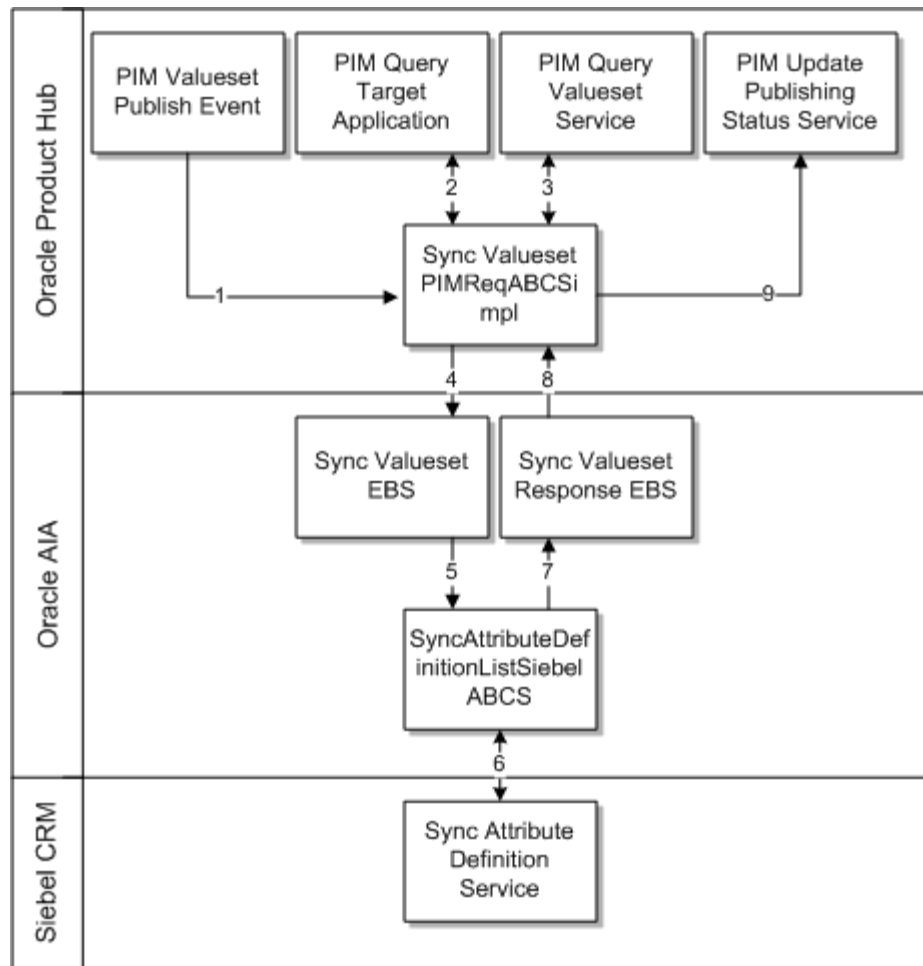
## Synchronization of Valuesets Independently

Note the following:

- The publication framework provides a separate user interface to search and to publish the valuesets. Multiple versions of the valueset exist in Oracle Product Hub, but at any point, only one active released version of a valueset exists. Only the active released version and the future-dated versions of the valuesets can be published. Oracle Product Hub enables you to publish one or more valuesets per publish process to the downstream participating applications. The publishing of valuesets is a manual process performed by the Oracle Product Hub product administrator.
- The publish action invokes the integration process and provides basic information about valuesets. The integration queries Oracle Product Hub for the complete details of all the valuesets.
- In cases in which a version valueset is republished to Siebel CRM, the updates on the attribute definition do not create a new version in Siebel CRM.
- The versions of valuesets in Oracle Product Hub are associated with the effective start and end dates. Only the active released versions or future-dated versions are published.
- Draft versions-versions that are not yet released-cannot be published from Oracle Product Hub.
- Once the attribute definitions are successfully created in Siebel CRM for versionable valuesets of Oracle Product Hub, a status is propagated back to Oracle Product Hub. Oracle Product Hub maintains the publication history for all the versions of valuesets for each downstream application.
- Oracle Product Hub supports these data types:
  - Char
  - Number
  - Standard Date
  - Standard Date time
  - BOOL (Oracle Product Hub does not support the Boolean type. If the valueset name has a BOOL suffix, it is treated as a Boolean type)
  - INT (Oracle Product Hub does not support the Integer type. If the valueset name has an INT suffix, it is treated as an Integer type)
- Siebel CRM supports all of these data types as well as Boolean and Integer data types except Datetime.

Figure 3–3 illustrates the valueset publish from Oracle Product Hub to Siebel CRM.

Figure 3-3 Synchronizing Valuesets



These are the steps depicted in the diagram.

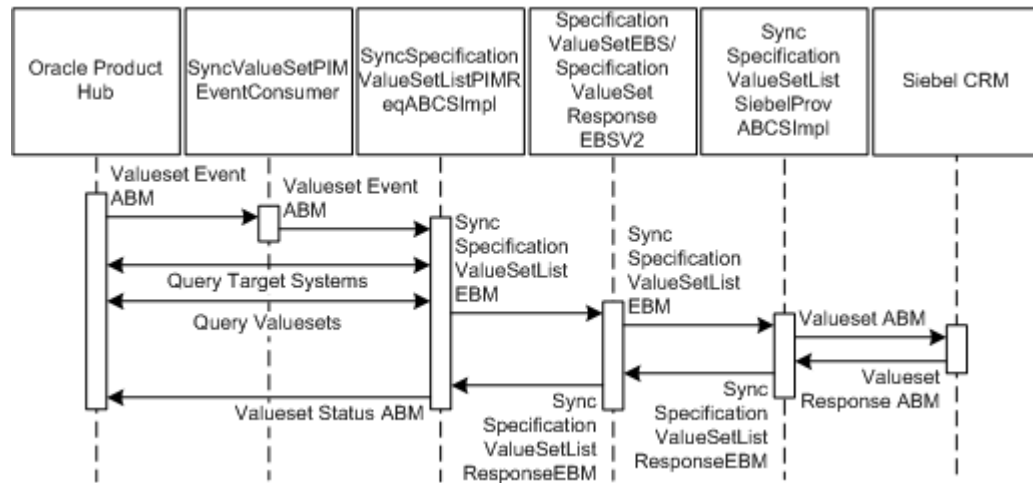
1. The product administrator in Oracle Product Hub publishes one or more valuesets from the publication-framework user interface. The process publishes an event and provides basic information about the valuesets. The event consumer process invokes the requester ABCS for Oracle Product Hub
2. The requester ABCS queries the Oracle Product Hub valueset service.
3. The Oracle Product Hub valueset service returns a single payload with all the valuesets to the Oracle Product Hub requester ABCS.
4. The Oracle Product Hub requester ABCS extracts all the valuesets and transforms them into a single SpecificationValuesetListEBM. It invokes an enterprise business service with the synchronization operation and provides the SpecificationValuesetListEBM as input.
5. The enterprise business service routes to the appropriate provider ABCS.
6. The Siebel CRM provider ABCS transforms the incoming SpecificationListEBM into attribute definitions. The Siebel CRM provider ABCS invokes the Siebel CRM interfaces.
7. The Siebel CRM interface creates the corresponding attribute definitions. It checks the workspace auto-release flag: If it is *Y*, it calls Siebel CRM Web service to release workspace. If it is *N*, the workspace is not released. In case of *D* (default behavior),

it reads release workspace from AIAConfigurationProperties.xml. The interface returns to the caller.

8. The Siebel CRM provider ABCS sends the status through a response EBS.
9. The response enterprise business service routes to the appropriate update valueset publish status Oracle Product Hub provider ABCS.
10. The update valueset publish status Oracle Product Hub provider ABCS invokes the Oracle Product Hub update status Web service and provides the valuesets and the corresponding statuses. The Oracle Product Hub update status service updates the publication history to reflect the status of publish.

Figure 3–4 illustrates the services and their interactions:

**Figure 3–4 Sequence for Synchronizing Valuesets**



The following are the sequence diagram steps and descriptions:

1. Oracle Product Hub valueset events are raised when the product administrator publishes the valuesets from the Oracle Product Hub system. Valuesets can be published in batches.
2. There are two sub-steps in this step.
  - a. SyncValueSetPIMEventConsumer listens to the business event and receives the WF\_EVENT\_T\_msg event payload for the valueset event. SyncValueSetPIMEventConsumer routes to SyncSpecificationValueSetListPIMReqABCsImpl with complete event payload.
  - b. The Oracle Product Hub requester ABC implementation service, SyncSpecificationValueSetListPIMReqABCsImpl, takes the batch ID from the WF\_EVENT\_T ABM payload, queries the get target systems service, and calls the Query valueset Oracle Product Hub service to get the response. SyncSpecificationValueSetListPIMReqABCsImpl service also calls the query target application's Oracle Product Hub service and gets the response back. It counts the target applications from response, and based on the count it constructs the EBM for applications. SyncSpecificationValueSetListPIMReqABCsImpl service transforms the Query valueset Oracle Product Hub service response ABM into

QuerySpecificationValueSetList EBM and calls the SpecificationValueSetEBS Service.

3. SpecificationValueSetEBS Service gets SyncSpecificationValueSetList EBM as its input and routes it to the SyncSpecificationValueSetListSiebelProvABCImpl service.
4. There are two sub-steps in this step.
  - a. SyncSpecificationValueSetListSiebelProvABCImpl receives the SyncSpecificationValueSetList EBM message as input. SyncSpecificationValueSetListSiebelProvABCImpl transforms the SyncSpecificationValueSetList EBM message into a Siebel CRM request ABM message and calls the synchronize attribute definition Siebel CRM Web service.

This service also filters the MLS mapping. A filter condition uses DVM to get the details on how many languages a provider system is supporting and another DVM is used for language code translation across applications. Only supported languages are synchronized to the provider system, for example, Siebel CRM.

Synchronize attribute definition Siebel CRM Web service returns the response to SyncSpecificationValueSetListSiebelProvABCImpl service, which has the status detail.

- b. SyncSpecificationValueSetListSiebelProvABCImpl receives the response from synchronize attribute definition Siebel CRM Web service and transforms the Siebel CRM response ABM message into SyncSpecificationValueSetListResponseEBM and calls the SpecificationValueSetResponseEBS service.

The valueset synchronization is based on all or none basis. If Oracle Product Hub is sending a batch of 10 valuesets out of which five fail, this status is sent back to Oracle Product Hub that batch failed and the error that comes from the provider service is logged in error-handling framework.

5. SpecificationValueSetResponseEBS routes the SyncSpecificationValueSetListResponseEBM to SyncSpecificationValueSetListPIMReqABCImpl.
6. SyncSpecificationValueSetListPIMReqABCImpl service transforms the SyncSpecificationValueSetListResponseEBM message into a Oracle Product Hub valueset status-update Web-service input message and calls the Oracle Product Hub value set status-update Web service.
7. This Web service takes the input message from the SyncSpecificationValueSetListPIMReqABCImpl service and updates the status in Oracle Product Hub as:
  - Success
  - Failed

## About Other Supported Features

This section lists other features supported by the Siebel CRM option.

## Multi-Event Product Synchronization from Oracle Product Hub to Siebel CRM

Products that have multiple charges and events are created as customizable products in Siebel CRM. Events are created as separate products and are added as child products of the main product. The component products represent the charges and events and have the billing type of event that is set in Siebel CRM. The item synchronization process from horizontal implementation supports both create and update of products in Siebel CRM.

To support create and update of pricelist, the process integration to Siebel CRM reuses the communications pricelist Siebel CRM-provider-connector services. The Oracle Product Hub requester item ABCS must create the corresponding item and pricelist line cross-references.

## Support for Class Type Relationship in Product Synchronization for Siebel CRM

The item definition in Oracle Product Hub (for product) supports two types of parent-child relationships:

- Relationship of domain type as product: The product relationship represents another item that is related to the root item.
- Relationship of domain type as class: The class relationship provides a list of items that can be treated as options. In Oracle Product Hub, these items are defined as a BOM with BOM item type as option class. The options are added as components of the BOM. During the promotion definitions, the item that has a class relationship is reused across promotions and one or more items in the options are included or excluded based on the promotion modeling.

---

**Note:** Domain type is a UDA associated with the item that identifies the type of relationship.

---

For more information about Telco seeded library attributes and their corresponding valuesets, see the discussion of seeded item metadata libraries in *Oracle Product Hub Implementation Guide*.

Table 3–5 provides an example of how product and class type relationships are supported in Oracle Product Hub and how the process integration creates them in Siebel CRM.

Root item (item with a model type of BOM)

Wireless router (item)

Bluetooth devices (item with an option class type of BOM and entity type of option group)

Samba Bluetooth Headset

B-Micro Bluetooth Headset

**Table 3–5 Example of How Product and Class Type Relationships are Supported in Oracle Product Hub**

Item Catalog Category	Item	Item Components
Wireless Router Class	Wireless Router	NA
Wireless Device Accessory	Bluetooth devices	NA
Wireless Device Accessory	NA	Samba Bluetooth Headset
Wireless Device Accessory	NA	B-Micro Bluetooth Headset

Table 3–6 depicts the Oracle Product Hub definition of the relationship in the seeded attribute group: version structure associated with the root item. These attributes are component attributes for which values are set in context of the root item.

The type is expected to be set to Option Group. If the type is not set to Option Group, the class relationship is not set in Siebel CRM; instead, it is associated as a product.

**Table 3–6 Oracle Product Hub Definitions**

Relationship Name	Domain Type	Product	Class	Default Cardinality
Relationship1	Product	Wireless Router	NA	1
Relationship2	Class	Bluetooth devices	Wireless device accessory	1

For product relationship, the process integration creates a relationship of type product in Siebel CRM for the root item. This relationship has an empty relationship domain in Siebel CRM.

For class relationship, the process integration creates a relationship of type class and adds all the components of the Bluetooth devices as the relationship domain. The product (Bluetooth devices) is not included as a part of the relationship in Siebel CRM.

Table 3–7 provides an example of the support for relationship for products in Siebel CRM.

Root Item

Relationship1 = Wireless Router

Relationship2 = Wireless Device Accessory

Samba Bluetooth Headset

B-Micro Bluetooth Headset

The structure attributes of the root item in Siebel CRM are:

**Table 3–7 Attributes for Root Items in Siebel CRM**

Relationship Name	Domain Type	Product	Class	Default Cardinality
Relationship1	Product	Wireless Router	NA	1
Relationship2	Class	NA	Wireless device accessory	1

Table 3–8 shows the relationship domain and quantity.

**Table 3–8 Relationship Domain**

Relationship Domain	Quantity
Samba Bluetooth Headset	X
B-Micro Bluetooth Headset	X

To update the relationship domain, you can add more product components to the product Bluetooth devices and all the items that have a class relationship with Bluetooth devices must be synchronized to Siebel CRM for the relationship domain to be updated. Updating the Bluetooth devices and synchronizing them is a prerequisite.

Siebel CRM has a relationship type attribute called domain type, and a DVM is created to map the domain type.

Any change in the relationship creates a new version of root item in Oracle Product Hub. The process integration updates the corresponding root item and the relationship in Siebel CRM.

---

---

**Note:** A BOM item with the same item catalog category or a simple item cannot be added as a component of any root item more than once for any parent-child relationship in Oracle Product Hub.

---

---

## Support for Controlling Auto-Release of Entities Published in the Same Batch

The Product Master Data Management integration enables the product administrator to control the automatic release of all the entities within the project workspace in Siebel CRM at a more granular level than the Siebel CRM system parameter. This enables different behaviors with regard to entity auto-release based on the publishing style and the needs of the various product administrators.

Oracle Product Hub passes a flag to allow a more granular control at the batch level for the release of all the entities within the project workspace. This flag specifies whether the entities within the project workspace should be auto-released, should not be auto-released, or should use the default behavior set in Siebel CRM with regard to auto-release.

- While Oracle Product Hub sets the flag at the batch level, Siebel CRM cannot use it at the batch level because within a batch, multiple Siebel CRM services are called for product and discounts, pricelist items, promotions, and structures. If the flag is set at service invocation level, inconsistencies may exist, for example, the service for product and discounts may succeed and all the product and discounts released in Siebel CRM, but the service for the associated structures may fail and the overall batch publishing would fail; however, the entities for product and discounts are released in Siebel CRM. Hence, the workspace cannot be released until all the entities in the batch that need to be synchronized are successfully synchronized.
- Oracle Product Hub also passes a workspace name. The Siebel CRM ABCS uses the workspace name received from Oracle Product Hub as the product workspace name. The workspace name allows creating all the entities from the same batch within the same workspace.
- Oracle Product Hub passes the batch auto-release flag and workspace name with each payload (item payload and BOM payload) at the header level. Values of these parameters are consistent in different payloads associated with the same batch ID.
- The batch auto-release flag is set to one of these values:
  - Y - indicates that the batch is released automatically
  - N - indicates that the batch is released manually
  - D - indicates that the release of the batch should be controlled by the Siebel CRM system parameter. This is the default.

Oracle Product Hub always publishes the batch auto-release flag with a value of *D*.

- Siebel CRM services for the creation of product and discounts, promotions, pricelist items, and structure take these at the header level:
  - Workspace name
  - Workspace reuse flag
  - Auto-release flag



If a workspace does not exist, Siebel CRM creates a new workspace; however, if it exists, the workspace reuse flag is set to *Y* and the workspace is reused. Otherwise, if the workspace exists and the workspace reuse flag is set to *N*, a new workspace is created the name of which is a concatenation of the value in the workspace name and a time stamp.

- When the first Siebel CRM service within a batch is invoked, which typically is the service for the creation of products and discounts:
  - The value passed from Oracle Product Hub in the workspace name is used for the Siebel CRM workspace name.
  - The value passed in the batch auto-release flag is used for the Siebel CRM auto-release flag (upon DVM conversion), and the Siebel CRM workspace reuse flag is set to *N*.
- When subsequent Siebel CRM services are invoked for the same batch:
  - The value passed in the batch name is used for the Siebel CRM workspace name.
  - The value passed in the batch auto-release flag is used for the Siebel CRM auto-release flag (upon DVM conversion), and the Siebel CRM workspace reuse flag is set to *Y*.
- Finally, when all the Siebel CRM services pertinent to the same batch have been invoked and if the batch has succeeded, one more service for the creation of product and discounts is invoked with an empty sequence of products and discounts. It has these values for the header attributes:
  - The workspace name is the value passed in the batch name.
  - The Workspace reuse flag is *Y*.
  - The Auto-release flag is *Y*.
- Within Siebel CRM, an unpublished workspace is not discarded, because administrators may want to access them for investigation. Periodic cleansing of unpublished workspaces is an administrator activity in Siebel CRM.

---



---

**Note:** The auto-release flag is always passed as *Default* from Oracle Product Hub.

---



---

## Synchronization of Promotions from Oracle Product Hub to Siebel CRM

Promotions are defined as items in Oracle Product Hub with entity type as promotion. The process integration creates product promotions for every such item in Siebel CRM.

Promotions have model BOM as the structure. The model BOM can have one or more items that have model BOMs as structure or can have items that represent promotion level discounts. The process integration adds the corresponding customizable products as components of the promotions in Siebel CRM.

- Promotions cannot have an option class BOM associated with the root item. The immediate components of the promotion can have a BOM of type *option class*. The items or components that are associated with BOMs of type *option class* are reused to create different promotions.
- Promotions support *n*-level nesting of BOMs. The typical value for *n* is between 1 (one) and 6 (six).

- In addition to the seeded UDA that is set for the item, these seeded user-defined attributes (UDAs) that are specific to items of entity type promotions are set during the promotion definition in Oracle Product Hub. See *Sellable Product Information Library* under *Telco Seeded Library Attributes* for more information.
  - Promotion: More Information
  - Charge Plan: Non Recurring Charge Details
  - Charge Plan: Recurring Charge Details
  - Charge Plan: Charges, Adjustment, Usage Plan Details
  - Commitment: Charges Credits
  - Product Promotions: Upgrade
- The process integration to Siebel CRM sets the corresponding fields of the promotion in Siebel CRM.
- Whenever a BOM is created for the promotion (root item) and the components are added, these seeded component UDAs are set during the promotion definition in Oracle Product Hub. See *Component UDA for Item Synchronization* under *Telco Seeded Library Attributes*.
  - Product Promotions: Components (Component UDA)
  - Product Promotions: Component Pricing (Component UDA)
  - Product Promotions: Pricing: Components: Adjustments (Component UDA)
- Once the promotions and all the associated BOMs and their components are successfully synchronized to Siebel CRM, the status is updated in Oracle Product Hub publication history. Even if one of the components or child components of the BOM within the promotion fails, or is partially successful, the status on the promotion item is *failed*.
- Adding, deleting, or updating the components that are directly associated with the promotions, or updating the fields of promotions such as promotion name, description, and so on, or updating the promotion-related information such as seeded and component UDA creates a new revision of the promotion in Oracle Product Hub. The new revision must be published to the downstream applications from the publication framework in Oracle Product Hub. For the deletion of components, the expectation is that the Oracle AIA configuration property synchronization flag be set to *Y*.

### **Populating Commitments on Sub-components: Level 3 Products**

Whenever there are level 3 products in a product, structure and data for commitments need to be populated. Access the Product Promotion components structure UDA and populate the commitments such as Grace Period, Grace period, UOM, ApplyComponentCharge, Commitment, Prorated Plan and so on in the override column instead of the New Column.

For example, M113 Options is an Option Class or a Class Aggregate as it is an immediate component of a Promotion which has one default product, P113 Voice Unlimited Service. For this Product P113 Voice Unlimited Service, you need to populate the Default Cardinality field in the Override Value column.

### **Populating the Default Products in a Promotion Aggregate**

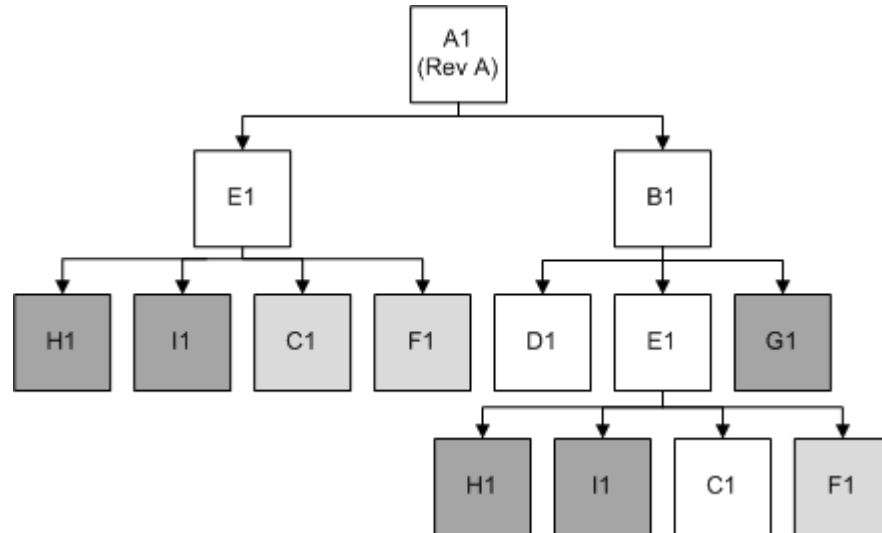
If a Promotion has Aggregate (Product Class/Product Line) and they have some products in the hierarchy, then populate the 'Default Cardinality' value in the override column of Version structure UDA of the corresponding products.

For example, if you need to populate the commitments, do that in the same column in the Product Promotions Components UDA which is above the Version Structure. This is for products under both the Bundle as well as Aggregates.

### Component Exclusion in Context of Promotions

Whenever a promotion is defined in Oracle Product Hub, some of the components of the promotion or subcomponents under the root item can be excluded. See [Figure 3–5](#) for an example of component exclusion.

**Figure 3–5 An Example of Component Exclusions**



Oracle Product Hub publishes the entire promotion definition with all the components, and wherever exclusions exist, it explicitly marks the exclusions by providing the path of exclusion from the root item (promotion).

The process integration to Siebel CRM handles the exclusions as follows:

- Siebel CRM provides a contextual framework to control the promotion definition. In Siebel CRM, the complete promotion definition with components at all levels can be updated in context of the promotion without affecting the original definition of the customizable products that are included in the promotion definition. The integration identifies the components that are excluded, which is specified in the payload that is published from Oracle Product Hub. Siebel CRM does not remove or delete the components of customizable products in context of the promotions. This is a documented Siebel CRM limitation.

The process integration uses the cardinality associated with the components to exclude the components in context of the promotion.

The promotion definition in Oracle Product Hub can specify the exclusions in these ways:

- Promotion definition can set the minimum cardinality and maximum cardinality of the component UDA to be 0 (zero).
- Promotion definition can mark the exclusions of the components.

**Note:** Siebel CRM provides the context-specific framework only in case of product promotions, not for customizable or component-type customizable products.

**Exclusions Within Option Class BOMs**

Table 3–9 explains the exclusions within option class BOMs.

**Table 3–9 Exclusions Within Option Class BOMs**

Level	Description
1	VoIP Limited Package
1.1	VoIP Core Bundle
1.1.1	VoIP Core Service
1.1.1.1	VoIP Service Options
1.1.1.1.1	Basic VoIP
1.1.1.1.2	VoIP w/unlimited calls
1.1.1.1.3	Fast Dialing
1.1.1.2	VoIP Voicemail
1.1.1.3	VoIP Caller ID
1.1.1.4	VoIP Fax Service
1.1.2	VoIP Adaptor
1.1.3	VoIP Phone
1.1.4	VoIP Laptop Phone
1.2	VoIP Recurring Discount

**Case 1: Exclude components of item that represents promotions**

The component items that are directly under the promotion are excluded by removal of the components.

These are the immediate components of the promotion VoIP Limited Package

1. VoIP Core Bundle
2. VoIP Recurring Discount

If any of these components have to be excluded, both the minimum and the maximum cardinality in the component UDA - Product Promotions: Components must be set to 0 (zero) in Oracle Product Hub and synchronized to Siebel CRM.

**Case 2: Exclude components of items that represent bundles within the promotion**

The component items that are not immediate children of the promotions but are subcomponents in the promotion definition are excluded by specifying the minimum and maximum cardinality as 0 (zero).

These are the subcomponents of the promotion VoIP Limited Package:

1. VoIP Core Service
2. VoIP Service Options

3. VoIP Voicemail
4. VoIP Caller ID
5. VoIP Fax Service
6. VoIP Adapter
7. VoIP Phone
8. VoIP Laptop Phone

The component items that are not immediate children of the promotions but are subcomponents in the promotion definition are excluded by right click on the component and exclude from current revision onwards in Product workbench of Oracle Product Hub. In Siebel CRM, the excluded component's Default quantity is blanked out.

### Case 3: Exclude component of option class items

The components under an option class item have to be excluded in Oracle Product Hub by means of the exclusion method provided in the product workbench. Default cardinality in the component UDA - Version: Structure must be specified in Oracle Product Hub for all the items that are not excluded.

These are the components of the option class item VoIP Service Options:

1. Basic VoIP
2. VoIP with unlimited calls
3. Fast Dialing

If one of the components of the option class has to be made optional in context of the promotion, the default cardinality must override to 0 (zero).

---



---

**Note:** If a component has to be excluded from the option class, right click on the Component and exclude from Current Revision onwards in the Product Workbench of Oracle Product Hub.

---



---

If an override is not specified to any of the components, none of the components are excluded and the default quantity in the target application is set to 0 (zero), thus making all the components available as optional items as part of the same promotion.

### Component Overrides in Context of Promotions

- The promotions contain various levels of BOMs, where the model BOM is the root BOM. The components association with the BOMs within the context of the promotion can be overridden. For example, the minimum cardinality set at the immediate parent item can be overridden in context of the promotion.
- This UDA is used to update the component association at every level of the BOM: Version: Structure (Component UDA).
- The process integration sets the component overrides in context of promotions in Siebel CRM.

### Transaction Attribute Value Exclusions in the Context of Promotions

- A valueset is associated with every transaction attribute of the item. In Oracle Product Hub, a subset of values from the set of values of the valueset can be excluded for the transaction attribute of a component in context of the promotion.

The metadata synchronization process synchronizes all the valuesets from Oracle Product Hub to Siebel CRM. See "[About the Process Flow for Synchronizing Valuesets](#)".

- Whenever the items are added as components of the BOMs of parent items or added as components of BOMs that are included in the promotions, the product administrator in Oracle Product Hub can exclude a subset of values from the valueset associated with the transaction attribute in context of the BOM or the root item. During promotion synchronization to Siebel CRM, Oracle Product Hub provides the values that were excluded from the valueset. The component that is marked in green represents the excluded value. This is marked C in the diagram depicting components exclusion.
- The process integration to Siebel CRM must set the values as *exclusions* for the corresponding attributes published from Oracle Product Hub in context of the promotion.

#### Promotion Based Discounts

The discounts on the components of the promotion are specified by means of these component UDAs:

- Product Promotions: Component Pricing (Component UDA)
- Product Promotions: Pricing: Components: Adjustments (Component UDA)

For more information about how to define promotion-based discounts in Oracle Product Hub, see the whitepaper "Guidelines and Methodology to Define and Manage Entities in Oracle Product Hub for Oracle Product Master Data Management Integration" in article ID 1086492.1 on My Oracle Support.

## Siebel CRM Interfaces

Siebel CRM interfaces used by the integration are:

#### Inbound Siebel CRM Web Services: Synchronize Items or BOMs

- Service name: SWIProductIntegrationIO
  - Operation name: SWIProductImportUpsert
  - Request and response: SWIProductImportUpsert\_Input, SWIProductImportUpsert\_Output

#### Inbound Siebel CRM Web Services: Synchronize Pricelist

- Service name: SWIISSPriceListItemIO
  - Operation name: SWIPriceListItemUpsert
  - Request and response: SWIISSPriceListItemIO.xsd

#### Inbound Siebel CRM Web Services: Synchronize Promotions

- Service name: SWIPromotionIntegrationIO
  - Operation name: SWIPromotionUpsert
  - Request and response: SWIPromotionUpsert\_Input, SWIPromotionUpsert\_Output

#### Inbound Siebel CRM Web Services: Synchronize Attribute Definition

- Service name: SWIAttributeIntegrationIO.wsdl

- Operation name: SWIAttributeImportUpsert
- Request and response: **SWIAttributeIntegrationIO.xsd**

#### **Inbound Siebel CRM Web Services: Synchronize Product Class**

- Service name: **SWIProductClassIntegrationIO.wsdl**
  - Operation name: SWIClassImportUpsert
  - Request and response: **SWIProductClassIntegrationIO.xsd**

#### **Inbound Siebel CRM Web Services: Synchronize Product Line**

- Service name - **SWIProductLine.wsdl**
  - Operation Name - SWIProductLine\_Synchronize
  - Request and response - **SWIProductLine.xsd**

#### **Outbound Siebel CRM Web Services**

There are no outbound events from Siebel CRM for the Oracle Product Hub integration.

For more information about Siebel CRM web services, see *CRM Web Services Reference*.

## **Siebel CRM Integration Services**

These are the integration services required for Siebel CRM to integrate with Oracle Product Hub:

- SyncProductSiebelProvABCImpl
- SyncBillOfMaterialsListSiebelProvABCImpl
- SyncClassificationSchemeListSiebelProvABCImpl
- SyncSpecificationValueSetListSiebelProvABCImpl
- ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl
- SyncItemCompositionListSiebelCommsProvABCImpl

### **SyncProductSiebelProvABCImpl**

This single operation service accepts a SyncItemListEBM product message as a request. It does the following on receiving the ItemEBM based on the entity type specified with the item:

For type promotion, it transforms this message into a Siebel CRM promotion ABM and invokes the Siebel CRM promotion ABM. For the type as product or discount, it is assumed to be of type product.

For type as option\_group it is assumed as Option Group. For discount\_model it is assumed as Discount Model and for bundle it is assumed as Commercial Bundle. In case of non-promotion items-based on the entity type, the transformation conditionally transforms the corresponding elements required to support that entity.

This service transforms the message into Siebel CRM product ABM. It invokes the Siebel CRM product web service to synchronize with Siebel CRM.

## SyncBillOfMaterialsListSiebelProvABCImpl

This single operation service accepts a SyncBillOfMaterialsListEBM product message as a request and returns SyncBillOfMaterialsListResponseEBM message. The BillOfMaterialsEBS invokes SyncBillOfMaterialsListSiebelProvABCImpl.

This service receives SyncBillOfMaterialsListEBM, transforms it into a Siebel CRM product ABM, and invokes the Siebel CRM product web service to synchronize the item structure with Siebel CRM.

When an item is defined as an option class in Oracle Product Hub, the provider services has two distinct behaviors:

- Synchronize option class as a product with the option components as relationships with domain type of components.
- Option class item can be established as a child with domain type of class. The service to establish the same as class is based on the values set for the component UDAs associated with the product. Domain type must be set to class, and the relationship name and class name must be same as item catalog category name that is pre-established in Siebel CRM as product class.
- The components under the option class item in Oracle Product Hub are set as relationship domain for the class type relationship.

## SyncClassificationSchemeListSiebelProvABCImpl

This single operation service accepts a SyncClassificationListEBM product message as a request.

The responsibility of this service is to receive SyncClassificationListEBM, transform it to Siebel CRM product class ABM, and invoke the Siebel CRM product-class web service to synchronize it into Siebel CRM. Once the product class synchronization is done in Siebel CRM, it passes the response message to ClassificationResponseEBS.

## SyncSpecificationValueSetListSiebelProvABCImpl

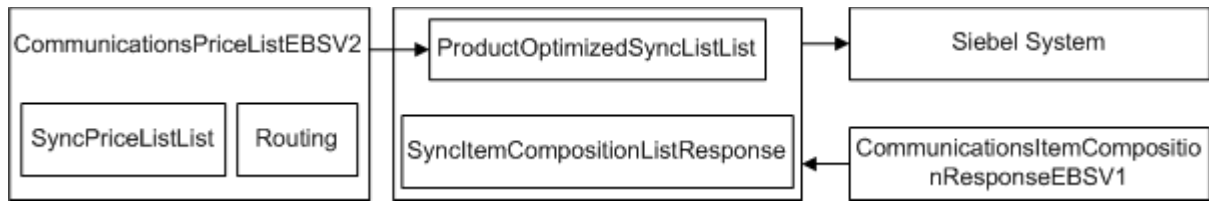
This single operation service accepts a SyncSpecificationValueSetListEBM product message as a request.

The responsibility of this service is to receive SyncSpecificationValueSetListEBM, transform it to Siebel CRM attribute definition ABM, and invoke the Siebel CRM attribute-definition web service to synchronize it into Siebel CRM. Once the attribute synchronization is done in Siebel CRM, it passes the response message to specification valueset response EBS.

## ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl

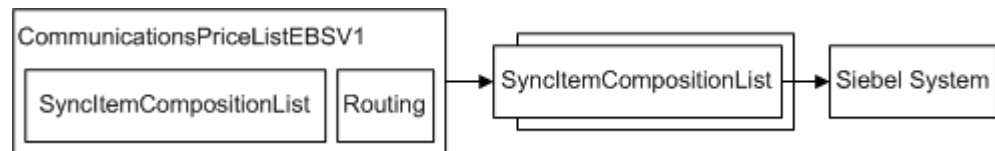
The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl service transforms the PriceList EBM into a Siebel CRM pricelist message and then calls the Siebel CRM pricelist web service on operation InsertOrUpdate. The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl transforms the PriceList EBM into a Siebel CRM product message and then calls the Siebel CRM product web service on operation Product\_spclmport\_spcUpdate. The Siebel CRM web service completes the request and returns a response message. ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl then transforms the Siebel CRM response message into the PriceList response EBM and sends it back to PriceListEBS.



**Figure 3–6 ProductOptimizedSyncPriceListListSiebelCommsProvABCSimpl**

### SyncItemCompositionListSiebelCommsProvABCSImpl

The `SyncItemCompositionListSiebelCommsProvABCSImpl` process accepts the `SyncItemCompositionListEBM`. It transforms `SyncItemCompositionListEBM` into the Siebel CRM product ABM. It then invokes the Siebel Product web service to create products and product structures in Siebel CRM.

**Figure 3–7 SyncItemCompositionListSiebelCommsProvABCSImpl**

## Assumptions and Constraints for the Siebel CRM Option

The assumptions and constraints for the Siebel CRM option are:

1. The Item with BOM item type = "Option Class" and its component items must have the same item catalog category.
2. Changes to the seeded static attribute groups are not supported. If you want to introduce new static attributes, you must define new attribute groups and static attributes provided that the internal name of the custom static attribute have a prefix of 'XX\_'. The integration creates these as flex attributes in Siebel CRM. If you introduce new valuesets in Oracle Product Hub for new UDA, the integration passes the information as free-form text.
3. The items and BOMs in the structure of an item catalog category must be published before the item catalog category is published.
4. The item catalog categories that are associated with the items must be synchronized to Siebel CRM before the items are published.
5. Every operating unit in Oracle Product Hub that has an associated order-entry item-validation organization needs to be defined as a business unit in Siebel CRM and the item validation organization defined as the inventory location associated with the business unit in Siebel CRM. This is required even if the operating unit and inventory organization role is associated with the same organization. For more information about establishing cross-references, see ["Setting Up Cross-References"](#). (This is applicable only if the Order to Cash: Siebel CRM - EBS integration is being planned or implemented with Product Master Data Management.)
6. The cross-references for inventory organization for item validation organization are set only for Siebel CRM and only those items from an item validation group that are associated with a single operating unit can be synchronized to Siebel CRM. (This is applicable if the Order to Cash: Siebel CRM - EBS integration is

being planned or implemented with or without Product Master Data Management).

7. Oracle Product Hub produces multiple BOM revisions, but it can identify whether the component item of the BOM is part of the current revision or a future revision of the BOM so that Siebel CRM can filter out the item components that are not part of the current BOM structure. Siebel CRM has a limitation that it uses only the current BOM revision.
8. The integration supports multi-tier pricing between Oracle Product Hub and Siebel CRM and product prices with different effective dates. One Rate plan can have multiple pricing with different non-overlapping effective dates.
9. For Siebel CRM, these entities and associated subcomponents need to be setup manually in Oracle Product Hub:
  - Proration plans for items of entity type of promotion.
  - Compatibility rule matrix.
  - Volume discounts.

For Siebel CRM, these entities and associated subcomponents need to be enriched post synchronization:

- Eligibility rules.
  - Configurator Rules
  - Sales Catalog
  - Volume discounts.
10. Once the products are successfully synchronized from Oracle Product Hub to Siebel CRM, they can also be updated to enrich the product definition. The existing products (subscription) that have references to pending quotes, orders, or assets in Siebel CRM or Oracle Communications Billing and Revenue Management (BRM) cannot be enabled as simple-service bundles. Similarly, the simple service bundles cannot be changed into subscription based billing products because by doing this, it impacts the existing asset cross-references. This is enforced by using the Siebel CRM validation only in the UI.

A mechanism or methodology must be defined for a product administrator in Oracle Product Hub to identify a given set of products that have pending quotes, orders, or assets in Siebel CRM (Order capture application). Once identified, the corresponding products must not be updated and synchronized to Siebel CRM unless the validation is added in the Siebel CRM application.

11. Pricelist:
  - A price list contains multiple product offerings and a product can appear in multiple price lists.
  - A Siebel CRM price list contains prices for all product offerings in one currency.
  - Changes to pricing either the current or future effectivity is done within the same price list
  - A single product definition is created for a specific asset (for example, handset) and separate one-time prices are defined in different price lists. For example, Consumer Price list and Business Price list. This helps in creating or using price lists for consumer, creating or using price list for business and

associating single product definition of asset to each of the pricelist and define prices in price lists.

- A single product definition is created for the monthly tariff and separate recurring prices are defined in different price lists. For example, Consumer Price list and Business Price list. This helps in creating or using Price list for consumer, creating or using price list for business and associating single product definition of asset to each of the pricelist and define prices in Price lists.
- Publishing the product definition from Oracle Product Hub:
  - Synchronizes the product definition and the different pricing information in the consumer and business price lists in Siebel CRM. Future date effectivity prices must be added as new pricelist line items within the existing price list.
  - Synchronizes the product definition and the different pricing information for the consumer and business users as separate rate plans in BRM. For example, Consumer Pricelist and Business Pricelist. To track the relationship of rate plans with the pricelist, separate rate plans and rate plan selector definitions are created in BRM. Future date effectivity prices must be added as new rate tiers within the existing rate plan.

## Multiple Price List Methodology

This section details the methodology of multiple price lists having Oracle Product Hub/BRM as its product master.

### About Price Lists and Rate Plans

In Siebel CRM, a price list is a set of standard prices for products and services. You can use multiple price lists to offer separate prices for the same product and you can specify a default price list. The price list specifies a price, the currency for that price, and the frequency with which the price is charged.

For example, you can use separate price lists to charge business customers US\$30 a month for internet service and to charge residential customers US\$50 a month for the same service. In this example, the residential price list specifies that the price is 30, the currency is US dollars, and the frequency is monthly and the business price list specifies that the price is 50, the currency is US dollars, and the frequency is monthly.

You can use multiple price lists to offer different prices in different market segments (such as consumer or business customers, as in the previous example), different currencies, different sales channels (such as products purchased online or at a store), or different geographic locations.

Siebel CRM price lists map to rate plans in Oracle Product Hub/BRM. You create the price lists in Siebel CRM and set up the mapping between price lists and rate plans in the PRICELIST domain value map (DVM) before creating products in Oracle Product Hub/BRM.

While creating products in Oracle Product Hub/BRM, you define rate plans to specify what to charge for the products. You associate the rate plans with corresponding price lists configured in Siebel CRM so that the integration can determine where Siebel CRM tracks charges.

---

---

**Note:** BRM also has a price list entity, but this is different from the Siebel CRM price list. When this document refers to price lists, it is referring to the Siebel CRM entity.

---

---

## Creating or Updating Rate Plans in Oracle Product Hub/BRM

The product master (Oracle Product Hub/BRM) stores the relationship between a rate plan and the pricelist. The integration uses this relationship to create or update pricing information in Siebel CRM.

- In Oracle Product Hub/BRM, for multiple Rate Plans the Rate Plan Selector is used to define the relationship between a Rate Plan and the Pricelist.
- To define price lists, you can either choose Standard pricing model or Billing pricing model. In Oracle Product Hub, the Rate Plan attribute group is used for Pricing Code Billing and Pricing:Simple Price List attribute group for Pricing Code Standard is used to define the relationship between a Rate Plan and the Pricelist.
- When no pricelist is explicitly specified, the configured default pricelist is used.

## Associating Rate Plans in Oracle Product Hub/BRM with Siebel CRM Price Lists

You associate rate plans in Oracle Product Hub/BRM with Siebel CRM price lists in Pricing Center using a rate plan selector.

To associate a rate plan with a price list using a rate plan selector:

1. In Pricing Center, follow the steps for defining rate plan selectors described in the Pricing Center Help.
2. When setting up columns for your rate plan selector, create a column called **EVENT.PIN\_FLD\_USAGE\_TYPE**.
3. Add a row for each rate plan and corresponding price list that you intend to use.
4. In the **EVENT.PIN\_FLD\_USAGE\_TYPE** column:
  - To associate a rate plan with a specific price list, enter the name of the price list exactly as it appears in the **PRICELIST DVM**.  
  
If you enter a name that does not appear in the **PRICELIST DVM**, an error will occur when you synchronize the products to Siebel CRM.
  - To associate a rate plan with the default price list, enter \* in the place of a price list name. The integration maps \* to the default price list. See [Table 3-11](#) for an example.
5. In the Rate Plan column, enter the name of the rate plans that correspond to the price lists that you entered in the **EVENT.PIN\_FLD\_USAGE\_TYPE** column.
6. Finish defining the rate plan selector as described in the Pricing Center Help.

### Example Rate Plan Structures

In this example:

- Two products have been synchronized from BRM to Siebel CRM: **Broadband** and **GSM**.
- A default price list has been set up in Siebel CRM, and entered into the **AIAConfigurationProperties.xml** file and the **PRICELIST DVM**, as described in "[Setting Up Cross-References](#)".

- Four additional price lists have been set up in Siebel CRM and entered into the PRICELIST DVM: ConsumerPL, BusinessPL, NewYorkPL, and CaliforniaPL.
- Five rate plans have been set up in Pricing Center: ConsumerRP, BusinessRP, NewYorkRP, CaliforniaRP, and StatesRP.

Table 3–10 shows the rate plan structure for the Broadband product. For this product, the product administrator uses two price lists to offer different prices for consumer and business customers.

**Table 3–10 Example Rate Plan Structure for Broadband Products**

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
ConsumerRP	ConsumerPL	1	01/01/2013	12/31/2013	\$40
BusinessRP	BusinessPL	1	01/01/2013	12/31/2013	\$30

Table 3–11 shows the rate plan structure for the GSM product. For this product, the pricing administrator uses the NewYorkPL and CaliforniaPL price lists to offer different prices for customers in New York and California and the default price list for customers in all other states. To make the integration use the default price list, the product administrator enters \* for the price list associated with the StatesRP rate plan.

**Table 3–11 Example Rate Plan Structure for GSM Products**

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
NewYorkRP	NewYorkPL	1	01/01/2013	12/31/2013	\$45
CaliforniaRP	CaliforniaPL	1	01/01/2013	12/31/2013	\$40
StatesRP	-	1	01/01/2013	12/31/2013	\$35

In Siebel CRM, the Broadband product is mapped to price list line items under the ConsumerPL and BusinessPL price lists and the GSM product is mapped to price list line items under the NewYorkPL, CaliforniaPL, and default price lists.

## Offering a Product in Multiple Currencies

To offer a product in multiple currencies:

1. In Siebel CRM, create price lists as described in "[Setting Up Cross-References](#)", and enter them in the PRICELIST DVM. Create a separate price list for each currency.
2. In Pricing Center, create rate plans that use the same currencies as the price lists in the PRICELIST DVM.
3. Define a rate plan selector for your product, associating the rate plans in the rate plan selector with the Siebel CRM price lists that use the corresponding currency. You must ensure that the currency in the rate plans matches the currency in the associated price lists. Currency matching is not validated by Siebel CRM or BRM.
4. Finish defining the rate plan selector and product as described in the Pricing Center Help.
5. Commit the product to the Oracle Product Hub/BRM database so that it is synchronized to Siebel CRM.

### Example of Offering a Product in Multiple Currencies

To offer a product called Broadband in Canadian dollars and U.S. dollars, the BRM product administrator uses a separate rate plan associated with a separate price list for each currency while creating the product.

In this example:

- A default price list has been set up in Siebel CRM and entered into the **AIAConfigurationProperties.xml** file and the PRICELIST DVM.
- Two additional price lists have been set up in Siebel CRM and entered into the PRICELIST DVM: CanadaPL and USAPL. The currency for the CanadaPL price list is Canadian Dollars (CAD) and the currency for the USAPL price list is U.S. dollars (USD).
- Two rate plans have been set up in Pricing Center: CanadaRP and USARP. The currency for the CanadaRP rate plan is Canadian dollars (CDN\$) and the currency for the USARP rate plan is U.S. dollars (US\$).

The product administrator uses the rate plan structure shown in [Table 3–12](#) when creating the product.

**Table 3–12 Offering the Broadband Product in Multiple Currencies**

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
CanadaRP	CanadaPL	1	01/01/2013	12/31/2013	CDN\$30
USARP	USAPL	1	01/01/2013	12/31/2013	US\$35

When the product administrator commits the Broadband product to the BRM database to synchronize it to Siebel CRM, the Broadband product is mapped to price list line items under the CanadaPL and USAPL price lists.

## Managing Pricing in Rate Plans and Price Lists

After you have synchronized your products from Oracle Product Hub/BRM to Siebel CRM, you can manage the prices in the rate plans in Oracle Product Hub/BRM and resynchronize the products to Siebel CRM to update the price lists. You can manage prices by changing a product from using multiple price lists to using the single default price list.

### Changing a Product from Multiple Price Lists to a Single Price List

To change a product in Oracle Product Hub/BRM that has already been synchronized to Siebel CRM from using multiple price lists to using the default price list:

- If the one of the rate plans in the rate plan selector is associated with the default price list (uses \* in the EVENT.PIN\_FLD\_USAGE\_TYPE column):

1. In Pricing Center, set the duration end dates to the current day for all of the rate plans for the product associated with non-default price lists. Leave the rate plan associated with the default price list as is.

See the discussion of defining the duration of a rate in the Pricing Center Help for more information about setting the duration end date.

2. Commit the product to the Oracle Product Hub/BRM database so that the product is resynchronized to Siebel CRM.

- If none of the rate plans in the rate plan selector are associated with the default price list:
  1. In Pricing Center, set the duration end dates to the current day for all of the rate plans associated with the product.  
See the discussion of defining the duration of a rate in the Pricing Center Help for more information about setting the duration end date.
  2. Commit the product to the Oracle Product Hub/BRM database so that the product is resynchronized to Siebel CRM.
  3. Under the Rate Plan Structure column for the product, select **Single Rate Plan**.
  4. Commit the product to the Oracle Product Hub/BRM database so that the product is resynchronized to Siebel CRM. The integration automatically associates the single rate plan structure with the default Siebel CRM price list.

### Examples of Changing a Product from Multiple Price Lists to a Single Price List

To change the GSM product with the rate plan structure shown in [Table 3–11](#) from using multiple price lists to using a single price list (the default price list), the Oracle Product Hub/BRM product administrator uses Pricing Center to edit the rate plan selector. The product administrator does the following:

- Changes the end dates of the NewYorkRP and CaliforniaRP rate plans to the current date. See [Table 3–13](#).

**Table 3–13** *Setting the End Date for Rate Plans for the GSM Product*

Rate Plan Name	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
NewYorkRP	NewYorkPL	1	01/01/2013	01/31/2013	\$45
CaliforniaRP	CaliforniaPL	1	01/01/2013	01/31/2013	\$40
StatesRP	-	1	01/01/2013	01/31/2013	\$35

- Commits the product to the Oracle Product Hub/BRM database to resynchronize the product to Siebel CRM and update the effectivity dates for the price list line items.

Setting the end date for the rate plans not associated with the default price list means that the integration only uses the default price list and StatesRP rate plan for that product.

To change the Broadband with the rate plan structure shown in [Table 3–11](#) from using multiple price lists to using a single price list (the default price list), the Oracle Product Hub/BRM product administrator uses Pricing Center to edit the rate plan selector. The product administrator does the following:

- Changes the end dates of the ConsumerRP and Business RP rate plans to the current date. See [Table 3–14](#).

**Table 3–14** *Setting End Date for Rate Plans for the Broadband Product*

Rate Plan Names	Price List Associated with the Rate Plan	Tier	Start Date	End Date	Monthly Cycle Forward Fee
ConsumerRP	ConsumerPL	1	01/01/2013	01/31/2013	\$40
BusinessRP	BusinessPL	1	01/01/2013	01/31/2013	\$30

- Commits the product to the Oracle Product Hub/BRM database to resynchronize the product to Siebel CRM and update the effectivity dates for the price list line items.
- Selects **Single Rate Plan** under the Rate Plan Structure column for the Broadband product.
- Commits the product to the Oracle Product Hub/BRM database to resynchronize the product to Siebel CRM.

Changing the rate plan structure to **Single Rate Plan** means that no price list is associated with the rate plan in Oracle Product Hub/BRM. The integration automatically associates this rate plan structure with the default Siebel CRM price list and maps the Broadband product to price list line items under the default price list.

### Support for Effectivity

The price lists effectivity period can be managed by changing the start and end dates on a rate plan. Any extensions can be possible by creating the same pricelist with new effective dates.

Similarly, the change in pricelist name can be achieved by setting the end date on incorrect Pricelist and recreating a new rate plan, map it to the new pricelist/tier group/rate data and balance impact.

## Synchronizing Sponsorship Items from Oracle Product Hub to Siebel CRM

Sharing groups are groups of services that share benefits. Sponsorship items are used to model splitting the cost of service. For example, using sharing groups, a company can pay for some of its employees' mobile phone usage and offer a discounted rate for calls between employees.

One of the use case for Sharing groups is for Charge sharing. Sponsorship objects represent the charge share object that defines the rules for sharing the charges. In Oracle Product Hub, you need to define the Sponsorship object and then synchronize to Siebel CRM, Pricing Design Center and BRM. After synchronization sponsorship item to Siebel CRM, further modelling of Sharing Groups is required. This includes defining Promotion Group and including sponsorship item in the promotion group. After synchronizing sponsorship item to Pricing Design Center and BRM, further enrichment is required to define the charge sharing model. Oracle Product Hub supports definition of Sponsorship and discounts objects. However, promotions groups are not supported.

The Oracle AIA Master Data Management Product pre-built integration interfaces between Oracle Product Hub, Siebel CRM, and BRM to process sharing group requirements using a connector. It maps the sharing group product and discount offerings between Oracle Product Hub, Pricing Design Center and BRM and also creates or updates Oracle AIA cross-references.

### Sharing Group Components

Each sharing group definition consists of associated components. You need to define these components in Oracle Product Hub and map to Siebel CRM, BRM, and Pricing Design Center. These components include:

- Sponsorships
- Discounts
- Members



- **Promotion Groups**

During run time, the owner of a sharing group needs to create the group at the owner account level. Members and services are added to the group by the owner. To avail the benefits of the sharing group, members need to subscribe to the sharing groups published by the owner.

## **Sponsorship**

Sponsorships are defined in Oracle Product Hub as items and mapped to billing type items of chargeshare in Siebel CRM. Chargeshare model have to be predefined in BRM. Sponsorship items are used to define rewards in promotion groups in Siebel CRM.

After a sharing group request is published from Oracle Product Hub, Oracle AIA looks at the billing service type to determine the type of sharing group to create in BRM. When a service creation plus a promotion group creation request comes in, Oracle AIA first creates the service and then the sharing group in BRM. During the initiate billing phase, the Oracle AIA BRM provider provisions the promotion group line items after all services have been sent to billing.

## **Discounts**

Discounts and chargeshares are created in BRM and special rating products created in Siebel CRM, added at design time as rewards to be shared by the group owner and members.

Discounts are created as items in Oracle Product Hub and synchronized to one or more Siebel CRM instances. The items and BOMs together with the seeded telecommunications library attributes are used to model various entities like products, discounts, bundles, promotions in Oracle Product Hub.

For more information on modeling the entities in Oracle Product Hub, see the whitepaper "Guidelines and Methodology to Define and Manage entities in Oracle Product Hub for Oracle Product Master Data Management Integration 2.5" Note ID - 1086492.1 on My Oracle Support.

For more information about the functional and technical details of this process flow, see "[BRM Interfaces](#)" and "[Synchronizing Discounts and Discount Models to Siebel CRM and BRM](#)".

## **Membership**

Member components are used to define items that represent members of a sharing group in Oracle Product Hub. Members are defined using the item type, Promotion Group Membership and maps to Product Type in Siebel CRM. Members can also be other accounts or assets. When synchronizing sharing groups, memberships are defined as service bundle or simple service bundle in Siebel CRM.

## **Promotion Groups**

Your customers can share charges, discounts, and special rating profiles by purchasing promotion groups. You create promotion groups at design time in Siebel CRM. Each promotion group definition includes an owner membership product, a member membership product, and one or more reward products.

For more information on how to model sharing groups using Siebel CRM promotion groups, refer to the Oracle® Application Integration Architecture Oracle Communications Order to Cash Integration Pack Implementation Guide for Siebel CRM, Oracle Communications Order and Service Management, and Oracle

Communications Billing and Revenue Management, Release 11.4, “Creating Promotion Groups”.

To process sharing group discounts, the Oracle AIA Pre-built Integration Pack integration does the following:

- Promotion group rewards with the billing service type of discount is purchased for the owner.
- Promotion group rewards with the billing service type of special rating Friends and Family ERA is created for the owner.
- Promotion group rewards with the billing service type of subscription is purchased for the owner.

You use APIs to create or update sponsorships in BRM when published from Oracle Product Hub to BRM.

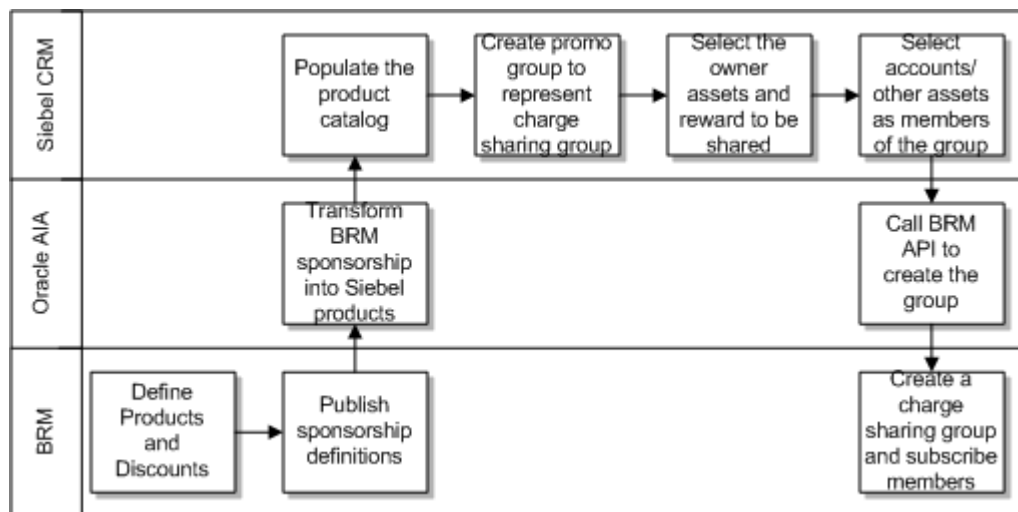
## Defining a Charge Sharing Group and Publishing to Siebel CRM

A charge sharing group allows the owner of the group to sponsor the charges of one or more members of the group. Members subscribe to the sharing group to avail sponsorship and discount benefits.

Charge sharing groups are used to determine how charges are shared among members. For example, an employer can sponsor the voice charges incurred by employees who are members of a charge sharing group.

This figure shows the high-level tasks to create a charge sharing group in Siebel CRM and how the integration is handled in Oracle AIA and BRM:

**Figure 3–8 Creating a Charge Sharing Group in Siebel CRM**



Here are the steps to define a charge sharing group in Oracle Product Hub and synchronize with Siebel CRM and BRM:

In Oracle Product Hub:

1. Define a sponsorship item type and specify the sponsorship attribute for chargeshare.
2. Map an event to the chargeshare model.
3. Create a Membership item type.

4. Create a Promotion Group item type.
5. Add one or more membership item to the promotion groups and define the cardinality of the membership products.
6. Add one or more products to each of the membership within the promotion groups.
7. Add one or more rewards products to the promotion groups and define the access control rules on the each of the membership within the promotion groups.
8. Define the promotion group validation rules such as the compatibility rules for promotion groups and membership in the context of the promotion groups.
9. Define the asset membership cardinality of the rewards products across promotion groups.
10. Associate the promotion groups, memberships and the rewards products to one of more pricelists.
11. Create or update the commitment terms and conditions for promotion groups and membership items.
12. Optionally, define pricing adjustments on the membership products, assets, and rewards in the promotion groups.
13. The promotion group, membership or rewards can be synchronized on their own or as part of the structure of the promotion group.
14. Publish and synchronize the following for promotion groups to Siebel CRM.
15. Publish and synchronize the Sponsorship Object from Oracle Product Hub to BRM/Pricing Design Center when Sponsorship object is published standalone or when it is published as part of the Promotion Group Structure.
16. Publish and synchronize the Membership Object from Oracle Product Hub to BRM/Pricing Design Center as billing product when there is a charge (recurring or onetime) for the member.

Oracle AIA synchronizes the sponsorship items from Oracle Product Hub to Siebel CRM. The product in Siebel CRM must indicate the billing type as sponsorship. Oracle AIA also synchronizes the sponsorship items from Oracle Product Hub to Pricing Design Center along with the charge share event map attributes and synchronizes the sponsorship items from BRM to Siebel CRM.

In Siebel CRM:

1. Map sponsorship item type to the billing type attribute in Siebel CRM.

In BRM:

1. Sync the sponsorship item to Siebel CRM using the billing item type.
2. Sync billing service type in BRM to these billing service types in Siebel CRM:
  - /account indicates that the charge sharing is at the account level.
  - /service/telco/gsm/telephony indicates charge sharing is for specific service.

## Defining a Charge Discount Group and Publishing to Siebel CRM

Discount sharing group is a way to share a discount or resource, such as free minutes between multiple accounts or services. A member is charged for usage, discounting credits on the member's account and free minutes are deducted from the owner.

For example, Joe receives a 10,000 free minute discount. Joe wants to share the free minutes with his family members. Joe is the promotion group owner who is charged a recurring or one-time fee or both. In this plan, a discount in the promotion group fee is offered as a reward for using the sharing group.

In Siebel CRM:

1. Create a promotion group and assign Joe as the owner.
2. Add the members to the promotion group.
3. Specify the membership domain.
4. Define the rewards.
5. Map the promotion group reward billing service type, Discount to BRM
6. sharing group type, Discount Sharing Group.

In BRM:

1. Map the billing type in Siebel CRM to Subscription or Item. The billing item is charged to the promotion group owner at account or service level. The fee product billing service type should match the sharing group owner service type. In Siebel CRM, these products are added as rewards.

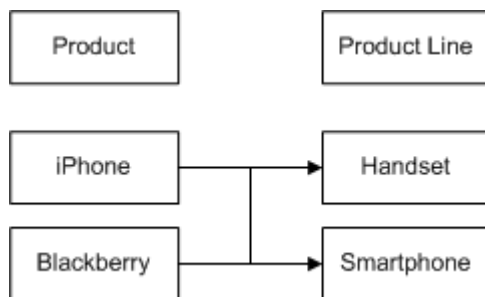
In this case, the reward for subscribing to the sharing group and paying the subscribing fee is 5GB data to be shared amongst the sharing group members.

When a service creation plus promotion group request comes in, Oracle AIA first creates the service and then sharing groups in BRM. During the billing phase, the Oracle AIA BRM provider processes the promotion group line items only after all services have been sent to billing.

## Synchronizing Multiple Product Lines from Oracle Product Hub to Siebel CRM

You can now set up multiple product lines for a product in Oracle Product Hub (Oracle Product Hub) and then synchronize this relationship from Oracle Product Hub to Siebel CRM. For example, you can set up an iPhone to belong to two product lines: mobile handsets and smart phones.

**Figure 3–9 Multiple Product Lines**



Here are the high-level steps to set up and synchronize multiple product lines for a product from Oracle Product Hub to Siebel CRM:

In Oracle Product Hub:

1. Define product line object in Oracle Product Hub with these attributes:
  1. Product Line (required)
  2. Product Line Description

3. Zero or more Product Line Managers
2. Product line managers can be mastered in Siebel CRM and then setup in Oracle Product Hub as value sets as part of post installation steps.
3. Associate product lines to entities such as billing products and discounts, promotions, and bundles in Oracle Product Hub. Specify an effective date.

In Oracle AIA when you publish a product line in Oracle Product Hub, the Product Master Data Management integration synchronizes the product line definition and the product line relationships with Siebel CRM and creates or appropriate cross reference for the product lines.

## Synchronizing Promotion Enhancements from Oracle Product Hub to Siebel CRM

Promotion aggregates allow you to group together products with common tariff plans. For example, you can group together handsets with common tariff plans.

Here are the high-level promotion aggregates integration steps between Oracle Product Hub and Siebel CRM:

In Oracle Product Hub:

1. Define a promotions product aggregate by product class.
2. Define one default product from a set of product options.
3. Specify one or more product line for a product offering.
4. Define promotion product aggregate by product line in Oracle Product Hub.
5. Specify a promotion adjustment by product line.

In Oracle AIA, synchronization occurs between a promotion and a product line or product class and promotion based aggregate pricing by product line or product class.

In Siebel CRM, web services are used to support integration of promotion aggregate definition and promotion pricing aggregate definition.

The Product Master Data Management Pre-built Integration Pack synchronizes the following promotion attributes from Oracle Product Hub to Siebel CRM:

- Splitting and merging promotion definition
- Promotions upgrade rules
- Action attributes
- Promotion upgrade attributes
- Commitment attributes of promotion components
- Commitment attributes of customizable product components
- Relationships between a promotion and a product line or product class
- Promotion based aggregate pricing by product line or product class

## Integration Services

The synchronize product and price business flow uses the following BRM interfaces:

- ProcessSalesOrderFulfillmentSiebelCommsReqABCImpl
- UpdateSalesOrderSiebelCommsProvABCImpl
- ProcessFulfillmentOrderBillingBRMCommsProvABCImpl

- ProcessFulfillmentOrderBillingBRMCommsAddSubProcess
- ProcessFulfillmentOrderBillingBRMCommsUpdateSubProcess
- ProcessFulfillmentOrderBillingBRMCommsDeleteSubProcess

## Cross References

The cross references for working with sharing groups are as follows:

- **BUNDLEDPROMOTION\_MEMBER\_ID.xref**: Used to maintain relationship between bundled promotion and its subcomponents.
- **PROMOTIONGROUP\_MEMBER\_ID.xref**: Used to maintain the relationship between the promotion group and its members.

## Assumptions and Constraints

The assumptions and constraints for working with the Oracle AIA Product Master Data Management pre-built integration connection between Oracle Product Hub, Siebel CRM, and BRM are as follows:

- Oracle AIA creates a new sharing group in BRM for each type of promotion group reward. Promotion group rewards are shared by all promotion group members.
- When synchronizing promotion group members to BRM, Oracle AIA directly associates them as SG members in BRM.
- Promotion Group can have only one owner at any point of time.
- When a promotion group ownership is changed on the Siebel CRM side, the system deletes the existing promotion group and adds new one.
- Suspend or resume functions are not available for promotion groups.
- If the promotion group owner is suspended, then all rewards are suspended for the promotion group members.
- Although there is no corresponding information in Siebel CRM SalesOrderABM, Oracle AIA populates the promotion group information for the promotion group header order line.
- The integration does not support initial load or batch load capability for item catalog categories. The recommended process is to publish the item catalog categories one at a time because this is performed design time.

## Product Master Data Management Integration Option for BRM

This chapter describes the Oracle Application Integration Architecture (Oracle AIA) Product Master Data Management integration option for Oracle Communications Billing and Revenue Management (BRM).

### Synchronization of Billing Products from Oracle Product Hub to BRM

Table 4–1 shows the types of billing products supported by the process integration.

**Note:** Although these examples are communications-specific, these are billing products that can be used, based on industry requirements.

**Table 4–1 Supported Billing Products**

Product Type	Applies To	Charge Type	Example	Comments
Item	Account	One-Time	Penalty for cancellation for the service.	Recurring charges or cycle-forward charges are not applicable for product type of item.
-	Service	One-Time	One-time service charges such as suspend service fee and resume service fee.	Recurring charges or cycle-forward charges are not applicable for product type of item.
Subscription	Account	One-Time	One-time purchase fee that is not related to the service being purchased.	NA
-	Account	Recurring	Service provider (SP) charges a monthly flat fee of \$10 for sending a hard copy of the invoice using FedEx.	NA
-	Service	One-Time	One-time fee related to the service being purchased. Activation or installation fee.	NA
-	Service	Recurring	Monthly charges associated with the service.	NA
-	Service	Usage	Charges are applied based on the usage generated in the billing application	NA

---

---

**Note:** BRM supports multi-tier rating to be associated with rate plans. The Oracle Product Hub pricing model in the seeded telecommunications library can be used to define multi-tier rating for an item. The process integration supports multi-tier rating between Oracle Product Hub and BRM; however, the integration to Siebel CRM does not support multi-tier rating. The canonical model (PriceListEBO) also supports multi-tier rating.

---

---

Every product is associated with events, which determine how much and how often to charge the customers. These events are called billable events. Products that have one billable event are called single-event product and products that have more than one billable event are called multi-event products in BRM. Every billable event is associated with the rate plan that defines the charge and the frequency of applying the charges.

The item name and item long description in Oracle Product Hub are used to define the billing product name and billing product description in BRM, respectively.

Oracle Product Hub provides a seeded UDA framework to define the billing products in Oracle Product Hub. These seeded attribute groups and attributes are used to define the billing product attributes in Oracle Product Hub:

- Additional Entity details
- Additional Billing Information
- Billing Products Attributes General
- Billing Products Event Map
- Rate Plan
- Rate Data
- Balance Impact
- Tier Group
- Day Time Range
- Days of the Week Range

## Multi-Event Product Synchronization from Oracle Product Hub to Siebel CRM

The products that have multiple charges and events are created as customizable products in Siebel CRM. The events are created as separate products and are added as child products of the main product. The component products represent the charges and events and have the billing type of event that is set in Siebel CRM. The item synchronization process from horizontal implementation supports both create and update of products in Siebel CRM.

To support create and update of pricelist, the process integration to Siebel CRM reuses the communications pricelist Siebel CRM-provider-connector services. The Oracle Product Hub requester item ABCS must create the corresponding item and pricelist line cross-references.

### Billing Products with Single Rate Plan

- Define the base-product-attribute values and define the rate plan. Ensure that the rate plan name is the same as the event name. Otherwise, the BRM service ends in



error. This indicates that a single rate plan has been associated with the charge of the billing product.

- For single rate plans, multi-tier rating can be defined in Oracle Product Hub using the seeded attribute framework. The process integration synchronizes the multi-tier rating from Oracle Product Hub to BRM. The Siebel CRM connector ignores the multi-tier rating and sets the price to 0 (zero). Mostly one-time and recurring charges have single-rate plans defined.
- For billing products with more than one charge, the pricing model in Oracle Product Hub provides multi-row attribute groups, which the product administrator must use to define multiple charges associated with the product.

### **Billing Products with Rate Plan Selector-Multiple Rate Plans**

- In case of BRM, multiple-rate plans can be associated by means of rate plan selectors or multiple single rate plans (custom event analysis).
- If rate plan selectors must be associated with the charges, then specify the rate-plan-selector name in the pricing model and leave the rate plan name as NULL.
  - In case of Multiple Rate Plans, associate each rate plan to a pricelist.
  - In case of multi-currency, create separate rate plans and associate it with different pricelists. Each pricelist represents prices of products in a single currency.
- If multiple single-rate plans (custom event analysis) must be associated with the charges, specify the rate plan names and do not specify the rate-plan-selector name in the pricing model. The rate plan names must be unique across events and charges.
- The charges for the billing products are defined by means of the UDA pricing model supported by Oracle Product Hub. The charges must be created as events in BRM and the corresponding prices must be associated with the events. Mostly one-time and recurring charges have multiple rate plans defined.

---



---

**Note:** The rate plan selector is defined in BRM. Subsequent updates made to the billing product in Oracle Product Hub and synchronized to BRM do not remove the rate plan selector enriched in BRM.

---



---

### **Billing Products with Usage Rating**

The billing products with usage rating can be defined as follows:

- A billing product where usage is the only event for the product.
- A billing product where usage is one of the many events associated with the product.

In Oracle Product Hub, only the name of the usage rate plan is defined and published to the target applications. BRM uses pipeline-based rating for usage charges. The usage-based rating is enriched in BRM. Only Delayed Telco GSM usage event/charge is supported as the product is delivered. You can add more usage type events and charges and define the corresponding usage rating in the target billing applications.

---



---

**Note:** The pipeline rate plan is defined in BRM. Subsequent updates made to the billing product in Oracle Product Hub and synchronized to BRM do not remove the pipeline rate plan enriched in BRM.

---



---

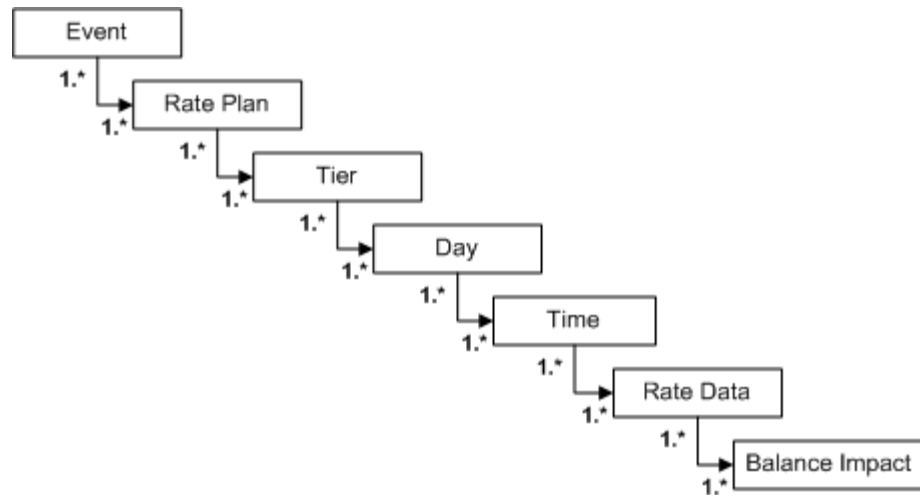
Table 4–2 summarizes the support offered for pricing in Oracle Product Hub.

**Table 4–2 Pricing in Oracle Product Hub**

Rating Type	Charge Type	Rating Method	Description
Real-Time Rating	One-time	Single Rate Plan	Complete definition in Oracle Product Hub. The rate plan can also be enriched in BRM. The rate plan name must not be changed in BRM because subsequent updates override the changes.
-	-	Rate Plan Selector	Oracle Product Hub sends the rate-plan-selector name and the rate plans are defined and enriched in BRM.
-	-	Custom event analysis	Complete definition in Oracle Product Hub. The rate plans can also be enriched in BRM. The rate plan names must not be changed in BRM because subsequent updates override the changes.
-	Recurring	Single Rate Plan	Complete definition in Oracle Product Hub. The rate plan can also be enriched in BRM. The rate plan name must not be changed in BRM because subsequent updates override the changes.
-	-	Rate Plan Selector	Oracle Product Hub sends the rate-plan-selector name and the rate plans are defined and enriched in BRM.
-	-	Custom event analysis	Complete definition in Oracle Product Hub. The rate plans can also be enriched in BRM. The rate plan names must not be changed in BRM because subsequent updates override the changes.
-	Usage	Single Rate Plan	Complete definition in Oracle Product Hub. The rate plan can also be enriched in BRM. The rate plan name must not be changed in BRM because subsequent updates override the changes.
-	-	Rate Plan Selector	Oracle Product Hub sends the rate-plan-selector name and the rate plans are defined and enriched in BRM.
-	-	Custom event analysis	Complete definition in Oracle Product Hub. The rate plans can also be enriched in BRM. The rate plan names must not be changed in BRM because subsequent updates override the changes.
Pipeline Rating	Usage	Pipeline rate plans	Oracle Product Hub sends only the rate plan name. Enrichment of pipeline usage rating is done in BRM. The enrichment done in BRM is not removed when the updates are synchronized.

### Pricing Hierarchy of BRM

Figure 4–1 depicts the real-time rating in BRM.

**Figure 4–1 Pricing Hierarchy**

The pricing hierarchy can be modeled in Oracle Product Hub by means of the seeded telecommunications library UDAs. The pricing hierarchy is defined in context of an item that represents the billing product and is published to the target BRM systems.

### One-Time and Penalty Charges

The one-time and penalty charges are defined as simple products (products with one event or charge with real-time single rate plan) in Oracle Product Hub. These have to be synchronized to all the billing instances because these are applied during MACD (move, add, change, delete) orders.

For more information about associating the one-time charges and penalty charges to the corresponding products in Siebel CRM, see the Siebel CRM documentation.

The one-time charges (suspend, resume, move, and so on) are modeled as service-level products and the penalty charges are modeled as account-level products. Some of the products of billing type as subscription (both single event and multi-event) are treated as service bundles. Ensure that the service instance-enabled flag is set for such products.

Service bundles represent products that contain other products and bundles. The billing service type for such products must be set to service bundle. For nonservice bundles, the billing type must not be set.

### Support for Multiple-Rate Tiers and Rate Tier Effectivity

Multiple-rate tiers can be defined for the billing products in Oracle Product Hub. These are synchronized to BRM and Siebel CRM. The multiple-rate tiers manifest as multiple pricelist lines on a product in Siebel CRM.

The effectivity on the rate tier can be absolute dates or relative dates based on the product purchase date. The integration from Oracle Product Hub to BRM supports both the absolute and relative effectivity.

The integration from Oracle Product Hub to Siebel CRM supports only absolute effectivity.

For more information about rate tiers and effectivity, see "[Support for Rate Tiers and Effectivity](#)".

For more information about synchronizing billing products and billing discount from Oracle Product Hub to BRM, see the Synchronization of Items and BOMs section in "[Product Master Data Management Integration Base Pack](#)".

For more information on modeling the billing products with examples in Oracle Product Hub, refer to the whitepaper "Guidelines and Methodology to define and manage entities in Oracle Product Hub for Oracle Product Master Data Management Integration" in article ID 1086492.1 on My Oracle Support.

For more information about Telco seeded Library attributes and their corresponding valuesets, see the discussion of seeded item metadata libraries in *Oracle Product Hub Implementation Guide*.

### **Time-Zone-Dependent and Time-Zone-Independent Fields**

Time-zone-dependent fields identify the absolute time when an event takes place or will take place. Such fields need to be translated to the time zone of the system or you to identify that specific time, for example, order submission, start date, and end date for a product or other entity, time of failure, and so on.

The time-zone-dependent fields are translated to UTC time zone by Oracle Product Hub requester. If Oracle Product Hub has captured the time as 10:00 AM PST, this is translated to 18:00 UTC. As all the time values are coming in UTC format, the BRM provider does not need to do further translation.

---

---

**Note:** In the Oracle Product Hub integration, most date and time fields published by Oracle Product Hub are translated by the Oracle Product Hub requester in UTC format; implying that they are time-zone-dependent.

---

---

Time-zone-independent fields are used to identify a time that should be used irrespective of the time zone in which you or system is. Typically, the fields that identify ranges fall in this category. For example, if a time range of 7 P.M. through 7 A.M. is set as *evening*, the time specified in this range should not be translated across systems that are in a different time zone because *evening* would always remain 7 P.M. through 7 A.M. irrespective of the time-zone differences.

The time-zone-independent fields are used as is by the Oracle Product Hub requester. If Oracle Product Hub has captured time as 10:00 AM PST, this is translated to 10:00 UTC (without adjusting the hour difference). The BRM provider identifies these as time-zone-independent values and does not translate these to UTC time zone. Instead, such values are converted to other data types (strings or numbers) by the BRM provider as requested by the BRM services.

Billing systems depend on these time-zone-independent ranges to apply special discounts, premium rates, or promotions in certain hours of the day, certain days of the week, or certain days of the month.

The time-zone-independent date and time fields are in these attribute groups:

- Day Time Range
- Days of the Week Range

## **Synchronizing Discounts and Discount Models to Siebel CRM and BRM**

This synchronization allows a centralized definition of common discount features in Oracle Product Hub so that discount entities are consistently defined across Siebel CRM and BRM. It allows modeling of discount models in Oracle Product Hub to allow

generalization and reuse of features used by multiple discounts and mapping to discount model entities in BRM.

Discount models are used in BRM to increase reusability, improve performances in calculating discounts, and facilitate maintenance of discounts. Discount models capture information such as the type and value of the discount that is applied.

The billing-discount event map is a conceptual entity that associates the discount with its discount model in the context of a billing event to which the discount is applied. Oracle Product Hub uses a simplified model and only two entities are captured: discounts and discount models.

Discounts and discount models are represented as items in Oracle Product Hub. The attribute entity type in the item entity is set to discount or discount model respectively. The billing-discount event map entity is modeled as a multi-row attribute UDA within the discount entity in Oracle Product Hub.

Table 4–3 shows the seeded attribute groups that are used to model billing discount in Oracle Product Hub.

**Table 4–3 Seeded Attributes**

Library	Attribute Group
Communications Services Billing Library - Vertical	Billing Discount Attributes, Billing Discount Event Map

For more information on the library and the attribute groups, see *Oracle Product Hub Implementation Guide*.

Each row of the multi-row UDAs captures this information:

- Event - The same discount may apply to multiple billable events. A single discount can have multiple events (purchase and cycle). Each event can have separate models, which identify the percentage or absolute discount for that event.
- SnowBall - A flag that indicates whether the discount/event combination is a snowball discount. A snowball discount allows distribution of group discounts to group members.
- StopDiscounting - This is an enumerator defining run-time behavior regarding when to stop giving the discount if the discount is inactive or canceled. Seeded values are:
  - Never
  - When Canceled
  - When Inactive
  - When Inactive or Canceled
- Discount Structure Type - This is an enumerator identifying the type of discount model associated with the product. Possible values are:
  - Discount model - Only one discount model can be associated with the event.
  - Discount model selector - More than one discount model can be selected from a selector.
- Discount Model Name - The name of the discount model or model selector associated with the discount for the specific event. The possible values are those returned by querying of the name of the items with the entity type as discount model.

- BRM does not provide a separate API/Service for the creation and update of discount models. Instead, it uses an implicit creation or update of discount models within the API/service provided for discounts. When the BRM service for discount is invoked, the service looks at the discount model name; if it does not exist, it creates.
- Only the discount model name is captured in Oracle Product Hub, which has minimal impact on the participating applications from the integration perspective.

---

**Note:** The discount model name is a free text field that needs to be entered manually in Oracle Product Hub. Currently, no cross-validation is implemented by Oracle Product Hub between this field and the name of the discount-model entity.

---

The name for the discount model should be descriptive because the product administrator in Oracle Product Hub can rely only on the name to assign the discount model to a discount. In Oracle Product Hub, the description field in the item can also be used to add additional information for the discount model; however, this information is limited to Oracle Product Hub and is not visible to the BRM administrator because no description field exists for the discount model in the BRM services.

Two alternative approaches are supported:

1. You can define a discount model in BRM and then create in Oracle Product Hub the corresponding item with the *discount model* as the value for entity type, making sure that the names match.
2. You can first create the discount model in Oracle Product Hub and then enrich in BRM through the synchronize discount flow. Discount models are manually configured in Oracle Product Hub and are not explicitly synchronized to BRM.

When you define the model first in BRM, the benefit is that a discount model is used only after it has been fully defined in BRM and thus no risk is involved in releasing a discount with a shell discount model.

For more information on modeling Billing discounts and for examples in Oracle Product Hub, refer to the whitepaper "Guidelines and Methodology to define and manage entities in Oracle Product Hub for Oracle Product Master Data Management Integration 2.5" Note ID - 1086492.1 on My Oracle Support.

## Viewing Publishing Status of Items

For more information about how to view publishing status of items, see the discussion of viewing the publishing history in the Oracle Product Information Management User's Guide.

## BRM Interfaces

The BRM integration for Oracle Product Hub uses this service: `PCM_OP_PRICE_SET_PRICE_LIST.xsd`

For more information about BRM services, see the BRM documentation.

## BRM Integration Services

These are the integration services required for BRM to integrate with Oracle Product Hub.

- SyncItemListBRMProvABCImpl
- SyncPriceListListBRMProvABCImpl

### SyncItemListBRMProvABCImpl

This single operation service receives SyncItemListEBM as a request from ItemEBS. This input is transformed to BRM Product ABM to invoke BRM PCM\_OP\_PRICE\_SET\_PRICE\_LIST web service. When the web service returns back to the BPEL with BRM IDs, cross-reference values are updated with the BRM identifiers.

### SyncPriceListListBRMProvABCImpl

Single operation service receives SyncPriceListListEBM as input from PriceListEBS. This input is transformed to BRM Product ABM and BRM adapter is invoked to update the price information for the products. When web service returns back to the BPEL with success notification, cross-reference values are updated with BRM identifiers. A response EBM is constructed and used to invoke PriceListResponseEBS.

## Assumptions and Constraints for Oracle Communications BRM Option

The assumptions and constraints for the BRM option are:

1. No explicit synchronization flow exists for discount models from Oracle Product Hub to BRM.
2. Multiple rate plans using rate plan selectors are not supported for item definition in Oracle Product Hub. When an item is associated with multiple rate plans mapped to different Pricelists in Oracle Product Hub, the integration automatically creates the rate plan selector in BRM to maintain the mapping between the Rate plan and the pricelist information in BRM.  
  
This implies that for real-time rating either a single rate plan or multiple single rate plans must be associated with one-time, recurring, and usage charges in Oracle Product Hub and rate plans for the rate plan selectors must be enriched in BRM.
3. The usage-based products are defined in Oracle Product Hub as separate products and are not defined as a part of the multi-event product. The pipeline rating for usage products is defined and associated in BRM.
4. Usage-based rating using pipeline rate plans cannot be defined in Oracle Product Hub and hence they are not supported by integration. They must be defined in BRM.
5. Enrichment for one-time and cycle charges must not change rate plan names in BRM.
6. The discount models are associated with the items that represent the discounts in Oracle Product Hub. The complete definition of the discount model is done in BRM. The discount model name must not be updated in BRM because the subsequent updates override the value.

7. Pricelists: For more information, see ["Product Master Data Management Integration Option for Siebel CRM"](#) and ["Assumptions and Constraints for the Siebel CRM Option"](#).



---

---

# Product Master Data Management Integration Option for Design Studio

This chapter provides an overview of the Oracle Product Master Data Management Integration Pack Option for Oracle Communications Design Studio, supported features, and assumptions and constraints, and specifically, describes the process flows, interfaces, and integration services required in this integration.

## Overview

The new integration, Oracle Product Master Data Management integration Pack Option for Oracle Communications for Design Studio enables you to query or import new or updated item catalog categories and Transaction Attributes from Oracle Product Hub into Oracle Communications Design Studio through Oracle Application Integration Architecture (Oracle AIA).

This new integration offers the following advantages if you are using Oracle Product Hub as their Product Master:

- You can directly query or import the item catalog categories and Transaction Attributes into Oracle Communications for Design Studio instead of routing them through Siebel CRM.
- Lowers total cost of ownership (TCO) if you want to query or import the item catalog categories and Transaction Attributes directly from Oracle Product Hub into Design Studio.

The mapping between the Oracle Product Hub item catalog category and product specification in Design Studio enables communication service providers (CSPs) to achieve maximum reuse of a fulfillment process for products that share the same classification, for example, Broadband.

---

---

**Note:** Oracle Rapid Offer and Order Delivery deployments that have Oracle Product Hub, Siebel CRM and Design Studio, have an option of importing the product class definitions into the Design Studio directly from Oracle Product Hub by using Oracle Product Master Data Management Integration Pack Option for Oracle Communications Design Studio or the existing Design Studio integration with Siebel CRM.

To understand the configuration differences between Siebel CRM and Oracle Product Hub integration with Design Studio, see ["QueryProdClassAndAttributesSCEReqCommsABCSImpl"](#).

---

---

---

---

**Note:** The Product Master Data Management: Design Studio integration option needs the Communications Order to Cash Pre-built Integration Pack to invoke the requester.

---

---

## Assumptions and Constraints

The assumptions and constraints in Design Studio integration option are:

- The Design Studio queries the item catalog categories from Oracle Product Hub. Any updates that you make to the product specification in Design Studio are not synchronized back to Oracle Product Hub.
- The Design Studio maintains the mapping between the Oracle Product Hub item catalog category and the product specification.
- Only the item catalog category and the associated attributes are imported into the Design Studio.
- The item catalog categories published to Design Studio must be published to Siebel CRM to support the existing Order fulfillment processes.
- The product administrator is responsible for notifying the service-mapping specialist about the addition or modification of item catalog categories.

## Oracle Communications Design Studio Setup

The Oracle Communications Design Studio is a graphical tool that provides a ready-to-use integration to Oracle Product Hub and imports Oracle Product Hub item catalog categories for order fulfillment process. CSPs can define order fulfillment processes, tasks, order templates, dependencies and change management policies easily and accurately.

For more information about Design Studio, see the Design Studio documentation.

## Oracle Product Hub Web Services

The Oracle AIA composites invoke the following web services:

- item catalog category service  
This web service is used to query item catalog category definitions.
- Value set service  
This web service is used to query value sets.

For more information about invoking these web services, see the discussion of web services for publishing in Oracle Product Hub Implementation Guide.

## Integration Services

The integration option uses the following integration services:

- QueryClassificationListPIMCommsProvABCSImpl
- QuerySpecificationValueSetListPIMCommsProvABCSImpl
- QueryProdClassAndAttributesSCEReqCommsABCSImpl

## QueryClassificationListPIMCommsProvABCImpl

The QueryClassificationListPIMCommsABCImpl is a synchronous BPEL process and is the Oracle Product Hub provider Application Business Connector Services (ABCS) implementation and performs item catalog category related queries. This service follows all of the standards of ABCS provider. This service has one operation: QueryClassificationList. This accepts a QueryClassificationListEBM as a request and returns QueryClassificationListResponseEBM as response.

## QuerySpecificationValueSetListPIMCommsProvABCImpl

The QuerySpecificationValueSetListPIMCommsABCImpl is a synchronous BPEL process and is the Oracle Product Hub provider ABCS implementation and performs Valueset related queries. This service follows all of the standards of ABCS provider. This service has one operation: QuerySpecificationValueSetList. This accepts a QuerySpecificationValueSetListEBM as a request and returns QuerySpecificationValueSetListResponseEBM as response.

## QueryProdClassAndAttributesSCEReqCommsABCImpl

The QueryProdClassAndAttributesSCEReqCommsABCImpl is a synchronous BPEL process and is the Design Studio requester ABCS implementation and queries item catalog categories (or product classes in Siebel CRM) from Oracle Product Hub.

To retrieve product class data from the Product Master, you can either select Siebel CRM or Oracle Product Hub as their edge application. In a ready-to-use integration, Oracle AIA is configured to retrieve product class data from Siebel CRM.

The process to query item catalog categories (or product classes in Siebel CRM) depends on the **AIAConfigurationProperties.xml** file settings set by you. For example, to retrieve item catalog categories from Oracle Product Hub, you must modify the settings in the **AIAConfigurationProperties.xml** file to set the Routing.EdgeApplications property of composite, QueryProductClassAndAttributesSCEReqCommsReqABCImpl to System ID of Oracle Product Hub (in the **AIAConfigurationProperties.xml** file, change <Property name="Routing.EdgeApplications">SEBL\_01</Property> to <Property name="Routing.EdgeApplications">PIM\_01</Property>) and update it to the MDS repository.

This service follows all the standards of a requester ABCS. This service has one operation: QueryProdClassAndAttributes. This accepts a QueryClassificationListEBM as a request and returns QueryProdClassAndAttributesRespABM as a response.

## Understanding the Query Item Catalog Categories Business Flow

Product administrators define Item Catalog Categories and Transaction Attributes in Oracle Product Hub. The Oracle Product Hub administrator publishes the item catalog categories that sends a notification to the product-to-service mapping specialist. The product-to-service mapping specialist queries and imports the item catalog category and the associated Transaction Attributes to create or update a cartridge.

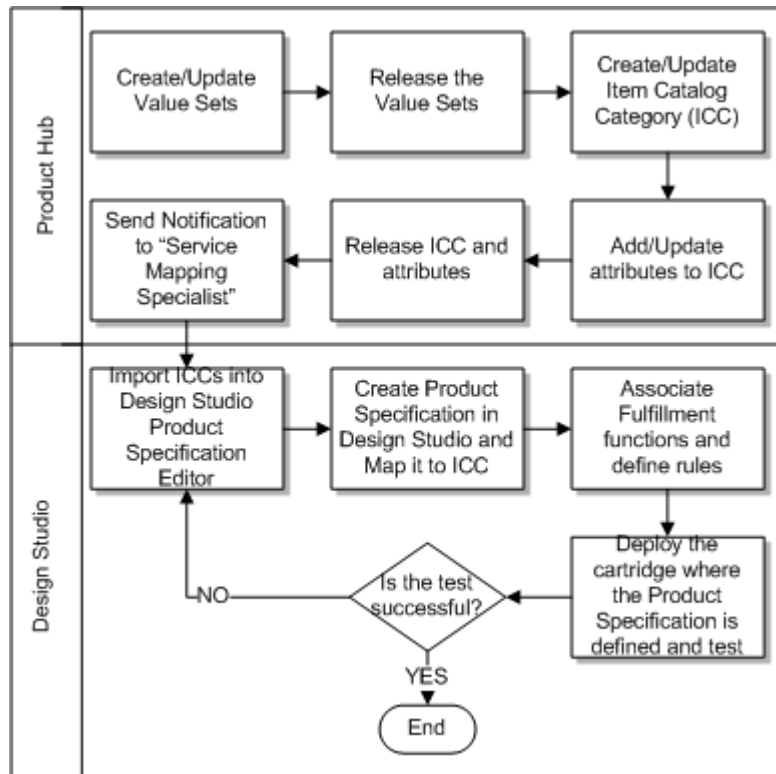
A cartridge in Design Studio is a jar (Java Archive) file that includes product specifications, orchestration of a product, item catalog category (or product classes in Siebel CRM) -to-product-specification mapping rules and so on that helps you meet their business needs and support ready-to-use features.

The product-to-service mapping specialist then maps the item catalog category to a product specification in the cartridge. The product-to-service mapping specialist uses

the product specification to associate decomposition rules, fulfillment functions, and their dependencies. After all the design time setup is complete, the product-to-service mapping specialist deploys the cartridge to the Oracle Order and Service Management (OSM) environments. The cartridge defines various fulfillment topologies to process order lines during order processing.

The [Figure 5–1](#) illustrates the planned business process flow.

**Figure 5–1 Business Process Flow**



The product administrator performs the following operations in the Oracle Product Hub application:

1. Creates a new or updates the existing valueset definition and generates a newer version of the valueset. A valueset is required for Order-time attribute that has a static list of values.
2. Creates or updates an item catalog category definition by adding, deleting or updating the transaction attribute definition.
3. Releases a newer version of the item catalog category.
4. Updates the attribute definitions (value sets) by adding or removing values from them. The product administrator can update the attribute value set associated with the attribute definition in the Oracle Product Hub application. This action automatically updates all the item catalog category and their subclasses.
5. The product administrator selects an item catalog category that is associated with the updated attribute definition and sends the notification.

In Design Studio, the product administrator can notify the product-to-service mapping specialist in one of the following ways:

- E-mail notification.

The product administrator can send template based e-mail notifications through E-Business Suite workflow, and can keep track of e-mail notification history.

To receive an e-mail notification from Oracle Product Hub, you must define a subscription for the business event `oracle.apps.ego.item.publishItemCatalogCategory`. For more information about how to setup e-mail notifications, see knowledge base article, MOS Doc ID 1491286.1, "Design Studio integration with Oracle Product Hub" in My Oracle Support.

---

---

**Note:** Email notification provides better traceability than company's email.

---

---

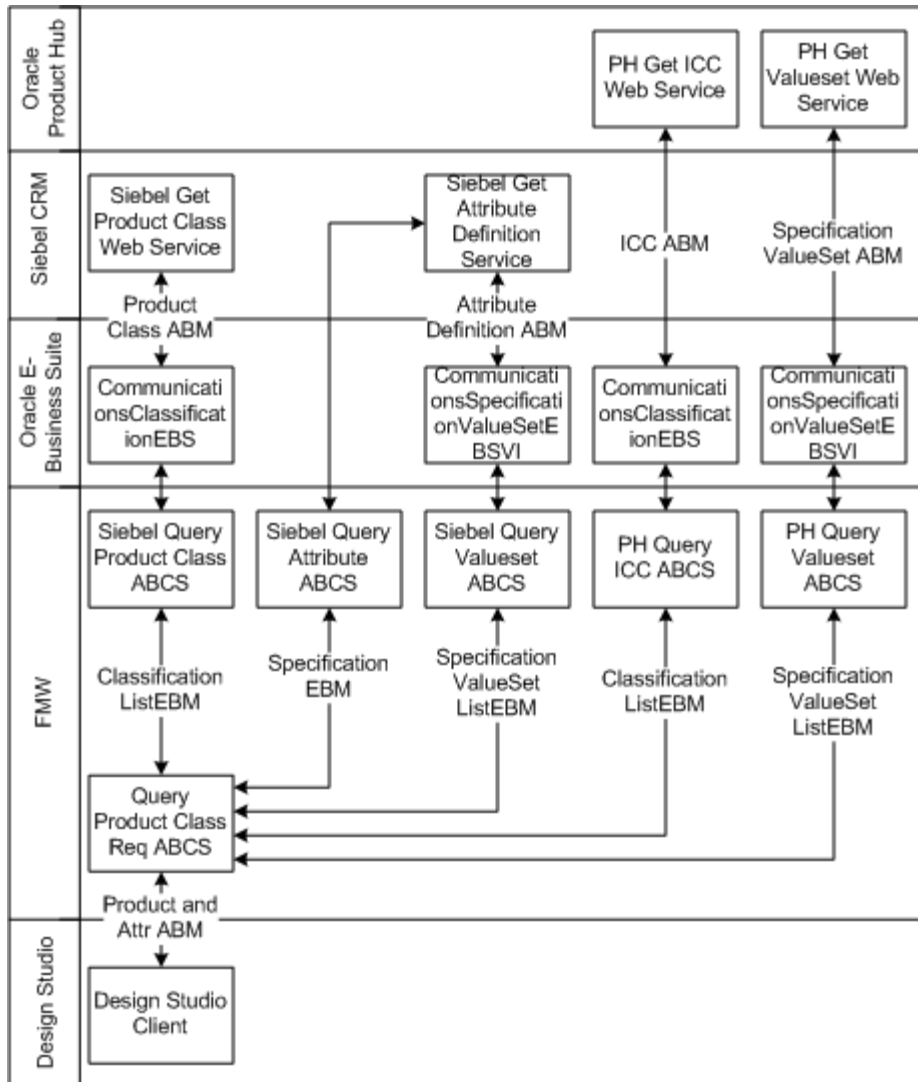
- Use a company's e-mail application  
The product administrator must manually create e-mails with correct information. This method is not dependent on a template and does not require additional configuration steps.
- 6. The product-to-service mapping specialist opens the Design Studio and queries the item catalog category and the Transaction Attribute details from Oracle Product Hub into a cartridge. The import process uses the item catalog category name. The Design Studio also offers the flexibility to query all the classes in the hierarchy associated with a item catalog category.
- 7. After the item catalog categories are successfully imported into a cartridge, all the Transaction Attributes and the associated value sets are stored in the Design Studio data dictionary. The item catalog category is mapped to a product specification in the cartridge.  
  
A new product specification can be created or an existing product specification is mapped to the item catalog category. After the mapping between the item catalog category and product specification is completed, the product to service mapping specialist will define and associate fulfillment functions and their dependencies to the product specification. After the validation and decomposition rules are defined, the cartridge is deployed into the OSM environments.
- 8. The Design Studio provides a function that tests the design-time configuration after you submit a sample test order. If there are issues with the product specification in the cartridge, a new product specification is created or the item catalog category is imported again. If a new product specification is created, then the configuration must be performed again. The cartridge must be re-deployed for the changes to become effective. The deployed cartridge is used by the OSM to process order lines that are submitted for fulfillment.

## Query Item Catalog Categories: Implementation

This section describes the implementation details of the `QueryProdClassAndAttributesSCEReqCommsABCSImpl` service.

The [Figure 5-2](#) illustrates the item catalog categories query flow.

**Figure 5–2 item catalog category Query Flow**



The process to query item catalog categories (or product classes in Siebel CRM) depends on the **AIAConfigurationProperties.xml** file settings set by the FMW server administrator. Based on these settings, the product-to-service mapping specialist performs one of the following:

- Query or import new or updated product classes from Siebel CRM (or the equivalent item catalog categories in Oracle Product Hub) into Design Studio.  
For more information about how to query product classes in Siebel CRM, see the discussion of the Query Product Classes business flow in - Implementation Overview in the *Oracle AIA Oracle Communications Order to Cash Integration Pack Implementation Guide*.
- Query or import the item catalog categories and Transaction Attributes from Oracle Product Hub into Design Studio through Oracle AIA.

## Query Item Catalog Categories and Transaction Attributes

When the product-to-service mapping specialist opens Design Studio and imports the item catalog categories and Transaction Attributes from Oracle Product Hub into Design Studio, the following events occur:

---



---

**Note:** The product-to-service mapping specialist must include all the necessary item catalog category Names in the query dialogue in Design Studio and invoke the query operation.

---



---

1. The Design Studio Client calls the Query Product Class Application Business Connector Service (ABCS) and provides one or more item catalog category Names as input. To import product classes from PH, you must provide an Oracle AIA URL in Design Studio. The following example contains a sample Oracle AIA URL:

### **Example 5–1 Sample Oracle AIA URL**

```
http://aia-server-name:aia-server-port/soa-infra/services/default/QueryProductClassesAndAttributesSCECommsReqABCSEImpl/QueryProductClassAndAttributesSCECommsReqABCSEImpl
```

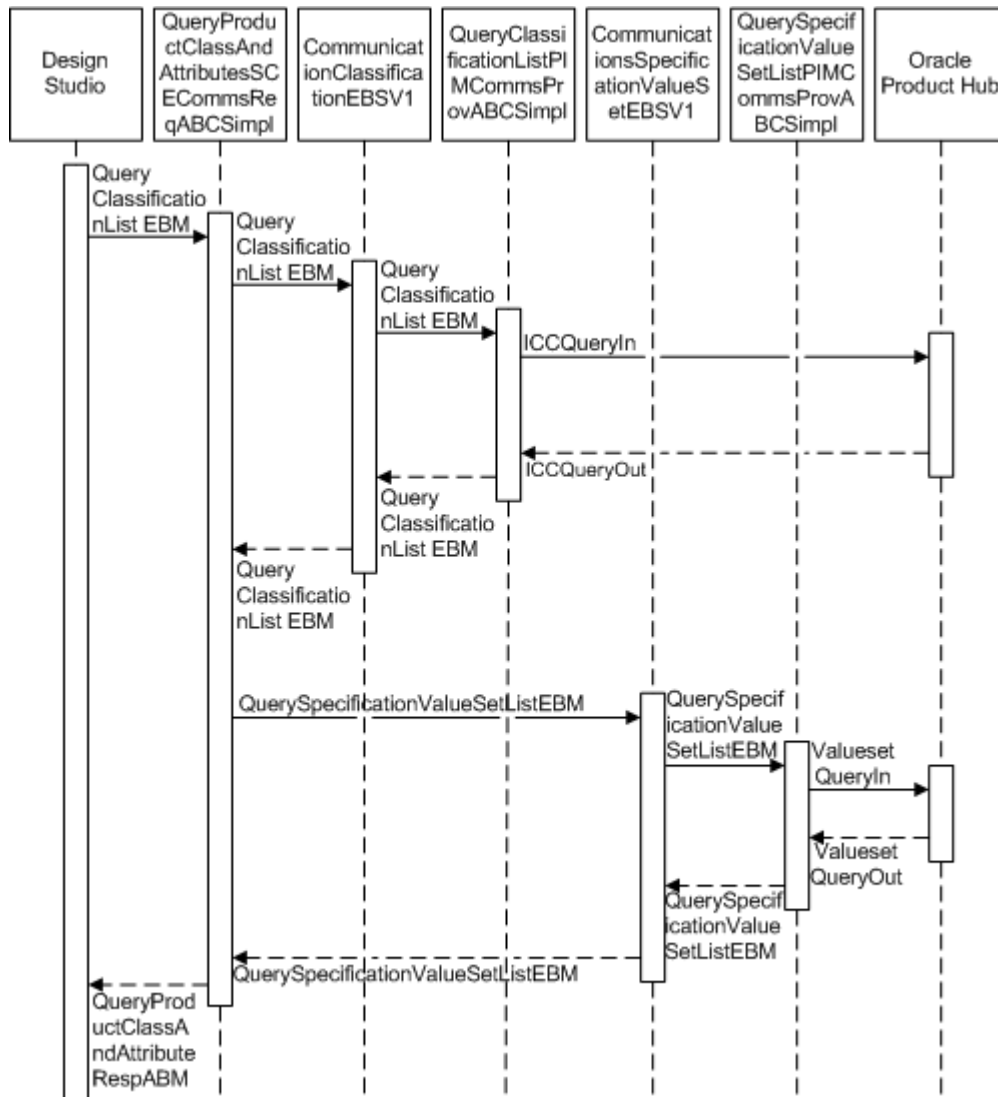
- In Design Studio, when selecting the item catalog category Names, you also have the option to query all the parent classes for that item catalog category.
2. The Query Product Class ABCS provides the Classification EBM (Enterprise Business Message).
3. The Query Product Class ABCS routes the Classification EBM to Oracle Product Hub Query item catalog category ABCS provider.
4. The Oracle Product Hub Query item catalog category ABCS transforms the Classification EBM, extracts the item catalog category name, and invokes the Oracle Product Hub Get item catalog category Web Service.
5. The Oracle Product Hub Get item catalog category Web Service returns a list of item catalog category details for all input item catalog categories to the Oracle Product Hub Query item catalog category ABCS.
6. The Oracle Product Hub Query item catalog category ABCS transforms the response message into the Classification EBM.
7. The Oracle Product Hub Query item catalog category ABCS routes the Classification EBM to the DS Query Product Class ABCS.
8. The Query Product Class ABCS routes the request to the Oracle Product Hub Query Valueset ABCS.
9. The Oracle Product Hub Query Valueset ABCS performs the transformation, filters the necessary fields, and constructs the ValuesetEBM.
10. The Oracle Product Hub Query Valueset ABCS routes the ValuesetEBM back to the DS Query Product Class ABCS.
11. The Query Product Class ABCS composes the Classification EBM and the ValuesetEBM and returns it to the Design Studio Client application.

## Query Item Catalog Categories: Integration Flow

This section describes the sequence of control flows that are required among the various integration services when querying item catalog categories and its attributes from Oracle Product Hub and publishing them into Design Studio.

The [Figure 5-3](#) illustrates the item catalog categories integration flow.

**Figure 5-3 item catalog category Integration Flow**



When this process initiates, the following events occur:

1. The Design Studio user (a product-to-service mapping specialist) triggers the Query item catalog category and attribute definitions integration flow using the Design Studio client, which provides the item catalog category Name and the Oracle Fusion Middleware (FMW) URL. In the case of an update, Design Studio also provides the item catalog category Names for all of the subclasses.
2. The Design Studio invokes the `QueryProdClassAndAttributesSCEReqCommsABCSImpl` with the `QueryClassificationListEBM`, which contains the item catalog category Names.
3. `QueryProdClassAndAttributesSCEReqCommsABCSImpl` passes through the `QueryClassificationListEBM` to the `QueryICCListPIMCommsProvABCSImpl`.
4. `QueryICCListPIMCommsProvABCSImpl` transforms the `QueryClassificationListEBM` into the `IccQueryIn` definition IO application business message (ABM).



5. The ICC web service returns the complete item catalog category information and associated Attribute IDs for all input item catalog category Names. The QueryICCListPIMCommsProvABCImpl transforms the Oracle Product Hub IccQueryOut into the QueryClassificationsList EBM.
6. The QueryClassificationList EBM goes all the way back to the QueryProdClassAndAttributesSCEReqCommsABCImpl.
7. The QueryProdClassAndAttributesSCEReqCommsABCImpl then invokes the QueryValueSetListPIMCommsProvABCImpl with the QuerySpecificationValueSetListEBM. The QueryValueSetListPIMCommsProvABCImpl transforms the QuerySpecificationValueSetListEBM into the Oracle Product Hub attribute definitions IO (ABM).
8. The ICC service returns the complete attribute valueset information for one or more attribute IDs. The QueryValueSetListPIMCommsProvABCImpl transforms the response into the QuerySpecificationValueSetListRespEBM.
9. The QuerySpecificationValueSetListEBM go all the way back to the QueryProdClassAndAttributesSCEReqCommsABCImpl. The QueryProdClassAndAttributesSCEReqCommsABCImpl merges the QuerySpecificationValueSetListEBM and the QueryClassificationList EBM and maps them to the QueryProdClassAndAttributesRespABM.
10. The QueryProdClassAndAttributesSCEReqCommsABCImpl replies to Design Studio along with the QueryProdClassAndAttributesRespABM, which contains the item catalog category and attribute details for all the item catalog category Names provided by the product to the service mapping specialist.

## Support for Effectivity during Design-Time - Methodology

You can use the item catalog categories to represent unique product specifications. Product specifications represent a type of product offering and can only be sold through a product offering. Product specifications represent the unique entities that must be fulfilled. Not all product classes are created for this purpose. For the subset of product classes created to represent product specifications, at design time you must map the product class name and the corresponding fulfillment pattern name in OSM in its Central Order Management role. Consequently, changes to these product classes affects the mapping for both design time and run time order management handling. This section describes an important aspect of this mapping effectivity: how the time of a new mapping or a change to an existing mapping takes effect in coordination across Order Capture (Oracle Product Hub) and Central Order Management (OSM).

References to product classes in this section are limited to those used to represent product specifications. To distinguish these, it is recommended to use a naming convention, such as ending the name with the term *ProdSpec*. You can create and update product classes directly in Siebel CRM or from a Product Master, such as Oracle Product Hub for Communications. In this section, whenever you see Siebel design time product class changes, it is also applicable to product master when one is used.

Specify the effectivity for the item catalog categories in Oracle Product Hub when:

- You create a new item catalog category.
- You update an existing item catalog category.
- You make an existing item catalog category inactive.

Whenever these scenarios or any combination of these scenarios occur, you must query the item catalog categories in the Design Studio, which is the design time tool for OSM from the Oracle Product Hub application. After the item catalog categories are successfully queried, item catalog categories and the product specification are updated manually, and a following condition occurs:

- The item catalog category is mapped to a new product specification. The new product specification definition involves defining the fulfillment metadata.
- The item catalog category is mapped to an existing product specification.
- The item catalog category mapping is changed to a different product specification.

Item catalog category effectivity must be the same as the product specification effectivity. You update product specification effectivity manually in the Design Studio after the item catalog category is successfully queried and the product specification is mapped to it.

When the mapping between the item catalog category and product specification is updated (when the effectivity of the item catalog category and product specification changes), the cartridge version in SCE Studio must be updated and the cartridge must be redeployed to the environment.

For more information about updating the cartridge version and the various deployment options, see *Oracle Communications Order and Service Management Cartridge Guide for Oracle Application Integration Architecture*.

## Deciding on Effectivity and Cartridge Deployment

Whenever the previously mentioned scenarios or their combination occur, you must create a new version of the cartridge and redeploy it. All orders the OSM system has submitted for processing uses the existing cartridge version. Any subsequent new orders (including revision orders, follow-on orders, and change orders) or existing orders that are not yet submitted for processing by the OSM system uses the new version of the deployed cartridge. You should group item catalog category changes and set effectivity for a date-time that is suitable for deploying a new cartridge version, such as an off-peak hour.

---

---

**Note:** You deploy a new version of the cartridge only if you introduced a new mapping or changed an existing mapping.

---

---

The support for effectivity on the item catalog category and the effectivity on the product specification is manifested in OSM by the effectivity of deployment of the cartridge. Effectivity is defined and controlled by the deployment dates for the cartridge. To manage effectivity across the applications, you must consider the following issues:

- The effective dates on the item catalog category are the same as the deployment dates on the cartridge.

In this case, the Oracle Product Hub administrator and the product-to-service mapping specialist in SCE Studio must reach a consensus on the effectivity dates of the item catalog category and the deployment dates of the cartridge, respectively.

- The effective dates on the item catalog category differ from the deployment dates on the cartridge.

In this case, the deployment dates on the cartridge control effectivity. You can handle effectivity as follows:

- Create a new item catalog category or update an existing item catalog category. Import the item catalog category and define the mappings to the product specification in the cartridge. Deploy it based on when the changes must be applied to the order lines in the OSM fulfillment system.
- Create a new item catalog category, the mappings to the product specification are not yet done in the cartridge. If the item catalog category is subsequently updated, Oracle Product Hub queries the updated version of the item catalog category and defines the mappings in the cartridge before deploying it. In this case, multiple versions of the item catalog category may be defined with different effective dates and queried before the mappings are defined and the cartridge is deployed. Again, the deployment is based on when the changes must be applied to the order lines in the OSM fulfillment system.

---

---

**Note:** If you create a new product specification in the cartridge, you must configure the product specification before deploying the cartridge.

---

---



---

---

# Oracle Product Master Data Management Integration Option for PDC

This chapter provides an overview of the Oracle Product Master Data Management Integration Pack Option for Oracle Communications Pricing Design Center (PDC), supported features, and assumptions and constraints, and specifically, describes the process flows, interfaces, and integration services required in this integration.

## Overview

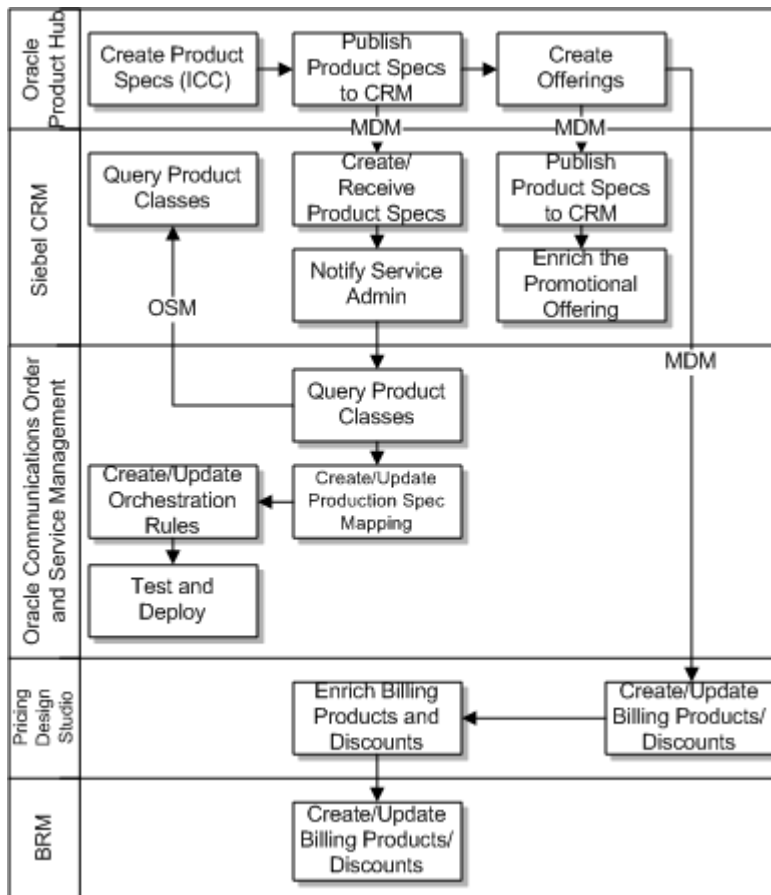
Oracle Product Hub publishes products, discounts, and sponsorships offerings in batch to various systems, such as Siebel CRM, Pricing Design Center, and Design Studio. The Oracle Application Integration Architecture (Oracle AIA) Product Master Data Management pre-built Integration receives this input, prepares payload for each system, and invokes the appropriate Pricing Design Center services.

Products offerings are synchronized from Oracle Product Hub to Pricing Design Center and auto-promoted to Oracle Communications Billing and Revenue Management (BRM). Once the product is successfully committed to the BRM database, it publishes the message to BRM AQ. The integration then updates the Oracle AIA Cross reference with the BRM Identifier. If the offerings are enriched in Pricing Design Center, then Pricing Design Center administrators manually submit these items from Pricing Design Center to BRM. When these objects are created or updated successfully in BRM, a message is automatically published to indicate successful transaction.

## Integration Process Flow

Product offerings are driven by business needs and implemented by product administrators. These offerings are defined in Oracle Product Hub or other third party product master application and synchronized with edge systems, such as Siebel CRM and BRM. Product managers validate each product offering before releasing the products in Pricing Design Center.

**Figure 6–1 Process flow from Oracle Product Hub to BRM through Pricing Design Center and Oracle AIA**



Here is the business process flow from Oracle Product Hub to Pricing Design Center:

1. Product, discounts, and sponsorships are mastered in Oracle Product Hub.
2. Oracle Product Hub batch publishes product, discount, and sponsorship definitions to Oracle AIA.
3. Product Master Data Management Pre-built Integration Pack for Siebel CRM synchronizes product, discount and sponsorship offerings to Siebel CRM and creates Oracle AIA cross-references.
4. A Product Master Data Management Pre-built Integration Pack for Pricing Design Center Oracle AIA connector synchronizes the product, discount and sponsorship offerings to Pricing Design Center and creates Oracle AIA cross-references.
5. Data is enriched in Pricing Design Center using the transformation engine to synchronize the product, discount and sponsorship offerings to BRM.
6. After the product, discount and sponsorship offerings are synchronized in BRM, the BRM identifiers are updated in Oracle AIA cross-references.

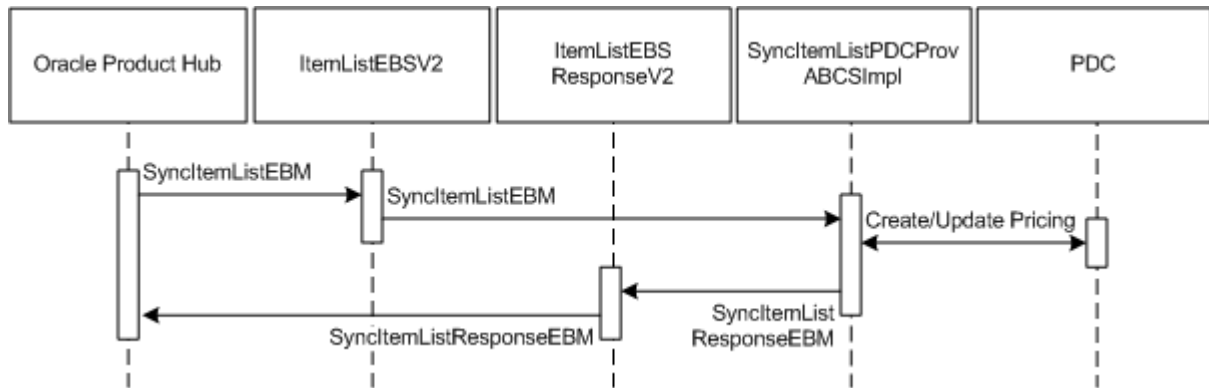
## Synchronization of Oracle Product Hub and Pricing Design

The Oracle Product Hub to Pricing Design Center synchronization flow lets you create product offerings in Oracle Product Hub and publish them to Pricing Design Center using the Oracle AIA Product Master Data Management pre-built integration.

The Oracle AIA Product Master Data Management pre-built integration receives the published items from Oracle Product Hub and sends them to Pricing Design Center by invoking Pricing Design Center services.

Figure 6–2 shows synchronization between Oracle Product Hub and Pricing Design Center.

**Figure 6–2 Oracle Product Hub to Pricing Design Center Synchronization**



Integration synchronizes items from Oracle Product Hub to Pricing Design Center as follows:

- Product administrators create or edit products/Discounts/ChargeShares in Oracle Product Hub and publish the batch to Oracle AIA by selecting the target systems.
- SyncItemPIMEEventConsumer consumes the message and invokes SyncItemPIMReqABCSImpl
- SyncItemPIMReqABCSImpl queries Oracle Product Hub for the list of target systems with which the message needs to be synced using QueryItemListPIMAdapter.
- After identifying the target system, SyncItemPIMReqABCSImpl calls ItemEBSV2 service to route the message to target systems based on their system IDs.
- Since the message is intended for Pricing Design Center, ItemEBS routes the message to SyncItemPDCProvABCSImpl.
- SyncItemPDCProvABCSImpl transforms the SyncItemEBM message into Pricing Design Center PricingObject and sends it to Pricing Design Center.
- Once successful, SyncItemPIMReqABCSImpl calls PriceListEBSV2 service to route the message to all the target systems based on their system IDs.
- Since the message is intended for Pricing Design Center, PriceListEBS routes the message to SyncPriceListPDCProvABCSImpl which updates all the pricing related information such as rate to Pricing Design Center.
- SyncPriceListPDCProvABCSImpl transforms the SyncPriceListEBM message into Pricing Design Center PricingObject and sends it to Pricing Design Center.
- Once successful, the success message is sent to SyncItemPIMReqABCSImpl, which in turn updates Oracle Product Hub with the message status. In case of failure, an error message with all required details is sent back to Oracle Product Hub.
- Once the product/discount/sponsorship items are synced to Pricing Design Center successfully, they are auto synced to BRM.

- After the product/discount/sponsorship items are committed to BRM database successfully, BRM send a message to the AQ.
- Corresponding AQ Consumer picks up the message and populates the Xref (ITEM\_ITEMID.xref) with the relevant values.

When a sponsorship object is synced from Oracle Product Hub to Pricing Design Center, Oracle AIA integration creates a dummy sponsorship model (charge share model) in Pricing Design Center which is submitted to BRM automatically. Any enhancements to the sponsorship model is done from Pricing Design Center and submitted to BRM.

Other information such as description, minimum, maximum, and quantity has to be enhanced in Oracle Product Hub. When sponsorship details are changed (other than model), Oracle AIA integration updates Pricing Design Center and BRM.

For information on Pricing Design Center to BRM synchronization see, "About Pricing Design Center Application Architecture", *Oracle Communications Pricing Design Center User's Guide*.

## Integration Services

The integration services used by the integration option are the following:

- SyncItemListPDCProvABCSImpl
- SyncPriceListPDCProvABCSImpl
- CreateProductXrefBRMAQ
- CreateSponsorshipXrefBRMAQ
- CreateDiscountXrefBRMAQ

### SyncItemListPDCProvABCSImpl

This provider ABCS synchronizes item list from Oracle Product Hub to Pricing Design Center.

### SyncPriceListPDCProvABCSImpl

This provider ABCS synchronizes pricelist from Oracle Product Hub to Pricing Design Center.

### CreateProductXrefBRMAQ

This process creates products in the cross-reference files.

### CreateSponsorXrefBRMAQ

This process creates sponsorship items in the cross-reference files. When a sponsorship item is synced from Oracle Product Hub, Pricing Design Center integration spoke creates sponsorship with details. Pricing Design Center provides web service for creating the sponsorship offering object.

### CreateDiscountXrefBRMAQ

This process creates discount items in the cross-reference files.



## Terminology Mapping Between Pricing Design Center and BRM

Table 6–1 lists common terms and description used in Pricing Design Center and BRM:

**Table 6–1 Mapping Between BRM and Pricing Design Center Terms**

BRM	Pricing Design Center
ChargeShare Model	Charge share
ChargeShare	Charge share offer
Deal	Bundle
Discount model	Discount
Discount	Discount offer
EBal(n) discount expression	Variable(n) discount expression
EVAL("function_name") discount expression	Function("function_name") discount expression
Plan	Package
Plan list	Package list
Product	Charge offer
Rate plan	Charge
Rate plan selector	Charge selector
Resource	Balance element
Subscription group	Service

## Pre-requisites for Multi-BRM Setup

The following are the pre-requisites whenever the multi-BRM setup is used:

1. Only one Pricing Design Center environment can be associated with one BRM. For two BRM environments, set up two Pricing Design Center environments.
2. For integration using two Pricing Design Center environments, enter the values as provide PDC\_02 and PDC\_03 in the drop down of the destination values on the Destination System Specification page. Similarly, enter the values as PDC\_02 and PDC\_03 when publishing items.
3. Add PDC\_02 and PDC\_03 in the systems page in Oracle AIA and restart server.
4. Add the PDC\_02 and PDC\_03 columns and values in the respective DVMs and cross-references (XREFS) with same values as mentioned in PDC\_01.
5. Add the PDC\_02 and PDC\_03 endpoint URL in AIAConfigurationProperties.xml.
6. In the AIAConfigurationProperties.xml, change the default.BRMID to BRM\_02 or BRM\_03 as required. For a single BRM instance, use BRM\_01.

Since Oracle Product Hub does not send all the discount/ChargeShare model information, the pre-built integration include these values by default (Table 6–2) as they are mandatory columns in Pricing Design Center for creating models.

**Table 6–2 Default Discount/ChargeShare Element Values**

Element	Discount	Sponsorship
pricingProfileName	Standard Discount	ChargeSharing
priceListName	Default	Default
Default	0	0
endDate	Inf	Inf
applicableChargeAndQuantity	REMAINING_CHARGE	NA
name	Pricing	Pricing
distributionMethod	FROM_BAL_IMPACT	FROM_BAL_IMPACT
tierBasis	FALSE	FALSE
lowerBound	0.0	0.0
upperBound	NO_MAX	NO_MAX
price	0.0	0.0
unitOfMeasure	NONE	NONE
balanceElementNumCode	840	840
priceType	GRANT	GRANT
startValidityMode	IMMEDIATE	IMMEDIATE
endValidityMode	RELATIVE_TO_START	RELATIVE_TO_START
relativeStartOffset	-1	-1
relativeEndOffset	1	1
relativeStartOffsetUnit	BILLING_CYCLE	BILLING_CYCLE
relativeEndOffsetUnit	BILLING_CYCLE	BILLING_CYCLE

## Assumptions and Constraints

Assumptions and constraints in Oracle Product Hub to Pricing Design Center integration are:

- Pricing Design Center does not allow creating products with same rate plan name.
- Unlike BRM integrations, time models are mastered in Pricing Design Center for usage events. Oracle AIA integration does not send objects related to time models.
- Although, Pricing Design Center provides an alternate method to create and update products, discounts, and sponsorship objects synced from Oracle Product Hub, the recommended method is to use the CreateSubmit and UpdateSubmit APIs to ensure proper updates to BRM.
- Product item information coming from Oracle Product Hub includes rate plan, rate data, balance impact information. Discount item information coming from Oracle Product Hub include discount, discount model information. Sponsorship item information coming from Oracle Product Hub includes sponsorship, sponsorship model information.
- Oracle Product Hub week and time ranges are applicable to only usage rate plans in Pricing Design Center.
- Time model object is applicable only to usage rate plan in Pricing Design Center. While time models are enhanced in the Pricing Design Center, the Oracle AIA

---

integration does not send the day ranges and time ranges to Pricing Design Center. For usage events, all rate plans are mastered in Pricing Design Center. The Oracle AIA integration sends rate plan objects with rate tier and balance impact to Pricing Design Center. Time models need to be enhanced in Pricing Design Center.

- Even though Oracle Product Hub accepts the same minimum and maximum quantity values for the same rate plan in the balance impact, Pricing Design Center restricts same quantity tiers in the date ranges.
- As part of master event definition, changes are required on the event details and not for rate plans and rate tiers.
- Reuse of rate plans is supported by Pricing Design Center. Reuse of existing rate plans in different products is not.
- Only basic information is synced to Pricing Design Center for usage based rate plan. You need to further enhance the time model part and other information in the rate plan if required. You cannot update the same rate plan information from Oracle Product Hub to Pricing Design Center as any enhancement done for usage rate plans will be lost.
- While sending payload to Pricing Design Center, rate plan (charge) information is sent as an object to Pricing Design Center in the same payload along with charge offering object. This charge rate plan is unique within the charge offering.
- Pricing Design Center IDs do not change when products are migrated from one Pricing Design Center environment to another.
- Pricing Design Center accepts only default price list values in the charge offering and charge rate plans.
- All offers are subscription profiles. This applies to onetime and recurring charges as well. However, for usage and discounting, you can use different profiles.



# Part II

---

## Implementing the Delivered Process Integrations

Part II contains the following chapters and appendixes:

- [Configuring the Oracle Product Hub Integration](#)
- [Frequently Asked Questions](#)
- [Support for Rate Tiers and Effectivity](#)
- [User Defined Attributes Framework](#)
- [Scenarios](#)



---

---

# Configuring the Oracle Product Hub Integration

This chapter provides information on configuring Oracle Product Hub, Siebel CRM, and Oracle E-Business Suite for integration.

## Adding a New Spoke Instance

To add a new spoke or target instance for the integration:

1. Add the instance name to the destination system to pick target systems while publishing.
  - a. Login to Oracle Product Hub.
  - b. Navigate to the responsibility Oracle Product Hub Data Librarian.
  - c. Under Batch Import, navigate to Import Workbench.
  - d. Click the System Setup tab.
  - e. Click Define Source System.
  - f. Provide the Name and Code for the new Target Instance. For example, if the new target system is JD Edwards:  
Name: JD Edwards  
Code: JDE
2. Add the same entry on the destination system valueset Destination\_System in Oracle Product Hub.
  - a. Select the responsibility Setup Workbench (Setup)
  - b. Click the Value Sets tab.
  - c. Search for the valueset Destination\_System.
  - d. In values section, click Create.
  - e. Add the display value and internal value for the new target system. Example:  
Value: JDE (This value must match the code value defined above).  
Translated Value: JD Edwards (This must match the name previous value).  
Description: New Target System JD Edwards.
3. Add the entry for the instance in the System Setting in Oracle AIA.
  - a. Login to the System Setting page.
  - b. Click the Setup tab.

- c. Click the System tab.
  - d. Click Create.
  - e. Add the entry for the new Target System. Here is an example:  
 System Code: JDE\_01. (The code that identifies the system.)  
 System Description: JD Edwards Instance  
 System Type: JDE. (The type of system for example: JDE, SIEBEL, EBIZ and so forth. This value is used in the routing rules.)  
 Application Type: MFG. (The type of application. For example: MFG, CRM, ERP and so forth.)
4. Add the entry in TARGET\_ID DVM with the COMMON column pointing to the system defined in System Setting with PIM\_01 column pointing to the system defined in Oracle Product Hub system (Step 1). [Table 7-1](#) shows an example:
- COMMON: JDE\_01  
 PIM\_01: JDE  
 Entry in TARGET\_ID DVM:

**Table 7-1 Sample DVMs**

COMMON Column	PIM_01 Column
EBIZ_01	EBSSS
SEBL_01	Siebel
BRM_01	BRM
IDE_01	IDE

5. Add the entry in ENTITY\_TO\_TARGET\_APPLICATION DVM, for processing Pricelist and Bill of Materials (BOM) for the new instance. Here is an example:
- TARGET\_ID: JDE\_01 (Common ID of the application instance)  
 RECEIVE\_PRICELIST\_FLAG: N  
 RECEIVE\_BOM\_FLAG: Y

**Table 7-2 Examples for System Name and Codes**

System	System Name and Code in Oracle Product Hub	System Setting
Ebiz	Name: Ebiz System Code: EBSSS	Internal Id: Ebiz System Code: EBIZ_01 System Type: EBIZ Application Type: EBIZ
Siebel	Name: Siebel CRM Code: Siebel	Internal Id: Siebel System Code: SEBL_01 System Type: SIEBEL Application Type: SIEBEL
BRM	Name: BRM System Code: BRM	Internal Id: BRM System Code: BRM_01 System Type: BRM Application Type: BRM
JDE	Name: JDE System Code: JDE	Internal Id: JDE System Code: JDE_01 System Type: JDE Application Type: JDE



---

**Note:** In the System Setting values, System Code is the value used for cross-reference and DVM lookups. In the routing rules, the Application Type value is checked.

Seeded DVM and cross-reference files are available in the build under the following directories:

*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/dvm*

*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/xref*

They can also be looked at by logging into MDS.

---

6. Build and deploy the provider connector service for processing item/itemstructure/pricing for the new spoke system. Here is an example:

SyncItemListJDEProvABCImpl: For processing items.

SyncPricelistJDEProvABCImpl (if applicable): For processing pricelists.

SyncBillOfMaterialsListJDEProvABCImpl: For processing item structures.

7. Add the routing rules to the new instances.

Example: Add routing rule in ItemEBSV2 for Invoking SyncItemListJDEProvABCImpl

Filter Expression:

```
{aia:getServiceProperty('|lcb;|http://xmlns.oracle.com/EnterpriseServices/
Core/Item/V2|rcb;|ItemEBSV2', 'Routing.ActiveRuleset', false())='DEFAULT'
and count(/ebo:SyncItemListEBM/ebo:DataArea) > 0 and
(/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction
/corecom:EnvironmentCode='PRODUCTION' or not
(/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction
/corecom:EnvironmentCode/text())) and
(aia:getSystemType(/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Target
/corecom:ID)='MFG') and
(aia:getSystemType(/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Sender
/corecom:ID) != 'MFG')}; {namespace
ebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/Item/V2 namespace
aia=http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.
AIAFunctions namespace
corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2}
```

8. Add similar rules for PriceListEBSV2 and BillOfMaterialsEBS.
9. Run the scripts to create the columns corresponding to the instance in the cross-reference tables. The new spoke system needs to be added to the cross-reference and DVM files and updated in MDS. To do this:
  - a. Identify the cross-references and DVMs being used in the connectors for the new spoke system.
  - b. For example: Add JDE\_01 to the xref file ITEM\_ITEMID.xref under *Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/xref/ITEM\_ITEMID.xref* and to the DVM file DVM under *Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/dvm/ITEM\_TYPE.dvm*
  - c. Update MDS. Modify *domain\_home/soa/aia/MDSUtils/UpdateMetaDataDP.xml* to include the files modified

in step b. Then navigate to the folder *MW\_home/soa/aiafp/Install/AID/* and run:

```
ant -f MW_home/soa/aiafp/Install/AID/UpdateMetaData.xml
-DdeploymentPlan=domain_home/soa/aia/MDSUtils/UpdateMetadataDP.xml
-DPropertiesFile=domain_home/soa/aia/bin/AIAInstallProperties.xml
```

- d. Once complete, login to the Oracle AIA console, navigate to *setup\AIA Configuration* and click the Reload button to load the changes.

## Post Installation Configuration Set Up

Inventory organizations can be broadly classified as one that is associated as item validation organization for a specific operating unit, and the other as a list of all the various organizations that have been classified as inventory organization to which product can be associated.

Other aspects are:

- An operating unit in Oracle Product Hub is mapped to an Organization/Business Unit in Siebel CRM.
- An inventory organization in Oracle Product Hub is mapped to inventory location in Siebel CRM, which needs to be child of a business unit.
- If Oracle Product Hub and E-Business Suite co-exist, the organization and inventory organization needs to be set up in Oracle Product Hub exactly the same way as in E-Business Suite. You need to perform the steps mentioned here to extract data (IDs) for creating cross-references.

---



---

**Note:** For more information on operating units or inventory organizations, see the Oracle E-Business guides.

---



---

## Extract Inventory Location and Operating Unit IDs in Oracle Product Hub

To extract inventory location IDs:

1. Log in to Oracle Product Hub database.
2. Run a SQL statement to query for all the organizations classified as inventory organization. For example:

```
select A.ORGANIZATION_ID from HR_ORGANIZATION_INFORMATION A, INV_ORGANIZATION_
NAME_V B where A.ORGANIZATION_ID = B.ORGANIZATION_ID and A.ORG_
INFORMATION1='INV' and B.ORGANIZATION_CODE = 'V1'
```

---



---

**Note:** Replace the organization code specific to your installation.

---



---

3. Use the IDs from step 2 for cross-referencing the data for inventory location using the XREF\_TABLE\_NAME: INVENTORY\_LOCATION\_ID.

To extract operating unit IDs:

1. Log in to the Oracle Product Hub database.
2. Run a SQL statement to query for all the organizations classified as inventory organization. For example:

```
select A.ORGANIZATION_ID from HR_ORGANIZATION_INFORMATION A, INV_ORGANIZATION_
NAME_V B where A.ORGANIZATION_ID = B.ORGANIZATION_ID and A.ORG_
```

---

```
INFORMATION1='OPERATING_UNIT' and B.ORGANIZATION_CODE = 'V1'
```

---



---

**Note:** Replace the organization code specific to your installation.

---

3. Use the IDs from step 2 for cross-referencing the data for operating units using the XREF\_TABLE\_NAME: ORGANIZATION\_ID.
4. Identify Item Validation Organization for Oracle Product Hub for the operating unit you want.

### Identify Item Validation Organization for Oracle Product Hub

To identify the item validation organization from the names of the organizations you noted in Extract Inventory Organization and Operating Unit IDs in E-Business Suite.

Publishing a product is on-demand by a product administrator from Oracle Product Hub. As products can be associated with various inventory Organizations under a given operating unit, it is expected that the items associated with the Inventory Organization - Item Validation Org in Order Management is only published to Siebel CRM.

1. Log in to Oracle Product Hub. The Oracle Applications home page will appear.
2. Confirm that you have the responsibilities set up for your operating units of interest, for example, order management.

---

**Note:** Vision Operations has been used as an example. You can use operating units of your choice.

---

3. Select Order Management Super User, Vision Operations (USA).
4. Perform these steps using the System Parameters screen for each of the order management responsibilities associated with various operating units that are of interest to you:
  - a. Select Values from Setting Parameter Values.
  - b. Enter Vision Operations in Operating Unit and GenericParameters in Category.
  - c. Write down the value that appears in Item validation organization. (Examples: Vision Operations, Vision Germany, and Vision France).

## Mapping Siebel CRM

To map Siebel CRM organizations to E-Business Suite operating units:

1. Log in to Siebel CRM Applications.
2. Click Site Map.
3. Select Administration-Groups, Organizations.
4. For every operating unit in E-Business Suite, create a corresponding organization in Siebel CRM.

For more information, see the Siebel CRM product documentation.

### Creating Inventory Locations in Siebel CRM

To map Siebel CRM Inventory Locations to E-Business Inventory Organizations:

1. Log into Siebel CRM applications.
2. Click on Site Map.
3. Select Inventory, Inventory Locations.

Make sure to create a 'Warehouse' Inventory Type for the Oracle Inventory Organizations which will validate, check availability for orders, and ship items during order fulfillment. The corresponding operating unit should also be associated to the Inventory Location as 'Organization'.

### Extract Business Unit and Inventory Location IDs in Siebel CRM

To extract inventory location details for Siebel CRM:

1. Log in to the Siebel CRM database as the table owner.
2. Run this query and note the IDs for the business unit from the search result:

```
select row_id, name from s_org_ext where name like '%Vision%'
```

3. Run this query and note the IDs for the inventory locations:

```
(select row_id, name from s_invloc where name like '%Vision%')
```

## Setting Up BRM

These are the setups required for Oracle Communications Billing and Revenue Management (BRM):

- Install and configure BRM JCA adapter.  
For more information about installing and setting up the BRM JCA adapter, see the BRM documentation.
- Set up all metadata required for creating products, pricelist data and discounting. For example: Create services; events; resources; G/L IDs; ratable usage metrics (RUMs); Tax code; Impact categories; Timezones; map event types to services; pipeline data, and so on.  
For more information about price lists, see *Oracle Communications Billing and Revenue Management Setting Up Pricing and Rating*.
- Set up pricing for friends and family functionality.  
For more information about setting up pricing, see *Oracle Communications Billing and Revenue Management Setting Up Pricing and Rating*.

## Setting Up Cross-References

Cross-references can be created after inventory locations have been set up in Siebel CRM, Oracle Product Hub, and Oracle E-Business Suite. The setup includes:

- [Populating Seed Values in Cross-Reference Tables](#)
- [Validating Cross-References](#)
- [Creating a Pricelist in Siebel CRM](#)

## Populating Seed Values in Cross-Reference Tables

The cross-reference data for the ORGANIZATION\_ID and INVENTORY\_LOCATION\_ID XRef entity must be added manually before you use the Pre-built Integration Pack, by using SQL insert statements. The following rows need to be created in the XRef data table:

- One row for Oracle Product Hub: PIM\_01
- One row for Common: COMMON.
- One row for Siebel CRM: SEBL\_01 (if the Siebel CRM option is installed).

The physical XRef table (XREF\_DATA) can be located in the database configured for FMW/SOA and contains the columns listed in [Table 7-3](#) and [Table 7-4](#):

**Table 7-3 Xref Table Format**

Column Name	Description
XREF_TABLE_NAME	This column stores the XRef data types. For example: <ul style="list-style-type: none"> <li>■ oramds:/apps/AIAMetaData/xref/ORGANIZATION_ID.xref</li> <li>■ oramds:/apps/AIAMetaData/xref/INVENTORY_LOCATION_ID.xref</li> </ul>
XREF_COLUMN_NAME	This column stores the values indicating the edge applications and COMMON value: <ul style="list-style-type: none"> <li>■ PIM_01</li> <li>■ EBIZ_01</li> <li>■ COMMON</li> <li>■ SEBL_01</li> </ul>
ROW_NUMBER	This column stores the GUID.
VALUE	This column stores the actual data that is cross-referenced.
IS_DELETED	This column indicates whether the column is deleted or not. For example, 'N'.
LAST_MODIFIED	This column stores last modified time.

**Table 7-4 Organizational XRef Table Value Format**

XRef_Column_Name	Value
COMMON	GUID
PIM_01	Organization Id from Oracle Product Hub. For example, 911
SEBL_01	Organization Id (Business Unit) from Siebel CRM. For example, 88-8KO98

### Sample SQL Insert Statement for INVENTORY\_LOCATION\_ID Table

For PIM\_01.

```
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_
DELETED, LAST_MODIFIED) VALUES('oramds:/apps/AIAMetaData/xref/ORGANIZATION_
ID.xref', 'PIM_01', 'BD618D40B30C11DEBFA5D9490F57512E', '911', 'N', SYSTIMESTAMP)
```

For SEBL\_01:

```
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_
DELETED, LAST_MODIFIED) VALUES('oramds:/apps/AIAMetaData/xref/ORGANIZATION_
ID.xref', 'SEBL_01', 'BD618D40B30C11DEBFA5D9490F57512E', '88-8KO98', 'N', SYSTIMESTAMP)
```

For COMMON:

```
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_
```

```
DELETED, LAST_MODIFIED) VALUES('oramds:/apps/AIAMetaData/xref/ORGANIZATION_
ID.xref', 'COMMON', 'BD618D40B30C11DEBFA5D9490F57512E', '2d37303730363632353936363232
3735', 'N', SYSTIMESTAMP)
```

**Table 7-5 Sample Data for XRef Organization\_ID Table**

XRef_Table	XRef_Column	Row_Number	Value	IS_Deleted	Last_Modified
oramds:/apps/AIAMetaData/xref/ORGANIZATION_ID.xref	SEBL_01	BD618D40B30C11DEBFA5D9490F57512E	88-8KO98	N	2012-4-11.11.29 .54.210000000
oramds:/apps/AIAMetaData/xref/ORGANIZATION_ID.xref	PIM_01	BD618D40B30C11DEBFA5D9490F57512E	911	N	2012-4-11.11.29 .54.210000000
oramds:/apps/AIAMetaData/xref/ORGANIZATION_ID.xref	COMMON	BD618D40B30C11DEBFA5D9490F57512E	2d373037 30363632 35393636 32323735	N	2012-4-11.11.29 .54.210000000

**Table 7-6 INVENTORY\_LOCATION\_ID XRef Table Value Format**

XRef_Column_Name	Value
COMMON	GUID
PIM_01	INVENTORY_LOCATION_ID from Oracle Product Hub. For example, 911.
EBIZ_01	Ebiz INV ID. For example, 204.
SEBL_01	INVENTORY_LOCATION_ID from Siebel CRM. For example, 88-8KO98.

**Sample SQL Insert Statement for INVENTORY\_LOCATION\_ID Table**

For PIM\_01.

```
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_
DELETED, LAST_MODIFIED) VALUES('oramds:/apps/AIAMetaData/xref/INVENTORY_LOCATION_
ID.xref', 'PIM_01', 'BD618D40B30C11DEBFA5D9490F57512E', '911', 'N', SYSTIMESTAMP)
```

For SEBL\_01:

```
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_
DELETED, LAST_MODIFIED) VALUES('oramds:/apps/AIAMetaData/xref/INVENTORY_LOCATION_
ID.xref', 'SEBL_01', 'BD618D40B30C11DEBFA5D9490F57512E', '88-8KO98', 'N', SYSTIMESTAMP)
```

For COMMON:

```
INSERT INTO XREF_DATA (XREF_TABLE_NAME, XREF_COLUMN_NAME, ROW_NUMBER, VALUE, IS_
DELETED, LAST_MODIFIED) VALUES('oramds:/apps/AIAMetaData/xref/INVENTORY_LOCATION_
ID.xref', 'COMMON', 'BD618D40B30C11DEBFA5D9490F57512E', '2d37303730363632353936363232
3735', 'N', SYSTIMESTAMP)
```

**Table 7-7 Sample Data for XRef INVENTORY\_LOCATION\_ID Table**

XRef_Table	XRef_Column	Row_Number	Value	IS_Deleted	Last_Modified
oramds:/apps/AIAMetaData/xref/INVENTORY_LOCATION_ID	EBIZ_01	BD618D40B30C11DEBFA5D9490F57512E	204	N	2012-4-11.11.29. 54.210000000

**Table 7–7 (Cont.) Sample Data for XRef INVENTORY\_LOCATION\_ID Table**

XRef_Table	XRef_Column	Row_Number	Value	IS_Deleted	Last_Modified
oramds:/apps/AIAMetaData/xref/INVENTORY_LOCATION_ID	SEBL_01	BD618D40B30C11DE BFA5D9490F57512E	88-8KO98	N	2012-4-11.11.29. 54. 210000000
oramds:/apps/AIAMetaData/xref/INVENTORY_LOCATION_ID	PIM_01	BD618D40B30C11DE BFA5D9490F57512E	911	N	2012-4-11.11.29. 54. 210000000
oramds:/apps/AIAMetaData/xref/INVENTORY_LOCATION_ID	COMMON	BD618D40B30C11DE BFA5D9490F57512E	2d373037 303636323 539363632 323735	N	2012-4-11.11.29. 54. 210000000

## Validating Cross-References

To validate cross-references:

1. Log in to the AIA XRef database.
2. Use this query on the table XREF\_DATA to confirm there are three records each for organization and inventory location used in the XML Files.

Sample query to verify INVENTORY\_LOCATION\_ID:

```
select value||'::'||Xref_column_name from xref_Data where row_number in (select
row_number from xref_data where xref_table_name like '%INVENTORY_LOCATION_ID%'
and value in ('911'))
```

Sample query to verify ORGANIZATION\_ID:

```
select value||'::'||Xref_column_name from xref_Data where row_number in (select
row_number from xref_data where xref_table_name like '%ORGANIZATION_ID%' and
value in ('911'))
```

3. Replace the value '204' in the above queries with the values from your system. (The number of operating units and inventory locations depends on your set up.)

## Creating a Pricelist in Siebel CRM

To create a pricelist in Siebel CRM:

1. Create a pricelist in Siebel CRM.
2. Extract the pricelist RowID in Siebel CRM and update the Property PriceList.ID with the pricelist RowID of AIAConfigurationProperties for following:
  - a. SyncItemCompositionListSiebelCommsProvABCSImpl
  - b. ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl
3. If these pricelist(s) are used in standard pricing offer or billing pricing offer (rate plan selector in billing products) model in Oracle Product Hub, then extract the RowID of the pricelist(s) in Siebel CRM and add data for each pricelist (in PRICELIST.dvm). Thereafter, update the PIM\_01/BRM\_01 columns to map the corresponding pricelist identifiers used in Oracle Product Hub/BRM.

For more information, see ["Setting Up Configuration Properties"](#).

## Configuring the Integration to Support 100 Characters

To support 100 characters for product names during synchronization between Oracle Product Hub and Siebel CRM, refer to the following assumptions and perform the setup in Oracle AIA as described below:

### Assumptions

Consider these assumptions to support 100 character product names:

1. In Oracle Product Hub, Vision Operations is the default or master organization where products must be first defined. You can create new organization and configure as Master Organization in Oracle Product Hub.
2. In Siebel CRM, Default Organization is the master organization where products must first be defined. Do not to change the configuration of Default Organization.
3. The name of the master organizations in Siebel CRM and Oracle Product Hub can be different and the integration needs to maintain the same mapping.
4. You can create additional organization in Oracle Product Hub and Siebel CRM. Associate the products to these additional organizations as secondary organizations. However, the integration does not support multiple organization at this time.

### Setup in Oracle AIA

In the Oracle AIA xref for Organization, map the master organization ID in Oracle Product Hub to master organization ID in Siebel CRM. For example:

**Table 7–8 Master Organization Mapping - Oracle Product Hub to Siebel CRM**

Common	PIM_01	SEBL_01
3242424a3f313	12345 (Vision Operations ID)	0-R9NH (Default Organization ID)

With the above assumptions and setup done in Oracle AIA, in Siebel CRM, set the Product Business Unit to Default Organization, and the Primary Organization to Default Organization. If the Product Business Unit and Primary Organization values are identical, Oracle can support a 100 chars long product name through Product WS.

## Working with Cross-References

Cross-references map and connect the records within the application network and enable these applications to communicate in the same language. The integration server stores the relationship in a persistent way so that others can refer to it.

For more information about working with cross-references, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

[Table 7–9](#) lists the cross-references for Product Master Data Management integration:



**Table 7–9 Oracle Product Hub cross-references**

Table Name	Column Name	Description
ITEM_ITEMID	EBIZ_01,COMMON, SEBL_01, PIM_01, BRM_01	Item/Product Ids used in item synchronization and various other flows.
ORGANIZATION_ID	EBIZ_01,COMMON, SEBL_01, PIM_01	Organization/ business unit IDs used in various flows.
INVENTORY_LOCATION_ID	EBIZ_01,COMMON, SEBL_01, PIM_01	Inventory location IDs used in various flows.
CLASSIFICATION_CLASSIFICATIONID	COMMON, SEBL_01, PIM_01	Mapping of product class-ids or catalog-category ids between applications used in Item and item catalog category synchronization flows.
USER_NAME	EBIZ_01,COMMON, SEBL_01, PIM_01	User names of various applications used in various flows.
PRICELINE_ID	EBIZ_01,COMMON, SEBL_01, PIM_01	Used to cross reference pricelist Line Id.
PRICELINETYPE_ID	EBIZ_01,COMMON, SEBL_01, PIM_01	Used to cross-reference BRM product's events with Siebel price line ID. Also links to the COMMON of ITEM_ID.
SIEBELPRODUCTEV_ENTXREF	ITEM_ID_COMMON, LINEPRICETYPECODE	To cross-reference BRM Product's Event that is Associated with the main product in Siebel CRM.
PROMOTION_PROMOTIONID	COMMON, SEBL_01	During product synchronization Promotion Id is stamped in this cross-reference based Entity Type. During Product Structure Synchronization, this cross-reference is looked up to determine Product is promotion.

## Working with Domain Value Maps

Domain value maps (DVMs) are a standard feature of the Oracle SOA Suite that enables you to equate lookup codes and other static values across applications, for example, FOOT and FT or US and USA.

DVMs are static in nature, though administrators can add additional maps as needed. Transactional business processes never update DVMs, they only read from them. They are stored in XML files and cached in memory at run-time.

The DVM data should be in synchronization with what the participating applications use. This synchronization should be done before running any initial loads or initiating any incremental transactional flows.

Seeded DVM and cross-reference files are available in the build under the following:

*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/dvm*

*Oracle\_home/comms\_home/source/soainfra/apps/AIAMetaData/xref*

They can also be looked at by logging into MDS.

[Table 7–10](#) lists the DVMs for the Product Master Data Management integration:

**Table 7-10 List of DVMs**

<b>DVM Type</b>	<b>DVM Column Name</b>	<b>Description</b>
ADDRESS_COUNTRYID	COMMON, SEBL_01, PIM_01	Indicates ID of the countries.
BOM_EXPLODE_OPTION	SYSTEM, CURRENTFUTURE_FLAG	Exploding the BOM based on CURRENTFUTURE_FLAG (CURRENT, CURRENTFUTURE).
BOM_INDICATOR_NUM_TO_BOOL	EBIZ_01,COMMON	DVM to convert Boolean value to number.
DATA_TYPE_CODE	EBIZ_01,COMMON, SEBL_01	It specifies the data type of the valueset
DAYS_OF_WEEK	EBIZ_01,COMMON, SEBL_01, PIM_01, JDEE1_01, P6EPPM_01	Indicates the days of the week as Monday, Tuesday, and so on.
ENTITY_TO_TARGET_APPLICATION	TARGET_ID, RECEIVE_PRICELIST_FLAG, RECEIVE_BOM_FLAG	List of entities received by target applications, for example, Oracle E-Business Suite receives Items and BOMs; Siebel CRM receives item, pricelist and BOM; BRM receives items and pricelist.
ITEM_BILLING_SPECIAL_RATING_TYPE	COMMON, SEBL_01, PIM_01	Indicates the GL codes to record revenue.
ITEM_BOM_ITEMTYPE_CODE	EBIZ_01,COMMON, SEBL_01, PIM_01	Cross-reference value: BOM item-type code.
ITEM_CONSIGNMENT_ITEM_INDICATOR	COMMON, EBIZ_01, PIM_01	Cross-reference value: ConsignmentItemIndicator.
ITEM_CONTRACT_TYPE_CODE	EBIZ_01,COMMON, PIM_01	Cross-reference value: Contract type code.
ITEM_CREATE_FIXED_ASSET_INDICATOR	EBIZ_01,COMMON, PIM_01	Cross-reference value CreateFixedAssetIndicator.
ITEM_CURRENTFUTURE_PAST_STATUS	COMMON, SEBL_01, PIM_01	Status: Current, future, past.
ITEM_EFFECTIVITY_CONTROL_CODE	EBIZ_01,COMMON, PIM_01	Cross-reference value: EffectivityControlCode.
ITEM_INVENTORY_PLANNING_CODE	EBIZ_01,COMMON, PIM_01, JDEE1_01	Cross-reference value: InventoryPlanningCode.
ITEM_LOGISTICS_SHIP_CARRIER	COMMON, SEBL_01, PIM_01	This indicates carrier information, for example, DHL.
ITEM_LOGISTICS_SHIP_METHOD	COMMON, SEBL_01, PIM_01	This indicates mode of transport, for example, Air.
ITEM_MAKEORBUY_CODE	EBIZ_01,COMMON, PIM_01, JDEE1_01	Cross-reference value: MakeOrBuyCode.
ITEM_RECEIVING_ROUTING_CODE	EBIZ_01,COMMON, PIM_01	This indicates: ReceivingRoutingCode.
ITEM_REPLENISHMENT_SOURCE_CODE	EBIZ_01,COMMON, PIM_01, JDEE1_01	Cross-reference value: ReplenishmentSourceCode.
ITEM_RESERVATION_ALLOWED_INDICATOR	EBIZ_01,COMMON, PIM_01	Cross-reference value: ReservationAllowedIndicator.
ITEM_SERVICE_REQUEST_ENABLED_INDICATOR	EBIZ_01,COMMON	Cross-reference value: ServiceRequestEnabledIndicator.
ITEM_TARGET_INDUSTRY	COMMON, SEBL_01, PIM_01	Indicates the type of industry, for example, Manufacturing.

**Table 7–10 (Cont.) List of DVMs**

<b>DVM Type</b>	<b>DVM Column Name</b>	<b>Description</b>
ITEM_TRACK_INSTANCE_INDICATOR	EBIZ_01,COMMON,PIM_01	Cross-reference value: TrackInstanceIndicator.
ITEM_TYPE	EBIZ_01,COMMON,SEBL_01,PIM_01,JDEE1_01,OTM_01,JDEW_01,CRM_01	Indicates item-type code.
ITEM_ENTITY_TYPE	SEBL_01,PIM_01,COMMON	Identifies the type of entity that Oracle Product Hub is publishing. Examples: Item, Promotion, Discount.
ITEM_UOM_CODE	EBIZ_01,COMMON,AGILE_01,SAP_01,JDEE1_01	Cross-reference value item unit-of-measure code.
ITEM_WIP_SUPPLY_CODE	EBIZ_01,COMMON,SEBL_01,PIM_01,AGILE_01	Cross-reference value: WIPSupplyTypeCode.
LANGUAGE_CODE	EBIZ_01,COMMON,SEBL_01,PIM_01,RETL_01,PSFT_01,JDEE1_01,SAP_01,OTM_01	Specifies the language codes of Oracle Product Hub, Siebel CRM, and Common.
LINEPRICE_TYPECODE	COMMON,SEBL_01,PIM_01	Indicates the type of the list-price of the item, for example, Base Price, One-time, Activation, Quarterly, Monthly, or so on.
PLANTIER_RESTRICTIONTYPE	COMMON,SEBL_01,PIM_01	Indicates the rate restriction used in the plan tier, for example, No Restrictions, Restriction based on days only, Restriction based on days/time slots.
PRICE_BILLINGPERIODCODE	COMMON,SEBL_01,PIM_01	Indicates the frequency of applying the charge, for example, One-Time, Weekly, Monthly, Yearly.
PRICE_DISCOUNTTYPECODE	COMMON,SEBL_01,PIM_01	Indicates the type of discount, for example, Percent or Amount.
PRICE_MARKUPBASISCODE	COMMON,SEBL_01,PIM_01	Indicates the basis for calculating the price markup of the product from the covered product, for example, List Price, Net Price.
PRICE_OVERRIDEYPECODE	COMMON,SEBL_01,PIM_01	Indicates whether a discount override or a price override is defined for the charge, for example, Committed - Price override, Dynamic - Discount.
PRICE_TYPE	COMMON,SEBL_01,PIM_01	Indicates the type of price, for example, One-Time, Recurring, and so on.
PRICECHARGETYPE	EBIZ_01,COMMON,SEBL_01,PIM_01	Indicates the price-charge type, for example, Usage, Product Purchase Fee, Monthly Cycle Forward, and so on.
PRODUCT_STATUS	EBIZ_01,COMMON,SEBL_01,PIM_01,OTM_01,CRM_01	Cross-reference value: product status.
RATEPLAN_TYPECODE	COMMON,SEBL_01,PIM_01	Indicates the type of rate plan associated with the charge of the product, for example, Single Rate Plan, Rate Plan Selector, Custom event analysis, and so on.
RATEPLAN_UNITCODE	COMMON,PIM_01	Indicates the rate plan unit code.
STRUCTURE_NAME_ALTERNATE_BOM	EBIZ_01,COMMON,PIM_01	Mapping between structure name and alternate BOM.
TARGET_ID	COMMON,UCM_01,PIM_01	Specifies the common IDs of the systems for sync. This DVM is used to map the system defined in Oracle Product Hub and OER for publication.

**Table 7–10 (Cont.) List of DVMs**

<b>DVM Type</b>	<b>DVM Column Name</b>	<b>Description</b>
UNIT_OF_MEASURE	EBIZ_01,COMMON, SEBL_01, PIM_01,OTM_01,JDEE1_01,JDEE1_01_RATECODE, P6EPPM_01_RATECODE, EBS_PROJECTS_01_UNIT_OF_MEASURE,SAP_01	Mapping of unit of measure (UOM) codes among various applications.
ITEM_DUAL_UOM_TRACKING_INDICATOR	COMMON, PIM_01,EBIZ_01, JDEE1_01	Dual UOM tracking indicator.
ITEM_UOM_CONVERSION_USAGE_CODE	COMMON, PIM_01,EBIZ_01	UOM conversion usage code.
ITEM_STATUS_CODE	EBIZ_01,COMMON, SEBL_01, PIM_01	Value cross-reference item-status code.
ITEM_SERIALIZATION_EVENT_CODE	COMMON, PIM_01,EBIZ_01, JDEE1_01	Indicates serialization-event code.
ITEM_INDICATOR	COMMON, SEBL_01, PIM_01, EBIZ_01, OTM_01, AGILE_01, JDEE1_01	Value cross-reference Indicator as YES/NO.
ACTIVE_FLAG_SEBL	COMMON,SEBL_01	Indicates active status in Siebel CRM as active, pending, or closed.
ITEM_PRIMARYCLASSIFICATIONCODE	COMMON,EBIZ_01	Used for mapping item catalog group code.
ITEM_BILLING_TYPE_CODE	EBIZ_01,COMMON, SEBL_01, PIM_01, AGILE_01, BRM_01	This DVM maps the Billing Type of a product, for example, subscription, item, system, and so on.
APPLIES_TO	PIM_01, COMMON, BRM_01, SEBL_01	This DVM is used to map discount billing service type, for example, /service/telco/gsm/telephony.
PARTIAL_PURCHASE	PIM_01, COMMON, BRM_01	This DVM maps the partial purchase allowed indicator, for example, Y/N.
PROVISIONING_TAG	PIM_01, COMMON, BRM_01	This DVM maps the provisioning tag, for example, Voice, Corporate, Premium, and so on.
DISCOUNT_VALIDITY	COMMON, BRM_01	This DVM maps the discount validity, for example, PRORATED_DISCOUNT, NO_DISCOUNT, FULL_DISCOUNT.
MULTIPLE_DISCOUNT_PER_EVENT	PIM_01, COMMON, BRM_01	This DVM maps the multiple discount per event field, for example, PARALLEL, SEQUENTIAL
CURRENCY_CODE	EBIZ_01,COMMON, SEBL_01, PIM_01,PORTAL_01, UCM_01, PSFT_01, RETL_01, JDEE1_01, CRMOD_01, OTM_01, P6EPPM_01, JDEW_01, SAP_01, BRM_01	Indicates the currency codes, for example, USD, EUR, and so on.
RATEPLAN_UNITCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the unit code of the rate plan, for example, days, month, and so on.
RATEDATA_PURCHASEPROVISIONINGCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates how to charge customers when they purchase the product during the accounting cycle, for example, Do not Charge for the cycle; Calculate Charge based on the amount used, etc.
RESOURCE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates non-currency resource codes, for example, minutes, loyalty points, and so on.

**Table 7–10 (Cont.) List of DVMs**

<b>DVM Type</b>	<b>DVM Column Name</b>	<b>Description</b>
GL_CODE	PIM_01, COMMON, BRM_01, SEBL_01	Identifies the set of products in general ledger.
PRICE_IMPACTCLASSIFICATIONCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the impact category associated with the rate, for example, Default, Premium, Standard, Free, and so on.
PLANIMPACT_UNITCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the scaled unit codes, for example, Second, Minute, Hour, and so on.
RATEDATA_DISCOUNTBRACKETCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the discount brackets to charge different amounts based on quantity, for example, Rate Dependent, Continuous, Resource, and so on.
PLANTIER_EFFECTIVITYMODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the associated effectivity mode for the tier's date and time, for example, Absolute, Relative.
RATEPLAN_TAXAPPLICABILITYCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates when to apply the tax, for example, No Taxes, Event Time, End of Cycle, and so on.
PRICETYPE_EVENT	COMMON, SEBL_01, PIM_01, PRICECHARGETYPE, PRICECHARGETYPEUOM, PIM_01, COMMON, BRM_01	Maps the event name, for example, Purchase, Cancel, and so on.
LINEPRICE_UOMCODE	PIM_01, COMMON, BRM_01, SEBL_01	Maps the price unit of measure code. Indicates the ratable usage metric (RUM). RUM defines the quantity being measured and the units used, for example, Occurrence.
GRACE_PERIOD_UOM	SEBL_01, COMMON	Indicates Grace Period UOM for Promotion Ex.Per Day,Months,Years
ITEM_INSTANCES	COMMON, SEBL_01, PIM_01	Indicates Promotion Instances
ITEM_PROMOTION_TYPE	COMMON, SEBL_01, PIM_01	Indicate Promotion Type ex: Bundle Promotion
RECURRING_ADJUST_UOM	COMMON, SEBL_01, PIM_01	Indicate Recurring Adjust UOM
ITEM_CHARGE_CREDITS_TYPE	COMMON, SEBL_01, PIM_01	Indicate Item Charge Credit Type. For example: Penalty.
ITEM_PROMOTION_COMMITMENT_START	COMMON, SEBL_01, PIM_01	Indicate Promotion Commitment Start. For example: Now, Original Start, Original End.
ITEM_PROMOTION_DURATION	COMMON, SEBL_01, PIM_01	Indicates Promotion Duration. For example: New Duration.
PRICE_ADJUSTMENT_TYPE	COMMON, SEBL_01, PIM_01	Indicates price Adjustments in Promotion Ex. Price Override, Discount Amount.
PRICELIST	PIM_01, COMMON, SEBL_01, BRM_01	This DVM maps the pricelists created in Siebel CRM/ BRM from E-Business Suite / Oracle Product Hub instance to Siebel CRM/BRM.

The considerations for setting up DVMs include:

- STRUCTURE\_NAME\_ALTERNATE\_BOM: This is used to map the STRUCTURE\_NAME in Oracle Product Hub to an ALTERNATE\_BOM in E-Business Suite. There is no Siebel CRM equivalent for STRUCTURE\_NAME\_ALTERNATE\_BOM.
- STRUCTURE\_TYPE: Used to identify type for structures defined in Oracle Product Hub.

For every structure name defined in Oracle Product Hub, an entry needs to be made in this DVM against STRUCTURE NAME column. In the STRUCTURE TYPE column, the structure name needs to be identified as SALES or Others. When a BOM Event is raised in Oracle Product Hub, the routing rule using the structure name associated with the BOM event and the corresponding type defined in this DVM is used to route the SALES BOM to Siebel CRM.

Table 7–11 provides sample data for the STRUCTURE\_TYPE domain value map:

**Table 7–11 Sample Data for STRUCTURE\_TYPE DVM**

Structure Name	Structure Type
Pbom	Sales
Ebom	BOM
PrimaryCommon	Sales

- An entry should be made in this DVM even for a primary BOM.

## Handling Errors

Based on the roles defined for the services, the program sends email notifications if a service errors out. There are no Oracle AIA-specific errors thrown by the services for this integration. For more information about the errors thrown by Siebel CRM or E-Business Suite, see the respective product documentation.

For more information about Oracle AIA error handling, see the discussion of setting up error handling in *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack*.

## Setting Up Configuration Properties

Set these properties in the *Comms\_home/source/soainfra/apps/AIAMetaData/config/AIAConfigurationProperties.xml* file.

For more information about requirements for working with *AIAConfigurationProperties.xml*, see the discussion of how to set up Oracle AIA Workstation in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Core Extension*.

Table 7–12 shows the properties for the SyncItemPIMReqABCImpl service.

**Table 7–12 SyncItemPIMReqABCImpl Properties**

System Property	Default Value	Description
Default.SystemID	-	It is the responsibility of the requestor application to send the system ID. If any requester application fails to send this, Oracle AIA picks the default system ID from this configuration property.
Routing.QueryItemListPIMAdapter.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.
Routing.QueryBillOffMaterialsListPIMAdapter.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.
Routing.GetListofTargetSystems.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.

**Table 7–12 (Cont.) SyncItemListPIMReqABCImpl Properties**

System Property	Default Value	Description
Routing.PublicationService.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.
Routing.ItemEBSV2.SyncItemList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION /CAVS	Sets the environment code element to the value PRODUCTION.
Routing.BillOfMaterialsEBS.SyncBillOfMaterialsList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION /CAVS	Sets the environment code element to the value PRODUCTION.
Routing.PriceListEBSV2.SyncPriceListList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION /CAVS	Sets the environment code element to the value PRODUCTION.
ABCSExtension.PreXformABMtoEBMItemListABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvokeItemListEBSSyncItemListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformEBMtoABMItemListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvolkePublicationServiceItemEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvokePriceListEBSSyncItemEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformEBMtoABMPriceListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvolkePublicationServicePriceListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreXformABMtoEBMIBillOfMaterialsListABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvokeBillOfMaterialsListEBSSyncBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformEBMtoABMBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvolkePublicationServiceBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
NUMBER_OF_PARALLEL_ITEM_TARGETS	-	When synchronizing items to multiple targets, this property controls Item processing for parallel targets
NUMBER_OF_PARALLEL_BOM_TARGETS	-	This property controls number of parallel targets to be processed for BOMs to multiple targets

Table 7-13 shows the properties for the SyncSpecificationValueSetListSiebelProvABCImpl service.

**Table 7-13 SyncSpecificationValueSetListSiebelProvABCImpl Properties**

System Property	Default Value	Description
Default.SystemID	I	It is the responsibility of the requestor application to send the system ID. If any requestor application fails to send this, Oracle AIA picks the default system ID from this configuration property.
ABCSExtension.PreXformEBMtoABMSpecificationValueSetListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformABMtoEBMSpecificationValueSetListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
Routing.SWIAAttributeImport.RouteToCAVS	false/true	Determines whether to invoke Oracle AIA session pool manager to retrieve / release sessions based on CAVS property. In case of simulating the service, a call to AIASessionPoolManager is unnecessary and it is ignored.
Routing.QueryBillOffMaterialsListPIMAdapter.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.
Routing.GetListofTargetSystems.RouteToCAVS	false/true	Determines whether to invoke Oracle AIA session pool manager to retrieve / release sessions based on CAVS property. In case of simulating the service, a call to AIASessionPoolManager is unnecessary and it is ignored.
Routing.GetListofTargetSystems.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.
Routing.PublicationService.RouteToCAVS	false/true	Determines whether to invoke Oracle AIA session pool manager to retrieve / release sessions based on CAVS property. In case of simulating the service, a call to AIASessionPoolManager is unnecessary and it is ignored.
Routing.PublicationService.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.
Routing.SWIAAttributeImport.SEBL_01.EndpointURI	-	Siebel CRM web service end-point location (SOAP endpoint URL).
Routing.SWIAAttributeImport.MessageProcessingInstruction.EnvironmentCode	PRODUCTION /CAVS	Sets the environment code element to the value PRODUCTION.
Routing.SpecificationValueSetResponseEBS.SyncSpecificationValueSetListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION /CAVS	Sets the environment code element to the value PRODUCTION.
Routing.SpecificationValueSetResponseEBS.SyncSpecificationValueSetListResponse.RouteToCAVS	false/true	Determines whether to invoke Oracle AIA session pool manager to retrieve / release sessions based on CAVS property. In case of simulating the service, a call to AIASessionPoolManager is unnecessary and it is ignored.
ABCSExtension.PreInvokeABSSyncAttributeDefinitionABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostInvokeABSSyncAttributeDefinitionABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.



**Table 7–13 (Cont.) SyncSpecificationValueSetListSiebelProvABCSImpl Properties**

System Property	Default Value	Description
OPERATION_TYPE	Y/N; N	This property indicates whether Siebel CRM web service is to be used in synchronize or upsert mode.  OPERATION_TYPE = 'Y': Updates elements in payload if they already exist. If not, they are inserted. If the element is not present in payload, but is present in system then it is not deleted.  OPERATION_TYPE = 'N' (Default): Updates elements in payload if they already exist. If not, they are inserted. If the element is not present in payload, but is present in system then it is not deleted.
UTC_FORMAT_ENABLE	Y/N; N	If the value is set to 'N' (Default), then the datetime elements are not changed to the UTC format and the values are passed through. If the value is set to 'Y', then the datetime elements are changed to the UTC format.
ABCSExtension. PreXformABMtoEBMItemListABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension. PreInvokeItemListEBSSyncItemEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension. PostXformEBMtoABMItemListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension. PreInvolkePublicationServiceItemListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension. PreInvokePriceListEBSSyncItemListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension. PostXformEBMtoABMPriceListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension. PreInvolkePublicationServicePriceListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension. PreXformABMtoEBMIBillOfMaterialsListABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvokeBillOfMaterialsListEBSSyncBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformEBMtoABMBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvolkePublicationServiceBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
NUMBER_OF_PARALLEL_ITEM_TARGETS	-	This property controls number of parallel targets to be processed for Item when synchronizing items to multiple targets.
NUMBER_OF_PARALLEL_BOM_TARGETS	-	This property controls number of parallel targets to be processed for BOMs to multiple targets.

Table 7–14 shows the properties for the SyncItemListBRMProvABCSImpl service.

**Table 7–14 SyncItemListBRMProvABCImpl Properties**

System Property	Default Value	Description
Default.SystemID	BRM_01	It is the responsibility of the requestor application to send the system ID. If any requester application fails to send this, Oracle AIA picks the default system ID from this configuration property.
ABCSExtension.PreXformEBMtoABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformEBMtoABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
Routing.BRMPRICINGServicett.RouteToCAVS	false/true	Determines whether to route the endpoint URL to either the end application service or CAVS for simulating the service.
Routing.BRMPRICINGServicett.CAVS.EndpointURI	NA	Sets the endpoint URL for the CAVS simulator.
Routing.BRMPRICINGServicett.BRM_01.EndpointURI	eis/BRM1	Sets the endpoint URL for the target application.
Routing.BRMPRICINGServicett.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
ABCSExtension.PreInvokeABS	false/true	Used as an extension point before application business service is invoked. This property determines whether to make invocation of service at the extension point depending on if it is true or false.
ABCSExtension.PostInvokeABS	false/true	Used as an extension point before application business service is invoked. This property determines whether to make invocation of service at the extension depending on if it is true or false.

Table 7–15 shows the properties for the SyncPriceListListBRMProvABCImpl service.

**Table 7–15 SyncPriceListListBRMProvABCImpl Properties**

Property Name	Property Value	Description
Default.SystemID	BRM_01	It is the responsibility of the requestor application to send the system ID. If any requester application fails to send this, Oracle AIA picks the default system ID from this configuration property.
ABCSExtension.PreXformEBMtoABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformEBMtoABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
Routing.BRMPRICINGServicett.RouteToCAVS	NA	Determines whether to route the EndpointURI to either the end application service or CAVS for simulating the service.
Routing.BRMPRICINGServicett.CAVS.EndpointURI	NA	Sets the EndpointURI for the CAVS simulator
Routing.BRMPRICINGServicett.BRM_01.EndpointURI	eis/BRM1	Sets the EndpointURI for the target application

**Table 7–15 (Cont.) SyncPriceListListBRMProvABCImpl Properties**

Property Name	Property Value	Description
Routing.BRMPRICINGServ ice_ptt.MessageProcessingIns tr uction.EnvironmentCode	PRODUCTION	Sets the EnvironmentCode element to the value PRODUCTION.
ABCSExtension.PreInvokeA BS	false/true	Used as an extension point before application business service is invoked. This property determines whether to make invocation of service at the extension point depending on if it is true or false.
ABCSExtension.PostInvoke ABS	false/true	Used as an extension point before application business service is invoked. This property determines whether to make invocation of service at the extension depending on if it is true or false.

Table 7–16 shows the properties for the QueryBillOfMaterialsListPIMAdapter service.

**Table 7–16 QueryBillOfMaterialsListPIMAdapter Properties**

System Property	Default Value	Description
Default.SystemID	-	It is the responsibility of the requestor application to send the system ID. If any requester application fails to send this, Oracle AIA picks the default system ID from this configuration property.
BATCH_SIZE	any integer	This property is used for defining the number of structures to be processed in sub batch (chunking)
Routing.ItemService. PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location (SOAP endpoint URL).

Table 7–17 shows the properties for the QueryItemListPIMAdapter service.

**Table 7–17 QueryItemListPIMAdapter Properties**

System Property	Default Value	Description
Default.SystemID	-	It is the responsibility of the requestor application to send the system ID. If any requester application fails to send this, Oracle AIA picks the default system ID from this configuration property.
BATCH_SIZE	any integer	Used for defining the number of sub-batch process items (chunking)
Routing.ItemService. PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location (SOAP endpoint URL).

Table 7–18 shows the properties for the SyncSpecificationValueSetListPIMReqABCImpl service.

**Table 7–18 SyncSpecificationValueSetListPIMReqABCImpl Properties**

System Property	Default Value	Description
Default.SystemID	-	It is the responsibility of the requestor application to send the system ID. If any requester application fails to send this, Oracle AIA picks the default system ID from this configuration property.
TRACE.LOG.ENABLED	-	This property is used enabling trace log
ABCSExtension.PreInvokeValueSetSer viceABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.

**Table 7–18 (Cont.) SyncSpecificationValueSetListPIMReqABCImpl Properties**

System Property	Default Value	Description
ABCSExtension.PostInvokeValueSetServiceABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvokeEBSSpecificationValueSetListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
Routing.SpecificationValueSetEBS.SyncSpecificationValueSetList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.PublicationService.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.
Routing.ValueSetService.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.

Table 7–19 shows the properties for the SyncItemCatalogCategoryPIMReqABCImpl service.

**Table 7–19 SyncItemCatalogCategoryPIMReqABCImpl Properties**

System Property	Default Value	Description
Default.SystemID	-	It is the responsibility of the application to send the system ID from which the request is being sent. If any requester application fails to send this, Oracle AIA picks the default system ID from this configuration property.
ABCSExtension.PreXformABMtToEBMItemCatalogCategoryABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformABMtToEBMItemCatalogCategoryABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvokeEBSSyncClassificationSchemeListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostInvokeEBSSyncClassificationSchemeListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
Routing.ItemCatalogCategoryService.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.
Routing.PublicationService.PIM_01.EndpointURI	-	Oracle Product Hub web service end-point location. This is a SOAP endpoint URL.
Routing.ClassificationSchemeEBS.SyncClassificationSchemeList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.SpecificationValueSetEBS.SyncSpecificationValueSetList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
TRACE.LOG.ENABLED	-	This property is used enabling trace log.

Table 7–20 shows the properties for the SyncClassificationSchemeListSiebelProvABCImpl service.

**Table 7–20 SyncClassificationSchemeListSiebelProvABCImpl Properties**

System Property	Default Value	Description
Default.SystemID	-	It is the responsibility of the requestor application to send the system ID. If any requester application fails to send this, Oracle AIA picks the default system ID from this configuration property.
REUSE_WORKSPACE	-	For reusing the workspace.
ABCSExtension.PreXformEBMtoABMClassificationSchemeListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PreInvokeABSyncProductClassABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformABMtoEBMClassificationSchemeListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
Routing.SyncProductClassService.SEBL_01.EndpointURI	-	Siebel CRM web service end-point location. This is a SOAP endpoint URL.
Routing.ClassificationSchemeResponseEBS.SyncClassificationSchemeListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION /CAVS	Sets the environment code element to the value PRODUCTION.
TRACE.LOG.ENABLED	-	This property is used enabling trace log.

Table 7–21 shows the properties for the SyncClassificationSchemeListSiebelProvABCImpl service.

**Table 7–21 SyncClassificationSchemeListSiebelProvABCImpl Properties**

System Property	Default Value	Description
Default.SystemID	-	It is the responsibility of the requestor application to send the system ID. If any requester application fails to send this, Oracle AIA picks the default system ID from this configuration property.
Routing.SWIAttributeImport.RouteToCAVS	false/true	Determines whether to invoke Oracle AIA session pool manager to retrieve / release sessions based on CAVS property. In case of simulating the service, a call to AIASessionPoolManager is unnecessary and it is ignored.
Routing.SpecificationValueSetResponseEBS.SyncSpecificationValueSetListResponse.RouteToCAVS	false/true	Determines whether to invoke Oracle AIA session pool manager to retrieve / release sessions based on CAVS property. In case of simulating the service, a call to AIASessionPoolManager is unnecessary and it is ignored.
ABCSExtension.PreXformEBMtoABMSpecificationValueSetListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostXformABMtoEBMSpecificationValueSetListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
Routing.SWIAttributeImport.SEBL_01.EndpointURI	-	Siebel CRM web service end-point location. This is a SOAP endpoint URL.

**Table 7–21 (Cont.) SyncClassificationSchemeListSiebelProvABCSImpl Properties**

System Property	Default Value	Description
Routing.SWIAttributeImport.MessageProcessingInstruction.EnvironmentCode	PRODUCTION /CAVS	Sets the environment code element to the value PRODUCTION.
Routing.SpecificationValueSetResponseEBS.SyncSpecificationValueSetListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION /CAVS	Sets the environment code element to the value PRODUCTION.
ABCSExtension.PreInvokeABSSyncAttributeDefinitionABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
ABCSExtension.PostInvokeABSSyncAttributeDefinitionABM	false/true	This property is used as an extension point before EBM is transformed to ABM to determine whether to invoke service depending on if it is true or false.
OPERATION_TYPE	Y/N; N	This property indicates whether Siebel CRM web service is to be used in synchronize or upsert mode.  OPERATION_TYPE = 'Y': Updates elements in payload if they already exist. If not, they are inserted. If the element is not present in payload, but present in system then it is not deleted.  OPERATION_TYPE = 'N' (Default): Updates elements in payload if they already exist. If not, they are inserted. If the element is not present in payload, but present in system then it is not deleted.
UTC_FORMAT_ENABLE	Y/N; N	Currently the default value is N. If the value is set to 'N', then the datetime elements are not changed to the UTC Format and the values are passed through. If the value is set to 'Y', then the datetime elements are changed to the UTC format.

Table 7–22 shows the properties for the ProductOptimizedSyncPriceListListSiebelCommsProvABCS service.

**Table 7–22 ProductOptimizedSyncPriceListListSiebelCommsProvABCS Properties**

System Property	Default Value	Description
Routing.Target.Default.SiebelProductService.SystemID	SEBL_01	Siebel CRM system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.Target.Default.SiebelProductService.EndpointURI	No default value.	Siebel CRM system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.SiebelProductService.RouteToCAVS	True/False. Default = false.	If true, it invokes the actual target system whose end point is indicated by the service-level property Routing.Target.Default.SiebelProductService.EndpointURI. If false, it invokes the verification system whose end point is indicated by the system-level property SyncResponseSimulator.Soap.EndpointURL.
Siebel.BusinessUnit	No default value.	All the products created will belong to this business unit in the Siebel CRM system. The value for this property should be the ID of the business unit in the Siebel CRM system. Set this before running product sync.
Siebel.Product.WorkspaceName	Demo Workspace	Name of the workspace to be used by Siebel CRM. Create a workspace and update this file with that workspace name.
Siebel.Product.WorkspaceReleaseFlag	Y/N. Default = N	Indicates whether the workspace needs to be released after the product is synchronized.

**Table 7–22 (Cont.) ProductOptimizedSyncPriceListListSiebelCommsProvABCS Properties**

System Property	Default Value	Description
Siebel.Product.WorkspaceReuseFlag	Y/N. Default = Y	Indicates whether the workspace needs to be reused for product to be synced.
Siebel.PriceList.ID	No default value.	All the products created by this sync belongs to this pricelist in the Siebel CRM system. The value for this property should be the ID of the pricelist in the Siebel CRM system. Set this before running product sync.
Siebel.PriceList.Currency	USD	Currency code of the pricelist mentioned in the preceding property. If the currency of the prices in PriceListEBM does not match this currency, price in Siebel CRM will be set to 0(zero). Set this before running product sync.

Table 7–23 shows the properties for the SyncItemCompositionListSiebelCommsProvABCSImpl service.

**Table 7–23 SyncItemCompositionListSiebelCommsProvABCSImpl Properties**

System Property	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel CRM system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.CommunicationsItemCompositionResponseEBSV1.SyncItemCompositionListResponse.RouteToCAVS	true/false. Default = false	Controls whether CommunicationsItemCompositionResponseEBSV1 routes messages to the verification system or to the Provider ABCS implementation.
Routing.CommunicationsItemCompositionResponseEBSV1.SyncItemCompositionListResponse.CAVS.EndpointURI	http://<http.hostname>:<http.port>/AIAValidationSystemServlet/asyncreponserecipient	The endpoint URI of the CAVS simulator.
Routing.CommunicationsItemCompositionResponseEBSV1.SyncItemCompositionListResponse.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTION /Any Value Default = PRODUCTION	If CAVS, then the message is routed to CAVS. For other values, the message is routed to the Provider ABCS Implementation.
Routing.SiebelProductService.SEBL_01.EndpointURI	http://<http.hostname>:<http.port>/eaienu/start.swe?SWEExtSource=SecureWebService&amp;SWEExtCmd=Execute&WSSOAP=1	Siebel CRM Product Import web service end point location. This is a SOAP end point URL. If the request message contains the target URL, then that takes precedence.
Routing.SiebelProductService.RouteToCAVS	true/false. Default = false	If true, it invokes the actual target system whose end point is indicated by the service-level property Routing.SiebelProductService.SEBL_01.EndpointURI. If false, it invokes the verification system whose end point is indicated by the system-level property SyncResponseSimulator.Soap.EndpointURL.
Routing.SiebelProductService.CAVS.EndpointURI	http://<http.hostname>:<http.port>/AIAValidationSystemServlet/asyncreponsesimulator	The endpointURI of the CAVS simulator.
Siebel.SEBL_01.BusinessUnit	No default value.	All the products created belong to this business unit in Siebel CRM system. The value for this property should be the ID of the business unit in Siebel CRM system. Set this before running product sync.

**Table 7–23 (Cont.) SyncItemCompositionListSiebelCommsProvABCSImpl Properties**

System Property	Value/Default Value	Description
Siebel.Product.WorkspaceName	Demo Workspace	Name of the workspace used by Siebel CRM. Create a work space and update it with that workspace name.
Siebel.Product.WorkspaceReleaseFlag	Y/N. Default = N	Indicates whether the workspace must be released after the product is synchronized.
Siebel.Pricelist.ID	No default value.	All the products created by this sync belongs to this pricelist in Siebel CRM system. The value for this property should be the ID of the Siebel CRM system pricelist. Set this before running product sync.
ABCSExtension.PreXformEBMtoABM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (before the EBM to ABM transformation.)
ABCSExtension.PostXformABMtoEBM	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (after the ABM to EBM transformation).
ABCSExtension.PreInvokeABS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvokeApplication).
ABCSExtension.PostInvokeABS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvokeApplication).

## Setting Up the Data for Seeded Attributes of the Telecommunications Library

The entities in [Table 7–24](#) have to be set up in target applications, and the corresponding attribute valueset must be updated with the values in Oracle Product Hub.

**Table 7–24 Attributes**

Attribute Group Name	Attribute Name
Product Details: Logistics	Vendor Part Number (Siebel CRM)
Subject compatibility rules	Matrix Name (Siebel CRM)
Pricing: Simple Pricelist (For non-billing products)	Pricelist Name (Siebel CRM), Volume Discount (Siebel CRM), Attribute Adjustment (Siebel CRM)
Pricing: Mutiple Pricelist	-
Promotion: Commitment: Charges Credits	Plan (Siebel CRM)
Billing Products Attributes General	Provisioning Tag (BRM), Supplier Tax ID (BRM)
Balance Impact	Pricelist Name (BRM), GL_ID (BRM), Impact Category (BRM)
Billing discount attributes	Provisioning Tag (BRM)

[Table 7–25](#) lists the attributes and respective DVMs in which the values for these attributes must be added:



**Table 7–25 DVM values**

<b>Attribute Name</b>	<b>DVM</b>
GLID	GL_CODE
Provisioning Tag	PROVISIONING_TAG
Impact Category	PRICE_IMPACTCLASSIFICATIONCODE

## Setting up List of Values in Siebel CRM

Once the Oracle Product Hub integration is installed and the configuration setup is complete, add these values to the PROD\_CD list of values (LOV) in Siebel CRM application:

- Commercial Bundle
- Discount
- Option Group
- Discount Model



---

---

## Frequently Asked Questions

1. **Q:** Is it possible to synchronize items from Oracle Product Hub to Siebel CRM that are associated with multiple inventory organizations that in turn are associated with the same operating unit?

**A:** No. Uniqueness of a product in Siebel CRM is based on a combination of product, business unit, and vendor; therefore, if such synchronization is done, the key does not remain unique.

2. **Q:** Why only set up the cross-references for inventory locations associated with item validation organization for Siebel CRM?

**A:** Item synchronization to Siebel CRM is based on the requirement that Siebel CRM is able to use that item as part of the order capture process in the Order to Cash: Siebel CRM - Oracle E-Business Suite integration. In E-Business Suite, an order can refer to products that are part of item validation organization only. For this reason, you must populate the cross-references for inventory location only with item validation organization.

3. **Q:** How to extend the Siebel CRM provider service to consume the UDAs published by Oracle E-Business Suite?

**A:** For SyncProductSiebelProvABCSImpl and SyncBillOfMaterialsListSiebelProvABCSImpl, set the flag enableCustomExtensions in XformSyncItemListEBMToSiebelProdABM\_Custom\_UDA.xml or XformSyncBillOfMaterialsListEBMToSiebelProdABM\_Custom\_UDA.xml to *true*/. Then add custom contents in these files.

4. **Q:** How to configure the STRUCTURE\_TYPE and STRUCTURE\_NAME\_ALTERNATE\_BOM DVMs?

**A:** STRUCTURE\_TYPE is composed of Structure Name and Structure Type. Structure Name value is the common value associated with a Structure Name in the DVM STRUCTURE\_NAME\_ALTERNATE\_BOM. Structure Type can be Sales for Sales BOM and any other value for all other BOMs. The Sales BOM needs to be identified by Sales as type, if this is changed the routing rules need to be changed to reflect the new value, since this value determines if the product will be routed to Siebel CRM or not.

5. **Q:** Does the service support sharing of inventory organization across multiple operating units?

**A:** No. This release of the integration does not support sharing of Inventory Organization across multiple operating units.

6. **Q:** Would there be an issue if the same item validation organization were shared across multiple operating units?

---

A: For answer, see question and answer 5.

7. Q: What are the default values for Master Data Management-product related entries in the Oracle AIA XML?

A: See "[Setting Up Configuration Properties](#)".

8. Q: When invoking Oracle Product Hub service, the Oracle Product Hub errors out with a message "No Access to Method [Method Name]".

A: To resolve this, here is a workaround:

1. Remove the existing grants to service methods on **Irep**.
  2. Go to Launch Functional Administrator, Core Services, Caching Framework, Select Total Cache Components, Select All.
  3. Click Clear all cache.
  4. Recreate grants.
  5. Click Redeploy on Irep service page to redeploy the service.
9. Q: When invoking Oracle Product Hub services multiple times, the subsequent invocation may take long time and the service request ends with a SocketTimeOut error

A: To resolve this, here is a workaround:

1. Launch Integrated SOA Gateway, Launch SOA Monitor.
  2. Click Purge.
  3. Click Turn off Web Service Monitor.
10. Q: The following error occurred during Item sync to E-Business Suite instance:  
*Unexpected error occured in package EGO\_ITEM\_PVT procedure Process\_Items:  
ORA-06502: PL/SQL: numeric or value error: character string buffer too small. Please contact your system administrator or support representative.*

A: To resolve this issue, compile System Items KFF {Key Flexfield } by following these steps:

1. Log on to **Ebiz Application**. Navigate to **System Administrator -> Application -> Flexfield -> Key -> Segments**
2. In Key Flexfield Segment form, search for application with keyword as **Inventory** and then search for Flexfield Title with keyword as **System Items**.
3. Click **Compile**. Run a background Concurrent Program to compile KPF. If it is completed successfully, the KFF is compiled.
4. Compile the package bodies INV\_EBI\_CHANGE\_ORDER\_PUB and INV\_EBI\_ITEM\_HELPER

---

**Note:** While compiling, if you receive an error *:Unable to extend index SYS.I\_PLSCOPE\_SIG\_ACTION\$ by 128 in tablespace SYSAUX*, then increase the application's table space and recompile.

---

## Support for Rate Tiers and Effectivity

Table B-1 depicts the support for rate tiers and effectivity.

**Table B-1 Support for Rate Tiers and Effectivity**

Feature	Promotion	Account Product	Bundle	Product	Tier 1	Balance Impact	Tier 2	Account Discount
Rate defined in Oracle Product Hub?	No	Yes	No*2	Yes	Yes	Yes	Yes	NA*5
Rate sync'd to Billing and Revenue Management?	No	Yes	No	Yes	Yes	Yes	Yes	NA
Rate sync'd to Siebel CRM?	No	Yes	No	Yes	No*3	Yes	No*3	NA
Revision defined in Oracle Product Hub?	Yes	Yes	Yes	Yes	NA	NA	NA	Yes
Revision sync'd to Billing and Revenue Management?	No	No	No	No	NA	NA	NA	No
Revision sync'd to Siebel CRM?*1	Yes	Yes	Yes	Yes	NA	NA	NA	Yes*6
Effectivity defined in Oracle Product Hub (absolute + relative)	Yes	Yes	Yes	Yes	Yes	Yes (Only absolute)	Yes	Yes
Effectivity sync'd to Billing and Revenue Management (absolute + relative)	No	Yes	No	Yes	Yes	Yes (Only absolute)	Yes	Yes
Absolute effectivity sync'd to Siebel CRM	No	Yes	Yes	Yes	Yes	NA*4	Yes	Yes
Relative effectivity sync'd to Siebel CRM (Only Rates and Balance Impact)	NA	NA	NA	NA	No	No	No	NA

There is no one-to-one mapping between revisions of items in Oracle Product Hub and versions of products in Siebel CRM.

The pricing for the bundles is calculated as the *sum of the charges for the component products*. If you decide not to use the recommended Communications Product Definition methodology on bundles, the pricing can be specified on the item that represents the bundle.

For more information about the communications-product-definition methodology, see the *Oracle Application Integration Architecture Oracle Communications Order to Cash Integration Pack Implementation Guide*.

---

The rate tier information is not synchronized to Siebel CRM. The rate tier only stores the price specified in the balance impact attribute group (AG) associated with the rate tier as the list price on the pricelist line.

The balance impact effectivity is defined only for non-currency resources and it is relative to the purchased product date.

Only discount models can be associated for discount. Only the name of the discount model is associated to the discount in Oracle Product Hub and synchronized to Billing and Revenue Management (BRM). The actual definition is enriched in BRM.

Discounts are synchronized as products to Siebel CRM.

---

---

## User Defined Attributes Framework

This appendix provides information about the user defined attributes (UDA) framework.

### Extending for Custom UDAs

You can classify user defined attributes as:

- Seeded attributes.
- Custom attributes.
- Transaction attributes.

For more information about the different types of attributes and how to handle them, see "[About the Integration Base Pack](#)".

### Handling UDAs

Product Master Data Management integration provides the following mappings:

- Operational attributes of the item (First Class Attributes) in Oracle Product Hub are mapped to the fixed elements in the itemEBO which are mapped to the item definition attributes in Oracle E-Business Suite and product definition attributes in Siebel CRM. Note that operational attributes are not mapped to product/discount definition in Oracle Communication Billing and Revenue Management (BRM).
- Seeded attributes of the item (user defined attributes) in Oracle Product Hub are mapped to the fixed elements in the itemEBO which are mapped to the product definition attributes in Siebel CRM and product/discount attributes in BRM. Note: Seeded attributes are not synchronized to E-Business Suite.
- Custom attributes of the item (User Defined Attributes) in Oracle Product Hub are mapped to the Specification group element in the itemEBO. Those custom attributes whose attribute group (AG) name start with XX are mapped to the user defined attributes in Siebel CRM and all other custom attributes are ignored in the target application connector services for Siebel CRM, BRM and E-Business Suite.
- Transaction attributes in Oracle Product Hub are mapped to the specificationEBO which are mapped to the attribute definition in Siebel CRM. Note that the transaction attributes are not synchronized to Oracle E-Business Suite and BRM.

All downstream systems need not consume all the UDAs. The UDA service will dynamically populate the Oracle Product Hub ABM only with UDAs instantiated as part of the particular Item Event raised by Oracle Product Hub.

- When an item catalog category that has a wide set of UDAs creates an item, it is possible that only a subset of the same would have to be mapped to the participating applications or only a subset would have been initialized as part of the particular item. As part of the service, only the attributes initialized are mapped to the itemEBO as a first class attribute (FCA) or as SpecificationGroup.

## Mapping Artifacts

The following artifacts specify the mapping from Oracle Product Hub to the canonical models.

### Mapping Operational Attributes to ItemEBM

Artifacts:

- **XformItemResponsePIMABMToSyncItemListEBM.xml**
- **XformItemListResponsePIMABMToSyncItemListEBM\_Custom.xml**

This is the mapping between Oracle Product Hub's operations attribute to EBM. Customer specific transformations related extensions should be done in **XformItemListResponsePIMABMToSyncItemListEBM\_Custom.xml**.

### Mapping Seeded Set of UDAs for Communications library to EBM

Artifacts:

- **XformUDAAGInterfaceCommsPIMBOMRespABMToSyncBOMEBM.xml**
- **XformUDAAGInterfaceCommsPIMItemRespABMToSyncItemListEBM.xml**
- **XformUDACommsPIMBOMRespABMToSyncBOMEBM.xml**
- **XformUDACommsPIMBOMRespABMToSyncBOMEBM\_Custom.xml**
- **XformUDACommsPIMItemRespABMToSyncItemListEBM.xml**
- **XformUDACommsPIMItemRespABMToSyncItemListEBM\_Custom.xml**
- **XformRestructureSyncBillOfMaterialsListEBM.xml**
- **XformRestructureSyncItemListEBM.xml**

### Mapping Pricelist related UDAs for Communications Library to EBM

- **XfromListOfItemsToSyncPriceListListEBM.xml**

The entire communications library seeded UDAs are mapped to EBM. For each AG, a mapping template is declared in the Interface xml.

The following xls are used to implement the same:

- **XformUDAAGInterfaceCommsPIMBOMRespABMToSyncBOMEBM.xml**
- **XformUDAAGInterfaceCommsPIMItemRespABMToSyncItemListEBM.xml**

The mappings for UDAs are defined in the following xml for each of the attributes within the AG:

- **XformUDACommsPIMBOMRespABMToSyncBOMEBM.xml**
- **XformUDACommsPIMItemRespABMToSyncItemListEBM.xml**

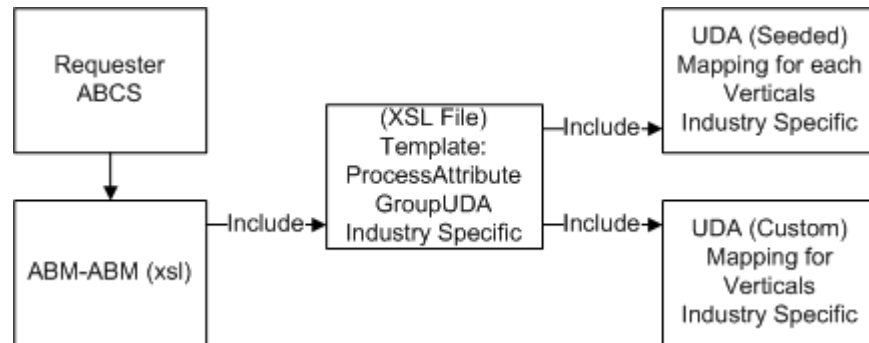
The mapping related to the discount and promotion type entities are passed from Oracle Product Hub UDA to a Specification group in the EBM.



## Implementation of UDA Transformation

Figure C-1 illustrates the UDAs to FCAs transformation in ItemEBM:

**Figure C-1 Adding UDA to a FCA in Item EBM**




---

**Note:** It is not recommended to add or modify the mapping from seeded user defined attributes to the first class attributes.

---

To add UDA to an FCA in Item EBM:

1. Document all the new UDAs.

Table C-1 shows an example of a UDA AG:

**Table C-1 Example of UDA AG**

Attribute Group Display Name	Attribute Group Internal Name
Item Manufacture	ItemManufacture
ID	ItemManufacture_ID
Priority	ItemManufacture_Priority
Status	ItemManufacture_Status
StartDate	ItemManufacture_StartDate
EndDate	ItemManufacture_EndDate

2. Define transformation for each attribute of the AG in industry/application specific XSL file.

For example:

```

<xsl:template name=" ItemManufactureAttrGeneral ">
<corecomEBO:ItemManufacture>
<corecom:Identification>
<corecom:ID>
<xsl:value-of select="ItemUDA[Name='ItemManufacture_ID']/CharValue"/>
</corecom:ID>
</corecom:Identification>
<corecom:PriorityCode>
<xsl:value-of-select="ItemUDA[Name='ItemManufacture_Priority']/CharValue"/>
</corecom:PriorityCode>
<corecom:EffectiveTimePeriod>
<corecom:StartDateTime>
<xsl:call-template name="formatUDADate">

```

```

<xsl:with-param name="formatDateParam" select="ItemUDA[Name='ItemManufacture_
StartDate']/DatetimeValue"/>
</xsl:call-template>
</corecom:StartDateTime>
<corecom:EndDateTime>
<xsl:call-template name="formatUDADate">
<xsl:with-param name="formatDateParam" select="ItemUDA[Name='ItemManufacture_
EndDate']/DatetimeValue"/>
</xsl:call-template>
</corecom:EndDateTime>
</corecom:EffectiveTimePeriod>
<corecom:Status>
<xsl:value-of-select="ItemUDA[Name='ItemManufacture_Status']/CharValue"/>
</corecom:Status>
</corecomEBO:ItemManufacture>
</xsl:template>

```

3. Declare each template, which processes the AG in interface file, under ProcessAttributeGroupUDA template.

For example:

```

<xsl:template name="ProcessAttributeGroupUDA">
<xsl:when test="$Name='ItemManufacture'">
<xsl:call-template name="ItemManufactureAttrGeneral"/>
</xsl:when>
</xsl:template>

```

4. Rearrange the UDA elements in restructure transformation file as per the schema.

Example: re-Arrange the UDA Elements as per the schema in restructure transformation file (**XformRestructureSyncItemListEBM.xsl**).

In this file, the reference to the attribute must follow the same sequence of elements as defined in EBM. In this example, if a UDA is mapped to ItemManufacturer, the EBM element ItemManufacturer in the **EBM.XSD** is after Item Supplier. The newly mapped element must follow the same sequence.

For example:

```

<xsl:template match="/">
<ns0:SyncItemListEBM>
<xsl:for-each select="/ns0:SyncItemListEBM/ns0:DataArea">
<ns0:DataArea>
<xsl:apply-templates select="corecom:Sync"/>
<ns0:SyncItemList>
<corecomEBO:ItemSupplier>
</corecomEBO:ItemSupplier>
<xsl:apply-templates select="ns0:SyncItemList/corecom:Revision/corecomEBO:Item
Manufacturer"/>
<xsl:apply-templates select="ns0:SyncItemList/corecom:ItemIdentification/corec
om:Revision/corecomEBO:RelatedItem"/>
<corecomEBO:ItemRevisionHistory>
</ns0:SyncItemList>
</ns0:DataArea>

```

5. Re-deploy the Oracle Product Hub requester service.

## Mapping Custom UDAs to EBM

---

**Note:** This is the recommended approach for handling custom UDA.

---

- Define all the custom AG with Prefix of "XX." For example: XX\_Suppliers.SupplierName.
- In case of custom UDAs not defined by prefixing XX, the attributes will be transformed to specification group in EBM and it will be the responsibility of implementation to handle the transformation of the same as part of the provider service custom extensions.
- The name of the custom attributes must prefix the AG name to avoid duplicate attributes across the AGs. For example, a new attribute SupplierName added to the AG Suppliers will have the internal name Suppliers\_SupplierName.
- It is the responsibility of the provider to consume the data as required. Ready-to-use these will be mapped to Flexfield attributes (User defined attributes user interface) in Siebel CRM.
- No change in the Oracle Product Hub requester is needed to support this.

### Artifacts (Requester ABCS)

The following artifacts can be found under PIPS\Core\PIM\RequestorABCS\SyncItemListPIMReqABCImpl\bpel where the integrations have been downloaded and are installed:

- XformUDAAGInterfaceCommsPIMBOMRespABMToSyncBOMEBM.xml
- XformUDAAGInterfaceCommsPIMItemRespABMToSyncItemListEBM.xml
- XformUDACommsPIMBOMRespABMToSyncBOMEBM\_Custom.xml
- XformUDACommsPIMItemRespABMToSyncItemListEBM\_Custom.xml
- XformRestructureSyncBillOfMaterialsListEBM.xml
- XformRestructureSyncItemListEBM.xml

### Artifacts (Siebel CRM Provider ABCS):

In Provider ABCS, specification group name starting with XX is mapped to Flexfield attributes (User defined attributes user interface) in Siebel CRM.

- XformSyncItemListEBMToSiebelProdABM.xml
- XformUDAInterfaceCommsSiebelPromotionEBMToABM.xml
- XformUDACommsSiebelProductEBMToABM.xml
- XformUDACommsSiebelProductEBMToABM\_Custom.xml
- XformRestructureProductABM.xml
- XformSyncItemListEBMToSiebelPromotionABM.xml
- XformUDAInterfaceCommsSiebelProductEBMToABM.xml
- XformUDACommsSiebelPromotionEBMToABM.xml
- XformUDACommsSiebelPromotionEBMToABM\_Custom.xml
- XformRestructurePromotionABM.xml

## Performance Optimization Using Chunking

As per the statistics and architecture proposal, the input payload for orchestration process should not exceed 2MB or it might lead to transaction time out/heap memory issue. To handle this issue, the Oracle Product Hub Req ABCS service invokes the Oracle Product Hub Query Adapter service to fetch the payload in fixed chunks or size. It processes the list of items/item structure in sub-batch. As part of the Oracle Product Hub Query Adapter services to query the Items and Bill of Materials (BOMs), the services refer to the value of BATCH\_SIZE as defined in Oracle Application Integration Architecture (Oracle AIA) Configuration property at the service level to decide the chunk size.

SQLX technology has been used to query the chunk of item/item structure from Oracle Product Hub batch output table. PL/SQL API: AIA\_MDMITEMBATCH.

### Specifying the Batch Size

To specify the batch size for chunking for items or item structures:

1. QUERY\_ITEM: Modify the value for the configuration property BATCH\_SIZE in AIAConfigurationProperties.xml- for the service QueryItemListPIMAdapter. Set the number of items to be processed in each chunk. This value needs to be set based on typical number of items in batch/payload size and server configuration.

Example: Number of Items in Batch: 30; BATCH\_SIZE: 5

2. QUERY\_BOM: Modify the value for the configuration property BATCH\_SIZE in AIAConfigurationProperties.xml for the service QueryBillOfMaterialsListPIMAdapter. Set the number of item structures to be processed in each chunk. This value needs to be set based on three considerations:
  - Number of item structure in batch/payload size.
  - "The number BOM component that has version structure UDA with Domain\_Type value as 'CLASS' in the particular BOM.
  - "Server configuration.

Example:

Number of item structures (without option class at component level) in batch: 15;  
BATCH\_SIZE: 5

Or

Number of item structures (with three option class at component level) in batch: 15;  
BATCH\_SIZE: 2

This appendix discusses the following scenarios for synchronizing items from Oracle Product Hub to Oracle E-Business Suite when item versions are different:

- [ASCII Code of the Item Version is Higher in Oracle Product Hub](#)
- [ASCII Code of the Default Item Version is Higher in Oracle E-Business](#)

### ASCII Code of the Item Version is Higher in Oracle Product Hub

In this scenario, the API used by Product Master Data Management pre-built integrations has a wrapper on the base API used to create items in E-Business Suite.

To compare effective date of incoming item messages, wrapper API uses this logic:

- If the effective date on the incoming item message is less than the current system datetime, it updates the item message date to system datetime plus 2 seconds.
- If, when you synchronize an item from Oracle Product Hub to E-Business Suite for the first time, the item version number is set higher than the default version for items in E-Business Suite, then the E-Business Suite API first creates a default version of the items and then creates a version synchronized from Oracle Product Hub.

Product Master Data Management pre-built integrations API uses current system datetime as effective date for the default version and then creates the next version of item with the effective date on the item from original system datetime plus 2 seconds set by wrapper API. Effective datetime for the default version of the item must be less than the effective datetime of versions of the item created in E-Business Suite.

There are two possible assumptions concerning the default version of the item:

1. The processing time for the item is less than 2 seconds. See [Table D-1](#).
2. The processing time for the item is more than 2 seconds. See [Table D-2](#).

**Table D-1 When Assumption 1 Applies**

Oracle Product Hub	E-Business Suite	Comments
Item version = 1 Effective datetime=04/15/11 10:00:00	Default version set = 0 System datetime = 04/15/11 10:00:10	Assumes item synchronization takes 10 seconds.
-	Item version = 1 Effective datetime= 04/15/11 10:00:12	Wrapper API sets effective date time to system datetime +2 seconds as the effective datetime is less than system datetime.
-	Item version = 1 Effective start date=04/15/11 10:00:12 Effective end date=04/15/11 10:00:12	Base API creates the default version first. Assumption 1 applies.
-	Item version = 1 Effective start date=04/15/11 10:00:12 Effective end date =	Base API creates next version of the item. Note: Start date of item version 1 is higher than the start date of default item version 0.

**Table D-2 When Assumption 2 Applies**

Oracle Product Hub	E-Business Suite	Comments
Item version = 1 Effective datetime=04/15/11 10:00:00	Default version set = 0 System datetime = 04/15/11 10:00:10	Assumes item synchronization from takes 10 seconds.
-	Item version = 1 Effective datetime=04/15/11 10:00:12	Wrapper API sets effective date time to system datetime +2 seconds as the effective datetime is less than system datetime.
-	Item version = 1 Effective start date=04/15/11 10:01:10 Effective end date =	Base API creates the default version first. Assumption 2 applies as processing of default version takes +2 seconds.
-	Item version = 1 Effective start date=04/15/11 10:01:10 Effective end date =	Base API tries to create next version of the item, effective start date of version 1 is less than the default version, item creation fails.

## Workaround

If your system takes more than 2 seconds to process and create the default version of the item, you must update the value based on their individual requirements.

For example, to update the time interval to 60 seconds, follow these steps:

1. Back up the **Inveipitb.pls** file from your E-Business Suit instance.
2. Open the **Inveipitb.pls** file, and then search for the following:

```
IF (l_item.main_obj_type.effectivity_date IS NULL OR l_item.main_obj_
type.effectivity_date = fnd_api.g_miss_date OR l_item.main_obj_
type.effectivity_date <= sysdate) THEN
l_item.main_obj_type.effectivity_date := sysdate+(2/(60*60*24));
END IF;
```

3. Replace with this code snippet (changes in bold):

```
IF (l_item.main_obj_type.effectivity_date IS NULL OR
l_item.main_obj_type.effectivity_date = fnd_api.g_miss_date OR
l_item.main_obj_type.effectivity_date <= sysdate) THEN
l_item.main_obj_type.effectivity_date := sysdate+(60/(60*60*24));
END IF;
```

4. Save the changes, and then recompile the **Inveipitb.pls** file.

## **ASCII Code of the Default Item Version is Higher in Oracle E-Business**

This scenario is not supported by Oracle.

