

Oracle® Application Integration Architecture

Oracle Product Master Data Management Integration
Implementation Guide

Release 11.1

E27426-03

September 2012

The Oracle Product Master Data Management (MDM) Integration implementation guide is a valuable resource for administrators and developers involved in the implementation, administration and deployment of application integration architecture (AIA) based enterprise-data-management solutions provided by Oracle Application Integration Architecture (AIA) Pre-Built Integrations Release 11.3.

The first part of this guide addresses the integration options available for various participating applications such as Siebel CRM, Oracle E-Business Suite and Oracle Communications Billing and Revenue Management (BRM). It also lists the various assumptions and constraints, process flows, interfaces and integration services used by Product MDM Pre-built integrations.

The second part of this guide discusses how to configure the Oracle Product Hub (OPH) integration which includes adding new spoke instances , post installation configuration steps, cross-references required for integrating the participating applications with Product MDM.

Oracle Application Integration Architecture Oracle Product Master Data Management Integration
Implementation Guide, Release 11.1

E27426-03

Copyright © 2001, 2012, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

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, the following notice is applicable:

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

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

Contents

Preface	xi
What's New in this Guide	xi
Common Oracle AIA Pre-Built Integration Guides	xi
Documentation Accessibility	xii
Additional Resources	xii

Part I Understanding the Delivered Integrations

1 Understanding the Oracle Product Master Data Management Integration

1.1	Overview	1-1
1.1.1	Terminology	1-2
1.2	Process Flows.....	1-4
1.3	Core AIA Components	1-6
1.4	Integration Services	1-7
1.4.1	ItemEBSV2	1-7
1.4.2	ItemResponseEBSV2	1-8
1.4.3	BillOfMaterialsEBS	1-8
1.4.4	BillOfMaterialsReponseEBS	1-8
1.4.5	PriceListEBSV2.....	1-9
1.4.6	PriceListResponseEBSV2	1-9
1.4.7	SpecificationValueSetEBS.....	1-9
1.4.8	SpecificationValueSetResponseEBS	1-10
1.4.9	ClassificationSchemeEBS.....	1-10
1.4.10	ClassificationSchemeResponseEBS	1-10
1.4.11	CommunicationsPriceListEBSV2	1-10
1.4.12	CommunicationsPriceListResponseEBSV2.....	1-11
1.4.13	CommunicationsItemCompositionEBSV1	1-11
1.4.14	CommunicationsItemCompositionResponseEBSV1	1-12

2 Oracle Product Master Data Management Integration Base Pack

2.1	Supported Features and Process Flows	2-1
2.1.1	Common Features.....	2-2
2.2	Synchronization of Items and BOMs	2-2
2.3	Common Features Supported by the Base Pack across Process Flows.....	2-10
2.3.1	Support for Routing Driven Through Explicit Specification of Target Systems	2-10

2.3.2	Support for Multi-Language for Item Synchronization.....	2-11
2.3.3	Support for Telecommunications Seeded Library Attributes	2-11
2.3.3.1	Sellable Product Information Library-Horizontal	2-11
2.3.3.2	Product Management Library-Horizontal.....	2-13
2.3.3.3	Communications Services Billing Library-Vertical	2-13
2.3.3.4	Communications Product Details Library - Vertical	2-14
2.3.3.5	Component UDA for Item Synchronization.....	2-14
2.3.3.6	Support for Multi-Row Attribute Groups.....	2-15
2.3.4	Controlling Publishing of Items to Target System Instances	2-16
2.3.5	Methodology to Set the Value for the Description System Specification Attribute Group 2-16	
2.3.6	Support for Updating Publishing Status in OPH upon Publication of Items.....	2-18
2.4	Oracle Product Hub Interfaces.....	2-19
2.5	Oracle Product Hub Integration Services.....	2-20
2.5.1	Synchronize Item	2-20
2.5.2	Synchronize Price List.....	2-20
2.5.3	Synchronize Bill of Materials	2-21
2.5.4	QueryItemListPIMAdapter	2-21
2.5.5	QueryBillOfMaterialsListPIMAdapter	2-21
2.5.6	SyncSpecificationValueSetListPIMEventConsumer.....	2-21
2.5.7	SyncSpecificationValueSetListPIMReqABCImpl.....	2-21
2.6	Assumptions and Constraints for OPH	2-22

3 Oracle Product Master Data Management Integration Option for Siebel CRM

3.1	Supported Features.....	3-1
3.1.1	Synchronization of Metadata	3-2
3.1.2	Synchronization of Item Catalog Categories	3-2
3.1.3	Synchronization of Relationships and Structure Under Item Catalog Categories.....	3-3
3.1.4	Synchronization of Attribute Groups as Part of Item Catalog Categories.....	3-6
3.1.5	Synchronization of Customer UDA as Part of Item Catalog Categories.....	3-6
3.1.6	Synchronization of Transaction Attributes as Part of Item Catalog Categories.....	3-7
3.1.7	Transaction Attributes without Valuesets	3-8
3.1.8	Synchronization of Item Catalog Category Hierarchies	3-8
3.1.9	Association of Item Catalog Category with Items	3-9
3.1.10	Publishing of Item Catalog Category from OPH to Siebel CRM Implementation Flow .. 3-9	
3.1.11	Synchronization of Valuesets.....	3-14
3.1.12	Synchronization of Valuesets as Part of Item Catalog Categories.....	3-14
3.1.13	Synchronization of Valuesets Independently.....	3-15
3.1.14	Batch and Initial Load	3-18
3.1.15	Multi Language Support for Valueset Synchronization	3-19
3.1.16	Synchronization of Products, Discounts, and Associated Structures from Oracle Product Hub to Siebel CRM 3-19	
3.1.17	Multi-Event Product Synchronization from OPH to Siebel	3-19
3.1.18	Support for Class Type Relationship in Product Synchronization for Siebel CRM 3-19	
3.1.19	Support for Controlling Auto-Release of Entities Published from Oracle Product Hub in the Same Batch for Siebel 3-21	
3.1.20	Synchronization of Promotions from Product Hub to Siebel CRM	3-23

3.2	Siebel CRM Interfaces.....	3-27
3.3	Siebel CRM Integration Services.....	3-28
3.3.1	SyncProductSiebelProvABCImpl.....	3-28
3.3.2	SyncBillOfMaterialsListSiebelProvABCImpl	3-28
3.3.3	SyncClassificationSchemeListSiebelProvABCImpl	3-29
3.3.4	SyncSpecificationValueSetListSiebelProvABCImpl	3-29
3.3.5	ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl.....	3-29
3.3.6	SyncItemCompositionListSiebelCommsProvABCImpl.....	3-29
3.4	Assumptions and Constraints for the Siebel CRM Option.....	3-30

4 Oracle Product Master Data Management Integration Option for Oracle E-Business Suite

4.1	Overview	4-1
4.1.1	Support for Operational Attributes	4-1
4.2	Synchronizing Items and BOMs from Oracle Product Hub to Oracle E-Business Suite	4-2
4.3	Oracle E-Business Suite Interfaces.....	4-2
4.4	Oracle E-Business Suite Integration Services.....	4-3
4.4.1	SyncItemListEbizProvABCImpl.....	4-3
4.4.2	SyncBillOfMaterialsListEbizProvABCImpl.....	4-3
4.4.3	SyncItemListEbizAdapter.....	4-4
4.4.4	SyncBillOfMaterialsListEbizAdapter.....	4-4
4.4.5	TransformAppContextEbizService	4-4
4.4.6	QueryResponsibilityEbizAdapter	4-4
4.5	Assumptions and Constraints for the Oracle E-Business Suite Option	4-4

5 Oracle Product Master Data Management Integration Option for Oracle Communications BRM

5.1	Synchronization of Billing Products from Oracle Product Hub to Oracle BRM	5-1
5.1.1	Multi-Event Product Synchronization from OPH to Siebel	5-3
5.1.2	Synchronization of Discounts and Discount Models from Oracle Product Hub to Siebel CRM and Oracle BRM	5-7
5.1.3	Viewing Publishing Status of Items	5-9
5.2	Oracle BRM Interfaces.....	5-9
5.3	Oracle BRM Integration Services	5-9
5.3.1	SyncItemListBRMProvABCImpl.....	5-9
5.3.2	SyncPriceListListBRMProvABCImpl	5-9
5.4	Assumptions and Constraints for the Oracle Communications BRM Option.....	5-10

Part II Implementing the Delivered Process Integrations

6 Configuring the Oracle Product Hub Integration

6.1	Adding a New Spoke Instance.....	6-1
6.2	Post Installation Configuration Set Up	6-4
6.2.1	Extract Inventory Organization and Operating Unit IDs in Oracle E-Business Suite	6-5

6.2.1.1	Identify the Item Validation Organization for Oracle E-Business Suite.....	6-5
6.2.2	Extract Inventory Location and Operating Unit IDs in Oracle Product Hub.....	6-7
6.2.2.1	Identify Item Validation Organization for Oracle Product Hub.....	6-7
6.2.3	Mapping Siebel CRM.....	6-9
6.2.4	Extract Business Unit and Inventory Location IDs in Siebel CRM.....	6-10
6.3	Setting Up Oracle BRM.....	6-10
6.4	Setting Up Cross-References.....	6-10
6.4.1	Populating Seed Values in Cross-Reference Tables.....	6-11
6.4.2	Validating Cross-References.....	6-14
6.4.3	Creating a Pricelist in Siebel.....	6-15
6.5	Working with Cross-References.....	6-15
6.6	Working with Domain Value Maps.....	6-16
6.7	Handling Errors.....	6-22
6.8	Setting Configuration Properties.....	6-22
6.8.1	Settings for the SyncItemListEbizProvABCServiceImpl service.....	6-22
6.8.2	Settings for the SyncProductSiebelProvABCServiceImpl service.....	6-24
6.8.3	Settings for the SyncBillOfMaterialsListEbizProvABCServiceImpl service.....	6-26
6.8.4	Settings for the SyncBillOfMaterialsListSiebelProvABCServiceImpl service.....	6-27
6.8.5	Settings for the SyncItemListPIMReqABCServiceImpl service.....	6-30
6.8.6	Settings for SyncSpecificationValueSetListSiebelProvABCServiceImpl service.....	6-33
6.8.7	Settings for the SyncItemListBRMProvABCServiceImpl service.....	6-36
6.8.8	Settings for the SyncPriceListListBRMProvABCServiceImpl service.....	6-37
6.8.9	Settings for the QueryBillOfMaterialsListPIMAdapter service.....	6-38
6.8.10	Settings for the QueryItemListPIMAdapter service.....	6-38
6.8.11	Settings for the SyncSpecificationValueSetListPIMReqABCServiceImpl service.....	6-39
6.8.12	Settings for the SyncItemCatalogCategoryPIMReqABCServiceImpl service.....	6-40
6.8.13	Settings for the SyncClassificationSchemeListSiebelProvABCServiceImpl service.....	6-42
6.8.14	Settings for the SyncClassificationSchemeListSiebelProvABCServiceImpl service.....	6-43
6.8.15	Settings for the ProductOptimizedSyncPriceListListSiebelCommsProvABCServiceImpl service.....	6-45
6.8.16	Settings for the SyncItemCompositionListSiebelCommsProvABCServiceImpl service.....	6-46
6.9	Setting Up the Data for Seeded Attributes of the Telecommunications Library.....	6-47
6.10	Setting up List of Values in Siebel.....	6-48

A Frequently Asked Questions

B Support for Rate Tiers and Effectivity

C User Defined Attributes Framework

C.1	Extending for Custom UDAs.....	C-1
C.1.1	Handling UDAs.....	C-1
C.1.2	Mapping Artifacts.....	C-2
C.1.2.1	Mapping Operational Attributes to ItemEBM.....	C-2
C.1.2.2	Mapping Seeded Set of UDAs for Communications library to EBM.....	C-2
C.1.2.3	Mapping Pricelist related UDAs for Communications Library to EBM.....	C-2
C.2	Implementation of UDA Transformation.....	C-3
C.2.1	Mapping Custom UDAs to EBM.....	C-5

C.2.2	Artifacts (Requester ABCS)	C-5
C.2.3	Artifacts (Siebel Provider ABCS):.....	C-6
C.3	Performance Optimization Using Chunking.....	C-6
C.3.1	Specifying the Batch Size	C-6

D Scenarios

D.1	ASCII Code of the item version is set same in OPH and Oracle E-Business Suite	D-1
D.2	ASCII Code of the item version is set higher in OPH and is set lower in Oracle E-Business Suite	D-1
D.2.1	Assumption	D-2
D.2.2	Workaround	D-3
D.3	ASCII Code of the default item version is set higher in Oracle E-Business Suite and is set lower in OPH	D-3

List of Tables

1-1	Terminology.....	1-2
1-2	Terminology Used by Participating Applications	1-4
2-1	Item and BOM integration sequence steps.....	2-7
2-2	Promotion example.....	2-16
2-3	Example	2-17
3-1	Product and class types.....	3-4
3-2	Attributes	3-5
3-3	Structure attributes	3-5
3-4	Relationship domain	3-5
3-5	Sequence steps.....	3-11
3-6	Sequence steps.....	3-17
3-7	Example of how product and class type relationships are supported in OPH	3-20
3-8	OPH definitions.....	3-20
3-9	Attributes for root items in Siebel.....	3-20
3-10	Relationship domain	3-21
3-11	Exclusions within option class BOMs	3-25
5-1	Supported billing products	5-1
5-2	5-4
5-3	Seeded attributes.....	5-7
6-1	Sample DVMs.....	6-2
6-2	Examples for System Name & Codes	6-3
6-3	XRef Table Format	6-11
6-4	Organization_ID XRef Table Value Format	6-11
6-5	Sample Data for XRef Organization_ID Table.....	6-12
6-6	INVENTORY_LOCATION_ID XRef Table Value Format	6-13
6-7	Sample Data for XRef INVENTORY_LOCATION_ID Table	6-14
6-8	OPH cross-references	6-15
6-9	List of DVMs	6-16
6-10	Sample Data for STRUCTURE_TYPE DVM:	6-22
6-11	SyncItemListEbizProvABCImpl	6-22
6-12	SyncProductSiebelProvABCImpl	6-24
6-13	SyncBillOfMaterialsListEbizProvABCImpl	6-26
6-14	SyncBillOfMaterialsListSiebelProvABCImpl.....	6-27
6-15	SyncItemListPIMReqABCImpl	6-30
6-16	SyncSpecificationValueSetListSiebelProvABCImpl	6-33
6-17	SyncItemListBRMProvABCImpl	6-36
6-18	SyncPriceListListBRMProvABCImpl.....	6-37
6-19	QueryBillOfMaterialsListPIMAdapter	6-38
6-20	QueryItemListPIMAdapter	6-38
6-21	SyncSpecificationValueSetListPIMReqABCImpl.....	6-39
6-22	SyncItemCatalogCategoryPIMReqABCImpl.....	6-40
6-23	SyncClassificationSchemeListSiebelProvABCImpl	6-42
6-24	SyncClassificationSchemeListSiebelProvABCImpl	6-43
6-25	ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl	6-45
6-26	SyncItemCompositionListSiebelCommsProvABCImpl.....	6-46
6-27	Attributes	6-48
6-28	DVM values	6-48
B-1	Support for rate tiers and effectivity	B-1
C-1	Example of UDA AG	C-3
D-1	When assumption 1 applies	D-2
D-2	When assumption 2 applies	D-2

List of Figures

1-1	Business process flow	1-5
2-1	Synchronization of items and BOMs from OPH to Oracle BRM, Siebel CRM and Oracle E-Business Suite 2-4	
2-2	Item and BOM integration sequence.....	2-7
3-1	Synchronization of ICC/Product Class created in OPH.....	3-10
3-2	Synchronization of ICC in OPH.....	3-11
3-3	Valueset publish from OPH to Siebel CRM	3-16
3-4	Valueset publish from OPH to Siebel CRM	3-17
3-5	An example of component exclusions	3-24
3-6	ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl	3-29
3-7	SyncItemCompositionListSiebelCommsProvABCImpl.....	3-30
5-1	Pricing Hierarchy	5-5
5-2	Conceptual model for discount entities.....	5-7
6-1	Oracle applications home page example.....	6-6
6-2	System parameters.....	6-6
6-3	Oracle applications home page example.....	6-8
6-4	System parameters.....	6-8
6-5	Administration - Group tab in Siebel.....	6-9
6-6	Creating Inventory Locations in Siebel.....	6-10
C-1	Adding UDA to an FCA in Item EBM	C-3
C-2	Code Sample	C-4
C-3	Code sample	C-4
C-4	Code sample	C-5

Preface

Welcome to the Oracle Product Master Data Management Integration Implementation Guide.

What's New in this Guide

- The Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations is restructured into a general installation chapter with an individual configuration and deployment chapter for each pre-built integration.
- The term *process integration pack* is replaced with the term *pre-built integrations*.
- The implementation guides are restructured into two parts: design and set up.
 - Part I - Design: This part provides functional overviews, activity diagrams, assumptions and constraints, and technical sequence diagrams and steps.
 - Part II - Set up: This part provides prerequisites, data requirements, and configuration steps.
- Starting with this release, these integrations are no longer available:
 - Oracle CRM On Demand Integration Pack for JD Edwards EnterpriseOne: Lead to Order
 - Oracle Workforce Administration Integration Pack for PeopleSoft Human Resources

Common Oracle AIA Pre-Built Integration Guides

Oracle Application Integration Architecture Pre-Built Integrations 11.1 includes the following guides shared by all products delivered with this release:

- Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations Release 11.1
 - This guide provides an overview of the installation process, including how to install, configure, and deploy your pre-built integrations. The steps required to upgrade your pre-built integrations to the latest release are also provided.
- Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide
 - This guide describes:
 - How to work with and configure Session Pool Manager (SPM), which is a service in the Oracle SOA Suite web server whose primary function is to manage a pool of web server session tokens that can be reused by BPEL flows.

- How to deploy and configure the AIACompositeScheduler. This is a utility component that is used by pre-built integrations to schedule a service-oriented architecture (SOA) composite to be invoked at the specified time interval.
- Oracle Application Integration Architecture Pre-Built Integrations 11.1: Product-to-Guide Index

The Product-to-Guide index lists the guides that provide information for each product delivered in this release.

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 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.

Additional Resources

The following resources are also available:

- **Oracle Application Integration Architecture Foundation Pack:**
Oracle AIA Pre-Built integrations require Foundation Pack 11.1.1.5.0 to be installed. Refer to the Foundation Pack documentation library on OTN to download the Foundation Pack guides at http://download.oracle.com/docs/cd/E21764_01/aia.htm.
- **Oracle Application Integration Architecture: Product-to-Guide Index:**
Oracle Technology Network: <http://www.oracle.com/technetwork/index.html>
- **Known Issues and Workarounds:**
My Oracle Support: <https://support.oracle.com/>
- **Release Notes:**
Oracle Technology Network: <http://www.oracle.com/technetwork/index.html>
- **Documentation updates:**
Oracle Technology Network: <http://www.oracle.com/technetwork/index.html>

Part I

Understanding the Delivered Integrations

Part I contains the following chapters:

- Chapter 1, "Understanding the Oracle Product Master Data Management Integration"
- Chapter 2, "Oracle Product Master Data Management Integration Base Pack"
- Chapter 3, "Oracle Product Master Data Management Integration Option for Siebel CRM"
- Chapter 4, "Oracle Product Master Data Management Integration Option for Oracle E-Business Suite"
- Chapter 5, "Oracle Product Master Data Management Integration Option for Oracle Communications BRM"

Understanding the Oracle Product Master Data Management Integration

This chapter provides an overview of the Oracle Product Master Data Management (MDM) Integration.

This chapter includes the following sections:

- [Section 1.2, "Process Flows"](#)
- [Section 1.3, "Core AIA Components"](#)
- [Section 1.4, "Integration Services"](#)

1.1 Overview

The Product MDM integration provides industry-neutral and application integration architecture (AIA) based enterprise-data-management solution that enables customers to centralize all product information from heterogeneous systems to create a single view of product information. This product-mastering solution harnesses the power and the flexibility of the best-in-class front-office and back-office applications.

Oracle Product Hub (OPH) (previously Product Information Management [PIM]) provides a flexible user-defined attribute (UDA) framework that enables information to be modeled in a number of ways so that a single definition of an entity can be established and subsequently propagated across multiple participating applications.

The Product MDM integration supports concept-to-launch processes in which the metadata, items, and bill of materials (BOM) are defined in Oracle Product Hub and then published to the registered target applications. The Product MDM integration also supports component exclusions and overrides in the context of items and BOMs, and the telecommunications library (a set of seeded attributes in OPH to support cross-industry application entities). Use the telecommunications library to model communications-related entities such as billable products, discounts, pricing, rules such as compatibility, upgrade and downgrade, promotions, promotion commitments and charges, promotion based discounting, and so on.

Customers can select one or more participating applications when installing OPH. In addition, the Product MDM integration can be configured to route the messages to only the intended applications, which eliminates the need for publishing the entire payload to applications that are not affected by the data change. The integration sends the publishing status back to the Oracle Product Hub, which is maintained for each target application at the entity level.

The details about the integration flows from Oracle Product Hub to the integration options are described in Chapter 2: Oracle Product Master Data Management Integration Base Pack.

Similarly, the integration flows from the Oracle Product Hub to an integration option along with the services and interfaces involved are described in:

[Chapter 3, "Oracle Product Master Data Management Integration Option for Siebel CRM"](#).

[Chapter 4, "Oracle Product Master Data Management Integration Option for Oracle E-Business Suite"](#).

[Chapter 5, "Oracle Product Master Data Management Integration Option for Oracle Communications BRM"](#).

The application and middleware components supported by Product MDM integration are:

- Oracle Fusion Middleware.
- AIA Foundation Pack.
- Oracle Product Hub.
- Siebel Customer Relationship Management (CRM).
- Oracle E-Business Suite.
- Oracle BRM.

For more information about required software, versions, patches, and installing the participating applications (integration options) download the latest version of the *Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations* available on Oracle Technology Network. This guide is continually updated.

1.1.1 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 Terminology

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 mechanism to inherit attributes and relationships to subcategories and products belonging to categories.

Table 1–1 (Cont.) Terminology

Term	Definition
Transaction attributes	<p>These attributes are associated with the item catalog category. The attribute definition and associated valuesets (domain, values) for the attributes are defined at the time of definition of the item catalog category. The attributes are then inherited by the product.</p> <p>The value of the attribute in the product is captured at run time during a business transaction such as a sales order or a quote. These kinds of attributes capture characteristics and properties of a transaction involving a specific instance of a product.</p>
Static attributes	<p>These attributes are associated with the item catalog category. The attribute definition and associated valuesets (domain, values) for the attributes are defined at the time of definition of the item catalog category. The attributes are then inherited by the product.</p> <p>The value of the attribute in the product is captured at design time or when the product is being defined. These attributes capture characteristics, properties, or features of a product.</p>
Structure	<p>A structure or a bill of material (BOM) is a listing of all the subassemblies, intermediates, parts, and raw materials that make up the parent assembly, showing the quantities of each subassembly required to make the assembly. A structure (BOM) is used to compose complex products out of simpler products.</p>
Relationships	<p>A relationship captures the relationship of one entity to another for business purposes, for example, substitute items, up-sell, cross-sell, and so on. The items and BOMs are added as relationships to the item catalog category (ICC).</p>
User-defined attributes	<p>User-defined attributes capture all the detailed information (such as cost information) about an object (items, change requests, or change orders). The item catalog enables you to create user-defined attributes with validation logic to support the needs of your organization. To do so, create valuesets and associate the valuesets with user-defined attributes. Attributes can have a static or dynamic list of values or a range of values.</p>
Valuesets	<p>Valuesets define the possible set of values that can be assigned for an attribute. For the UDA, the values are assigned at design time and for transaction attributes the values are assigned at run time</p>

For more information about the terminology used in this guide, see the product documentation for the relevant participating application.

This table lists some key differences in the terminology used by each application while referring to the same entity:

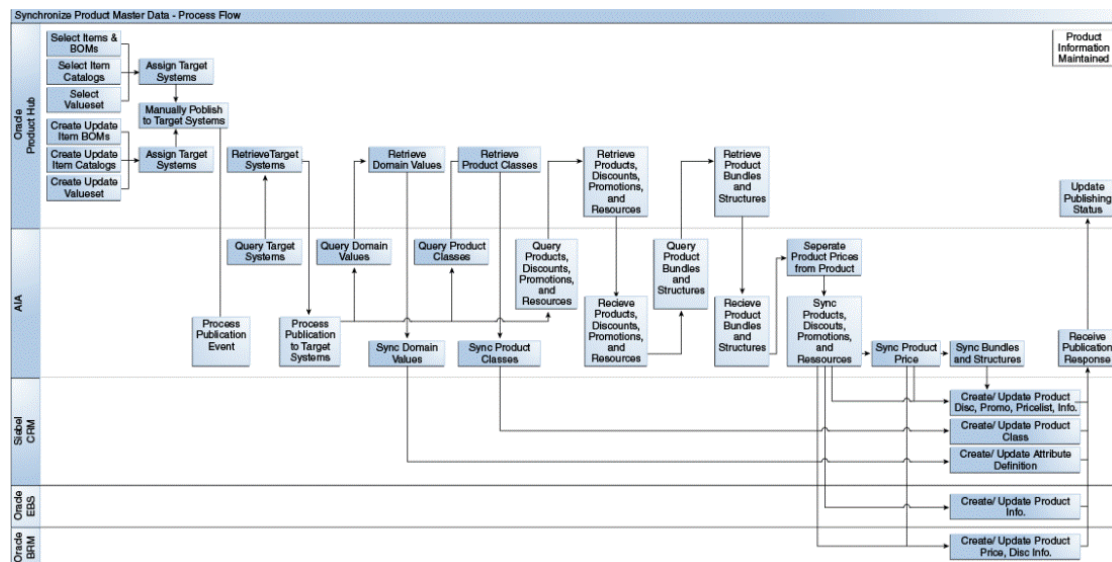
Table 1–2 Terminology Used by Participating Applications

Entity	OPH	Siebel CRM	Oracle E-Business Suite	Oracle BRM
Bundles	Item with BOM	Customizable product	Item with BOM	N/A
Promotions	Item	Promotion	Item	N/A
Inventory location or inventory organization	Inventory organization	Inventory location	Inventory organization	NA
Product structure	BOM	Customizable product	BOM	NA
Business unit or operating unit	Operating unit	Business unit	Operating unit	NA
Product	Item	Product	Item	Product
Discount	Item	Product	NA	Discount
Product class	Item catalog category	Product class	NA	NA
Attribute groups	Attribute groups	NA	NA	NA
Static attributes	Static attributes (seeded)	Fields or first class attributes of product	Operational attributes	Fields or first class attributes of product
Transaction attributes	Transaction attributes	Attributes of product class	NA	NA
Valueset of static attributes	Valuesets of static attributes	List of values	NA	NA
Dynamic or versioned valuesets	Valuesets (dynamic or versioned value sets)	Attribute definitions	NA	NA
Customer added static UDA	Customer added static UDA	Flexible attributes	NA	Customer added fields
Structure elements of ICC	Structure elements of the ICC	Relationships/structure of the product class	NA	NA

1.2 Process Flows

This diagram illustrates the business process flow between the Oracle Product Hub and one or more participating applications, as applicable.

Figure 1-1 Business process flow



These are the business-process-related tasks that provide a contextual framework for the solution:

1. The product administrator defines a common product data model using item catalog category (ICC) in Oracle Product Hub with a comprehensive definition of the products. Once the ICC is defined, the product administrator creates or updates new product definitions through the life cycle of the product from concept to launch and then on to obsolescence.

The valuesets can be published independently or during the ICC publish from the Oracle Product Hub publication framework.

The publication framework provides a user interface to publish item catalog categories to one or more downstream participating applications. The downstream applications must be registered within publication framework.

2. The launching of a new product and its maintenance includes:

Item catalog definition and hierarchy ICC (Siebel product classes).

Attribute definitions and their association with ICC and attribute metadata that may include the attribute's valueset or any other data of the attribute.

Basic item information such as name, product line, description, and so on.

This information is captured in primary and user-defined attributes in Oracle Product Hub. In addition, Oracle Product Hub supports seeded operational attributes that control the behavior of the item in various functional Oracle E-Business Suite modules such as Order Management and Procurement. The operational attributes are part of the definition of the item and are intended for consumption by Oracle E-Business Suite.

Item bill of material (BOM) information (hierarchical structure of the product with components).

This structure also includes the component-level attributes and attributes with regard to the relationship between parent and child items in a hierarchical relationship. You may want to have BOM synchronization with item definitions and a separate BOM synchronization process without item definitions that may be needed in certain cases.

3. After creating or updating the common set of data, the product administrator publishes the information to the participating applications. The publish mechanism is triggered through the publication framework provided by Oracle Product Hub. The integration sends the publishing status back to the Oracle Product Hub. The status is maintained for each target application spoke at each entity level.

For more information about the concepts and terms related to Oracle Product Hub, see the *Oracle Product Hub Implementation Guide*.

1.3 Core AIA Components

The core application integration architecture (AIA) components used by the integration are:

- ItemEBO
- SyncItemListEBM
- SyncItemListResponseEBM
- BillOfMaterialsEBO
- SyncBillOfMaterialsListEBM
- SyncBillOfMaterialsListResponseEBM
- PriceListEBO
- SyncPriceListEBM
- SyncPriceListResponseEBM
- SpecificationValueSetEBO
- SyncSpecificationValueSetListEBM
- SyncSpecificationValueSetListResponseEBM
- ClassificationSchemeEBO
- SyncClassificationSchemeListEBM
- SyncClassificationSchemeListResponseEBM
- Communications/PriceListEBO
- Communications/ItemCompositionEBO
- SyncItemCompositionListEBM
- SyncItemCompositionListResponseEBM

The core EBO and EBM XSD files can be located by EBO within the `$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/` parent folder.

The core EBS WSDL files can be located by EBO within the `$AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/` parent folder.

For detailed documentation of individual EBOs and EBMs, click the AIA Reference Doc link on EBO and EBM detail pages in the Oracle Enterprise Repository (OER).

For more information about using the OER and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application*

Integration Architecture Foundation Pack, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."

EBOs can be extended, for instance, to add new data elements. These extensions are protected and will remain intact after a patch or an upgrade.

For more information, see *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack, "Understanding Extensibility."*

1.4 Integration Services

These are the AIA integration services delivered with this integration:

- ItemEBSV2
- ItemResponseEBSV2
- BillOfMaterialsEBS
- BillOfMaterialsReponseEBS
- PriceListEBSV2
- PriceListResponseEBSV2
- SpecificationValueSetEBS
- SpecificationValueSetResponseEBS
- ClassificationSchemeEBS
- ClassificationSchemeResponseEBS
- CommunicationsPriceListEBSV2
- CommunicationsPriceListResponseEBSV2
- CommunicationsItemCompositionEBSV1
- CommunicationsItemCompositionResponseEBSV1

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."*

1.4.1 ItemEBSV2

ItemEBSV2 is implemented as a Mediator routing service. It exposes all of the enterprise operations that can be performed with an item enterprise business object. The EBS is used to route the SyncItemEBM received from OPH to the various participating applications.

In the Oracle Product Hub integration, this service implements the SyncItemList operation. The SyncItemList operation on ItemEBS invokes SyncItemListEbizProvABCImpl and SyncProductSiebelProvABCImpl.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack, "Designing and Developing Enterprise Business Services"* and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack, "Understanding Enterprise Business Services."*

1.4.2 ItemResponseEBSV2

ItemResponseEBSV2 is implemented as a routing service. It exposes all of the enterprise operations that can be performed with an item enterprise business object.

In the Oracle Product Hub integration, this service implements the SyncItemListResponse operation. The SyncItemListResponse operation on ItemResponseEBS transfers the response message from SyncProductListSiebelProvABCImpl and SyncItemListEbizProvABCImpl to SyncItemListPIMReqABCImpl.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

1.4.3 BillOfMaterialsEBS

BillOfMaterialsEBS is implemented as a Mediator routing service. It exposes all of the enterprise operations that can be performed with a BillOfMaterials enterprise business object.

In the Oracle Product Hub integration, this service implements the SyncBillOfMaterialsList operation. The SyncItemListPIMReqABCImpl invokes the SyncBillOfMaterials operation on the BillOfMaterialEBS. The BillOfMaterialsEBS calls SyncBillOfMaterialsListSiebelProvABCImpl and SyncBillOfMaterialsListEbizProvABCImpl (and other participating applications provider services).

This service synchronizes the product structure into the Siebel CRM and Oracle E-Business Suite systems. The assumption is that the components that are part of the product structure are already synchronized into Siebel CRM and Oracle E-Business Suite as part of the item synchronization flow.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

1.4.4 BillOfMaterialsResponseEBS

BillOfMaterialsResponseEBS is implemented as a Mediator routing service. It exposes all of the enterprise operations that can be performed with a BillOfMaterials enterprise-business-object.

In the Oracle Product Hub integration, this service implements the SyncBillOfMaterialsListResponse operation. The SyncBillOfMaterialsListResponse operation on BillOfMaterialsResponseEBS transfers the response message from SyncBillOfMaterialsListSiebelProvABCImpl and SyncBillOfMaterialsListEbizProvABCImpl (and other participating applications provider services) to SyncItemListPIMReqABCImpl.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

1.4.5 PriceListEBSV2

PriceListEBSV2 is implemented as a Mediator routing service. It exposes all of the enterprise operations that can be performed with a price list enterprise-business-object.

In the Oracle Product Hub integration, this service implements the SyncPriceListList operation. The SyncItemListPIMReqABCImpl invokes the SyncPriceListList operation on the PriceListEBSV2. The PriceListEBSV2 has routing rules for the various systems. For Siebel, the message is routed to the CommunicationsPricelistEBSV2 and SyncPriceListList for handling the synchronization of billing pricelists to Siebel.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

1.4.6 PriceListResponseEBSV2

PriceListResponseEBSV2 is implemented as a Mediator routing service. It exposes all of the enterprise operations that can be performed with a price list enterprise-business-object.

In the Oracle Product Hub integration, this service implements the SyncPriceListListResponse operation. The SyncPriceListListResponse operation on PriceListResponseEBSV2 transfers the response message from CommunicationsPricelistEBSV2 to the SyncItemListPIMReqABCImpl.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

1.4.7 SpecificationValueSetEBS

SpecificationValueSetEBS is implemented as a lightweight Mediator routing service. It exposes all of the enterprise operations that can be performed with a SpecificationValueSet enterprise-business-object.

In the Oracle Product Hub integration, this service implements the SyncSpecificationValueSetList operation. The SyncItemCatalogCategoryPIMReqABCImpl and SyncSpecificationValueSetListPIMReqABCImpl invoke the SyncSpecificationValueSetList operation on the SpecificationValueSetEBS to synchronize valuesets. The routing rule in the operation routes to the Siebel provider for valueset SyncSpecificationValueSetListSiebelProvABCImpl.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

1.4.8 SpecificationValueSetResponseEBS

SpecificationValueSetResponseEBS is implemented as a lightweight Mediator routing service. It exposes all of the enterprise operations that can be performed with a SpecificationValueSet enterprise business object.

In the Oracle Product Hub integration, this service implements the SyncSpecificationValueSetListResponse operation. The SyncSpecificationValueSetListResponse operation on SpecificationValueSetResponseEBS transfers the response message to the SyncSpecificationValueSetListPIMReqABCImpl or SyncItemCatalogCategoryPIMReqABCImpl, depending on which invokes the message.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

1.4.9 ClassificationSchemeEBS

In the Oracle Product Hub integration, this service implements the SyncClassificationSchemeList operation. The SyncItemCatalogCategoryPIMReqABCImpl service invokes the SyncClassificationSchemeList operation on the ClassificationSchemeEBS to synchronize ICCs from OPH. The routing rule in the operation routes to the Siebel provider for valueset SyncClassificationSchemeListSiebelProvABCImpl.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

1.4.10 ClassificationSchemeResponseEBS

ClassificationSchemeResponseEBS exposes all of the enterprise operations that could be performed with a ClassificationResponse enterprise-business-object.

In the Oracle Product Hub integration, this service implements the SyncClassificationListResponse operation. The SyncClassificationListResponse operation on ClassificationSchemeResponseEBS transfers the response message to SyncItemCatalogCategoryPIMReqABCImpl.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

1.4.11 CommunicationsPriceListEBSV2

CommunicationsPriceListEBSV2 is implemented as a Mediator routing service. It exposes all of the enterprise operations that can be performed with a communications pricelist enterprise-business-object.

In the Oracle Product Hub integration, this service implements the SyncPriceListList operation. The PricelistEBSV2 invokes this operation, and it transfers the request to

ProductOptimizedSyncPriceListListSiebelCommsProvABCSEImpl for billing pricelists that need to be synchronized to Siebel.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

For more information about Communications-specific services or flows, see the *Oracle Communications Order to Cash Integration Pack for Siebel CRM*, *Oracle Communications Order and Service Management*, and *Oracle Communications Billing and Revenue Management Implementation Guide*, or the *Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide*.

1.4.12 CommunicationsPriceListResponseEBSV2

CommunicationsPriceListResponseEBSV2 is implemented as a Mediator routing service. It exposes all of the enterprise operations that can be performed with a communications pricelist enterprise-business-object.

In the Oracle Product Hub integration, this service implements the SyncPriceListListResponse operation. It transfers the response to PricelistEBSResponseEBSV2.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

For more information about Communications-specific services or flows, see the *Oracle Communications Order to Cash Integration Pack for Siebel CRM*, *Oracle Communications Order and Service Management*, and *Oracle Communications Billing and Revenue Management Implementation Guide*, or the *Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide*.

1.4.13 CommunicationsItemCompositionEBSV1

CommunicationsItemCompositionEBSV1 performs the entire item/product-related actions such as create product/item, update product/item, and synchronize product/item. Based on the routing rules set up, it invokes a provider application business connector service (ABCS).

In the Oracle Product Hub integration, this service implements the SyncItemCompositionList operation. The ProductOptimizedSyncPriceListListSiebelCommsProvABCSEImpl invokes this operation, and it transfers the request to SyncItemCompositionListSiebelCommsProvABCSEImpl for Siebel.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

For more information about Communications-specific services or flows, see the *Oracle Communications Order to Cash Integration Pack for Siebel CRM*, *Oracle Communications Order and Service Management*, and *Oracle Communications Billing and Revenue Management Implementation Guide*, or the *Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide*.

1.4.14 CommunicationsItemCompositionResponseEBSV1

CommunicationsItemCompositionResponseEBSV1 is implemented as a Mediator routing service. It exposes all of the enterprise operations that can be performed with a communications pricelist enterprise-business-object.

In the Oracle Product Hub integration, this service implements the SyncItemCompositionListResponse operation. It transfers the response to ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl.

For more information about this EBS, see the *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Designing and Developing Enterprise Business Services" and *Oracle Fusion Middleware Concepts and Technologies Guide for Oracle Application Integration Architecture Foundation Pack*, "Understanding Enterprise Business Services."

For more information about Communications-specific services or flows, see the *Oracle Communications Order to Cash Integration Pack for Siebel CRM*, *Oracle Communications Order and Service Management*, and *Oracle Communications Billing and Revenue Management Implementation Guide*, or the *Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide*.

Oracle Product Master Data Management Integration Base Pack

The Oracle Product Hub (OPH) provides single, blended records through consolidation, cleansing, and governance of product information and publishes the enriched, cleansed data to the systems that need this information for individual, application-specific functionality.

For more information about the concepts and terms related to Oracle Product Hub, see *Oracle Product Hub Implementation Guide* documentation.

This chapter includes the following sections:

- [Section 2.1, "Supported Features and Process Flows"](#)
- [Section 2.2, "Synchronization of Items and BOMs"](#)
- [Section 2.3, "Common Features Supported by the Base Pack across Process Flows"](#)
- [Section 2.4, "Oracle Product Hub Interfaces"](#)
- [Section 2.5, "Oracle Product Hub Integration Services"](#)
- [Section 2.6, "Assumptions and Constraints for OPH"](#)

2.1 Supported Features and Process Flows

The following features are supported by the base pack. These are implemented as end-end process flows for various applications in the Product MDM integration.

1. Synchronization of Metadata.

- Synchronization of Item Catalog Categories:
 - Synchronization of Relationships/Structure under Item Catalog Categories.
 - Synchronization of Attribute Groups as part of Item Catalog Categories.
 - Synchronization of Customer UDA as part of Item Catalog Categories.
 - Synchronization of Transaction attributes as part of Item Catalog Categories.
 - Synchronization of Item Catalog Category Hierarchies.
- Synchronization of Valuesets:
 - Synchronization of Valuesets as part of Item Catalog Categories.
 - Synchronization of Valuesets Independently.

- Multi Language Support for Valuesets.
- Options Supported
Siebel integration option: Since these are implemented as end-end process flows, see [Chapter 3, "Oracle Product Master Data Management Integration Option for Siebel CRM"](#) for functional and technical explanations of the features.

2. Synchronization of Items and BOMs

Options supported:

- Oracle E-Business Suite Option: these flows are implemented as end-to-end process flows. For functional and technical explanations of the features, see [Chapter 4, "Oracle Product Master Data Management Integration Option for Oracle E-Business Suite"](#).
- Siebel Option: these flows are implemented as end-to-end process flows. For functional and technical explanations of the features, see [Chapter 3, "Oracle Product Master Data Management Integration Option for Siebel CRM"](#).
- BRM Option: these flows are implemented as end-to-end process flows. For functional and technical explanations of the features, see [Chapter 5, "Oracle Product Master Data Management Integration Option for Oracle Communications BRM"](#).

2.1.1 Common Features

These features are supported by all of the process flows for all of the integration options:

1. Support for routing driven through explicit specification of target systems.
2. Support for multi-language for item synchronization.
3. Support for telecommunications seeded library attributes.
4. Support for updating publishing status in OPH upon publication of items.

For more information about the features applicable to an integration option, see the chapter for that integration option.

2.2 Synchronization of Items and BOMs

The synchronization of items and BOMs is used by all of the integration options supported by the Product MDM integration.

This synchronization enables definition of products, discounts, and promotions as items in the Oracle Product Hub and publishes these to one or more Siebel CRM instances to be used by the order capture process as well as other processes. It enables you to organize simple products into structures and to select the desired structure to publish along with products and promotions.

The synchronization reduces manual effort in keeping definitions of product, discounts, promotions, and other entities aligned across multiple CRM systems and ensures that a common set of products is accurately represented and available across all systems to facilitate transactional processes such as order capture, order fulfillment, and so on.

The synchronization service is triggered by the publication of a synchronize item event by Oracle Product Hub, which contains this information:

- BATCH_ID
- STRUCTURE_FLAG
- Workspace Name
- Auto-Release

Upon receiving the publishing event, the requestor ABCS queries from OPH the list of target systems to which the batch has to be published. See [Section 2.3.1, "Support for Routing Driven Through Explicit Specification of Target Systems"](#).

At the batch level, a list of target systems is captured to identify the systems that should receive the list of items payload. This list of target systems is used for routing the list of items payload to the consuming applications. This list of target systems is queried by the requestor after the publishing event is received.

If only the items need to be published, then the STRUCTURE_FLAG is *N*. If a structure is published, STRUCTURE_FLAG is *Y* and the list of items payload contains the data for both the root items and the other items in the published structures.

Complete information is always published for each item in the list of items payload. The receiving applications are expected to perform a synchronization of the received entity. For each entity, the fields that are passed in with a value are updated with the provided value. Those passed in with NULL or empty values are nullified. The fields that are not passed in are left as is.

Complete information is always published for each item in the list of items payload. The receiving applications are expected to perform a synchronization of the received entity. For each entity, the fields that are passed in with a value are updated with the provided value. Those passed in with NULL or empty values are nullified. The fields that are not passed in are left as is.

Within the published payload, some of the items may not apply to some of the target systems, for example, promotions to Oracle BRM. For each item, the list of items payload also contains a multivalued attribute that is used to identify the destination systems for the items (implemented in OPH as a multi-row UDA); this attribute identifies the systems that should process the specific item within the batch. The destination systems attribute is used by filtering rules for a given target system; these filtering rules are used to identify the payload entities pertinent to the specific target system.

Various types of items are published by OPH. OPH uses the UDA Entity type to capture the type of the item. Possible values are:

- Product
- Discounts
- Discount Model (this is not published)
- Promotion
- Option Group
- Bundle

OPH provides a service with an operation that, given the batch ID published in the item publishing event, returns the payload for containing all the items published in that batch. The integration uses this service to retrieve the application payload for the items that need to be published. This service has another operation by which it takes the batch ID published in the item-publishing event and returns a payload with a list of all the BOMs (structures) published in the batch; however, if the structure flag is *N*, it does not retrieve the list of BOMs payload.

The payload for the list of items and list of BOMs is the complete payload containing all attributes and entities even when the publishing takes place to update existing entities.

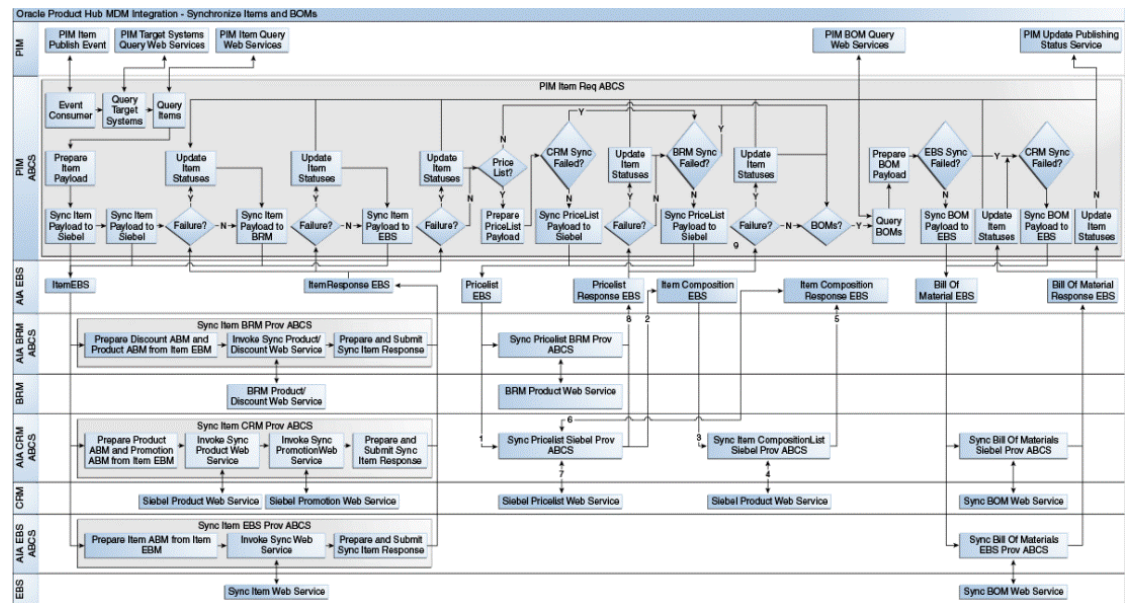
The list of BOMs payload contains the structures associated with the root items. The type of structure (for example, sales BOM, manufacturing BOM, marketing BOM, support BOM, and so on) is the same for all the BOMs in the payload as it was defined by the header-level attribute structure name. No multi-row, UDA destination systems are associated directly with BOMs. All BOMs go to all providers that consume BOMs. The DVM setting defines whether a provider consumes or does not consume BOMs. For a given target system, the synchronization of BOMs does not take place if the synchronization of items fails.

OPH-published messages can be generated from any number of organizations that have been established. The integration solution is based on manual pre-synchronization of organizations between OPH, Siebel, and EBS and creation of appropriate cross-references. The organization used for publication in OPH is submitted and mapped to the corresponding organization in the target application, if any.

For more information, see [Section 6.4, "Setting Up Cross-References"](#).

[Figure 2-1](#) shows the process flow from Oracle Product Hub to the participating applications (integration options) for item and BOM synchronization:

Figure 2-1 Synchronization of items and BOMs from OPH to Oracle BRM, Siebel CRM and Oracle E-Business Suite



The business process flow is as follows:

1. Oracle Product Hub publishes an item publish event to application integration architecture (AIA).
2. The event consumer within the OPH application business connector service (ABCS) (requestor) receives and processes the event. The event contains the batch ID, which is used to query data from OPH.
3. The OPH requestor ABCS queries the list of target systems from OPH based on the batch ID.

4. The OPH requestor ABCS queries the list of items from OPH based on the batch ID.
5. The requestor ABCS prepares an enterprise business message (EBM) to be sent to each target system. The EBS filters based on the destination system.

For each target system, the requestor ABCS invokes ItemEBS, which filters the items in the EBM based on the destination system user defined attributes (UDA) and routes the processing request to the appropriate provider. This is done sequentially for one provider after the other as described here:

Siebel CRM Provider ABCS

- a. Receives the ItemList EBM and separates items into products and promotions based on the UDA entity type. Items with entity type as product or entity type as discount are processed as products, and items with entity type as promotion are processed as promotions. Two application business messages (ABM) are produced, one for Siebel products, and one for Siebel promotions.
- b. Invokes separate web services on the application to synchronize the products, promotions, and price list.
- c. Prepares the response with the publishing status for the individual items, which is sent back to the requestor ABCS.

Oracle E-Business Suite Provider ABCS

- a. Receives the ItemList EBM and produces one ABM for Oracle E-Business Suite items.
- b. Invokes a web service on the application to synchronize the items.
- c. Prepares the response with the publishing status for the batch, which is sent back to the requestor ABCS.

Oracle BRM Provider ABCS

- a. Receives the ItemList EBM and produces one ABM containing both items for which the entity type is product and for which the entity type is discount.
 - b. Invokes a web service on the application to synchronize the discounts and the products. The same web service is also invoked when the product is updated with the pricing information from the price list.
 - c. Prepares the response with the publishing status for the individual items, which is sent back to the requestor ABCS.
6. The requestor ABCS receives the publishing status update from the provider ABCS. If a failure occurs, the requestor ABCS updates the publishing status in OPH for the items and continues with the processing.
 7. The requestor ABCS extracts pricing information from the items and prepares price list EBM for synchronizing pricing information.
 8. The requestor ABCS invokes pricelist EBS to send pricing information for the items in the batch to Oracle BRM, where it is associated with the products by means of the synchronize product web services.
 9. The price list response EBS is invoked to return a response to the requestor ABCS so that in case of failure, it updates the publishing status in OPH.
 10. The requestor ABCS invokes the price list EBS to send pricing information for the products in the batch to Siebel. This is conditionally based on the ENTITY_TO_TARGET_APPLICATION domain value map (DVM) setup.

11. The Siebel provider ABCS receives the price list EBM and invokes the item composition EBS for creating additional products associated with billing events.
12. The item composition EBS invokes the SyncItemCompositionList provider service for creating the event products in Siebel (billing type as Event) in cases in which multiple charges are associated with the items that represent the billable products.
13. The item composition response EBS invokes the synchronize price list provider ABCS for Siebel to synchronize the pricing information on the product and the additional product that has been created for billing events.
14. The price list response EBS is invoked to return a response to the requestor ABCS so that in case of failure, it updates the publishing status in OPH.
15. If a BOM was published in the batch, the requestor ABCS queries the BOMs associated with the root item in the batch.
16. The requestor ABCS prepares a BOM EBM and for each target system (Siebel CRM and Oracle E-Business Suite) that needs to process the BOM, it invokes the BillOfMaterial EBS, which invokes the provider ABCS. Each provider ABCS (Siebel CRM and Oracle E-Business Suite) based on the ENTITY_TO_TARGET_APPLICATION DVM that is set up for every target application:
 - a. Receives the BillOfMaterials EBM and prepares the BOM ABM.
 - b. Invokes a web service on the application to synchronize the BOM.
 - c. Prepares the response with the publishing status for the individual items, which is sent back to the requestor ABCS.
17. The requestor ABCS updates the publishing status in OPH for the root items of the BOM published in the batch.

The assumption is that OPH always specifies target systems at the batch level. The GetTargetSystems OPH service returns all the target systems to which the batch should be published. If the target system is not listed in the GetTargetSystems response, the payload is not sent to it even if the system name is defined at the entity level. If an error occurs in the item adapter while the adapter is getting data from OPH, the synchronization process is stopped for the entire batch.

When chunking is enabled, the process splits the total entities in a batch to multiple chunks or sub-batch based on the BATCH_SIZE specified in the configuration property. Each of these chunks or sub-batches is processed as a batch by itself.

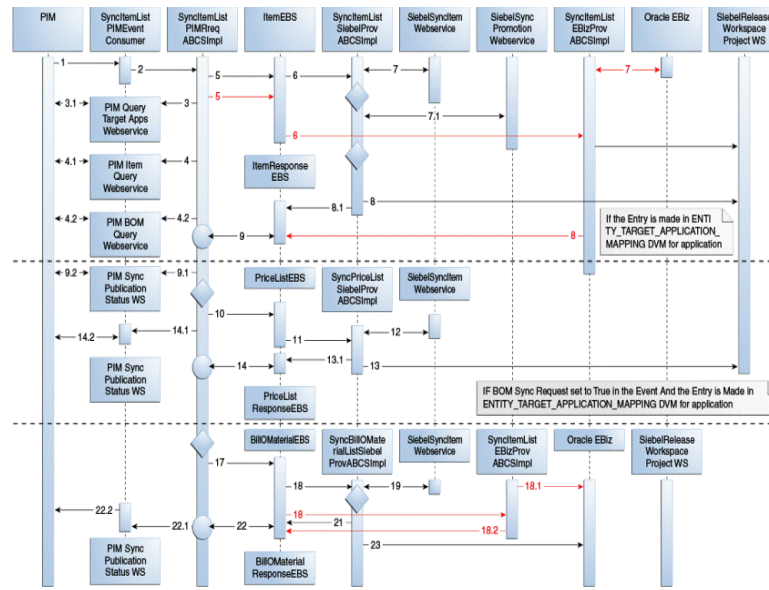
For more information, see [Section 6.8, "Setting Configuration Properties"](#).

Chunking can be enabled at the item or the BOM level, or for both. When chunking is enabled at the BOM level, the chunking extracts as many BOMs as defined in the chunk size. Additionally, if a BOM has an option class, the option class BOMs are also extracted into the same sub-batch. When chunking is enabled as part of the item synchronization process, if a sub-batch fails, all the items continue to be processed in that batch, but the pricelist or BOM is not processed. OPH provides a batch-level update that is called from the SyncItemPIMReqABCSEImpl in case the batch synchronization fails.

For more information on optimizing performance by enabling item chunking or BOM chunking, see [Appendix C, "User Defined Attributes Framework"](#).

[Figure 2–2](#) shows the technical implementation of the process:

Figure 2–2 Item and BOM integration sequence



Note: This diagram is also applicable to [Chapter 3, "Oracle Product Master Data Management Integration Option for Siebel CRM"](#).

Table 2–1 describes the steps depicted in the sequence diagram.

Table 2–1 Item and BOM integration sequence steps

Name	Step Description
OPH item event is raised	OPH item events are raised when the administrator publishes the items from the OPH system. This includes create/update to items or primary BOM associated with items. The event contains a Batch_ID to represent a set of items that are published as a single payload.
SyncItemListsPIMEventConsumer	SyncItemListsPIMEventConsumer listens to business events and receives the WF_EVENT_T_msg event payload for item events. SyncItemListsPIMEventConsumer routes to SyncItemListsPIMReqABCSImpl with complete event payload.

Table 2–1 (Cont.) Item and BOM integration sequence steps

Name	Step Description
SyncItemListPIMReqABCImpl	<p>The OPH requester ABC implementation SyncItemListPIMReqABCImpl receives the event message from OPH and queries the target systems. It then transforms the event message to the OPH ItemQueryABM and calls the QueryItemListPIMAdapter to get the ItemListABM from OPH.</p> <p>The QueryItemListPIMAdapter response is then converted into SyncItemListEBM and SyncPriceListEBM and routed to the corresponding EBS service. Once the items/price list is synchronized in the participating applications, the status of the synchronization is sent back to the SyncItemListPIMReqABCImpl, which updates OPH with the status. If the item synchronization is successful, only then is the BOM synchronized for that target.</p> <p>The SyncItemListPIMReqABCImpl invokes the QueryBillOfMaterialsListPIMAdapter to get the BillOfMaterialsListABM, transforms the same into the SyncBillOfMaterialsListEBM, and routes it to the corresponding EBS. Once the BOM is synchronized in the participating applications, the status of the synchronization is sent back to the SyncItemListPIMReqABCImpl, which updates OPH with the status.</p>
QueryItemListPIMAdapter	<p>This adapter service is implemented in business process enterprise library (BPEL). It is invoked from the SyncItemListPIMReqABCImpl. It takes the batch ID as input and responds with the ItemListABM payload in that batch. It also implements the chunking of items into smaller batches based on the BATCH_SIZE configuration property in the AIAConfigurationProperties.xml.</p>
QueryBillOfMaterialsListPIMAdapter	<p>This adapter service is implemented in BPEL. It is invoked from the SyncItemListPIMReqABCImpl. It takes the Batch_ID as input and responds with the BillOfMaterialsListABM payload in that batch.</p>
ItemEBSV2	<p>Invoking the ItemEBSV2 with the SyncItemList operation routes the SyncItemListEBM to the SyncItemListSiebelProvABCImpl, SyncItemListEbizProvABCImpl, and SyncItemListBRMProvABCImpl based on the target ID.</p>
SyncItemListEbizProvABCImpl	<p>The SyncItemListEbizProvABCImpl transforms the SyncItemListEBM into the appropriate Oracle E-Business Suite item ABM and invokes the appropriate application programming interface (API). The response from the Oracle E-Business Suite API is then transformed into the SyncItemListResponseEBM and sent back to the ItemResponseEBSV2. In this process, the cross-reference is linked to the Oracle E-Business Suite IDs of the product in the form of InventoryItemID::OrganizationID::OperatingUnitID.</p>

Table 2–1 (Cont.) Item and BOM integration sequence steps

Name	Step Description
SyncProductSiebelProvABCImpl	<p>The SyncProductSiebelProvABCImpl transforms the SyncItemListEBM into the Siebel product message (without Promotion) and calls the Siebel product web service SWIPProductImport to synchronize the product with all the items in the list, and performs these actions in sequence:</p> <ul style="list-style-type: none"> ■ Constructs the Siebel promotion message from the SyncItemListEBM. ■ Transforms the Siebel promotion message into the Siebel promotion ABM. ■ Calls the Siebel promotion web service SWIPromotionImport to synchronize the promotions. ■ Transforms the Siebel item and promotion response message into the SyncItemListResponseEBM and sends it back to the ItemResponseEBSV2. <p>In this process, the cross-reference is linked to the Siebel product IDs into the ITEM_ITEMID cross-reference. In addition, the cross-reference is linked to the Siebel promotion IDs into the PROMOTION_PROMOTIONID cross-reference.</p>
ItemResponseEBSV2	<p>The ItemResponseEBSV2 transmits the SyncItemListResponseEBM message back from the SyncItemSiebelProvABCImpl, SyncItemEbizProvABCImpl, and SyncItemBRMProvABCImpl to the SyncItemListPIMReqABCS, which then invokes the OPH status update service to update the item publication status in OPH.</p>
PriceListEBSV2	<p>The SyncItemListPIMReqABCS converts the OPH item query response ABM into the SyncPriceListEBM based on the specification name Prod_Details_Definitions with a UDA Pricing_Code value other than NONE. It then passes the EBM message to the PriceListEBS, which routes the message to the ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl.</p>
ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl	<p>The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl calls the Siebel SWIPProductImport to synchronize the pricelist in Siebel and then sends the response back to the SyncItemPIMReqABCSImpl using the PriceListResponseEBS.</p>
PriceListResponseEBSV2	<p>The PriceListResponseEBS routes the PriceListResponseEBM from ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl to the SyncItemListPIMReqABCS, which then updates the status to OPH.</p>
BillOfMaterialsEBS	<p>The SyncItemListPIMReqABCS invokes the BillOfMaterialsEBS with the asynchronous operation SyncBillOfMaterialsList if the item payload has the structure associated with it. The operation routes the SyncBillOfMaterialsListEBM message to Siebel CRM and Oracle E-Business Suite. Based on the routing rule, the BillOfMaterialsEBS invokes the SyncBillOfMaterialsListEbizProvABCImpl and the SyncBillOfMaterialsListSiebelProvABCImpl.</p>
SyncBillOfMaterialListSiebelProvABCImpl	<p>The SyncBillOfMaterialListSiebelProvABCImpl receives the SyncBillOfMaterialEBM from the BillOfMaterialsEBS and calls the Siebel product service to synchronize the product structure in Siebel. It calls the Siebel promotion service to synchronize promotion structure, structure component exclusion, and transaction attribute exclusion and sends the response back using the BillOfMaterialListResponseEBS.</p>

Table 2–1 (Cont.) Item and BOM integration sequence steps

Name	Step Description
BillOfMaterialListResponseEBS	The BillOfMaterialListResponseEBS routes the synchronize BillOfMaterialListResponseEBM from the SyncBillOfMaterialListSiebelProvABCImpl and the SyncBillOfMaterialListEbizProvABCImpl to the SyncItemListPIMReqABCS, which calls the OPH update service to update the status of BOM published in OPH.
SyncBillOfMaterialListEbizProvABCImpl	This service receives the BillOfMaterialEBM and calls the PLM API to synchronize the BOM in Oracle E-Business Suite. The SyncBillOfMaterialsListResponseEBM is then passed back to the SyncItemListPIMReqABCImpl.

2.3 Common Features Supported by the Base Pack across Process Flows

This section includes the following sections:

- [Section 2.3.1, "Support for Routing Driven Through Explicit Specification of Target Systems"](#)
- [Section 2.3.2, "Support for Multi-Language for Item Synchronization"](#)
- [Section 2.3.3, "Support for Telecommunications Seeded Library Attributes"](#)
- [Section 2.3.4, "Controlling Publishing of Items to Target System Instances"](#)
- [Section 2.3.5, "Methodology to Set the Value for the Description System Specification Attribute Group"](#)
- [Section 2.3.6, "Support for Updating Publishing Status in OPH upon Publication of Items"](#)

2.3.1 Support for Routing Driven Through Explicit Specification of Target Systems

This feature enables product administrators to specify the target systems explicitly in the Oracle Product Hub. Some of the changes made in Oracle Product Hub to a product or another entity apply only to one target system. Explicit specification prevents unneeded publishing of the payload to those target systems not affected by the changes.

- This feature enables a more flexible, data-driven, and explicit approach for routing, and by doing so, reduces administration and maintenance costs and increases flexibility. At the batch level, a list of target systems is captured to identify the target systems that should receive the payload (for items, value sets, or product classes).
- The list of target systems is used for routing the payloads to the target applications. AIA first maps the system IDs published by OPH to common system IDs and then uses these to route the payload to the target applications.
- Within the published payload, some of the entities may not apply to some of the target systems. Each entity in the payload has an associated multi-row attribute that is used to identify the subscribing systems for the entity (implemented in OPH as a multirow UDA [AG: Destination_Sys_Specification]). This attribute identifies the systems that should process the specific entity within the batch. If

Destination_Sys_Specification is instantiated and the destination system has the synchronize item flag set to Y, the items are synchronized to a participating application. If Destination_Sys_Specification is not instantiated, item is filtered out. The subscribing systems attribute is used for filtering rules for a given target system; these filtering rules are used to identify the payload entities pertinent to the specific target system.

2.3.2 Support for Multi-Language for Item Synchronization

The multi-language support (MLS) feature enables multi-language values for attributes to be published by OPH to target applications.

- The MLS provides language codes and all the values in the corresponding languages for attributes that support language translation.
- Some of the free-text fields on the items are translatable in the target applications, for example, Item Name, Description, and so on. For each translatable free-text field, OPH provides all the values in the supported languages and the corresponding language code for each value. The target applications may not support all the languages supported by OPH and thus consume only those values that belong to languages supported by the corresponding target applications.
- Some of the free-text fields on the items are not translatable in the target applications. In this case, although OPH provides all the language codes and the corresponding values, the target application consumes in a single default language. This default language supported by the target applications must be configurable.
- For the fields taking values from a valueset for which values are translatable, OPH provides the language-independent code and all the values in all the languages; however, AIA passes only language-independent code only to the target application.
- For the fields with valuesets for which values are not translatable, OPH provides the code for the value that must be consumed by the target application.
- For customer UDA, the values are synchronized with all the language codes and the corresponding values. The target applications have to consume these attributes based on the support offered for extended attributes.

2.3.3 Support for Telecommunications Seeded Library Attributes

The seeded attribute groups have attributes that define the characteristics of an item or the characteristics of an entity that the item represents. These attribute groups have been seeded in OPH. The process integration sets the values of these attributes in the corresponding target applications.

For more information about telecommunications seeded library attributes and their corresponding valuesets, see Appendix F: Seeded Item Metadata Libraries in the *Oracle Product Hub Implementation Guide*.

2.3.3.1 Sellable Product Information Library-Horizontal

- Product Details
 - Definitions: This attribute group is used to define the additional characteristics of an item that represents the product in Siebel. These are mapped to the first class attributes of the product in Siebel.

- Marketing: This attribute group is used to define the marketing characteristics of an item that represents the product in Siebel. These are mapped to the first class attributes that represent the marketing details of the product in Siebel.
- Service: This attribute group is used to define the service characteristics of an item. The service here refers to the customer service information for the item. These are mapped to the first class attributes that represent the customer service of the product in Siebel.
- Logistics: This attribute group is used to store the logistics information of an item. These details are usually associated with the physical goods. These are mapped to the first class attributes that represent the logistics information of the product in Siebel.
- Additional Information: This attribute group is used to define any additional information associated with the item. These are mapped to the first class attributes that represent the additional information of a product in Siebel.
- Pricing
 - Simple Pricelists: This attribute group is used to define pricing-related data associated with the item in OPH. The pricing defined in this attribute group is not specific to an industry, and the model represents a framework to define a simple price on the items. The attribute group is used whenever a billing-specific, complex-rate-plan model is not necessary to represent the price on the item. The UDA pricing code is used to specify that the simple pricelists attribute group is used for defining the price on the item.
- Promotion: The item in OPH can also represent a promotion.
 - More information: When an item represents a promotion, this attribute group is used to define additional information associated with the promotion. These are stored as a specification group in the ItemEBO of the canonical layer. These are mapped to the first class attribute of the promotion entity in Siebel.
- Charge Plan: The item in OPH can also represent a promotion.
 - Nonrecurring Charge Details: When an item represents a promotion, this attribute group is used to define a charge plan associated with the promotion. This attribute group is specifically for the nonrecurring charges that are associated with the promotion. These are stored as specification groups in the ItemEBO of the canonical layer. These are mapped to the first class attribute of the promotion entity in Siebel.
 - Recurring Charge Details: When an item represents a promotion, this attribute group is used to define a charge plan associated with the promotion. This attribute group is specifically for the recurring charges that are associated with the promotion. These are stored as specification groups in the ItemEBO of the canonical layer. These are mapped to the first class attribute of the promotion entity in Siebel.
 - Charges, Adjustment, Usage Plan Details: When an item represents a promotion, this attribute group is used to define a charge plan associated with the promotion. This attribute group is specifically for the more advanced information of the promotion such as adjustments, certain usage plan details, and so on that are associated with the promotion. These are stored as specification groups in the ItemEBO of the canonical layer. These are mapped to the first class attribute of the promotional entity in Siebel.
- Commitment: The item in OPH can also represent a promotion.

- Charges Credits: When an item represents a promotion, this attribute group is used to define the commitment charges associated with the promotion. These are stored as specification groups in the ItemEBO of the canonical layer. These are mapped to the first class attribute of the promotion entity in Siebel.
- Terms: When an item represents a promotion, this attribute group is used to define the commitment terms associated with the promotion. These are stored as specification groups in the ItemEBO of the canonical layer. These are mapped to the first class attribute of the promotion entity in Siebel.
- Product Promotions
 - Upgrade: When an item represents a promotion, this attribute group is used to identify all the promotions to which the current promotion can be upgraded. These are stored as specification groups in the ItemEBO of the canonical layer. These are mapped to the first class attribute of the promotional entity in Siebel.
- Subject Compatibility Rules: This attribute group is used to define compatibility rules associated with the item. This is a multi-row attribute group in OPH where multiple rules can be defined. These are stored as specification groups in the ItemEBO of the canonical layer. These are mapped to the first class attribute of the product entity in Siebel.

2.3.3.2 Product Management Library-Horizontal

- Destination System Specification: This attribute group is used to specify 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, routing, or both.

2.3.3.3 Communications Services Billing Library-Vertical

- Billing Attributes General: This attribute group is used to define billing-related information of the item for telecommunications. These attributes are 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: This attribute group is used to associate the charges with the billing products. In billing applications, charges are modeled as events called billable events. The association of the events to the billing products is represented by this attribute group.
- These attribute groups together represent the complex rating structure associated with the communications-related billing products:
 - Rate Plan: The rate plan attributes associated with the billing products are represented in this attribute group.
 - Tier Group: This attribute group is used to define multi-tier rating for the billing products.
 - Day Time Range: This attribute group is used to define the day and time when the associated rate data must be applied. You define the day and time using calendar dates in this attribute group. This is a timezone-independent field in which the time range should not be translated across the systems in different time zones.
 - Days of the Week Range: This attribute group is used to define the day and time when the associated rate data must be applied. You define the day and time by specifying the weekdays in a range of calendar dates. This is a

timezone-independent field in which the time range should not be translated across the systems in different time zones.

- Rate Data: The actual rate or price is grouped under the rate data. You define certain discount brackets and proration details that apply to actual charges using this attribute group.
- Balance Impacts: This attribute group defines the actual amount or charges that are associated with the billing products. The charges or the amounts can be grouped based on the type of resources, effective dates, and categories that can be defined in the billing application.
- Billing Discount Attributes: When an item represents a billing discount, this attribute group is used to define the billing discount-related information.
- Billing Discount Event Map: The association of the actual discount model with the discount is done by means of this attribute group.

For more information about communications-specific terminology, see *Oracle Communications Billing and Revenue Management* documentation.

2.3.3.4 Communications Product Details Library - Vertical

- Product Details: This attribute group is used to define communications-specific and fulfillment-related information of the item. These attributes are mainly used during order fulfillment processing.

2.3.3.5 Component UDA for Item Synchronization

These component UDAs have been seeded as part of the telecommunications library and are synchronized to the target application whenever a BOM structure is synchronized with the items:

- Product Promotions: Components
- Product Promotions: Pricing: Components: Adjustments
- Component Pricing
- Version: Structure

Product Promotions:Components: This component UDA is associated with the BOM that represents a promotion. It is defined for the immediate components of the item that represents a promotion. It does not apply to the children of the components. The item synchronization process sets the corresponding values in the context of the parent-child relationship of the promotion within Siebel.

Product Promotions: Pricing: Components: Adjustments: This component UDA represents the adjustments that are applied to the components of the item that represents the promotion. It is applied at the leaf-level component and is valid only in the context of the promotion. The process integration must set the values in the context of a promotion entity in Siebel.

Component Pricing: This pricing-related information of the component and subcomponents of the item that represents the promotion can be updated in context of the promotion. The hierarchy of items within the item synchronization is aware of the components and the associated pricing-related UDA. This component hierarchy is published by OPH. The integration sets these context-specific values for the corresponding subcomponents of promotions in Siebel.

Version: Structure: The component UDA provides more information about the structure that is associated with the BOM. It represents the contextual information of the BOM components. The process integration creates these as relationships in Siebel.

The integration supports only two types of relationships: product and class. See Option Class Support in Item Synchronization for Siebel.

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, this information is set for the relationship attributes of the product. The structure can also be associated with the item catalog categories in OPH. See [Section 3.1.2, "Synchronization of Item Catalog Categories"](#) for information about defining the structure for ICC for Siebel.

2.3.3.6 Support for Multi-Row Attribute Groups

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

These multi-row attribute groups are seeded as part of the telecommunications library:

- Pricing: Simple Price List
- 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

During item publish, the multiple sets of attribute values associated with each of these attribute groups are published by OPH. The process 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 applications can perform *delete all* and *insert all* operations. The interfaces on the target applications must be carefully designed and implemented based on the usage of this set of attribute values in the corresponding applications. For example, in Siebel, *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 process integration includes them in the canonical model.

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

2.3.4 Controlling Publishing of Items to Target System Instances

There is support for Destination System Attribute Groups. Destination and target systems are defined below:

- **Destination Systems:** The systems that are associated to the items using the seeded attribute group "Destination System Specification." These are always set in context of the item definition. It consists of an attribute (Flag) called Sync_Item. When Sync_Item="Y", the item must be published to the corresponding system and when Sync_Item="N", the item must not be published to the corresponding system.
- **Target Systems:** The systems that are selected in the publication framework when a batch of items are published from Oracle Product Hub.

For example, a promotion may have bundles and other component items. The promotion and bundles must be published only to Siebel where the leaf level items must be published to both BRM and Siebel. In such cases, the destination system for items that represent the promotion and Bundles will have only Siebel whereas the leaf level items will have the Siebel and BRM. When the promotion is published from Oracle Product Hub in the publication framework, Siebel and BRM instances are specified as the target for the batch.

Table 2–2 shows a promotion that explains each option:

Table 2–2 Promotion example

Example

```
<xxx>Standard VoIP
  <xxx>VoIP Bundle
    <xxx>VoIP Service
      <xxx>VoIP Access Options
        <xxx>Basic VoIP Access
        <xxx>Premium VoIP Access
      <xxx>VoIP Voicemail
      <xxx>VoIP $10 Monthly Discount
    <xxx>VoIP Adaptor
```

2.3.5 Methodology to Set the Value for the Description System Specification Attribute Group

Whenever the destination systems are not set in context of the item, the items will be published to all the instances specified on the batch in the publication framework (target systems).

Whenever the destination systems are set in context of the item with Sync Item = "Y" and they match the target systems in the publication framework, then the items will be published to all the instances specified in the destination system attribute group with Sync Item = "Y" in context of the item (rare and upgrade cases).

Add only those systems to "Destination System Specification" attribute group for which the items are not published and set the Sync_item flag = N.

Use Table 2–3 to consider the following as the application instances: SEBL_1, BRM_1, BRM_2, BRM_3 as target application instances:

Table 2–3 Example

Item	Item Type	Destination System	Sync Item
<xxxx>Standard VoIP	Promotion	BRM_1	N
		BRM_2	N
		BRM_3	N
<xxxx>VoIP Bundle	Commercial Bundle	BRM_1	N
		BRM_2	N
		BRM_3	N
<xxxx>VoIP Service	Service Bundle	BRM_1	N
		BRM_2	N
		BRM_3	N
<xxxx>Basic VoIP Access	Product	BRM_2	N
		BRM_3	N
<xxxx>VoIP Voicemail	Product	BRM_2	N
		BRM_3	N
<xxxx>VoIP \$10 Monthly Discount	Product	BRM_2	N
		BRM_3	N
<xxxx>VoIP Adaptor	Product	BRM_1	N
		BRM_3	N

In the above example, only the systems for which the items will NOT be published are added into the destination system specification attribute group and have the flag set to N. Corollary - All items are always published to the Target system instance that does not exist in the "Destination System Specification" attribute group of the item.

- Case 1: When the Promotion <xxxx>StandardVoIP is published in the publication framework all the systems SEBL_1, BRM_1, BRM_2 are specified as target systems.
 - a. All the items are published to SEBL_1
 - b. <xxxx>Basic VoIP Access, <xxxx>VoIP Voicemail, <xxxx>VoIP \$10 Monthly Discount are published to SEBL_1 and BRM_1
 - c. <xxxx>VoIP Adaptor to SEBL1 and BRM_2.

The items are always published to the target systems specified in the publication framework that does NOT exist in the destination system attribute group.

Also note <xxxx>VoIP Adaptor is not published to BRM_3; BRM_3 exists in destination system attribute group, but is not specified as a target system.

- Case 2: When the Promotion <xxxx>StandardVoIP is published in the publication framework all the systems BRM_1, BRM_2, BRM_3 are specified as target systems.
 - a. <xxxx>Basic VoIP Access, <xxxx>VoIP Voicemail, <xxxx>VoIP \$10 Monthly Discount are published to BRM_1
 - b. <xxxx>VoIP Adaptor published to only BRM_2
 - c. None of the items are published to BRM_3

The items are **not** published to the target systems (BRM_3) that exist in the destination system attribute group.

Note: If Customers already have item set up where the destination systems include systems which have Sync Item = Y and Sync Item = "N", ensure that the item is published to the corresponding target systems specified in the publication framework during the item publish process.

2.3.6 Support for Updating Publishing Status in OPH upon Publication of Items

Update of the publish status in OPH enables a product administrator to know the results of the previous publish and allows republishing if needed. A publishing status is associated with an item and OPH provides services to update the publication history for items (for its publication to a given system within a batch). OPH provides a service to update the status of the entity. The publishing status values can be:

- Batch Submitted - Initial status set by OPH for items before publication.
- In Process - Status set by AIA once the data is fetched from OPH for processing.
- Succeeded - Indicates that the item in a batch is successfully published to the target applications.
- Failed - Indicates that the publication of the item to the target applications failed.
- Ignored - Indicates that the item was ignored by a specific system because it does not apply to it, for example, promotions for BRM. This is set as a default value on the OPH side API to save processing complexity on the AIA side.

AIA updates the publishing status in OPH for items published within a batch once all the entities in the batch have been processed (both items and structures).

When the target system processes the entire payload at once (entire ListItemEBM or entire ListBOMEEM), the service to update status at the batch level is used; otherwise, a more granular service that updates the status at the entity level is used. For example, if an error occurs within AIA and the entire EBO cannot be processed, then a batch-level service is used. If the synchronize promotions services in Siebel fail, the status needs to be reported for each individual entity. When the target system offers services that are more granular than the submitted payload, but less granular than the items (such as Siebel or BRM), the status is still updated at the item level.

The publishing message contains two parts:

- An *initial string* prepared by the AIA requestor giving the context of the service that the error occurred in.

The initial string is a message in English the purpose of which is to help the OPH product administrator understand in which context the message was generated. The publishing status is per combination of item and system to which the item is being published. In the context of an item publishing, multiple interactions may exist with a given system (for items, price list, and BOM). The prepended message provides this context and enables the product administrator to understand what went wrong.

- The *message returned* by the target system or by the service provider.

The message returned by the target system or by the service provider may be an error message in case of failed synchronization or an informational message. When the target system is Siebel, the message also contains the workspace ID that was used in Siebel to publish the entities in the batch.

Status is published back to OPH by means of the responses to item, price list, and BOM synchronizations.

2.4 Oracle Product Hub Interfaces

Oracle Product Hub artifacts used by this integration are:

Item Synchronization: Inbound Web Services

- Service Name: ItemService
 - Operation Name: GetItemInformation
 - Request and response: ItemService_GetItemInformation, ItemService_GetItemInformationResponse

BOM Synchronization: Inbound Web Services

- Service Name: ItemService
 - Operation Name: getStructureDetails
 - Request and response: ItemService_getStructureDetails, ItemService_getStructureDetailsResponse

Publication Service: Outbound Web Services

- Service Name: PublicationService_Service
 - Operation Name: PublicationService_UpdatePublicationStatus
 - Request and response: PublicationService_UpdatePublicationStatus_InputMessage, PublicationService_UpdatePublicationStatus_OutputMessage
 - Operation Name: PublicationService_GetBatchSystems
 - Request and response: PublicationService_GetBatchSystems_InputMessage, PublicationService_GetBatchSystems_OutputMessage

Item Catalog Synchronization: Inbound Web Services

- Service Name: ItemCatalogService
 - Operation Name: getICCDetails
 - Request and response: ItemCatalogService_getICCDetails, ItemCatalogService_getICCDetailsResponse
 - Operation Name: PublicationService_GetBatchSystems
 - Request and response: PublicationService_GetBatchSystems_InputMessage, PublicationService_GetBatchSystems_OutputMessage

Valueset Synchronization: Inbound Web Services

- Service Name: ValuesetService
 - Operation Name: getValueSetDetails
 - Request and response: ValuesetService_getValueSetDetails, ValuesetService_getValueSetDetailsResponse

Product Class: Inbound Web Services

- Service Name: SWIProductClassIntegrationIO
 - Operation Name: SWIClassImportUpsert

- Request and response: SWIClassImportUpsert_Input, SWIClassImportUpsert_Output

Attribute Definition: Inbound Services

- Service Name: SWIAttributeIntegrationIO
 - Operation Name: SWIAttributeImportUpsert
 - Request and response: SWIAttributeImportUpsert_Input, SWIAttributeImportUpsert_Output

2.5 Oracle Product Hub Integration Services

These are the services delivered with this integration:

- SyncItemCatalogCategoryPIMEventConsumer
- SyncItemCatalogCategoryPIMReqABCImpl
- SyncItemListPIMEventConsumer
- SyncItemListPIMReqABCImpl
- QueryItemListPIMAdapter
- QueryBillOfMaterialsListPIMAdapter
- SyncSpecificationValueSetListPIMEventConsumer
- SyncSpecificationValueSetListPIMReqABCImpl

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack: Development Guide*, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."

2.5.1 Synchronize Item

A unique EBM ID is assigned to the SyncItemEBM and sent over the ItemEBS to the providers that synchronize the items in the participating applications and sends the response message back as SyncItemListResponseEBM. If no pricelist or BOM is available to synchronize, the workspace auto-release flag is also set on the ItemEBM so that the provider can release the workspace after the item synchronization is successfully completed. The SyncItemListPIMReqABCImpl receives the response using correlation and then updates the OPH application with the item response status for every entity in the batch.

2.5.2 Synchronize Price List

All applications that receive the price list have to be configured in this DVM during setup. The SyncPriceListEBM is routed to the target application providers only if item synchronization is successful and the target system is configured to receive the price list (ENTITY_TARGET_APPLICATION_MAPPING DVM). If no BOM exists to be processed for that particular batch, the release workspace flag is also set and sent so that the provider can release the workspace after the price list synchronization is successfully completed. The provider ABCS synchronizes the pricelist in the respective applications and sends the response status back as the SyncPriceListResponseEBM. The SyncItemListPIMReqABCImpl receives the response using correlation and then updates the OPH application with the pricelist response status.

2.5.3 Synchronize Bill of Materials

If the event payload has `STRUCTURE_FLAG` as `Y`, the target system is configured to receive the bill of materials (`ENTITY_TARGET_APPLICATION_MAPPING DVM`), and the item synchronization is successful, then the `SyncItemListPIMReqABCS` calls the `QueryBillOfMaterialsListPIMAdapter` service. The `SyncBillOfMaterialsListEBM` is sent to the participating applications.

The `SyncItemListPIMReqABCImpl`, using correlation, receives the `SyncBillOfMaterialsListResponseEBM` that is returned by the participating applications and OPH is updated accordingly. The release workspace flag is also set and sent to the provider. The provider releases the workspace after successfully synchronizing the BOM.

2.5.4 QueryItemListPIMAdapter

This BPEL adapter service receives the session ID, batch ID, mode, and system ID from the `SyncItemListPIMReqABCImpl` and retrieves the item information from OPH in a chunk or in batch mode based on the configuration (`BATCH_SIZE`) defined in the `AIAConfigurationProperties.xml`. If the batch size is 0 (zero), then all the item details in batch are returned (in batch mode); otherwise, the number of items defined in the configuration (`BATCH_SIZE`) is returned in multiple iterations (in chunk mode).

2.5.5 QueryBillOfMaterialsListPIMAdapter

This BPEL adapter service receives the session ID, batch ID, mode, and system ID from the `SyncItemListPIMReqABCImpl` and retrieves the `BillOfMaterial` information from OPH in a chunk or in batch mode based on the configuration (`BATCH_SIZE`) defined in the `AIAConfigurationProperties.xml`. If the batch size is 0 (zero), then all the bill of material details in batch are returned (in batch mode); otherwise, the number of bill of materials defined in configuration (`BATCH_SIZE`) are returned in multiple iterations (in chunk mode).

2.5.6 SyncSpecificationValueSetListPIMEventConsumer

The `SyncSpecificationValueSetListPIMEventConsumer` is implemented as a lightweight Mediator service. The `SyncSpecificationValueSetListPIMEventConsumer` has an Oracle applications adapter configured to listen for an OPH valueset publish business event. One service is available with one operation to read the OPH message `SyncValueSetReqMsg` from the Oracle AQ WF_BPEL_Q: `SyncSpecificationValueSetListPIMEventConsumer`. It is implemented as a Mediator process with an Oracle applications adapter for listening to a business event. OPH synchronizes valueset business event: `oracle.apps.ego.extfwk.publishValueSet`.

2.5.7 SyncSpecificationValueSetListPIMReqABCImpl

This service is a BPEL process. This accepts a valueset business-event message as a request and synchronizes valuesets in Siebel. The `SyncSpecificationValueSetListPIMReqABCImpl` is a BPEL process that receives the OPH valueset event message from the `SyncValueSetPIMEventConsumer` service. This service is responsible for calling the OPH query (participating applications) valueset Web service, based on event payload. It transforms the OPH valueset ABM messages into the appropriate `SyncSpecificationValueSetListEBM`. It invokes `SpecificationValueSetEBS` (`SyncSpecificationValueSetList` operation) to synchronize valuesets in Siebel.

This asynchronous service accepts the OPH valueset event message as a request. It follows all the standards of a requester ABCS.

This single operation service accepts a SyncSpecificationValueSetListEBM message as a request. The responsibility of this service is to receive SyncSpecificationValueSetListEBM and transform the same into a Siebel attribute definition ABM and call the Siebel attribute-definition Web service. This service also filters the MLS mapping based on LANG_CODE_DVM. This DVM has the details about how many languages the provider system can support. Once the synchronization is done, it gets the response message and passes the response to SpecificationValueSetResponseEBS.

2.6 Assumptions and Constraints for OPH

The assumptions and constraints for Oracle Product Hub are:

1. Oracle Product Hub is the master for definition of items, BOMs, and metadata. Any changes or enrichment that may occur in the participating application are not synchronized back to Oracle Product Hub.
2. The integration does not support synchronization of products from an inventory organization that is shared among multiple operating units.
3. Item catalog category cannot be published as a batch. Each ICC has to be manually published to the downstream systems; however, all ICCs that are part of hierarchies can be published together.
4. The parent ICC associated with an item catalog category cannot be changed once it is released. The parent ICC cannot be changed even to create a new version. This is an OPH limitation.
5. The same item cannot be added as a component of a BOM Item. This is an OPH limitation.
6. The MLS support offered by OPH for the valuesets depends on the validation type associated with the valueset. The validation types independent and translatable are supported for this release.
7. OPH does not support Boolean and Integer data types. To support Boolean and integer datatypes, the product administrator needs to suffix the valueset name with the word `_BOOL` and set predefined values, for example, True or False. The product administrator can define any number of Boolean valuesets using the technique, but the values must always be same, and seeded.
8. Publishing alternate catalogs from Oracle Product Hub is not supported by the integration.
9. Variant type attribute group in OPH is not supported by the integration.
10. OPH does not support adding the same item as component of BOM more than once. Siebel and process integration do support these. This is an OPH limitation. If required, this can be achieved by using the operation sequence support provided by OPH.

Oracle Product Master Data Management Integration Option for Siebel CRM

Siebel CRM is available as an installation option in the Product MDM integration.

This chapter includes the following sections:

- [Section 3.1, "Supported Features"](#)
- [Section 3.2, "Siebel CRM Interfaces"](#)
- [Section 3.3, "Siebel CRM Integration Services"](#)
- [Section 3.4, "Assumptions and Constraints for the Siebel CRM Option"](#)

3.1 Supported Features

These are the supported features for the Siebel option:

- Synchronization of Metadata
- Synchronization of Item Catalog Categories
 - Synchronization of Relationships/Structure under Item Catalog Categories
 - Synchronization of Attribute Groups as part of Item Catalog Categories
 - Synchronization of Customer UDA as part of Item Catalog Categories
 - Synchronization of Transaction attributes as part of Item Catalog Categories
 - Synchronization of Item Catalog Category Hierarchies
- Synchronization of Valuesets
 - Synchronization of Valuesets as part of Item Catalog Categories
 - Synchronization of Valuesets Independently
 - Multi Language Support for Valuesets
- Synchronization of Items and BOMs

The following process flows are supported from Oracle Product Hub to Siebel CRM as part of the item and BOM Synchronization:

1. Synchronization of Products and Discounts and Associated Structures from Product Hub to Siebel CRM.
 - a. Support for Transaction Attributes in Product Synchronization.
 - b. Support for Class type structure/ relationship for Siebel CRM.

- c. Support for Controlling Auto-Release of Entities Published from Oracle Product Hub in the Same Batch for Siebel.
2. Synchronization of Promotions from Product Hub to Siebel CRM.

For more information about Siebel CRM, see the Siebel CRM product documentation.

3.1.1 Synchronization of Metadata

Metadata enables customers to predefine various characteristics for the products that facilitates faster introduction of new product variants, reduces the time to market, and results in significant savings on initial and maintenance costs and support for product differentiation offering cross-sell and up-sell opportunities.

The metadata synchronization supports these features:

- Synchronization of Item Catalog Categories
 - Synchronization of Relationships and Structure Under Item Catalog Categories
 - Synchronization of Attribute Groups as Part of Item Catalog Categories
 - Synchronization of Customer UDA as Part of Item Catalog Categories
 - Synchronization of Transaction Attributes as Part of Item Catalog Categories
 - Synchronization of Item Catalog Category Hierarchies
 - Association of Item Catalog Category with Items
 - Publish of Item Catalog Category from OPH to Siebel CRM Implementation Flow
- Synchronization of Valuesets
 - Synchronization of Valuesets as part of Item Catalog Categories
 - Synchronization of Valuesets Independently
 - Batch and Initial Load
 - Multilanguage Support for Valuesets

All these synchronization processes report the status to OPH publication history.

3.1.2 Synchronization of Item Catalog Categories

The publication framework of OPH provides a user interface to publish item catalog categories (ICC) to one or more participating applications. The participating applications have to be registered within the publication framework of Oracle Product Hub.

A new ICC is created in OPH as a *draft* version and it is released, creating a working or active version of the ICC. Similarly, during updates, the draft version is updated and released to create a new version of ICC in OPH. However, at any given point, only one active released version of the ICC exists.

During the ICC publish, the active released versions or the future-dated versions of the ICC are published one at a time to the downstream participating applications (that is, one ICC per publish process). The publishing of ICC is a manual process performed by the OPH product administrator.

The publish action invokes the integration process and provides basic information of ICC such as the ICC unique identifier. The integration process queries OPH and the

OPH interfaces provide complete details of the ICC and all the following associated entities.

1. Item Catalog Categories (ICC)
2. Attribute Groups
3. Seeded Attributes and Custom added Attributes
4. Transaction Attributes
5. Valuesets
6. Structure/Relationship associated with ICC

The integration supports both create and update operations for ICC.

- For the create operation, the integration creates a workspace version of the product class in Siebel CRM.
- For the update operation, the existing product class is updated. The updates include changes to the product-class core fields-adding, deleting, and updating structure and relationship, and adding, deleting, and updating transaction attributes and valuesets.

Once the product class has been successfully created in Siebel, the status is sent back to OPH. OPH maintains a publication history for all the versions of ICC for each downstream application.

In cases in which a version ICC is republished without any changes from OPH to Siebel CRM, the updates on the product class do not create a new version in Siebel CRM.

The versions of ICC in OPH are associated with the effective start and end dates. Only the active released version or future-dated versions can be published from OPH publication framework.

The following versions cannot be published from OPH:

- Expired or past versions - effective end dates are earlier than the current date.
- Draft version - versions that are not yet released.

A provision exists to revert to the earlier version of the ICC. For this, a new version with effective dates must be created in OPH. No impact occurs to the integration with the creation of this new version because the integration process handles this as new versions.

Auto-Release of Entities

The auto-release provision is provided by OPH. The auto-release is activated by default. The product administrator in OPH can update it based on the modeling. Siebel releases the entity if the value is set or it must create the workspace version of the entity. The auto-release is supported for synchronize operations.

For more information about auto-release, see [Section 3.1.19, "Support for Controlling Auto-Release of Entities Published from Oracle Product Hub in the Same Batch for Siebel"](#).

3.1.3 Synchronization of Relationships and Structure Under Item Catalog Categories

An ICC can be defined with relationship and structure that contains items and BOMs.

- The ICC can be defined with items or BOMs as the relationships. When the integration queries OPH for the ICC, the structures that are defined as

relationships with the ICC are also returned. The ICC has references only to the latest version of the structure because the structure has already been successfully published to the downstream application. The integration creates relationships for the product classes in Siebel.

- The items or BOMs are synchronized to the target applications before the ICC is synchronized; otherwise, the integration process fails.
- The query also returns the component UDA associated with the relationship. This component-UDA stores context-specific values for the relationship within the ICC, for example, min, max, default value, and so on. The integration uses the references to create relationships within Siebel, add corresponding products, and set the context-specific values specified by the component UDA within the context of the product class in Siebel.
- All types of BOMs can be added as the relationship for the item catalog categories.

This is an example of how various relationships under ICC 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.

The structure definition in OPH 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 OPH, 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 "Appendix F: Seeded Item Metadata Libraries, Communications Product Details Library - Vertical" in the *Oracle Product Hub Implementation Guide*.

Table 3–1 is an example of how product and class type relationships are supported in OPH and how the process integration creates them in Siebel:

Wireless Service ICC

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 OPH definition of the relationship in the seeded attribute group Version Structure associated with the Wireless Service ICC. These attributes are component attributes values the values of which are set in context of the Wireless Service ICC.

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 for the Wireless Service ICC. This relationship has an empty relationship domain in Siebel.

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.

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

Wireless Service ICC

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 ICC in Siebel:

Table 3–3 Structure attributes

Relationship Name	Domain Type	Product	Class	Default Cardinality
Relationship1	Product	Wireless Router	NA	1
Relationship 2	Class	NA	Wireless device accessory	1

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 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 ICC in OPH. The process integration updates the corresponding Wireless Service ICC and the relationship in Siebel CRM.

3.1.4 Synchronization of Attribute Groups as Part of Item Catalog Categories

Every ICC is associated with zero or more attribute groups. Attribute groups are not defined within a context of the ICC and are reused across ICCs. Attribute groups are used to categorize user-defined attributes and operational attributes. The child ICC inherits the attribute groups of the parent ICC. The attribute groups are published when the associated ICCs are published. In the current release, OPH does not maintain the publication status of the attribute group in the publication history.

Whenever the integration queries OPH for the ICC, the attribute groups associated with the ICC 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; therefore, they are ignored at the Siebel connector service in the integration.

Note: OPH 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). Updates to these seeded attribute groups and the associated attributes are not supported by the OPH. 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. Whenever customer-defined UDA is updated or new attributes are added, all the ICCs 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, OPH 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 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.)

3.1.5 Synchronization of Customer UDA as Part of Item Catalog Categories

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 ICC inherits these UDAs from the parent ICC.

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

Whenever integration queries OPH for ICC, the UDAs that are associated with the ICC through attribute groups are returned along with the ICC definition and other entities.

The attributes that exist across the attribute groups are extracted and grouped under canonical models of the attribute.

Siebel does not have attributes as separate entity. The static UDA and operational attributes that are defined within attribute groups are ignored in the Siebel connector. These attributes have fixed mapping with the product fields in Siebel 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 ICC has to be republished for the updates on the UDA.

All customer-defined UDAs are stored in the flexfield support provided by CRM; therefore, the customers must ensure that they 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 AIA layer. Customers have to extend the connector services to map these attributes to the target applications.

The OPH does not track the publication status for attributes in the publication history for UDA.

3.1.6 Synchronization of Transaction Attributes as Part of Item Catalog Categories

Note the following:

- The transaction attributes are defined within the context of the ICC in OPH. These are not versioned entities. The transaction attributes are associated with versioned valuesets in OPH. Context-specific values are set for every transaction attribute within the ICC. The child ICC inherits the transaction attributes and the valueset from the parent ICC.
- Whenever the integration queries the OPH for ICC, the transaction attributes that are associated with the ICC 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 and sets the corresponding context-specific values.
- Any changes in the transaction attributes associated with the ICC (add, delete, or update the context specific values) updates the ICC and a new version is released in OPH. When an updated ICC 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 connector and the transaction attributes are created as product class attributes.
- Duplicate transaction attributes cannot be defined within an ICC. If a similar transaction attribute is defined in a parent ICC and the one of its child ICC 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 ICC, the transaction attribute does not remain an inherited attribute. It becomes a native attribute of the child ICC.

3.1.7 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

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.

To support transaction attributes with no valuesets as part of the Item Catalog Category synchronization to Siebel the following design approach has been implemented:

1. Create an attribute definition with the data type = "Text" and domain type = "Freeform" in Siebel.
2. Update the details of the attribute definition in AIA Configuration file.

During the Item Catalog Synchronization from OPH to Siebel, if a transaction attribute has no valueset associated with it, then the process synchronization will read the attribute definition from the configuration file and will set the corresponding attribute definition as the valueset for the transaction attribute in Siebel.

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.

3.1.8 Synchronization of Item Catalog Category Hierarchies

Note the following:

- The publication framework for ICC provides flags that provide these publishing options for the product administrator in OPH:
 - Publish only the selected ICC.
 - Publish selected ICC and all its parents.
 - Publish selected ICC and all its children.
 - Publish selected ICC and all its parents and children.
- In cases in which a single ICC is selected for publishing, if that ICC has a parent ICC that is not published, the parent ICC must be published to the downstream applications. If not, the process integration fails. You must publish the parent ICC and republish the child ICC. In this case, the child ICC references the parent ICC. The integration process associates the child ICC and its parent in Siebel.

Alternately, whenever a parent ICC is updated, the product administrator must publish the complete hierarchy associated with the ICC.

- In cases in which the flag to publish the entire hierarchy (parents or children; or parents and children) is set, the integration process queries OPH for all the ICCs in the hierarchy. OPH returns the complete definition of all the ICCs 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.
- Whenever an ICC is updated in OPH and published to downstream applications, the integration updates the existing product classes. In addition, the corresponding hierarchy is updated in Siebel 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 ICC are passed to the child ICC. The inherited structure in the child ICC 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 ICC. The behavior is the same for transaction attributes. The process integration updates the corresponding entities in the downstream applications.

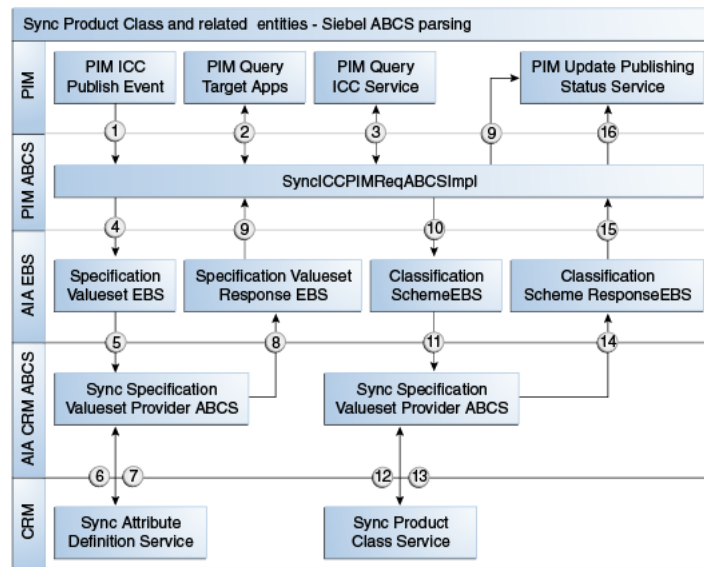
3.1.9 Association of Item Catalog Category with Items

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 ICC during the item synchronization process.

3.1.10 Publishing of Item Catalog Category from OPH to Siebel CRM Implementation Flow

Figure 3–1 shows the high-level integration flows for the item catalog category (ICC) synchronization:

Figure 3–1 Synchronization of ICC/Product Class created in OPH

These are the synchronization steps:

1. The product administrator in OPH publishes the ICC from the publication-framework user interface. The process publishes an event and provides basic information about ICC. The event consumer process invokes the requester ABCS for OPH.
2. The requester ABCS queries the target applications from OPH by calling the get target systems service.
3. Once the target systems are retrieved, the requester queries the ICC definition from OPH by calling the OPH Query ICC interface. The OPH Query ICC interface returns the complete information of the ICC 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 ICC.
4. The OPH requester connector service transforms the response from the OPH query ICC interface into the SpecificationValuesetListEBM and ClassificationSchemeListEBM. The ClassificationSchemeListEBM has complete definitions of ICCs, 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 provider ABCS.
6. The attribute definition class Siebel provider ABCS invokes the API provided by Siebel to create the corresponding attribute definitions in Siebel.
7. The Siebel 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 connector service invokes a response EBS.
9. The specification valueset response EBS returns the response to the requester ABCS. The OPH requester ABCS transforms the valueset response EBM and

invokes the OPH service to update the status. The OPH service updates the publication history for the valueset.

10. The OPH requester ABCS invokes the ClassificationSchemeEBS and provides ClassificationSchemeListEBM as input.
11. The ClassificationSchemeEBS routes to the appropriate synchronize product class Siebel provider ABCS. It invokes the API provided by Siebel 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.
12. The synchronize product class API returns the status to the connector service.
13. The Siebel connector service invokes a response EBS.
14. The ClassificationSchemeResponseEBS sends the status back to the requester ABCS.
15. The OPH requester ABCS transforms the classification scheme response EBM and invokes the OPH service to update the status. The OPH service updates the publication history for the ICC.

This diagram illustrates the services and their interaction.

Figure 3–2 Synchronization of ICC in OPH

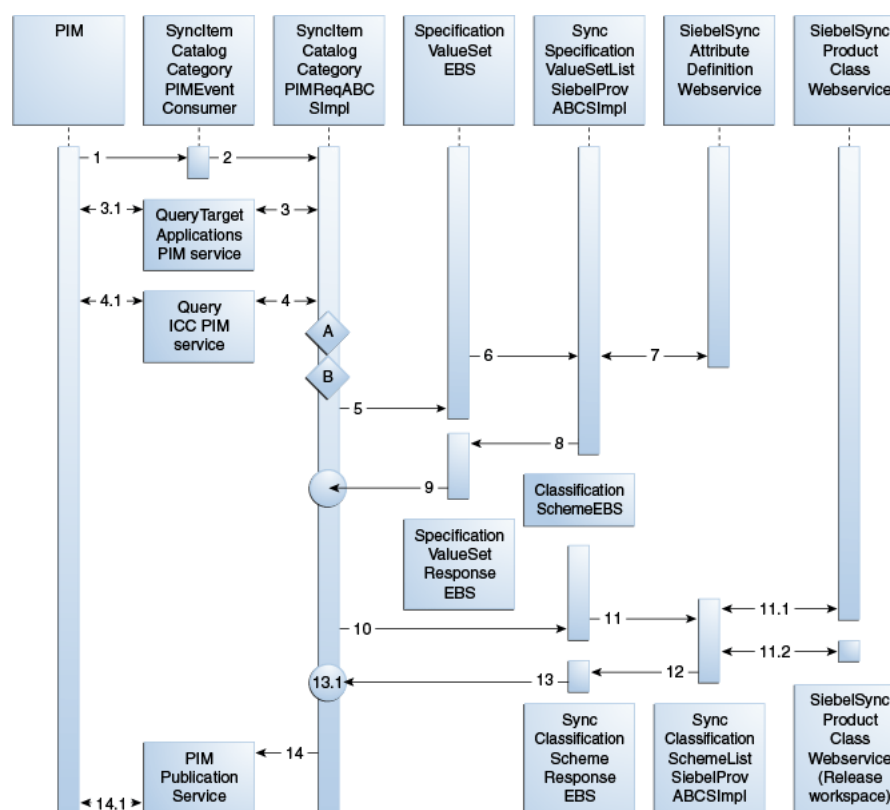


Table 3–5 Sequence steps

Steps	Name	Step Description
1	OPH ICC publish event is raised from the OPH System	OPH ICC publish events are raised when the product administrator publishes the ICC from the OPH system.

Table 3–5 (Cont.) Sequence steps

Steps	Name	Step Description
2	SyncItemCatalogCategoryPIMEventConsumer	<p>SyncItemCatalogCategoryPIMEventConsumer listens to the business event and receives the WF_EVENT_T_msg event payload.</p> <p>SyncItemCatalogCategoryPIMEventConsumer routes to SyncItemCatalogCategoryPIMReqABCImpl with complete event payload.</p>
3	SyncItemCatalogCategoryPIMReqABCImpl	<p>The OPH Requestor, SyncItemCatalogCategoryPIMReqABCImpl, receives the event payload and calls the OPH publication service to get the list of target applications by passing the batch ID. It then constructs the query ICC OPH ABM, calls the OPH query ICC web service, and gets the complete ICC details.</p> <p>It transforms the Query ICC response message to SyncSpecificationValueSetListEBM (only if valuesets are associated with T-Attr in OPH and the SyncValueSet flag is set to <i>True</i> in AIAConfigurationProperties.xml) and SyncClassificationSchemeListEBM.</p> <p>If valuesets are associated with T-Attr and the valueset synchronize property is set to <i>True</i> in the AIAConfigurationProperties.xml, SyncItemCatalogCategoryPIMReqABCImpl invokes SpecificationValueSetEBS, passing SyncSpecificationValueSetListEBM to synchronize valuesets in Siebel and other participating applications.</p> <p>Once the valueset synchronization is done successfully, it calls the classification scheme EBS passing SyncClassificationSchemeListEBM to synchronize the ICC.</p> <p>It gets the ICC synchronization response, constructs the OPH publication service ABM, and calls the OPH publication web service.</p>
3.1	OPH Publication Service (PublicationService_GetBatchSystems)	This service takes the batch ID and gives the list of the target systems to which the synchronization should be done.
4	OPH Query ICC web service	OPH Query ICC web service receives the batch ID, publishes parent hierarchy and child hierarchy flags, and returns the ICC details associated with that batch.
5	SpecificationValueSetEBS	SpecificationValueSetEBS receives SyncSpecificationValueSetListEBM as an input and routes it to the SyncSpecificationValueSetListSiebelProvABCImpl service to synchronize valuesets.

Table 3–5 (Cont.) Sequence steps

Steps	Name	Step Description
6	SyncSpecificationValueSetListSiebelProvABCServiceImpl	SyncSpecificationValueSetListSiebelProvABCServiceImpl receives SyncSpecificationValueSetListEBM. It transforms the EBM into the SyncAttributeDefinition ABM and calls the Siebel valueset-synchronization Web service. Once synchronization is done, it gets the response from the Siebel system, transforms SyncSpecificationValueSetEBM to SyncSpecificationValueSetResponseEBM, and invokes SpecificationValueSetResponseEBS.
7	Synchronize Attribute Definition Siebel Web service	Synchronize attribute definition Siebel Web service synchronizes the attribute definition in Siebel.
8	SpecificationValueSetResponseEBS	SpecificationValueSetResponseEBS receives SyncSpecificationValueSetListRespEBM as input and routes it to the SyncItemCatalogCategoryPIMReqABCServiceImpl service.
9	ClassificationSchemeEBS	Once the valueset synchronization is done successfully in Siebel, SyncItemCatalogCategoryPIMReqABCServiceImpl calls the ClassificationSchemeEBS passing SyncClassificationSchemeListEBM as its input and routes it to the SyncClassificationSchemeListSiebelProvABCServiceImpl service.
10	SyncClassificationSchemeListSiebelProvABCServiceImpl	SyncClassificationListSiebelProvABCServiceImpl receives SyncClassificationSchemeListEBM. It transforms the EBM into a SWIClassImportUpsert_Input message and calls the Siebel 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 IDs. Once this is done, it checks for any associated hierarchy. If it exists, it makes another call to the same Siebel Web service passing hierarchy information. Once synchronization is done, based on the workspace auto-release flag (Y), it calls the Siebel product-class synchronization service again and passes the workspace auto-release flag. It then constructs the SyncClassificationSchemeListResponseEBM and passes it to ClassificationSchemeResponseEBS.
11.1	Synchronize Product Class Siebel Web service	Synchronize product class Siebel Web service synchronizes the product class in Siebel and returns the response message.
11.2	Siebel workspace release Web service	Once the product class synchronization is done in Siebel, SyncClassificationSchemeListSiebelProvABCServiceImpl calls the Siebel product-class synchronization Web service again, passing workspace name, workspace refuse flag, and workspace auto-release flag to release the workspace.

Table 3–5 (Cont.) Sequence steps

Steps	Name	Step Description
12	ClassificationSchemeResponseEBS	ClassificationSchemeResponseEBS gets the SyncClassificationSchemeListResponse message and routes it to SyncItemCatalogCategoryPIMReqABCImpl.
13	OPH Publication Service	Once the OPH requestor gets the ICC synchronization response message, it calls the OPH ICC publish status Web service to update the publication status in OPH. These are the possible status values: Failed Succeeded Partial Fail (in case valueset fails)

3.1.11 Synchronization of Valuesets

The valuesets can be published independently or during the ICC publish from the OPH 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 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. [Section 3.1.15, "Multi Language Support for Valueset Synchronization"](#)

3.1.12 Synchronization of Valuesets as Part of Item Catalog Categories

Note the following:

- Whenever the integration queries OPH for the ICC that is published, the valuesets that are associated with the attributes of the ICC 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, OPH returns a unique set (union) of valuesets associated with the ICC.

- 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.
- Even though the canonical model consists of valuesets associated with the UDA and the transaction attributes, the Siebel 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 ICC is published, the publication framework extracts the version of all the valuesets that were associated when the ICC was released, implying that the version of the valueset that was associated with the ICC when the ICC 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 ICC 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 OPH and the downstream applications. OPH provides validation to ensure that only the released version of the valueset can be associated with the attribute.

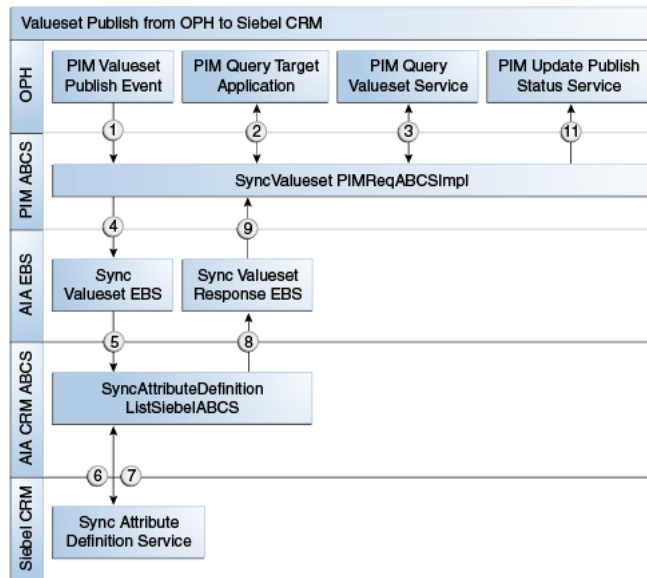
3.1.13 Synchronization of Valuesets Independently

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 OPH, 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. OPH 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 OPH product administrator.
- The publish action invokes the integration process and provides basic information about valuesets. The integration queries OPH 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 OPH 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 OPH.
- Once the attribute definitions are successfully created in Siebel for versionable valuesets of OPH, a status is propagated back to OPH. OPH maintains the publication history for all the versions of valuesets for each downstream application.
- OPH supports these data types:
 - Char
 - Number
 - Standard Date
 - Standard Date time
 - BOOL (OPH does not support the Boolean type. If the valueset name has a BOOL suffix, it is treated as a Boolean type)
 - INT (OPH does not support the Integer type. If the valueset name has an INT suffix, it is treated as an Integer type)
- Siebel supports all of these data types as well as Boolean and Integer data types except Datetime.

Figure 3–3 illustrates the valueset publish from OPH to Siebel CRM.

Figure 3–3 Valueset publish from OPH to Siebel CRM



These are the steps depicted in the diagram.

1. The product administrator in OPH 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 OPH
2. The requester ABCS queries the OPH valueset service.
3. The OPH valueset service returns a single payload with all the valuesets to the OPH requester ABCS.
4. The OPH 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 provider ABCS transforms the incoming SpecificationListEBM into attribute definitions. The Siebel provider ABCS invokes the Siebel interfaces.
7. The Siebel interface creates the corresponding attribute definitions. It checks the workspace auto-release flag: If it is *Y*, it calls Siebel 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 provider ABCS sends the status through a response EBS.
9. The response enterprise business service routes to the appropriate update valueset publish status OPH provider ABCS.
10. The update valueset publish status OPH provider ABCS invokes the OPH update status Web service and provides the valuesets and the corresponding statuses. The OPH 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 Valueset publish from OPH to Siebel CRM

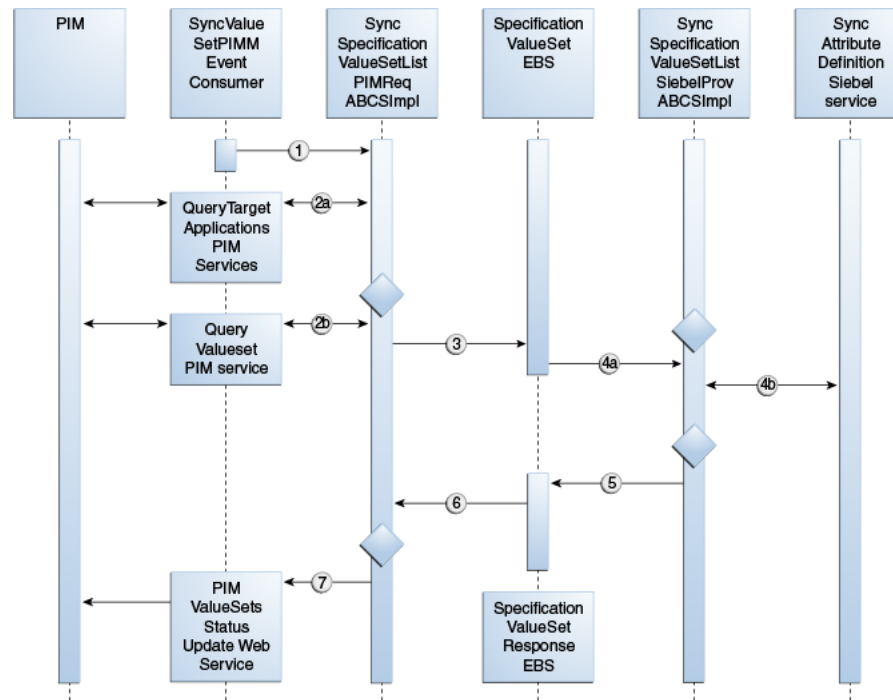


Table 3–6 shows the sequence steps:

Table 3–6 Sequence steps

Name	Description
OPH valueset event is raised from the OPH system	OPH valueset events are raised when the product administrator publishes the valuesets from the OPH system. Valuesets can be published in batches.
SyncValueSetPIMEventConsumer	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.
SyncSpecificationValueSetListPIMReqABCSImpl	The OPH 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 OPH service to get the response. SyncSpecificationValueSetListPIMReqABCSImpl service also calls the query target application's OPH 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 OPH service response ABM into QuerySpecificationValueSetList EBM and calls the SpecificationValueSetEBS Service.

Table 3–6 (Cont.) Sequence steps

Name	Description
SpecificationValueSetEBS	SpecificationValueSetEBS Service gets SyncSpecificationValueSetList EBM as its input and routes it to the SyncSpecificationValueSetListSiebelProvABCImpl service.
SyncSpecificationValueSetListSiebelProvABCImpl	SyncSpecificationValueSetListSiebelProvABCImpl receives the SyncSpecificationValueSetList EBM message as input. SyncSpecificationValueSetListSiebelProvABCImpl transforms the SyncSpecificationValueSetList EBM message into a Siebel request ABM message and calls the synchronize attribute definition Siebel 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. Only those languages are synchronized to the provider system, for example, Siebel. Synchronize attribute definition Siebel Web service returns the response to SyncSpecificationValueSetListSiebelProvABCImpl service, which has the status details.
SyncSpecificationValueSetListSiebelProvABCImpl	SyncSpecificationValueSetListSiebelProvABCImpl receives the response from synchronize attribute definition Siebel Web service and transforms the Siebel response ABM message into SyncSpecificationValueSetListResponseEBM and calls the SpecificationValueSetResponseEBS service. The valueset synchronization is based on all or none basis. If OPH is sending a batch of 10 valuesets out of which five fail, this status is sent back to OPH that batch failed and the error that comes from the provider service is logged in error-handling framework.
SpecificationValueSetResponseEBS	SpecificationValueSetResponseEBS routes the SyncSpecificationValueSetListResponseEBM to SyncSpecificationValueSetListPIMReqABCImpl
SyncSpecificationValueSetListPIMReqABCImpl	SyncSpecificationValueSetListPIMReqABCImpl service transforms the SyncSpecificationValueSetListResponseEBM message into a OPH valueset status-update Web-service input message and calls the OPH value set status-update Web service.
OPH value set status update Web service	This Web service takes the input message from the SyncSpecificationValueSetListPIMReqABCImpl service and updates the status in OPH as: Success Failed

3.1.14 Batch and Initial Load

The integration does not support initial load or batch load capability for item catalog categories. The recommended process is to publish the ICCs one at time because this is performed design time.

3.1.15 Multi Language Support for Valueset Synchronization

See [Section 2.3.2, "Support for Multi-Language for Item Synchronization"](#)

3.1.16 Synchronization of Products, Discounts, and Associated Structures from Oracle Product Hub to Siebel CRM

Products and discounts are created as items in Oracle Product Hub and are 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 MDM 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 [Chapter 2.2, "Synchronization of Items and BOMs"](#).

3.1.17 Multi-Event Product Synchronization from OPH to Siebel

Products that have multiple charges and events are created as customizable products in Siebel. 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. The item synchronization process from horizontal implementation supports both create and update of products in Siebel.

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

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

The item definition in OPH (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 OPH, 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 "Seeded Item Metadata Libraries," *Oracle Product Hub Implementation Guide*.

[Table 3–7](#) provides an example of how product and class type relationships are supported in OPH and how the process integration creates them in Siebel.

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 3–7 Example of how product and class type relationships are supported in OPH

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 3–8 depicts the OPH 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; instead, it is associated as a product.

Table 3–8 OPH 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 for the root item. This relationship has an empty relationship domain in Siebel.

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.

Table 3–9 provides an example of the support for relationship for products in Siebel.

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 are:

Table 3–9 Attributes for root items in Siebel

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

Table 3–9 (Cont.) Attributes for root items in Siebel

Relationship Name	Domain Type	Product	Class	Default Cardinality
Relationship 2	Class	NA	Wireless device accessory	1

Table 3–10 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 for the relationship domain to be updated. Updating the Bluetooth devices and synchronizing them is a prerequisite.

Siebel 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 OPH. 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 OPH.

3.1.19 Support for Controlling Auto-Release of Entities Published from Oracle Product Hub in the Same Batch for Siebel

The Product MDM integration enables the product administrator to control the automatic release of all the entities within the project workspace in Siebel at a more granular level than the Siebel 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.

OPH 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 with regard to auto-release.

- While OPH sets the flag at the batch level, Siebel cannot use it at the batch level because within a batch, multiple Siebel services are called for product and discounts, price list 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, 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. Hence, the workspace cannot be released until all the entities in the batch that need to be synchronized are successfully synchronized.

- OPH also passes a workspace name. The Siebel ABCS uses the workspace name received from OPH as the product workspace name. The workspace name allows creating all the entities from the same batch within the same workspace.
- OPH 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 system parameter. This is the default.

OPH always publishes the batch auto-release flag with a value of *D*.

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

If a workspace does not exist, Siebel 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 service within a batch is invoked, which typically is the service for the creation of products and discounts:
 - The value passed from OPH in the workspace name is used for the Siebel workspace name.
 - The value passed in the batch auto-release flag is used for the Siebel auto-release flag (upon DVM conversion), and the Siebel workspace reuse flag is set to *N*.
- When subsequent Siebel services are invoked for the same batch:
 - The value passed in the batch name is used for the Siebel workspace name.
 - The value passed in the batch auto-release flag is used for the Siebel auto-release flag (upon DVM conversion), and the Siebel workspace reuse flag is set to *Y*.
- Finally, when all the Siebel 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, 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.

Note: The auto-release flag is always passed as *Default* from OPH.

3.1.20 Synchronization of Promotions from Product Hub to Siebel CRM

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

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.

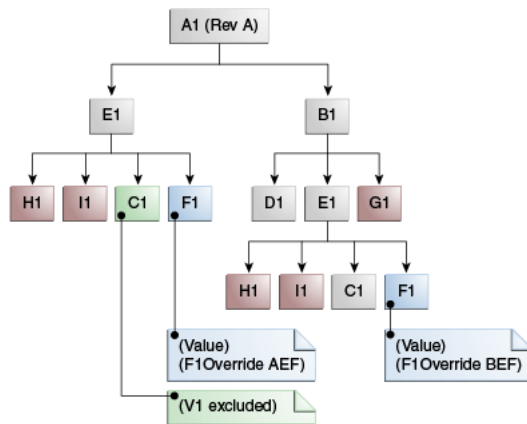
- Promotions cannot have an option class BOM associated with the root item. The immediate components of the promotion can have a BOM of type *option class*. The items or components that are associated with BOMs of type *option class* are reused to create different promotions.
- Promotions support *n*-level 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 OPH. 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 sets the corresponding fields of the promotion in Siebel.
- 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 OPH. 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, the status is updated in OPH 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 OPH. The new revision must be published to the downstream applications from the publication framework in OPH. For the deletion of components, the expectation is that the AIA configuration property synchronization flag be set to *Y*.

Component Exclusions in Context of Promotions

Whenever a promotion is defined in OPH, some of the components of the promotion or subcomponents under the root item can be excluded.

In [Figure 3–5](#) the components that are marked in red are excluded in context of the promotion *A*. In addition, item *E* with the associated BOM has been reused in the promotion definition under a different component *B*. These signify different paths to the components under the same root item (promotion) definition.

Figure 3–5 An example of component exclusions



OPH 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 handles the exclusions as follows:

- Siebel provides a contextual framework to control the promotion definition. In Siebel, 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 OPH. Siebel does not remove or delete the components of customizable products in context of the promotions. This is a documented Siebel limitation.

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

The promotion definition in OPH 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 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 3–11](#) explains the exclusions within option class BOMs.

Table 3–11 Exclusions within option class BOMs

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 OPH 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

If any of these components are to be excluded, both the minimum and the maximum cardinality in the component UDA - Version: Structure must be set as 0 (zero) in OPH and synchronized to CRM.

Case 3: Exclude component of option class items

The components under an option class item have to be excluded in OPH by means of the exclusion method provided in the product workbench. Default cardinality in the component UDA - Version: Structure must be specified in OPH 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 the VoIP with unlimited calls and fast dialing needs to be excluded, exclude these from the OPH product workbench and specify the override value for default cardinality of Basic VoIP.

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, the default cardinality of the included items must be overridden with a value in the context of the item that represents a promotion.

If the VoIP with unlimited calls needs to be excluded and fast dialing made optional, exclude the VoIP with unlimited calls from the OPH product workbench, specify the override value for default cardinality of Basic VoIP, and override the default cardinality of fast dialing to 0 (zero).

Note: 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.

Transaction Attribute Value Exclusions in the Context of Promotions

- A valueset is associated with every transaction attribute of the item. In OPH, 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 OPH to Siebel. See [Section 3.1.1, "Synchronization of Metadata"](#).

- 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 OPH 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, OPH 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 must set the values as *exclusions* for the corresponding attributes published from OPH 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 MDM Integration" in article ID 1086492.1 on My Oracle Support.

3.2 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

Outbound Siebel CRM Web Services

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

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

3.3 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

3.3.1 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 promotion ABM and invokes the Siebel 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 product ABM. It invokes the Siebel product web service to synchronize with Siebel.

3.3.2 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 product ABM, and invokes the Siebel product web service to synchronize the item structure with Siebel.

When an item is defined as an option class in OPH, 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 ICC name that is pre-established in Siebel as product class.
- The components under the option class item in OPH are set as relationship domain for the class type relationship.

3.3.3 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 product class ABM, and invoke the Siebel product-class web service to synchronize it into Siebel. Once the product class synchronization is done in Siebel, it passes the response message to ClassificationResponseEBS.

3.3.4 SyncSpecificationValueSetListSiebelProvABCImpl

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

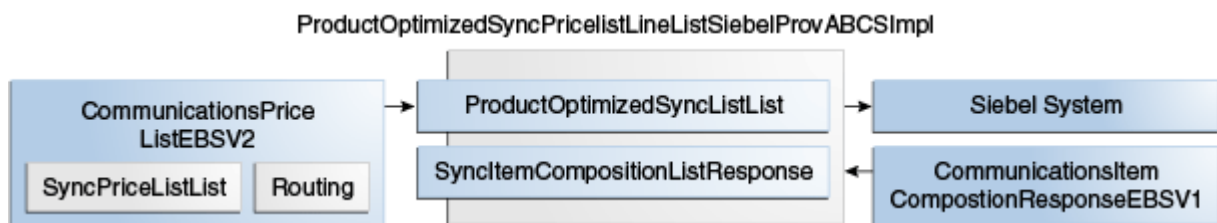
The responsibility of this service is to receive SyncSpecificationValueSetListEBM, transform it to Siebel attribute definition ABM, and invoke the Siebel attribute-definition web service to synchronize it into Siebel. Once the attribute synchronization is done in Siebel, it passes the response message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

3.3.5 ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl

The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl service transforms the PriceList EBM into a Siebel price list message and then calls the Siebel pricelist web service on operation InsertOrUpdate. The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl transforms the PriceList EBM into a Siebel product message and then calls the Siebel product web service on operation Product_spcImport_spcUpdate. The Siebel web service completes the request and returns a response message.

ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl then transforms the Siebel response message into the PriceList response EBM and sends it back to PriceListEBS.

Figure 3–6 ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl

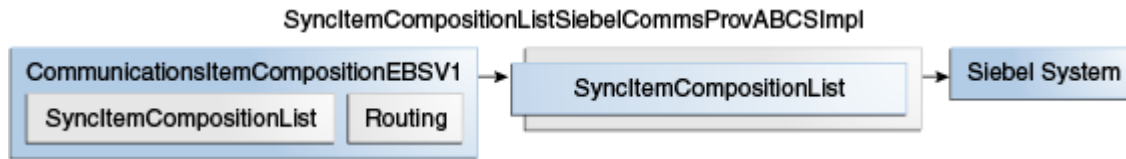


3.3.6 SyncItemCompositionListSiebelCommsProvABCImpl

The SyncItemCompositionListSiebelCommsProvABCImpl process accepts the SyncItemCompositionListEBM. It transforms SyncItemCompositionListEBM into the

Siebel product ABM. It then invokes the Siebel Product web service to create products and product structures in Siebel.

Figure 3–7 *SyncItemCompositionListSiebelCommsProvABCSImpl*



3.4 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 customers want to introduce new static attributes, they must define new attribute groups and static attributes. The integration creates these as flex attributes in Siebel. If new valuesets are introduced in OPH for new UDA, the integration passes the information as free-form text.
3. The items and BOMs in the structure of an ICC must be published before the ICC is published.
4. The item catalog categories that are associated with the items must be synchronized to Siebel 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 and the item validation organization defined as the inventory location associated with the business unit in Siebel. 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 [Section 6.4, "Setting Up Cross-References"](#). (This is applicable only if the Order to Cash: Siebel CRM - EBS integration is being planned or implemented with Product MDM.)
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 only if the Order to Cash: Siebel CRM - EBS integration is being planned or implemented with Product MDM.)
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 can filter out the item components that are not part of the current BOM structure. Siebel has a limitation that it uses only the current BOM revision.
8. The integration does not support multi-tier pricing between OPH and Siebel.
9. For Siebel CRM, these entities and associated subcomponents need to be locally enriched:
 - Eligibility rules.
 - Proration plans for items of entity type of promotion.

- Compatibility rule matrix.
 - Matrix discounts.
 - Volume discounts.
 - Attribute adjustments.
- 10.** Once the products are successfully synchronized from OPH to Siebel, 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 or Oracle 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 validation only in the UI.

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

Oracle Product Master Data Management Integration Option for Oracle E-Business Suite

This chapter provides an overview and discusses the process flow from OPH to Oracle E-Business Suite, as well as solution assumptions and constraints.

This chapter includes the following sections:

- [Section 4.2, "Synchronizing Items and BOMs from Oracle Product Hub to Oracle E-Business Suite"](#)
- [Section 4.3, "Oracle E-Business Suite Interfaces"](#)
- [Section 4.4, "Oracle E-Business Suite Integration Services"](#)
- [Section 4.5, "Assumptions and Constraints for the Oracle E-Business Suite Option"](#)

4.1 Overview

Oracle E-Business Suite is available as an installation option in the Oracle Product MDM integration.

The flows available in the Product MDM integration when you choose Oracle E-Business Suite as a participating application are described in [Chapter 2, "Oracle Product Master Data Management Integration Base Pack"](#).

For more information about Oracle E-Business Suite web services and documentation prior to Release 12.1.3, see the library on Oracle Technology Network:

<http://www.oracle.com/technetwork/documentation/applications-167706.html?>

For Oracle E-Business Suite documentation for R12.1.3 and beyond, see this library:

http://download.oracle.com/docs/cd/E18727_01/index.htm

4.1.1 Support for Operational Attributes

Oracle Product Hub supports seeded operational attributes that control the behavior of the item in various functional Oracle E-Business Suite modules such as Order Management and Procurement. The operational attributes are part of the definition of the item and are intended for consumption by Oracle E-Business Suite.

For more information about seeded operational attributes, see *Oracle Product Hub, Oracle Inventory User's Guide*.

4.2 Synchronizing Items and BOMs from Oracle Product Hub to Oracle E-Business Suite

Synchronization of items and BOMs from Oracle Product Hub to Oracle E-Business Suite is one-way and is triggered on demand by the publish mechanism of the Oracle Product Hub. The synchronization reduces the manual effort involved in maintaining a common set of products and facilitates transactional processes.

For create and update operations, routing rules need to be set up to identify the application services to invoke for the target systems. This setting should take precedence over any configuration settings defined earlier.

The synchronization enables having the item definition standardized between the standalone Oracle Product Hub and Oracle E-Business Suite that model inventory organizations and support supply chain and fulfillment processes.

Publication of an item associated with a BOM synchronizes all components for that item and BOM. The message routing rules use domain value maps that enable you to define the alternate structures that are of use to Siebel CRM. For an item, if a revision other than *A* or the first revision is published, Oracle E-Business Suite automatically creates revision *A* for that item.

The synchronization of an item also synchronizes the associated primary bill of material. This allows a single event for both the item and the related root BOM or primary BOM, but whether Oracle E-Business Suite uses this BOM is managed by configuration settings.

Primary item structure synchronization to Oracle E-Business Suite always has the effective date equal to the system date at the time of creation.

BOM synchronization involves two discrete services, namely, synchronization of component items and synchronization of structures to maximize performance and allow appropriate routing. Oracle E-Business Suite supports creating only future-dated BOM components.

For more information about the functional and technical details of this flow, see the [Section 2.2, "Synchronization of Items and BOMs"](#).

4.3 Oracle E-Business Suite Interfaces

These are the interfaces specific to Oracle E-Business Suite:

Item Synchronization: Inbound to EBS Web Services

- Service name: SyncItemListEbizAdapter
 - Operation name: SyncItemListEbizAdapter
 - Request and response: args_in_msg, args_out_msg

Item Synchronization: Outbound from EBS Event Interfaces

There are no outbound services.

BOM Synchronization: Inbound to EBS Web Services

- Service name: SyncBillOfMaterialsListEbizAdapter
 - Operation name: SyncBillOfMaterialsListEbizAdapter
 - Request and response: args_in_msg, args_out_msg

BOM Synchronization: Outbound from EBS Event Interfaces

There are no outbound services.

For more information about Oracle E-Business Suite web services and documentation prior to Release 12.1.3, see the library on Oracle Technology Network: <http://www.oracle.com/technetwork/documentation/applications-167706.html?> For Oracle E-Business Suite documentation for R12.1.3 and beyond, see this library: http://download.oracle.com/docs/cd/E18727_01/index.htm?

4.4 Oracle E-Business Suite Integration Services

These are the integration services required for Oracle E-Business Suite to integrate with Oracle Product Hub:

Note: Oracle E-Business Suite integration services can perform insert and update operations from Oracle Product Hub (OPH) to Oracle E-Business Suite.

- SyncItemListEbizProvABCImpl
- SyncBillOfMaterialsListEbizProvABCImpl
- SyncItemListEbizAdapter
- SyncBillOfMaterialsListEbizAdapter
- TransformAppContextEbizService
- QueryResponsibilityEbizAdapter

4.4.1 SyncItemListEbizProvABCImpl

This single operation service accepts a SyncItemListEBM product message as a request. This service transforms the SyncItemListEBM product message into an Oracle E-Business Suite application business message (ABM) format and runs Oracle E-Business Suite application programming interface (API) INV_EBI_ITEM_PUB. The PROCESS_ITEM_LIST through Oracle applications adapter synchronizes the item list in the Oracle E-Business Suite System.

SyncItemListPIMReqABCImpl invokes the service SyncItemListEbizProvABCImpl through ItemEBSV2. The response is sent back to SyncItemListPIMReqABCImpl through ItemResponseEBSV2 with the status after the synchronization, or in case of failure.

4.4.2 SyncBillOfMaterialsListEbizProvABCImpl

This single operation service accepts SyncBillOfMaterialsListEBM message as a request and returns a SyncBillOfMaterialsListResponseEBM as a response. This service follows all the standards of a provider ABCS.

SyncItemListPIMReqABCImpl invokes SyncBillOfMaterialsListEbizProvABCImpl through BillOfMaterialEBS. The response is sent back to SyncItemListPIMReqABCImpl with the status in case of success or failure through BillOfMaterialsResponseEBS.

The SyncBillOfMaterialsListEbizProvABCImpl service transforms the SyncBillOfMaterialsListEBM to SyncBillOfMaterialsListAdapter ABM. The SyncBillOfMaterialsListAdapter calls Oracle API BOM_BPEL_SYNCBILLOFMATERIALSLI.BOM_BO_PUB\$PROCESS_BOM through the Oracle applications adapter for synchronizing the BOM.

4.4.3 SyncItemListEbizAdapter

This Oracle applications adapter service invokes the Oracle E-Business Suite API INV_EBI_ITEM_PUB.PROCESS_ITEM_LIST.

4.4.4 SyncBillOfMaterialsListEbizAdapter

This is an Oracle applications adapter service that invokes the Oracle E-Business Suite API, BOM_BPEL_SYNCBILLOFMATERIALSLI.BOM_BO_PUB\$PROCESS_BOM.

4.4.5 TransformAppContextEbizService

This two-operation service is a BPEL service, which invokes another database adapter service QueryResponsibilityEbizAdapter that sets the AppsContext based on the integration user.

The user, responsibility, and the default operating-unit are picked from the AIAConfigurationProperties.xml file. The default user is OPERATIONS; default Responsibility is Order Management Super User, Vision Operations (USA); and default Operating Unit is 204.

4.4.6 QueryResponsibilityEbizAdapter

QueryResponsibilityEbizAdapter is a database adapter service that takes the responsibility as input and gives the responsibility ID, responsibility key, responsibility name, and description as output.

4.5 Assumptions and Constraints for the Oracle E-Business Suite Option

The assumptions and constraints for the Oracle E-Business Suite option are:

1. In case of the participating application being Oracle E-Business Suite and the items being associated with multiple inventory organizations, the item associated with the primary organization as defined in Oracle E-Business Suite must be synchronized prior to synchronization of items associated with any other organization.
2. In case of the participating application being Oracle E-Business Suite, the assumption is that the primary BOM associated with the BOM must be synchronized prior to invoking synchronization of alternate BOMs to Oracle E-Business Suite.
3. Oracle Product Hub must be deployed as an instance that is separate from Oracle E-Business Suite.
4. Synchronization of user-defined attributes (UDAs) to descriptive flexible fields in Oracle E-Business Suite is not supported as part of this release; therefore, a solution is delivered that does not map UDAs for Oracle E-Business Suite provider service.
5. For Oracle E-Business Suite, account IDs used for the integration remain synchronized for OPH and Oracle E-Business Suite. The locator code and subinventory need to be manually set up in Oracle E-Business Suite.
6. Item messages from Oracle Product Hub may come from multiple organizations that have been set up in Oracle Product Hub; therefore:
 - For any Oracle Product Hub organization that invokes the item synchronization, a corresponding organization (inventory organization) and operating unit must be

pre-established in Oracle E-Business Suite to provide a destination organization for the message.

- Oracle Product Hub published message must include the organization ID (inventory organization ID) and the operating unit associated with the organization ID that needs to consume the message.
- Cross-referencing is required even when organizations and items are identical between Oracle Product Hub and Oracle E-Business Suite.

For more information about Oracle E-Business Suite web services and documentation prior to Release 12.1.3, see the library on Oracle Technology Network:

<http://www.oracle.com/technetwork/documentation/applications-167706.html?>

For Oracle E-Business Suite documentation for R12.1.3 and beyond, see this library:

http://download.oracle.com/docs/cd/E18727_01/index.htm?

Oracle Product Master Data Management Integration Option for Oracle Communications BRM

Oracle Communications Billing and Revenue Management (BRM) is available as an installation option in the Product MDM integration.

For more information about Oracle BRM, see the Oracle BRM product documentation.

This chapter includes the following sections:

- [Section 5.1, "Synchronization of Billing Products from Oracle Product Hub to Oracle BRM"](#)
- [Section 5.2, "Oracle BRM Interfaces"](#)
- [Section 5.3, "Oracle BRM Integration Services"](#)
- [Section 5.4, "Assumptions and Constraints for the Oracle Communications BRM Option"](#)

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

[Table 5–1](#) 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 5–1 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.

Table 5–1 (Cont.) Supported billing products

Product Type	Applies To	Charge Type	Example	Comments
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 OPH 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 OPH and BRM; however, the integration to Siebel 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 OPH are used to define the billing product name and billing product description in BRM, respectively.

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

- 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

5.1.1 Multi-Event Product Synchronization from OPH to Siebel

The products that have multiple charges and events are created as customizable products in Siebel. 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. The item synchronization process from horizontal implementation supports both create and update of products in Siebel.

To support create and update of pricelist, the process integration to Siebel reuses the communications pricelist Siebel-provider-connector services. The OPH requestor 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 OPH using the seeded attribute framework. The process integration synchronizes the multi-tier rating from OPH to BRM. The Siebel 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 OPH 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.
- 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 OPH. 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 OPH 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 OPH, only the name of the usage rate plan is defined and published to the target applications. Oracle BRM uses pipeline-based rating for usage charges. The usage-based rating is enriched in Oracle 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 OPH and synchronized to BRM do not remove the pipeline rate plan enriched in BRM.

Table 5–2 summarizes the support offered for pricing in OPH.

Table 5–2

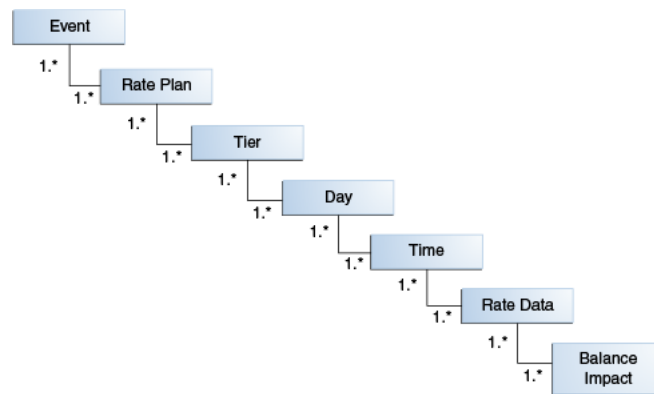
--	--	--	--
Real-Time Rating	One-time	Single Rate Plan	Complete definition in OPH. The rate plan can also be enriched in BRM. The rate plan name must not be changed in BRM because subsequent updates override the changes.
--	--	Rate Plan Selector	OPH sends the rate-plan-selector name and the rate plans are defined and enriched in BRM.
--	--	Custom event analysis	Complete definition in OPH. 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 OPH. 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	OPH sends the rate-plan-selector name and the rate plans are defined and enriched in BRM.
--	--	Custom event analysis	Complete definition in OPH. 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 OPH. 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	OPH sends the rate-plan-selector name and the rate plans are defined and enriched in BRM.
--	--	Custom event analysis	Complete definition in OPH. 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.

Table 5–2 (Cont.)

--	--	--	--
Pipeline Rating	Usage	Pipeline rate plans	OPH 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 5–1 depicts the real-time rating in Oracle BRM.

Figure 5–1 Pricing Hierarchy

The pricing hierarchy can be modeled in OPH 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 Oracle 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 OPH. 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, 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 OPH. These are synchronized to BRM and Siebel. The multiple-rate tiers manifest as multiple pricelist lines on a product in Siebel.

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 supports only absolute effectivity.

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

For more information about synchronizing billing products and billing discount from OPH to BRM, see the Synchronization of Items and BOMs section in [Chapter 2, "Oracle Product Master Data Management Integration Base Pack"](#).

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

For more information about Telco seeded Library attributes and their corresponding valuesets, see [Appendix F: Seeded Item Metadata Libraries, 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 the user 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 OPH requestor. If OPH 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 Oracle BRM provider does not need to do further translation.

Note: In the Oracle Product Hub integration, most date and time fields published by OPH are translated by the OPH requestor 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 the user 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 OPH requestor. If OPH has captured time as 10:00 AM PST, this is translated to 10:00 UTC (without adjusting the hour difference). The Oracle 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 Oracle BRM provider as requested by the Oracle 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

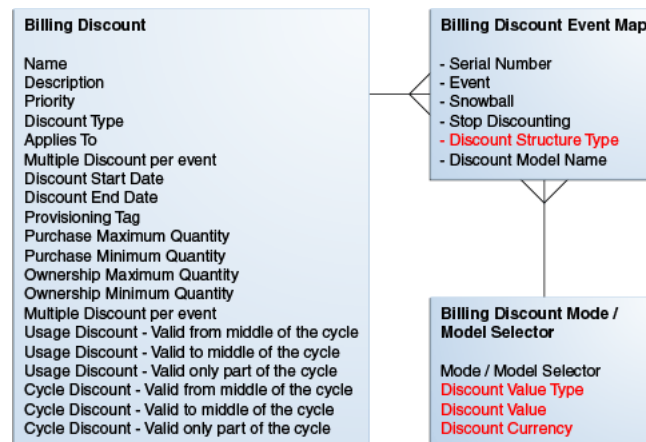
5.1.2 Synchronization of Discounts and Discount Models from Oracle Product Hub to Siebel CRM and Oracle BRM

This synchronization allows a centralized definition of common discount features in OPH so that discount entities are consistently defined across Siebel CRM and Oracle BRM. It allows modeling of discount models in OPH to allow generalization and reuse of features used by multiple discounts and mapping to discount model entities in Oracle 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.

Figure 5–2 is a conceptual simplification of the Oracle BRM model, which describes a discount and discount model entity and the way they are associated with each other.

Figure 5–2 Conceptual model for discount entities



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. OPH uses a simplified model and only two entities are captured: discounts and discount models.

Discounts and discount models are represented as items in OPH. 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 OPH.

Table 5–3 shows the seeded attribute groups that are used to model billing discount in OPH:

Table 5–3 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 the *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.
- Oracle 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 Oracle 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 OPH, 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 OPH. Currently, no cross-validation is implemented by OPH 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 OPH can rely only on the name to assign the discount model to a discount. In OPH, the description field in the item can also be used to add additional information for the discount model; however, this information is limited to OPH 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 OPH 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 OPH and then enrich in BRM through the synchronize discount flow. Discount models are manually configured in OPH 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 MDM Integration 2.5" Note ID - 1086492.1 on My Oracle Support.

5.1.3 Viewing Publishing Status of Items

For more information about how to view publishing status of items, see chapter "Viewing the Publishing History" in the

Oracle® Product Information Management

User's Guide.

5.2 Oracle BRM Interfaces

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

For more information about Oracle BRM services, see Oracle Communications Billing and Revenue Management (BRM) Documentation, BRM Documentation, Reference, API reference.

5.3 Oracle BRM Integration Services

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

- SyncItemListBRMProvABCImpl
- SyncPriceListListBRMProvABCImpl

5.3.1 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.

5.3.2 SyncPriceListListBRMProvABCImpl

This 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 the web service returns back to the BPEL with success notification, cross-reference values are updated with the BRM identifiers. A response EBM is constructed, which is used to invoke the PriceListResponseEBS.

5.4 Assumptions and Constraints for the Oracle Communications BRM Option

The assumptions and constraints for the Oracle BRM option are:

1. No explicit synchronization flow exists for discount models from OPH to Oracle BRM.
2. Multiple rate plans using rate plan selectors are not supported for item definition in OPH. 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 OPH and the rate plans for the rate plan selectors must be enriched in Oracle BRM.
3. The usage-based products are defined in OPH 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 Oracle BRM.
4. The usage-based rating using pipeline rate plans cannot be defined in OPH and hence they are not supported by the integration. They must be defined in Oracle BRM.
5. The enrichment for one-time and cycle charges must not change the rate plan names in Oracle BRM.
6. The discount models are associated with the items that represent the discounts in OPH. The complete definition of the discount model is done in Oracle BRM. The discount model name must not be updated in BRM because the subsequent updates override the value.

Part II

Implementing the Delivered Process Integrations

Part II contains the following chapters:

- [Chapter 6, "Configuring the Oracle Product Hub Integration"](#)
- [Appendix A, "Frequently Asked Questions"](#)
- [Appendix B, "Support for Rate Tiers and Effectivity"](#)
- [Appendix C, "User Defined Attributes Framework"](#)
- [Appendix D, "Scenarios"](#)

Configuring the Oracle Product Hub Integration

Oracle Product Hub, Siebel CRM, and Oracle E-Business Suite must be set up for the integration to work properly. This includes setting up organizations, setting up inventory locations, and so on.

This chapter includes the following sections:

- [Section 6.1, "Adding a New Spoke Instance"](#)
- [Section 6.2, "Post Installation Configuration Set Up"](#)
- [Section 6.3, "Setting Up Oracle BRM"](#)
- [Section 6.4, "Setting Up Cross-References"](#)
- [Section 6.5, "Working with Cross-References"](#)
- [Section 6.6, "Working with Domain Value Maps"](#)
- [Section 6.7, "Handling Errors"](#)
- [Section 6.8, "Setting Configuration Properties"](#)
- [Section 6.9, "Setting Up the Data for Seeded Attributes of the Telecommunications Library"](#)
- [Section 6.10, "Setting up List of Values in Siebel"](#)

6.1 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 OPH.
 - b. Navigate to the responsibility OPH 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. (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 OPH.
 - 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 value must match the name value defined above).

Description: New Target System JD Edwards.

3. Add the entry for the instance in the System Setting in 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 OPH system (Step 1). [Table 6-1](#) shows an example:

COMMON: JDE_01

PIM_01: JDE

Entry in TARGET_ID DVM:

Table 6-1 Sample DVMs

COMMON	PIM_01
EBIZ_01	EBSSS
SEBL_01	Siebel
BRM_01	BRM
JDE_01	JDE

5. Add the entry in ENTITY_TO_TARGET_APPLICATION DVM, for processing Pricelist and 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 6–2 Examples for System Name & Codes

System	System Name & Code in OPH	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:

<AIAHOME> // AIAMetaData/dvm/

<AIAHOME> // 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 <AIAHOME>/AIAMetaData/xref/ITEM_ITEMID.xref and to the DVM file DVM under <AIAHOME>/AIAMetaData/dvm/ITEM_TYPE.dvm
 - c. Update MDS. Modify <AIAHOME>/aia_instances/MDM/config/UpdateMetaDataDP.xml to include the files modified in step b. Then navigate to the folder <AIAHOME>/Infrastructure/Install/config and run:


```
ant -f UpdateMetaData.xml -DPropertiesFile<AIAHOME>/aia_instances/O2C/config/AIAInstallProperties.xml
```
 - d. Once complete, login to the AIA console, navigate to setup\AIA Configuration and click the Reload button to load the changes.

6.2 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 OPH is mapped to an Organization/Business Unit in Siebel CRM.
- An inventory organization in OPH is mapped to inventory location in Siebel CRM, which needs to be child of a business unit.
- If OPH and E-Business Suite co-exist, the organization and inventory organization needs to be set up in OPH 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: Creation of operating units or inventory organizations is beyond the scope of this document. See the Oracle E-Business guides for more information.

6.2.1 Extract Inventory Organization and Operating Unit IDs in Oracle E-Business Suite

Note: In Oracle E-Business Suite order management, the visibility to an item as part of the order capture is based on the association with an organization that in turn is associated with the system parameter item-validation-organization. As the product synchronization into Siebel is done to support order capture in Siebel, it is recommended that the various organizations defined as item validation organizations are cross-referenced for Siebel (and are created in Siebel); however, not all other inventory organizations may have a cross-reference to Siebel. See [Section 6.2.3, "Mapping Siebel CRM"](#)

If Oracle E-Business Suite is one of the participating applications in your implementation and the Order to Cash: Siebel CRM - EBS integration also is part of the implementation, then you need to perform these steps. After these steps are complete, you need to implement a similar organization hierarchy setup in OPH. If the Order to Cash integration is not in the set of pre-built integrations, then this step is not required.

To extract inventory organization details in Oracle E-Business Suite:

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

```
select ORGANIZATION_ID from HR_ORGANIZATION_INFORMATION
where ORG_INFORMATION1='INV'
```

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 the operating unit details in Oracle E-Business Suite:

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

```
select ORGANIZATION_ID from HR_ORGANIZATION_INFORMATION
where ORG_INFORMATION1='OPERATING_UNIT'
```

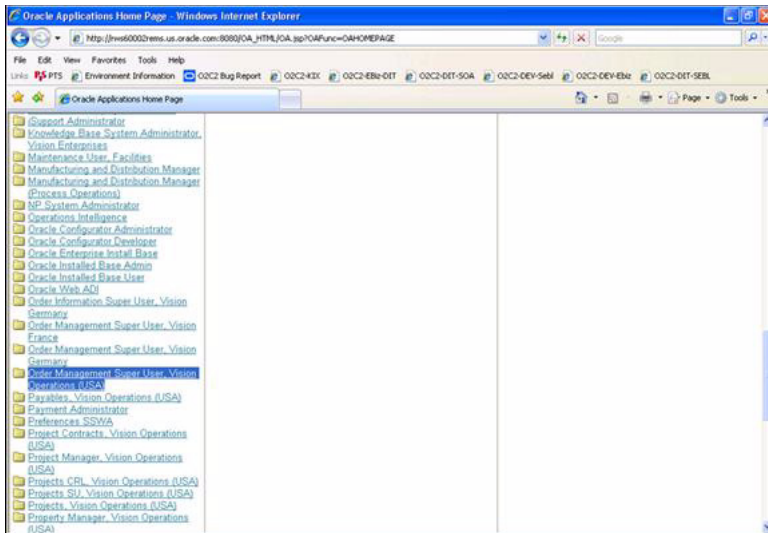
3. Identify the Item Validation Organization for Oracle E-Business Suite for the operating unit you want.

6.2.1.1 Identify the Item Validation Organization for Oracle E-Business Suite

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

1. Log in to Oracle E-Business Suite. The following page appears:

Figure 6–1 Oracle applications home page example

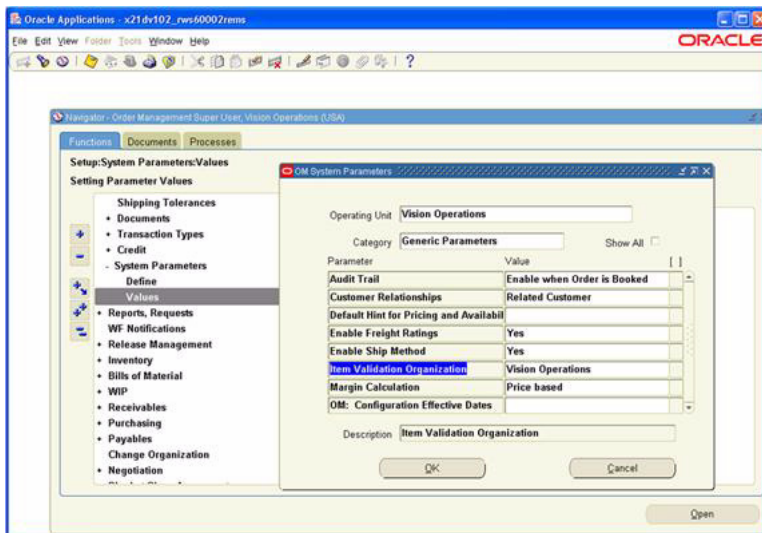


2. Confirm that you have the responsibilities set up for your operating units of interest, for example, order management.

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).

Figure 6–2 System parameters



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 the name of the Operating Unit and Generic Parameters in Category.

- c. Write down the value that appears in Item Validation Organization. (Examples: Vision Operations, Vision Germany, and Vision France).

6.2.2 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.

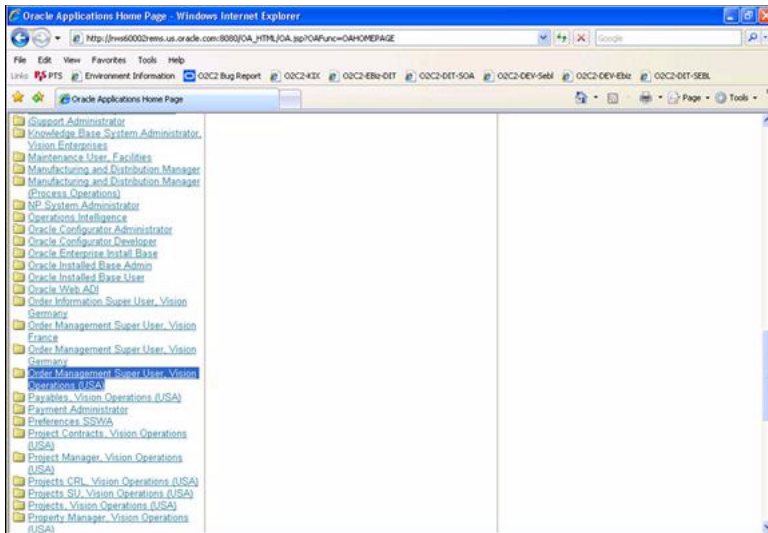
6.2.2.1 Identify Item Validation Organization for Oracle Product Hub

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

Publishing a product is on-demand by a product administrator from OPH. 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 OM is only published to Siebel.

1. Log in to Oracle Product Hub. The following page appears:

Figure 6–3 Oracle applications home page example

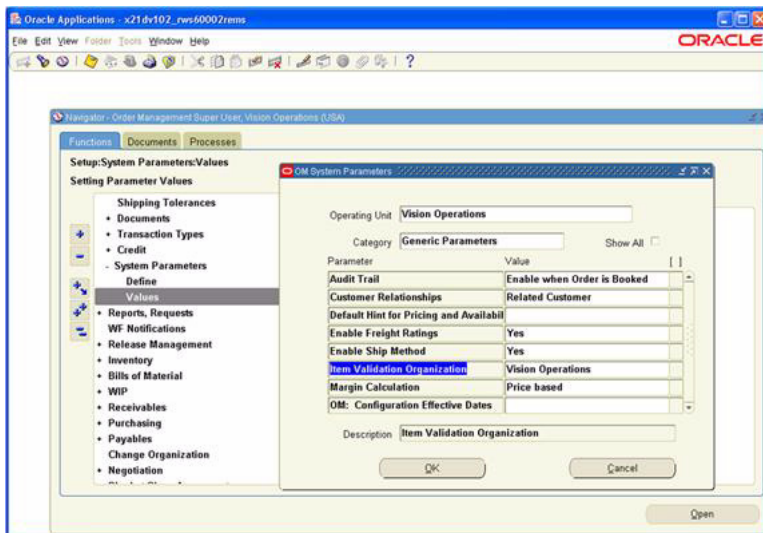


2. Confirm that you have the responsibilities set up for your operating units of interest, for example, order management.

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).

Figure 6–4 System parameters



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.

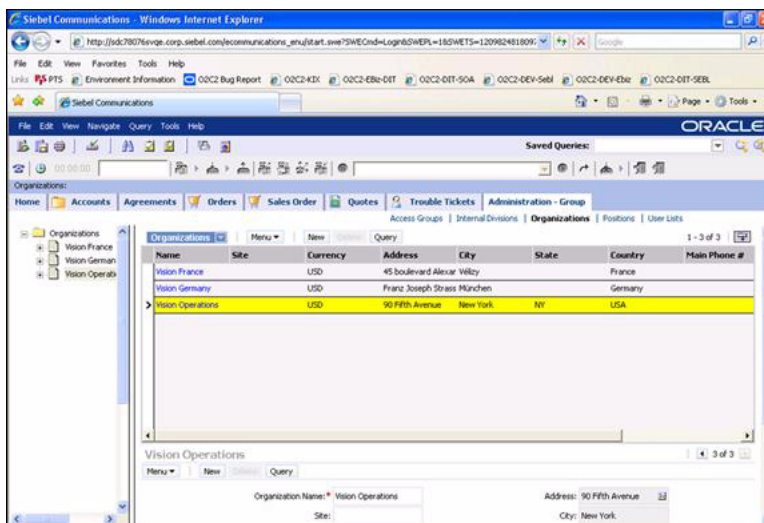
- c. Write down the value that appears in Item validation organization. (Examples: Vision Operations, Vision Germany, and Vision France).

6.2.3 Mapping Siebel CRM

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

1. Log in to Siebel Applications.
2. Click Site Map.
3. Select Administration-Groups, Organizations.
4. For every operating unit in Oracle E-Business Suite, create a corresponding organization in Siebel CRM. [Figure 6–5](#) shows a screen shot:

Figure 6–5 Administration - Group tab in Siebel



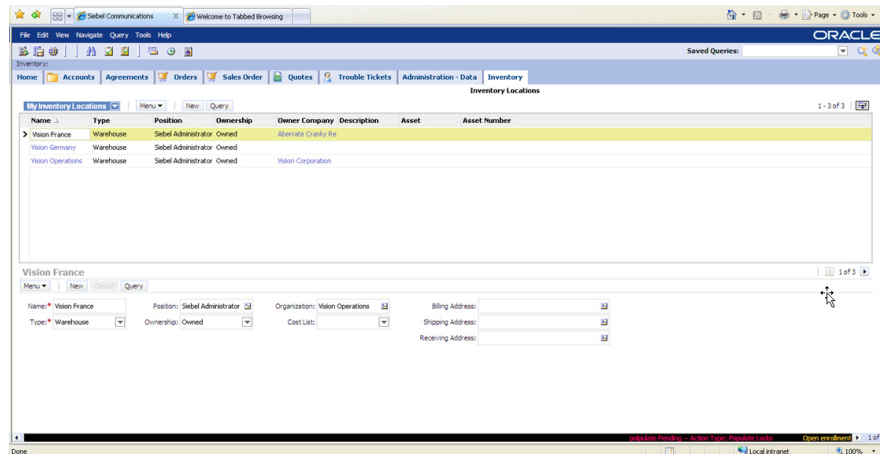
For more information, see the Siebel product documentation.

Creating Inventory Locations in Siebel

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

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

For the Oracle Inventory Organizations against which orders will be validated and availability is checked and also from where items are shipped during order fulfillment have to be created in this screen in Siebel with Inventory Type 'Warehouse'. The corresponding operating unit should also be associated to the Inventory Location as 'Organization' as shown in [Figure 6–6](#):

Figure 6–6 Creating Inventory Locations in Siebel

6.2.4 Extract Business Unit and Inventory Location IDs in Siebel CRM

To extract inventory location details for Siebel CRM:

1. Log in to the Siebel 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%')
```

6.3 Setting Up Oracle BRM

These are the setups required for Oracle BRM:

- Install and configure BRM JCA adapter.
For more information about installing and setting up the BRM JCA adapter, see "Installing the BRM JCA Resource Adapter" and "Service Integration Components, JCA Resource Adapter" in the Oracle BRM documentation.
- Set up all metadata required for creating products, price list 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 'About setting up a price list' in the Oracle BRM documentation.
- Set up pricing for friends and family functionality.
For more information about setting up pricing, see 'Setting Up Pricing and Rating, Working with extended rating attributes' in the Oracle BRM documentation.

6.4 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:

- Section 6.4.1, "Populating Seed Values in Cross-Reference Tables"
- Section 6.4.2, "Validating Cross-References"
- Section 6.4.3, "Creating a Pricelist in Siebel"

6.4.1 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 PIP, 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 Oracle E-Business Suite: EBIZ_01 (if the Oracle E-Business Suite option is installed).
- 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 following columns:

Table 6–3 XRef Table Format

Column Name	Description
XREF_TABLE_NAME	This column stores the XRef data types. For example: <ul style="list-style-type: none"> ■ oramds:/apps/AIAMetaData/xref/ORGANIZATION_ID.xref ■ oramds:/apps/AIAMetaData/xref/INVENTORY_LOCATION_ID.xref
XREF_COLUMN_NAME	This column stores the values indicating the edge applications and COMMON value: <ul style="list-style-type: none"> ■ PIM_01 ■ EBIZ_01 ■ COMMON ■ SEBL_01
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 6–4 Organization_ID XRef Table Value Format

XRef_Column_Name	Value
COMMON	GUID
PIM_01	Organization Id from OPH For example, 911
EBIZ_01	Ebiz Org ID For example, 204

Table 6–4 (Cont.) Organization_ID XRef Table Value Format

XRef_Column_Name	Value
SEBL_01	Organization Id (Business Unit) from Siebel CRM For example, 88-8KO98

Sample SQL Insert Statement for ORGANIZATION_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 EBIZ_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', 'EBIZ_01',
'BD618D40B30C11DEBFA5D9490F57512E', '204', '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', '2d373037303636323539363632323735', 'N',
SYSTIMESTAMP)
```

Table 6–5 Sample Data for XRef Organization_ID Table

XRef_ Table	XRef_ Column	Row_ Numb er	Value	IS_Deleted	Last_Modified
oramds:/a pps/AIA MetaData /xref/OR GANIZAT ION_ ID.xref	EBIZ_01	BD618 D40B3 0C11D EBFA5 D9490 F57512 E	204	N	2012-4-11.11.29. 54. 210000000
oramds:/a pps/AIA MetaData /xref/OR GANIZAT ION_ ID.xref	SEBL_01	BD618 D40B3 0C11D EBFA5 D9490 F57512 E	88-8K O98	N	2012-4-11.11.29. 54. 210000000

Table 6–5 (Cont.) 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	PIM_01	BD618D40B30C11DEBFA5D9490F57512E	911	N	2012-4-11.11.29. 54. 210000000
oramds:/apps/AIAMetaData/xref/COMMON	COMMON	BD618D40B30C11DEBFA5D9490F57512E	2d373036	N	2012-4-11.11.29. 54. 210000000

Table 6–6 INVENTORY_LOCATION_ID XRef Table Value Format

XRef_Column_Name	Value
COMMON	GUID
PIM_01	INVENTORY_LOCATION_ID from OPH 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 EBIZ_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', 'EBIZ_01', 'BD618D40B30C11DEBFA5D9490F57512E', '204', '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', '2d373037303636323539363632323735', 'N',
SYSTIMESTAMP)
```

Table 6–7 Sample Data for XRef INVENTORY_LOCATION_ID Table

XRef Table	XRef Column	Row Number	Value	IS_Deleted	Last_Modified
oramds:/a pps/AIA MetaData /xref/INV ENTORY_ LOCATIO N_ID	EBIZ_01	BD618 D40B3 0C11D EBFA5 D9490 F57512 E	204	N	2012-4-11.11.29. 54. 210000000
oramds:/a pps/AIA MetaData /xref/INV ENTORY_ LOCATIO N_ID	SEBL_01	BD618 D40B3 0C11D EBFA5 D9490 F57512 E	88-8K O98	N	2012-4-11.11.29. 54. 210000000
oramds:/a pps/AIA MetaData /xref/INV ENTORY_ LOCATIO N_ID	PIM_01	BD618 D40B3 0C11D EBFA5 D9490 F57512 E	911	N	2012-4-11.11.29. 54. 210000000
oramds:/a pps/AIA MetaData /xref/INV ENTORY_ LOCATIO N_ID	COMMON	BD618 D40B3 0C11D EBFA5 D9490 F57512 E	2d3730 373036 363235 393636 323237 35	N	2012-4-11.11.29. 54. 210000000

6.4.2 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.

Example 6–1 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 = 'INVENTORY_LOCATION_ID'
and value in ('204'))
```

Example 6–2 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 = 'ORGANIZATION_ID'
and value in ('204')
```

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.)

6.4.3 Creating a Pricelist in Siebel

To create a price list in Siebel CRM:

1. Create a price list in Siebel CRM.
2. Extract the RowID of the pricelist in Siebel and update the Property PriceList.ID with the RowID of the pricelist in AIAConfigurationProperties for the following:
 - a. SyncItemCompositionListSiebelCommsProvABCImpl
 - b. ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl

For more information, see [Section 6.8, "Setting Configuration Properties"](#).

6.5 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 cross-references, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite "Working with Cross References"*.

[Table 6–8](#) lists the cross-references for the Product MDM integration:

Table 6–8 OPH cross-references

Table Name	Column Name	Description	Use
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 ICC 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	To cross reference Price List Line Id	Mapping of pricelist line ids.
PRICELINETYPE_ID	EBIZ_01,COMMON, SEBL_01, PIM_01	To cross-reference Line Price Id.	To cross-reference BRM product's events with Siebel price line ID. Also links to the COMMON of ITEM_ID.

Table 6–8 (Cont.) OPH cross-references

Table Name	Column Name	Description	Use
SIEBELPRODUCTEV ENTXREF	ITEM_ID_ COMMON,LINEPRI CETYPECODE	To cross-reference Event Item that is Associated with the main Item.	To cross-reference BRM (Portal) Product's Event that is Associated with the main product in Siebel.
PROMOTION_ PROMOTIONID	COMMON, SEBL_01	To cross-reference Promotion ID	During product synchronization Promotion Id is stamped in this xref based Entity Type. During Product Structure Synchronization this xref is looked up to determine Product is promotion.

6.6 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 directories

```
<AIAHOME> // AIAMetaData / dvm /
```

```
<AIAHOME> // AIAMetaData / xref /
```

They can also be looked at by logging into MDS.

Table 6–9 lists the DVMs for the Product MDM integration:

Table 6–9 List of DVMs

DVM Type	DVM Column Name	Description
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.

Table 6–9 (Cont.) List of DVMs

DVM Type	DVM Column Name	Description
ENTITY_TO_ TARGET_ APPLICATION	TARGET_ID, RECEIVE_ PRICELIST_FLAG , RECEIVE_ BOM_FLAG	List of entities that each target application receives, for example, Oracle E-Business Suite receives Items and BOMs; Siebel receives item, price list 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_ CURRENTFUTUREP AST_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 6–9 (Cont.) List of DVMs

DVM Type	DVM Column Name	Description
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,CRMOD_01	Indicates item-type code.
ITEM_ENTITY_TYPE	SEBL_01, PIM_01, COMMON	Identifies the type of entity that OPH/PIM 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 OPH/PIM, Siebel, 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 kind of rate restriction used in the plan tier, for example, No Restrictions, Restriction based on days only, or Restriction based on days and 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, BRM_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, CRMOD_01	Cross-reference value: product status.

Table 6–9 (Cont.) List of DVMs

DVM Type	DVM Column Name	Description
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 to which synchronization is being done. This DVM is used to map the system defined in OPH/PIM for publication and system defined in OER.
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 as active, pending, or closed.
ITEM_ PRIMARYCLASSIFI CATIONCODE	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 for mapping 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.

Table 6–9 (Cont.) List of DVMs

DVM Type	DVM Column Name	Description
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, CASCADE, 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_ PURCHASEPRORAT IONCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates how to charge customers when they purchase the product in the middle of the accounting cycle, for example, Do not Charge for the cycle; Calculate Charge based on the amount used; and Charge for the entire cycle, and so on.
RATEDATA_ CANCELATIONPRO RATIONCODE	PIM_01, COMMON, BRM_01, SEBL_01	Maps rate data cancelation proration code. Indicates how to charge customers when they cancel the product in the middle of the accounting cycle, for example, Do not Charge for the cycle; Calculate Charge based on the amount used; Charge for the entire cycle; and so on.
RESOURCE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates non-currency resource codes, for example, minutes, loyalty points, and so on.
GL_CODE	PIM_01, COMMON, EBIZ_01, BRM_01	Identifies the set of products in general ledger.
PRICE_ IMPACTCLASSIFIC ATIONCODE	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_ DISCOUNTBRACKE TCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the discount brackets to charge different amounts based on quantity, for example, Continuous, Rate Dependent, Resource, and so on.

Table 6–9 (Cont.) List of DVMs

DVM Type	DVM Column Name	Description
PLANTIER_ EFFECTIVITYMODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the effectivity mode associated with the date and time of the tier, for example, Absolute, Relative.
RATEPLAN_ TAXAPPLICABILITY CODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates when the tax must be applied, 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.

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 Oracle E-Business Suite. There is no Siebel equivalent for STRUCTURE_NAME_ALTERNATE_BOM.
- **STRUCTURE_TYPE:** This is used to identify the type for every structure 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.

Table 6–10 provides sample data for the STRUCTURE_TYPE domain value map:

Table 6–10 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.

6.7 Handling Errors

Based on the roles defined for the services, email notifications are sent if a service errors out. There are no AIA-specific errors thrown by the services for this integration.

For more information about the errors thrown by Siebel CRM or Oracle E-Business Suite, see the respective product documentation.

For more information about AIA error handling, see the *Oracle Fusion Middleware Infrastructure Components and Utilities User's Guide for Oracle Application Integration Architecture Foundation Pack* "Setting Up Error Handling."

6.8 Setting Configuration Properties

Set these properties in the AIAConfigurationProperties.xml file. The file is located in AIA_HOME>/aia_instances/\$INSTANCE_NAME/AIAMetaData/config/.

For more information about requirements for working with AIAConfigurationProperties.xml, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Building AIA Integration Flows," How to Set Up AIA Workstation.

6.8.1 Settings for the SyncItemLEbizProvABCImpl service

Table 6–11 SyncItemLEbizProvABCImpl

System Property	Value/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 requestor application fails to send this, AIA picks the default system ID from this configuration property.	--
Routing.SyncItemLEbizAdapter.CAVS.Endpoint URI	This property sets the endpoint URL for the CAVS simulator.	--

Table 6–11 (Cont.) SyncItemListEbizProvABCImpl

System Property	Value/Default Value	Description
Routing.SyncItemListEbizAdapter.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SyncItemListEbizAdapter.EBIZ_01.EndpointURI	Endpoint URI of the Oracle E-Business Suite adapter	--
Routing.ItemResponseEBSV2.SyncItemListResponse.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.ItemResponseEBSV2.SyncItemListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.ItemResponseEBSV2.SyncItemListResponse.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
ABCSExtension.PoreXformEBMtoABMEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PoreInvokeSyncItemListABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeSyncItemListABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformABMtoEBMEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
TRACE.LOG.ENABLED	This property is used enabling trace log	--

6.8.2 Settings for the SyncProductSiebelProvABCImpl service

Table 6–12 *SyncProductSiebelProvABCImpl*

Property Name	Value/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 requestor application fails to send this, AIA picks the default system ID from this configuration property.
SIMPLE_PRODUCT_RELEASE_FLAG	Y/N	This property used for releasing simple product. To enable Siebel workspace auto release functionality, this flag must be set to N.
COMPLEX_PRODUCT_RELEASE_FLAG	Y/N	This property used for releasing complex product
Routing.SWIProductIntegration.SEBL_01.EndpointURI	Siebel web service end-point location. This is a SOAP endpoint URL.	--
Routing.SWIProductIntegration.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SWIProductIntegration.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.SWIPromotionIntegration.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SWIPromotionIntegration.SEBL_01.EndpointURI	Siebel web service end-point location. This is a SOAP endpoint URL.	--
Routing.SWIPromotionIntegration.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.ItemResponseEBSV2.SyncItemListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.ItemResponseEBSV2.SyncItemListResponse.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.ItemResponseEBSV2.SyncItemListResponse.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--

Table 6–12 (Cont.) SyncProductSiebelProvABCSImpl

Property Name	Value/Default Value	Description
ABCSExtension.PreXformEBMtoABMEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeSWIPProductImportUpsertABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeSWIPProductImportUpsertABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformABMtoEBMEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeSWIPromotionIntegrationABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeSWIPromotionIntegrationABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
Product.Source.PIP	MDM/O2C	This property is updated during installation. For MDM integration, the value must be set to MDM.
ABCSExtension.PreInvokeSWIPromotionIntegrationABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeSWIPromotionIntegrationABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.

Table 6–12 (Cont.) SyncProductSiebelProvABCImpl

Property Name	Value/Default Value	Description
OPERATION_TYPE	Y/N; N	<p>This property indicates whether Siebel web service is to be used in synchronize or upsert mode.</p> <p>OPERATION_TYPE = 'Y': The elements in payload are updated if they already exist in the system. Else, are inserted (if they do not already exist). If the element is not present in payload, but present in system then it is not deleted from the system.</p> <p>OPERATION_TYPE = 'N' (Default): The elements in payload are updated if they already exist. Else are inserted (if they do not already exist). If the element is not present in payload, but present in system then it is not deleted from system.</p>
UTC_FORMAT_ENABLE	Y/N; N	<p>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.</p>

6.8.3 Settings for the SyncBillOfMaterialsListEbizProvABCImpl service

Table 6–13 SyncBillOfMaterialsListEbizProvABCImpl

Property Name	Value/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 requestor application fails to send this, AIA picks the default system ID from this configuration property.	--
Routing.SyncBillOfMaterialsListEbizAdapter.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.SyncBillOfMaterialsListEbizAdapter.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SyncBillOfMaterialsListEbizAdapter.EBIZ_01.EndpointURI	Endpoint URI of the Oracle E-Business Suite adapter	--
Routing.BillOfMaterialsEBS.SyncBillOfMaterialsList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.

Table 6–13 (Cont.) SyncBillOfMaterialsListEbizProvABCImpl

Property Name	Value/Default Value	Description
Routing.BillOfMaterialsResponseEBS.SyncBillOfMaterialsListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.BillOfMaterialsResponseEBS.SyncBillOfMaterialsListResponse.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.BillOfMaterialsResponseEBS.SyncBillOfMaterialsListResponse.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
ABCSExtension.PreXformEBMtoABMEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeSyncBillOfMaterialsListEbizAdapterABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeSyncBillOfMaterialsListEbizAdapterABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformABMtoEBMEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.

6.8.4 Settings for the SyncBillOfMaterialsListSiebelProvABCImpl service

Table 6–14 SyncBillOfMaterialsListSiebelProvABCImpl

Property Name	Value/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 requestor application fails to send this, AIA picks the default system ID from this configuration property.	--

Table 6–14 (Cont.) SyncBillOfMaterialsListSiebelProvABCImpl

Property Name	Value/Default Value	Description
Routing.SWIPProductIntegration.SEBL_01.EndpointURI	Siebel web service end-point location. This is a SOAP endpoint URL.	--
Routing.SWIPProductIntegration.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SWIPProductIntegration.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.SWIPromotionIntegration.SEBL_01.EndpointURI	Siebel web service end-point location. This is a SOAP endpoint URL.	--
Routing.SWIPromotionIntegration.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SWIPromotionIntegration.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.BillOfMaterialsResponseEBS.SyncBillOfMaterialsListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.BillOfMaterialsResponseEBS.SyncBillOfMaterialsListResponse.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.BillOfMaterialsResponseEBS.SyncBillOfMaterialsListResponse.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
SIMPLE_PRODUCT_RELEASE_FLAG	This property used for Releasing Simple Product	--
COMPLEX_PRODUCT_RELEASE_FLAG	This property used for Releasing Complex Product	--
ABCSExtension.PreXformEBMtoABMEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeSWIPProductImportUpsertABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.

Table 6–14 (Cont.) SyncBillOfMaterialsListSiebelProvABCImpl

Property Name	Value/Default Value	Description
ABCSExtension.PostInvoiceSWIPProductImportUpsertABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformABMtoEBMEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvoiceSWIPromotionIntegrationABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvoiceSWIPromotionIntegrationABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
OPERATION_TYPE	Y/N; N	<p>This property indicates whether Siebel web service is to be used in synchronize or upsert mode.</p> <p>OPERATION_TYPE='Y': The elements in payload are updated if they already exist in the system. Else, are inserted (if they do not already exist). If the element is not present in payload, but present in system then it is not deleted from the system.</p> <p>OPERATION_TYPE='N' (Default): The elements in payload are updated if they already exist. Else are inserted (if they do not already exist). If the element is not present in payload, but present in system then it is not deleted from system.</p>
Default.SystemID	It is the responsibility of the application to send the system ID from which the request is being sent. If any requestor application fails to send this, AIA picks the default system ID from this configuration property.	--
Routing.SWIPProductIntegration.SEBL_01.EndpointURI	Siebel web service end-point location. This is a SOAP endpoint URL.	--
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.

6.8.5 Settings for the SyncItemListPIMReqABCSImpl service

Table 6–15 *SyncItemListPIMReqABCSImpl*

System Property	Value/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 requestor application fails to send this, AIA picks the default system ID from this configuration property.	--
Routing.QueryItemListPIMAdapter.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.QueryItemListPIMAdapter.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.QueryItemListPIMAdapter.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.QueryBillOfMaterialsListPIMAdapter.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.QueryBillOfMaterialsListPIMAdapter.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.QueryBillOfMaterialsListPIMAdapter.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.GetListofTargetSystems.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.GetListofTargetSystems.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.PublicationService.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.PublicationService.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--

Table 6–15 (Cont.) SyncItemListPIMReqABCImpl

System Property	Value/Default Value	Description
Routing.ItemEBSV2.SyncItemList.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.ItemEBSV2.SyncItemList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.ItemEBSV2.SyncItemList.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.BillOfMaterialsEBS.SyncBillOfMaterialsList.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.BillOfMaterialsEBS.SyncBillOfMaterialsList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.BillOfMaterialsEBS.SyncBillOfMaterialsList.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.PriceListEBSV2.SyncPriceListList.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.PriceListEBSV2.SyncPriceListList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.PriceListEBSV2.SyncPriceListList.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
ABCSExtension.PreXformABMtoEBMItemListABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeItemListEBSSyncItemListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformEBMtoABMItemListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.

Table 6–15 (Cont.) SyncItemListPIMReqABCImpl

System Property	Value/Default Value	Description
ABCSExtension.PreInvokePublicationServiceItemEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokePriceListEBSSyncItemEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformEBMtoABMPriceListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokePublicationServicePriceListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreXformABMtoEBMIBillofMaterialsListABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeBillofMaterialsListEBSSyncBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformEBMtoABMBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokePublicationServiceBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether 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.	--

6.8.6 Settings for SyncSpecificationValueSetListSiebelProvABCImpl service

Table 6–16 *SyncSpecificationValueSetListSiebelProvABCImpl*

System Property	Value/Default Value	Description
Default.SystemID	I	It is the responsibility of the application to send the system ID from which the request is being sent. If any requestor application fails to send this, AIA will select 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. It determines whether invocation of service at the extension point is to be made, depending on whether it is true or false.
ABCSExtension.PostXformABMtoEBMSpecificationValueSetListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made, depending on whether it is true or false.
Routing.SWIAtributeImport.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SWIAtributeImport.CAVS.EndpointURI		This property sets the endpoint URL for the CAVS simulator.
Routing.QueryBillOfMaterialsListPIMAdapter.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.QueryBillOfMaterialsListPIMAdapter.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.GetListofTargetSystems.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.GetListofTargetSystems.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.PublicationService.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.PublicationService.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.SWIAtributeImport.SEBL_01.EndpointURI		Siebel web service end-point location. This is a SOAP endpoint URL.

Table 6–16 (Cont.) SyncSpecificationValueSetListSiebelProvABCSImpl

System Property	Value/Default Value	Description
Routing.SWIAtributeImport.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 the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SpecificationValueSetResponseEBS.SyncSpecificationValueSetListResponse.CAVS.EndpointURI		This property sets the endpoint URL for the CAVS simulator.
ABCSExtension.PreInvokeABSSyncAttributeDefinitionABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made, depending on whether it is true or false.
ABCSExtension.PostInvokeABSSyncAttributeDefinitionABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made, depending on whether it is true or false.
OPERATION_TYPE	Y/N; N	<p>This property indicates whether Siebel web service is to be used in synchronize or upsert mode.</p> <p>OPERATION_TYPE = 'Y': The elements in payload are updated if they already exist in the system. Else, are inserted (if they do not already exist). If the element is not present in payload, but present in system then it is not deleted from the system.</p> <p>OPERATION_TYPE = 'N' (Default): The elements in payload are updated if they already exist. Else are inserted (if they do not already exist). If the element is not present in payload, but present in system then it is not deleted from system.</p>
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.
ABCSExtension.PreXformABMtoEBMItemListABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.

Table 6–16 (Cont.) SyncSpecificationValueSetListSiebelProvABCImpl

System Property	Value/Default Value	Description
ABCSExtension. PreInvokeItemEBSSyncItemEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension. PostXformEBMtoABMItemEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension. PreInvokePublicationServiceItemEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension. PreInvokePriceListEBSSyncItemEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension. PostXformEBMtoABMPriceListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension. PreInvokePublicationServicePriceListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension. PreXformABMtoEBMIBillofMaterialsListABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeBillofMaterialsListEBSSyncBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformEBMtoABMBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokePublicationServiceBOMListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.

Table 6–16 (Cont.) SyncSpecificationValueSetListSiebelProvABCSImpl

System Property	Value/Default Value	Description
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.

6.8.7 Settings for the SyncItemListBRMProvABCSImpl service

Table 6–17 SyncItemListBRMProvABCSImpl

System Property	Value/Default Value	Description
Default.SystemID	BRM_01	The application is responsible for sending the system ID from which the request is being sent. If any requestor application fails to send this, AIA picks the default system ID from this configuration property.
ABCSExtension.PreXformEBMtoABM	false/true	Used as an extension point before EBM is transformed to ABM. This property determines whether to make invocation of service at the extension point depending on whether it is true or false.
ABCSExtension.PostXformEBMtoABM	false/true	Used as an extension point before EBM is transformed to ABM. This property determines whether to make invocation of service at the extension point depending on whether it is true or false.
Routing.BRMPreICINGServiceRouteToCAVS	false/true	Determines whether to route the endpoint URL to either the end application service or CAVS for simulating the service.
Routing.BRMPreICINGServiceCAVS.EndpointURI	NA	Sets the endpoint URL for the CAVS simulator.
Routing.BRMPreICINGServiceBRM01.EndpointURI	eis/BRM1	Sets the endpoint URL for the target application.
Routing.BRMPreICINGServiceMessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.

Table 6–17 (Cont.) SyncItemListBRMProvABCImpl

System Property	Value/Default Value	Description
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 whether 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 whether it is true or false.

6.8.8 Settings for the SyncPriceListListBRMProvABCImpl service

Table 6–18 SyncPriceListListBRMProvABCImpl

Property Name	Property Value	Description
Default.SystemID	BRM_01	The application is responsible for sending the SystemID from which the request is being sent. If any requestor application fails to send this, AIA will pick the default SystemID from this config property.
ABCSExtension.PreXformEBMtoABM	false/true	Used as an extension point before EBM is transformed to ABM. This property determines whether to make invocation of service at the extension point depending on whether it is true or false
ABCSExtension.PostXformEBMtoABM	false/true	Used as an extension point After the EBM is transformed to ABM. This property determines whether to make invocation of service at the extension point depending on whether it is true or false
Routing.BRMPRICING Service_ptt.RouteToCAVS	NA	Determines whether to route the EndpointURI to either the end application service or CAVS for simulating the service.
Routing.BRMPRICING Service_ptt.CAVS.EndpointURI	NA	Sets the EndpointURI for the CAVS simulator
Routing.BRMPRICING Service_ptt.BRM_01.EndpointURI	eis/BRM1	Sets the EndpointURI for the target application
Routing.BRMPRICING Service_ptt.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Sets the EnvironmentCode element to the a 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 whether it is true or false.

Table 6–18 (Cont.) SyncPriceListListBRMProvABCImpl

Property Name	Property Value	Description
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 whether it is true or false.

6.8.9 Settings for the QueryBillOfMaterialsListPIMAdapter service

Table 6–19 QueryBillOfMaterialsListPIMAdapter

System Property	Value/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 requestor application fails to send this, AIA picks the default system ID from this configuration property.	--
BATCH_SIZE	<<any number>>	This property is used for defining the number of structures to be processed in sub batch (chunking)
Routing.ItemService.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.ItemService.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.ItemService.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--

6.8.10 Settings for the QueryItemListPIMAdapter service

Table 6–20 QueryItemListPIMAdapter

System Property	Value/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 requestor application fails to send this, AIA picks the default system ID from this configuration property.	--
BATCH_SIZE	<<any number>>	This property is used for defining the number of items to be processed in sub batch (chunking)

Table 6–20 (Cont.) QueryItemListPIMAdapter

System Property	Value/Default Value	Description
Routing.ItemService.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.ItemService.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.ItemService.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--

6.8.11 Settings for the SyncSpecificationValueSetListPIMReqABCImpl service

Table 6–21 SyncSpecificationValueSetListPIMReqABCImpl

System Property	Value/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 requestor application fails to send this, AIA picks the default system ID from this configuration property.	--
TRACE.LOG.ENABLED	This property is used enabling trace log	--
ABCSExtension.PreInvokeValueSetServiceABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeValueSetServiceABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeEBSpecificationValueSetListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeEBSpecificationValueSetListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
Routing.ValuesetService.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.

Table 6–21 (Cont.) SyncSpecificationValueSetListPIMReqABCImpl

System Property	Value/Default Value	Description
Routing.PublicationService.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.PublicationService.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.ValueSetService.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.SpecificationValueSetEBS.SyncSpecificationValueSetList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.PublicationService.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.ValueSetService.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.SpecificationValueSetEBS.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SpecificationValueSetEBS.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--

6.8.12 Settings for the SyncItemCatalogCategoryPIMReqABCImpl service

Table 6–22 SyncItemCatalogCategoryPIMReqABCImpl

System Property	Value/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 requestor application fails to send this, AIA picks the default system ID from this configuration property.	--
ABCSExtension.PreXformABMtoEBMItemCatalogCategoryABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.

Table 6–22 (Cont.) SyncItemCatalogCategoryPIMReqABCImpl

System Property	Value/Default Value	Description
ABCSExtension.PostXformABMtoEBMItemCatalogCategoryABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeEBSSyncClassificationSchemeListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeEBSSyncClassificationSchemeListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
Routing.ItemCatalogCategoryService.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.PublicationService.PIM_01.EndpointURI	OPH/PIM web service end-point location. This is a SOAP endpoint URL.	--
Routing.ClassificationSchemeEBS.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SpecificationValueSetEBS.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.ItemCatalogCategoryService.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.PublicationService.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.ClassificationSchemeEBS.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.SpecificationValueSetEBS.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.ItemCatalogCategoryService.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.PublicationService.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--

Table 6–22 (Cont.) SyncItemCatalogCategoryPIMReqABCSImpl

System Property	Value/Default Value	Description
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	--

6.8.13 Settings for the SyncClassificationSchemeListSiebelProvABCSImpl service

Table 6–23 SyncClassificationSchemeListSiebelProvABCSImpl

System Property	Value/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 requestor application fails to send this, 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. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PreInvokeABSSyncProductClassABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeABSSyncProductClassABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformABMtoEBMClassificationSchemeListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
Routing.ClassificationSchemeResponseEBS.SyncClassificationSchemeListResponse.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.

Table 6–23 (Cont.) SyncClassificationSchemeListSiebelProvABCImpl

System Property	Value/Default Value	Description
Routing.SyncProductClassService.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SyncProductClassService.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.ClassificationSchemeResponseEBS.SyncClassificationSchemeListResponse.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.SyncProductClassService.SEBL_01.EndpointURI	Siebel 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	--

6.8.14 Settings for the SyncClassificationSchemeListSiebelProvABCImpl service

Table 6–24 SyncClassificationSchemeListSiebelProvABCImpl

System Property	Value/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 requestor application fails to send this, 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. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostXformABMtoEBMSpecificationValueSetListEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
Routing.SWIAtributeImport.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.

Table 6–24 (Cont.) SyncClassificationSchemeListSiebelProvABCImpl

System Property	Value/Default Value	Description
Routing.SWIAAttributeImport.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
Routing.SWIAAttributeImport.SEBL_01.EndpointURI	Siebel web service end-point location. This is a 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 the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SpecificationValueSetResponseEBS.SyncSpecificationValueSetListResponse.CAVS.EndpointURI	This property sets the endpoint URL for the CAVS simulator.	--
ABCSExtension.PreInvokeABSSyncAttributeDefinitionABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
ABCSExtension.PostInvokeABSSyncAttributeDefinitionABM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines whether invocation of service at the extension point is to be made depending on whether it is true or false.
OPERATION_TYPE	Y/N; N	<p>This property indicates whether Siebel web service is to be used in synchronize or upsert mode.</p> <p>OPERATION_TYPE='Y': The elements in payload are updated if they already exist in the system. Else, are inserted (if they do not already exist). If the element is not present in payload, but present in system then it is not deleted from the system.</p> <p>OPERATION_TYPE='N' (Default): The elements in payload are updated if they already exist. Else are inserted (if they do not already exist). If the element is not present in payload, but present in system then it is not deleted from system.</p>

Table 6–24 (Cont.) SyncClassificationSchemeListSiebelProvABCImpl

System Property	Value/Default Value	Description
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.

6.8.15 Settings for the ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl service

Table 6–25 ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl

System Property	Value/Default Value	Description
Routing.Target.Default.SiebelProductService.SystemID	SEBL_01	Siebel 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 ProductImport 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.Target.Default.SiebelProductService.EndpointURI. If false, it invokes the verification system whose end point is indicated by the system-level property SyncResponseSimulator.Soap.EndpointURL.
Routing.Target.Default.SiebelPriceListService.SystemID	SEBL_01	Siebel system instance code (defined in BSR). This is used only if the request message does not contain the target system ID.
Routing.Target.Default.SiebelPriceListService.EndpointURI	No default value.	Siebel PriceList 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.SiebelPriceListService.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.SiebelPriceListService.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 system. The value for this property should be the ID of the business unit in the Siebel system. This value must be set before product sync is run.

Table 6–25 (Cont.) ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl

System Property	Value/Default Value	Description
Siebel.Product.Workspace Name	Demo Workspace	Name of the workspace to be used by Siebel. Create a workspace and update this file with that workspace name.
Siebel.Product.Workspace ReleaseFlag	Y/N. Default = N	Indicates whether the workspace needs to be released after the product is synchronized.
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 price list in the Siebel system. The value for this property should be the ID of the price list in the Siebel system. This value must be set before product sync is run.
Siebel.PriceList.Currency	USD	Currency code of the price list mentioned in the preceding property. If the currency of the prices in PriceListEBM does not match this currency, price in Siebel will be set to 0 (zero). This value must be set before the product sync is run.

6.8.16 Settings for the SyncItemCompositionListSiebelCommsProvABCSImpl service

Table 6–26 SyncItemCompositionListSiebelCommsProvABCSImpl

System Property	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel 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/asyncresponserecipient	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.

Table 6–26 (Cont.) SyncItemCompositionListSiebelCommsProvABCImpl

System Property	Value/Default Value	Description
Routing.SiebelProductService.SEBL_01.EndpointURI	http://<http.hostname>:<http.port>/eai_enu/start.swe?SWEEExtSource=SecureWebService&SWEEExtCmd=Execute&WSSOAP=1	Siebel 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 the Siebel system. The value for this property should be the ID of the business unit in the Siebel system. This value must be set before Product Sync is run.
Siebel.Product.WorkspaceName	Demo Workspace	Name of the workspace to be used by Siebel. Create a workspace and update this file with that workspace name.
Siebel.Product.WorkspaceReuseFlag	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 price list in the Siebel system. The value for this property should be the ID of the price list in the Siebel system. This value must be set before product sync is run.
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 (PreInvokeApplication).
ABCSExtension.PostInvokeABS	true/false. Default = false	Value determines whether the ABCS should invoke the Extension service (PostInvokeApplication).

6.9 Setting Up the Data for Seeded Attributes of the Telecommunications Library

The entities in [Table 6–27](#) have to be set up in target applications, and the corresponding attribute values set must be updated with the values in OPH.

Table 6–27 Attributes

Attribute Group Name	Attribute Name
Product Details: Logistics	Vendor Part Number (Siebel)
Subject compatibility rules	Matrix Name (Siebel)
Pricing: Simple Price List (For non-billing products)	Pricelist Name (Siebel)
	Volume Discount (Siebel)
	Attribute Adjustment (Siebel)
Promotion: Commitment: Charges Credits	Plan (Siebel)
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 6–28 lists the attributes and respective DVMs in which the values for these attributes must be added:

Table 6–28 DVM values

Attribute Name	DVM
GLID	GL_CODE
Provisioning Tag	PROVISIONING_TAG
Impact Category	PRICE_IMPACTCLASSIFICATIONCODE

6.10 Setting up List of Values in Siebel

Once the OPH integration is successfully installed and all the configuration setups are complete, these values need to be added to the PROD_CD list of values (LOV) in Siebel application:

- Commercial Bundle
- Discount
- Option Group
- Discount Model

Frequently Asked Questions

Q: Is it possible to synchronize items from OPH/PIM to Siebel 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 is based on a combination of product, business unit, and vendor; therefore, if such synchronization is done, the key does not remain unique.

Q: Why only set up the cross-references for inventory locations associated with item validation organization for Siebel?

A: Item synchronization to Siebel is based on the requirement that Siebel is able to use that item as part of the order capture process in the Order to Cash: Siebel CRM - EBS integration. In Oracle 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.

Q: How to extend the Siebel 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.

Q: How to configure the STRUCTURE_TYPE and STRUCTURE_NAME_ALTERNATE_BOM DVMs?

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 or not.

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.

Q: Would there be an issue if the same item validation organization were shared across multiple operating units?

A: See the answer for "Does the service support sharing of inventory organization across multiple operating units?"

Q: What are the default values for MDM-product related entries in the AIA Configuration properties XML?

A: See [Section 6.8, "Setting Configuration Properties"](#).

Q: When invoking OPH/PIM service, the OPH/PIM 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.

Q: When invoking OPH/PIM 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.

B

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

	Rate define d in OPH?	Rate sync'd to BRM?	Rate sync'd to SBL?	Revisi on define d in OPH?	Revision sync'd to BRM?	Revision sync'd to Siebel? *1	Effecti vity define d in OPH? (absol ute + relativ e)	Effectivity sync'd to BRM? (absolute + relative)	Absolute effectivity sync'd to Siebel?	Relative effectivity sync'd to Siebel? (Applies only to Rates and Balance Impact)
Promotion	No	No	No	Yes	No	Yes	Yes	No	Yes	N/A
Product (Acc level product/ SSB only)	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	N/A
Bundle Product	No *2	No	No	Yes	No	Yes	Yes	No	Yes	N/A
Tier 1	Yes	Yes	No *3	N/A	N/A	N/A	Yes	Yes	Yes	No
Balance Impact	Yes	Yes	Yes	N/A	N/A	N/A	Yes (Only absolut e)	Yes (Only absolute)	N/A *4	No
Tier 2	Yes	Yes	No *3	N/A	N/A	N/A	Yes	Yes	Yes	No
Balance Impact	Yes	Yes	Yes	N/A	N/A	N/A	Yes (Only absolut e)	Yes (Only absolute)	N/A*4	No
Discount (Acc level discount only)	N/A *5	N/A	N/A	Yes	No	Yes *6	Yes	Yes	Yes	N/A

There is no one-to-one mapping between revisions of items in OPH and versions of products in Siebel.

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 Communications Order to Cash Integration Pack for Siebel CRM, Oracle Communications Order and Service Management, and Oracle Communications Billing and Revenue Management 11.1 - Implementation Guide*.

The rate tier information is not synchronized to Siebel. Only the price specified in the balance impact attribute group (AG) associated with the rate tier is stored 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 OPH and synchronized to BRM. The actual definition is enriched in BRM.

Discounts are synchronized as products to Siebel.

User Defined Attributes Framework

This appendix includes the following sections:

- [Section C.1, "Extending for Custom UDAs"](#)
- [Section C.2, "Implementation of UDA Transformation"](#)
- [Section C.3, "Performance Optimization Using Chunking"](#)

C.1 Extending for Custom UDAs

The user defined attributes can be classified as:

- Seeded attributes.
- Custom attributes.
- Transaction attributes.

For more information about the different types of attributes and how they are handled, see 'Synchronization of Metadata' in [Section 2.1, "Supported Features and Process Flows"](#).

C.1.1 Handling UDAs

The Product MDM integration provides the following mappings:

- Operational attributes of the item (also referred as First Class Attributes) in OPH 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. Note that operational attributes are not mapped to product/discount definition in BRM
- Seeded attributes of the item (user defined attributes) in OPH are mapped to the fixed elements in the itemEBO which are mapped to the product definition attributes in Siebel and product/discount attributes in BRM. Note: Seeded attributes are not synchronized to EBIZ.
- Custom attributes of the item (User Defined Attributes) in Product Hub are mapped to the Specification group element in the itemEBO. Those custom attributes whose attribute group name start with XX are mapped to the user defined attributes in Siebel and all other custom attributes are ignored in the target application connector services for Siebel, Oracle BRM and Oracle E-Business Suite.
- Transaction attributes in OPH are mapped to the specificationEBO which are mapped to the attribute definition in Siebel. Note that the transaction attributes are not synchronized to Oracle E-Business Suite and Oracle BRM.

All downstream systems need not consume all the UDA attributes. The UDA service will dynamically populate the OPH/PIM ABM with only those UDA attributes that are instantiated as part of the particular Item Event raised by OPH.

- When an item is created from an item catalog category (ICC) that has a wide set of UDAs, 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.

C.1.2 Mapping Artifacts

The following artifacts specify the mapping from OPH to the canonical models.

C.1.2.1 Mapping Operational Attributes to ItemEBM

Artifacts:

- XformItemResponsePIMABMToSyncItemListEBM.xsl
- XformItemListResponsePIMABMToSyncItemListEBM_Custom.xsl

This is the mapping between OPH/PIM's operations attribute to EBM. Customer specific transformations related extensions should be done in XformItemListResponsePIMABMToSyncItemListEBM_Custom.xsl.

C.1.2.2 Mapping Seeded Set of UDAs for Communications library to EBM

Artifacts:

- XformUDAAGInterfaceCommsPIMBOMRespABMToSyncBOMEBM.xsl
- XformUDAAGInterfaceCommsPIMItemRespABMToSyncItemListEBM.xsl
- XformUDACommsPIMBOMRespABMToSyncBOMEBM.xsl
- XformUDACommsPIMBOMRespABMToSyncBOMEBM_Custom.xsl
- XformUDACommsPIMItemRespABMToSyncItemListEBM.xsl
- XformUDACommsPIMItemRespABMToSyncItemListEBM_Custom.xsl
- XformRestructureSyncBillOfMaterialsListEBM.xsl
- XformRestructureSyncItemListEBM.xsl

C.1.2.3 Mapping Pricelist related UDAs for Communications Library to EBM

- XfromListOfItemsToSyncPriceListListEBM.xsl

The entire communications library seeded UDAs are mapped to EBM. For each attribute group, a mapping template is declared in the Interface xsl.

The following xsls are used to implement the same:

- XformUDAAGInterfaceCommsPIMBOMRespABMToSyncBOMEBM.xsl
- XformUDAAGInterfaceCommsPIMItemRespABMToSyncItemListEBM.xsl

The mappings for UDAs are defined in the following xsl for each of the attributes within the attribute group:

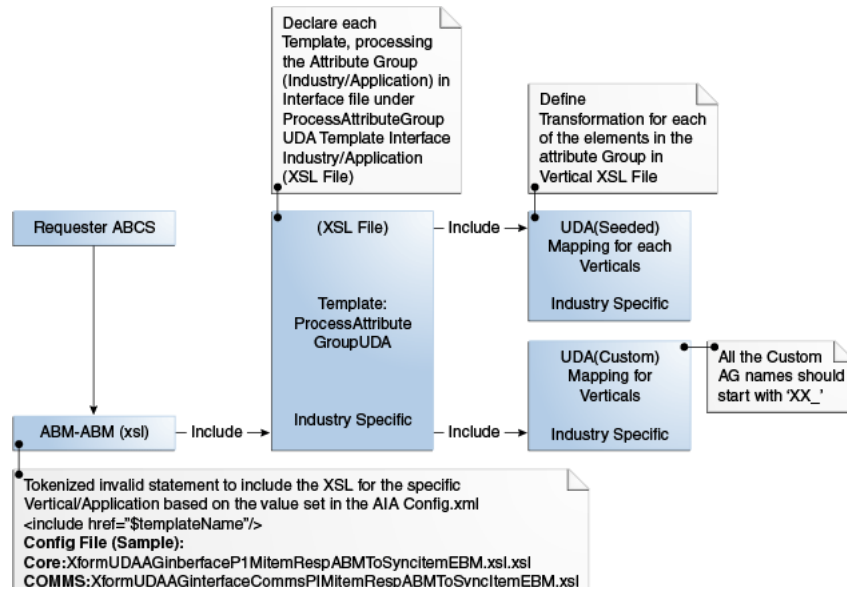
- XformUDACommsPIMBOMRespABMToSyncBOMEBM.xsl
- XformUDACommsPIMItemRespABMToSyncItemListEBM.xsl

The mapping related to the discount and promotion type entities are passed from OPH/PIM UDA to a Specification group in the EBM.

C.2 Implementation of UDA Transformation

Figure C-1 illustrates the UDAs to first class attributes (FCA) transformation in ItemEBM:

Figure C-1 Adding UDA to an 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 Attribute Group (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 attribute group in industry/application specific XSL file.

Figure C-2 provides an example of how to define transformations for each of the elements in the attribute group in the XformUDAComsPIMItemRespABMToSyncItemListEBM.xsl.

Figure C-2 Code Sample

```

<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 attribute group in interface file, under ProcessAttributeGroupUDA template.

Figure C-3 shows a code sample:

Figure C-3 Code sample

```

<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.

Figure C-4 shows a code sample:

Figure C-4 Code sample

```
<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:ItemManufacture
r"/>
<xsl:apply-templates
select="ns0:SyncItemList/corecom:ItemIdentification/corecom:Revision
/corecomEBO:RelatedItem"/>
<corecomEBO:ItemRevisionHistory>
</ns0:SyncItemList>
</ns0:DataArea>
```

5. Re-deploy the OPH/PIM requestor service.

C.2.1 Mapping Custom UDAs to EBM

Note: This is the recommended approach for handling custom UDAs.

- 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 attribute group name to avoid duplicate attributes across the attribute groups. For example, a new attribute SupplierName added to the attribute group Suppliers will have the internal name Suppliers_SupplierName.
- It is the responsibility of the provider to consume the data as required. Out-of-the-box these will be mapped to Flexfield attributes (User defined attributes user interface) in Siebel
- No change in the OPH/PIM Requestor is needed to support this.

C.2.2 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.xsl
- XformUDAAGInterfaceCommsPIMItemRespABMToSyncItemListEBM.xsl

- XformUDACommsPIMBOMRespABMToSyncBOMEBM_Custom.xml
- XformUDACommsPIMItemRespABMToSyncItemListEBM_Custom.xml
- XformRestructureSyncBillOfMaterialsListEBM.xml
- XformRestructureSyncItemListEBM.xml

C.2.3 Artifacts (Siebel Provider ABCS):

In Provider ABCS, specification group name starting with XX is mapped to Flexfield attributes (User defined attributes user interface) in Siebel.

- XformSyncItemListEBMToSiebelProdABM.xml
- XformUDAInterfaceCommsSiebelPromotionEBMToABM.xml
- XformUDACommsSiebelProductEBMToABM.xml
- XformUDACommsSiebelProductEBMToABM_Custom.xml
- XformRestructureProductABM.xml
- XformSyncItemListEBMToSiebelPromotionABM.xml
- XformUDAInterfaceCommsSiebelProductEBMToABM.xml
- XformUDACommsSiebelPromotionEBMToABM.xml
- XformUDACommsSiebelPromotionEBMToABM_Custom.xml
- XformRestructurePromotionABM.xml

C.3 Performance Optimization Using Chunking

As per the statistics and architecture proposal, the input payload for orchestration process should not exceed 2MB else it might lead to transaction time out/heap memory issue. To handle this issue, the OPH/PIM Req ABCS service invokes the OPH/PIM 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 OPH/PIM Query Adapter services to query the Items and BOMs, the services refer to the value of BATCH_SIZE as defined in 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 OPH/PIM batch output table. PL/SQL API: AIA_MDMITEMBATCH.

C.3.1 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 three scenarios for synchronizing items from Oracle Product Hub (OPH) to Oracle E-Business Suite when item versions in OPH and Oracle E-Business Suite are different.

The scenarios covered include the following:

- ASCII Code of the item version is set same in OPH and Oracle E-Business Suite.
- ASCII Code of the item version is set higher in OPH and is set lower in Oracle E-Business Suite.
- ASCII Code of the default item version is set higher in Oracle E-Business Suite and is set lower in OPH.

D.1 ASCII Code of the item version is set same in OPH and Oracle E-Business Suite

This is the Oracle recommended approach and has been tested and supported out-of-the-box (OOTB) in the Oracle Product Master Data Management (MDM) integration base pack. You must make sure to set the default item version number and the ASCII code of the item version number to the same value in OPH and Oracle E-Business Suite. For example, the value 0 is not same as 00.

D.2 ASCII Code of the item version is set higher in OPH and is set lower in Oracle E-Business Suite

In this scenario, the API used by the Product MDM pre-built integrations has a wrapper on the base API used to create items in the Oracle E-Business Suite.

To compare the effective date of an incoming item message, the wrapper API uses additional logic such as:

- If the effective date on the incoming item message is less than the current system datetime, then it updates the effective date for the item message to system datetime plus 2 seconds.
- If the item is synchronized from OPH to Oracle E-Business Suite for the first time, and the item version number is set higher than the default version for items in Oracle E-Business Suite, then the Oracle E-Business Suite API first creates a default version of the items and then creates a version synchronized from OPH.

The Product MDM pre-built integrations API uses the current system datetime as the effective date for the default version and then creates the next version of item with the effective date on the item which was the original system datetime plus 2 seconds set

by the wrapper API. The effective datetime for the default version of the item must be lesser than the effective datetime of subsequent versions of the item created in Oracle E-Business Suite.

D.2.1 Assumption

1. The processing time for the default version of the item is less than 2 seconds. The following table lists the examples.

Table D-1 When assumption 1 applies

OPH	Oracle 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	Assuming the item synchronization from OPH to Oracle E-Business Suite takes 10 seconds.
	Item version = 1 Effective datetime = 04/15/11 10:00:12	The Wrapper API sets the effective date time = system datetime + 2 seconds since the effective datetime is less than the system datetime.
	Item version = 1. Effective start date = 04/15/11 10:00:12. Effective end date = 04/15/11 10:00:12	The 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 =	The base API creates the next version of the item. Note: The effective start date of item version 1 is higher than the effective start date of default item version 0.

2. The processing time for the default version of the item is more than 2 seconds

Table D-2 When assumption 2 applies

OPH	Oracle 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	Assuming the item synchronization from OPH to Oracle E-Business Suite takes 10 seconds.
	Item version = 1 Effective datetime = 04/15/11 10:00:12	The Wrapper API sets the effective date time = system datetime + 2 seconds since the effective datetime is less than the system datetime.

Table D-2 (Cont.) When assumption 2 applies

OPH	Oracle E-Business Suite	Comments
	Item version = 1 Effective start date = 04/15/11 10:01:10 Effective end date =	The base API creates the default version first. Assumption 2 applies since the processing of default version takes more than 2 seconds. In this case, it takes 13 seconds.
	Itemversion = 1 Effective start date = 04/15/11 10:01:10 Effective end date =	When the base API tries to create the next version of the item, the effective start date of version 1 is less than the effective start date of the existing default version and the item creation fails.

D.2.2 Workaround

Customers whose Oracle E-Business Suite takes more than 2 seconds to process and create the default version of the item in Oracle E-Business Suite must update the value based on their individual requirements. For example, customers can increase the time interval to 60 seconds.

To update the time interval, follow these steps:

1. Backup the Inveipitb.pls file from customer's Oracle E-Business Suite instance.
2. Open the Inveipitb.pls file, and then modify the process_item API by following the steps listed below
 1. Replace the following code snippet with the code snippet listed in step 2.


```

                    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;
                
```
 2. 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;
                
```
 3. Save the changes, and then recompile the Inveipitb.pls file.

D.3 ASCII Code of the default item version is set higher in Oracle E-Business Suite and is set lower in OPH

This scenario is not supported by Oracle.

