

Oracle Product Master Data Management Integration

Implementation Guide

Release 2.5

E17549-04

January 2012

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.....	ii
Oracle Application Integration Architecture Foundation Pack Concepts and Technologies Guide	ii
Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide	iii
Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide	iii
Oracle Application Integration Architecture Process Integration Packs.....	iv
Additional Resources	iv
Chapter 1: Understanding the Product Master Data Management Integration	1
Overview.....	1
High-Level Process Flows.....	4
Core AIA Components	6
AIA Integration Services.....	7
Chapter 2: Oracle Product Master Data Management Integration Base Pack.....	13
Overview.....	13
Synchronization of Items and Bill of Materials from Oracle Product Hub	14
Synchronization of Metadata (value set and item catalog category) from Oracle Product Hub	59
Oracle Product Hub Interfaces	76
Oracle Product Hub Integration Services.....	78
Assumptions and Constraints.....	81
Chapter 3: Oracle Product Master Data Management Integration Option for Siebel CRM.....	84
Overview.....	84
Siebel CRM Interfaces	84
Siebel CRM Integration Services	85
Chapter 4: Oracle Product Master Data Management Integration Option for Oracle E-Business Suite	88
Overview.....	88
Oracle E-Business Suite Interfaces	88
Oracle E-Business Suite Integration Services	89
Set Up Users and Roles in Oracle E-Business Suite Instance	91

Chapter 5: Oracle Product Master Data Management Integration Option for Oracle Communications Billing and Revenue Management	93
Overview	93
Oracle BRM Interfaces	93
Oracle BRM Integration Services	94
Chapter 6: Implementing the Process Integration Pack for Oracle Product Hub	95
Adding a New Spoke Instance	95
Adding an Integration Option or Target Instance for MDM Integration	99
Extracting Organizations and Inventory Locations Data to Setup Cross-References	100
Setting Up Oracle BRM	108
Setting Up Cross-References	109
Identifying Cross-References	111
Describing Domain Value Maps	113
Handling Errors	118
Viewing EBO Implementation Maps (EIMs)	118
Setting Configuration Properties	119
Setting Up the Data for Seeded Attributes of PIM Telco Library	142
Frequently Asked Questions	145
Appendix A	147
Appendix B: User Defined Attributes Framework	149
Extending for Custom UDAs	149
Implementation of UDA Transformation	152
Performance Optimization Using Chunking	158
Appendix C: Cross References for the Process Integration for Product Management	160
Cross Reference Values	160
Integration Solution Cross References	162
Product Synchronization Flow	162
Discount Synchronization Flow	176
Index	179

Preface

This preface discusses:

- [The Oracle Application Integration Architecture - Foundation Pack: Concepts and Technologies Guide](#)
- [Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide](#)
- [Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide](#)
- [Oracle Application Integration Architecture Process Integration Packs](#)
- [Additional resources](#)

Oracle Application Integration Architecture Foundation Pack Concepts and Technologies Guide

The Oracle Application Integration Architecture - Foundation Pack: Concepts and Technologies Guide is a companion volume to the Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide and Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide. The Oracle Application Integration Architecture - Foundation Pack: Concepts and Technologies Guide provides definitions of fundamental Oracle Application Integration Architecture (AIA) concepts and discusses:

- Oracle AIA.
- Enterprise business objects and enterprise business messages.
- Enterprise business services.
- Application business connector services.
- Interaction patterns.
- Extensibility.
- Versioning.
- Business processes.
- Batch processing.
- Infrastructure services.
- Security.

Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide

Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide is a companion volume to Oracle Application Integration Architecture - Foundation Pack: Concepts and Technologies Guide and Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide.

The Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide discusses how to:

- Create an integration scenario.
- Define business service patterns.
- Design and develop enterprise business services.
- Design and develop enterprise business flows.
- Design and construct Application Business Connector Services (ABCS).
- Work with message transformation, enrichment, and configuration.
- Develop custom XPath functions.
- Design and construct JMS Adapter services.
- Work with enterprise message headers.
- Work with message routing.
- Work with transactions.
- Develop Oracle Application Integration Architecture (AIA) services to work with the CAVS.
- Configure Oracle AIA processes to be eligible for error handling and logging.
- Extend enterprise business objects (EBOs).

In addition, this book provides:

- AIA naming standards.
- Sample and template WSDLs for use with Oracle AIA.

Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide

Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide is a companion volume to Oracle Application Integration Architecture – Foundation Pack: Concepts and Technologies Guide and Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide.

Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide discusses how to:

- Work with the Composite Application Validation System (CAVS).
- Work with the Business Service Repository (BSR).
- Set up and use error handling and logging.
- Work with the diagnostics framework.

Oracle Application Integration Architecture Process Integration Packs

A process integration pack (PIP) is a prebuilt set of integrated orchestration flows, application integration logic, and extensible enterprise business objects and services required to manage the state and execution of a defined set of activities or tasks between specific Oracle applications associated with a given process. A PIP provides everything you need to deploy a selected integrated business process area. The PIP product offering is suited to those customers seeking for rapidly implementation of a discreet business process.

Additional Resources

These resources are available:

Resource	Location
AIA Pre-Built Integration Implementation Guides	Oracle Technology Network: http://download.oracle.com/docs/cd/E24010_01/index.htm
Foundation Pack Guides	Oracle Technology Network: http://download.oracle.com/docs/cd/E21764_01/aia.htm
Installation and Upgrade Guide	Oracle Technology Network: http://download.oracle.com/docs/cd/E24010_01/index.htm
Release Notes	Oracle Technology Network: http://download.oracle.com/docs/cd/E24010_01/index.htm

Visit the My Oracle Support website frequently to keep apprised of ongoing changes.

For other sources of documentation, visit [Oracle Technology Network: Oracle Documentation](#).

For training opportunities, visit [Oracle University](#).

Chapter 1: Understanding the Product Master Data Management Integration

This chapter provides an overview of the Oracle Product Hub process integration pack and covers:

- [Overview](#)
- [High-level process flows](#)
- [Core AIA Components](#)
- [AIA Integration Services](#)

Overview

The process integration pack for Oracle Product Hub (or Product Information Management [PIM]) 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 provides a flexible user-defined attribute (UDA) framework that enables this information to be modeled in several ways so that a single definition of an entity can be established and then later propagated across multiple participating applications.

Oracle Product Hub Process Integration Pack (PIP) supports concept to start process where the metadata, items, and BOMs are defined in Oracle Product Hub and published to the registered target applications. Oracle Product Hub PIP also supports component exclusions and overrides in context of items and BOMs, and PIM Telco library (a set of seeded attributes in Oracle Product Hub to support cross industry application entities). The PIM Telco library can be used 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 have an option to select one or more participating applications when they install Oracle Product Hub except for Oracle BRM, which is a patch installation. In addition, the OPH PIP can be configured to route the messages to only the intended applications. This eliminates the need for publishing the entire payload to applications that are not affected by the data change. The process 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 covered in [Chapter 2: Oracle Product Master Data Management Integration Base Pack](#).

Similarly, the integration flows from an integration option to Oracle Product Hub, as applicable, along with the services and interfaces involved are covered as under:

- [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 Billing and Revenue Management](#)

The application and middleware components supported by Oracle Product Hub PIP are as follows:

- Fusion Middleware
- Foundation Pack
- Oracle Product Hub
- Siebel Customer Relationship Management (CRM)
- Oracle E-Business Suite
- Oracle Billing and Revenue Management (BRM)

For more information about how to install participating applications (integration options) based on your business need, see *Oracle Application Integration Architecture 2.5: Installation and Upgrade Guide* and *Oracle Product Master Data Management Integration Option for Oracle Communications Billing and Revenue Management 2.5: Installation Guide Addendum*.

Terminology

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

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 sub-categories and products that belong to categories.
Transaction Attributes	<p>These attributes are associated to 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 characteristic/properties of a transaction involving a specific instance of a product.</p>
Static Attributes	These attributes are associated to 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

Term	Definition
	product. The value of the attribute in the product is captured at design-time while defining the product. These attributes capture characteristics/ properties or features of a product.
Structure	A structure or a bill of material (BOM) can be defined as a listing of the all the sub-assemblies, intermediates, parts, and raw materials that go into making the parent assembly that displays the quantities of each sub-assembly required to make the assembly. Structure (BOM) is used to compose complex products out of simpler products.
Relationships	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/BOMs are added as relationships to the ICC.
User-defined Attributes	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 that use validation logic to support the needs of your organization. To do this, 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.
Valuesets	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.

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

Some key differences in the terminology used by each application while referring to the same entity include the following:

Entity	OPH	Siebel CRM	Oracle E-Business Suite	Oracle BRM
Inventory location or inventory organization	Inventory organization	Inventory location	Inventory organization	N/A
Product structure	BOM	Customizable Product	BOM	N/A
Business unit or operating unit	Operating Unit	Business Unit	Operating Unit	N/A
Product	Item	Product	Item	Product
Discount	Item	Product	N/A	Discount
Product Class	Item Catalog Category	Product Class	N/A	N/A

Entity	OPH	Siebel CRM	Oracle E-Business Suite	Oracle BRM
Attribute Groups	Attribute Groups	N/A	N/A	N/A

Entity	OPH	Siebel CRM	Oracle E-Business Suite	Oracle BRM
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	N/A	N/A
Valueset of static attributes	Valuesets of static attributes	List of Values	N/A	N/A
Dynamic or Versioned valuesets	Valuesets (dynamic or versioned value sets)	Attribute Definitions	N/A	N/A
Customer Added Static UDA	Customer added static UDA	Flexible Attributes	N/A	Customer added Fields
Structure Elements of ICC	Structure elements of the ICC	Relationships/Structure of the Product Class	N/A	N/A

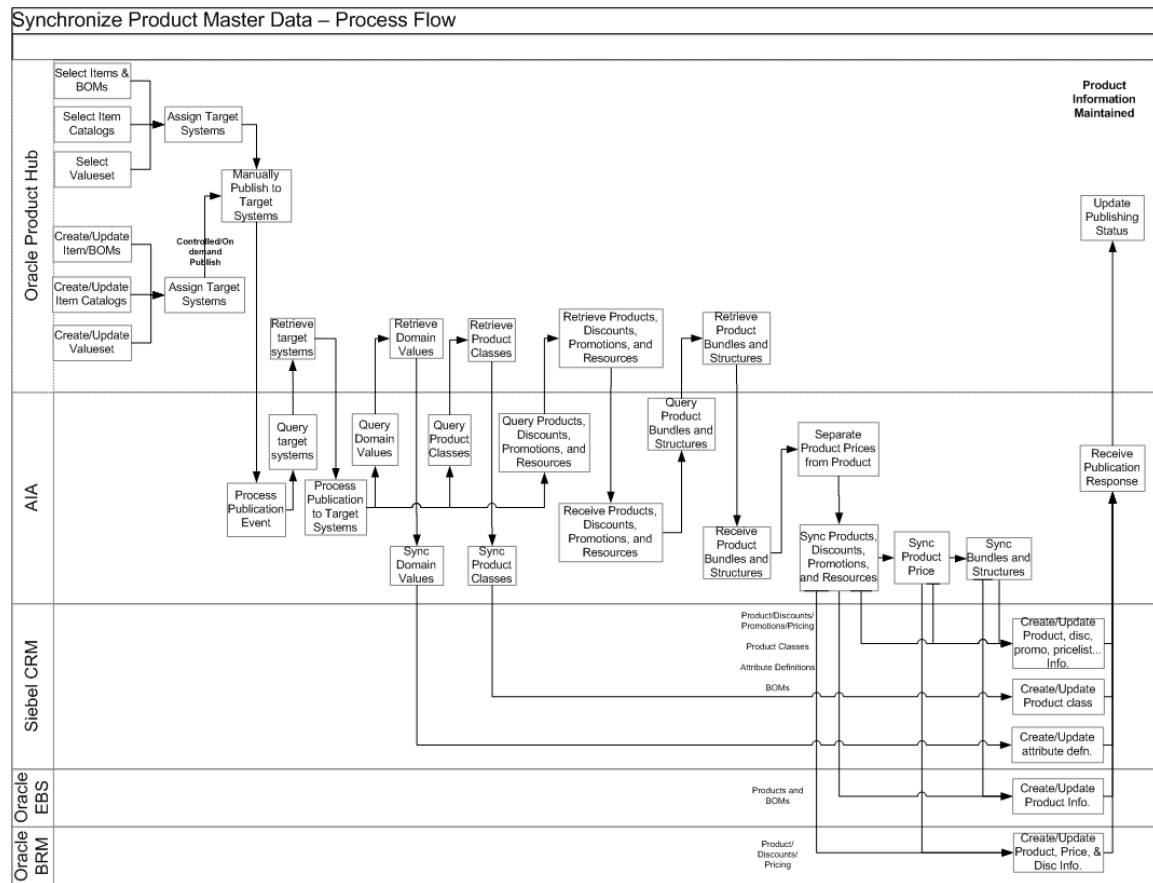
High-Level Process Flows

These are the high-level process flows as part of Oracle Product Hub PIP:

- [Synchronization of items and bill of materials.](#)
- [Synchronization of item catalog categories.](#)
- [Synchronization of valuesets.](#)

For more information about the synchronization flows from Oracle Product Hub to the integration options, see [Chapter 2: Oracle Product Master Data Management Integration Base Pack](#).

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



Business Process Flow

The business-process related tasks, which provide a contextual framework for the solution covers:

- The product administrator defines a common product data model by using item catalog category (ICC) in Oracle Product Hub with a comprehensive definition of the products. After 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 onto obsolescence.
 - The valuesets can be published independently or during the ICC publish from the Oracle Product Hub (PIM) publication framework.
 - The publication framework provides a user interface to publish item catalog categories to one or more downstream participating applications. The downstream applications have to be registered within publication framework.
- The launching of a new product and its maintenance includes the following:
 - Item catalog definition and hierarchy ICC (Siebel product classes).

Attribute definitions and their association with ICC and attribute metadata that may include attributes' 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 like 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 related to the relationship between parent and child items in a hierarchical relationship. It is desirable 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 you create or update 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 process 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 *Oracle Product Information Management* documentation.

Core AIA Components

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

- 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 this parent folder:
[http://\[HOST:PORT\]/AIAComponents/EnterpriseObjectLibrary/Core/EBO/](http://[HOST:PORT]/AIAComponents/EnterpriseObjectLibrary/Core/EBO/).

The core EBS WSDL files can be located by EBO within this parent folder:
[http://\[HOST:PORT\]/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/](http://[HOST:PORT]/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/).

For detailed documentation of individual enterprise business objects, click **EBO Name** link on the **Integration Scenario Summary** page in the Oracle AIA Console. You can also use the **Integration Scenario Summary** page to search for and view integration scenarios that use a particular EBO or EBS.

For more information, see *Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide*, “Using the BSR,” Using the BSR UI to View Integration Scenarios.

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

For more information, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer’s Guide*, “Extensibility for AIA Artifacts.”

AIA 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 about EBSs, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer’s Guide*, “Designing and Developing Enterprise Business Services” and *Oracle Application Integration Architecture – Foundation Pack: Concepts and Technologies Guide*, “Understanding Enterprise Business Services.”

ItemEBSV2

ItemEBSV2 is implemented as an ESB 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 PIM to the various participating applications.

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

ItemResponseEBSV2

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

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

BillOfMaterialsEBS

BillOfMaterialsEBS is implemented as an ESB routing service. It exposes all the enterprise operations that might be performed with a BillOfMaterials enterprise-business-object.

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

This service synchronizes the product structure into Siebel CRM and Oracle E-Business Suite system. It is assumed 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.

BillOfMaterialsResponseEBS

BillOfMaterialsResponseEBS is implemented as an enterprise service bus (ESB) routing service. It exposes all the enterprise operations that might be performed with a BillOfMaterials enterprise-business-object.

In Oracle Product Hub PIP, 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 SyncItemPIMReqABCImpl.

PriceListEBSV2

PriceListEBSV2 is implemented as an ESB routing service. It exposes all the enterprise operations that might be performed with a price list enterprise-business-object.

In Oracle Product Hub PIP, this service implements the SyncPriceListList operation. The SyncItemPIMReqABCImpl invokes 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.

PriceListResponseEBSV2

PriceListResponseEBSV2 is implemented as an ESB routing service. It exposes all the enterprise operations that might be performed with a price list enterprise-business-object.

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

SpecificationValueSetEBS

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

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

SpecificationValueSetResponseEBS

SpecificationValueSetResponseEBS is implemented as a lightweight enterprise service bus (ESB) routing service. It exposes all the enterprise operations that might be performed with a SpecificationValueSet enterprise business object.

In Oracle Product Hub PIP, this service implements the SyncSpecificationValueSetListResponse operation. The SyncSpecificationValueSetListResponse operation on SpecificationValueSetResponseEBS transfers the response message to the SyncSpecificationValueSetListPIMReqABCSImpl or SyncItemCatalogCategoryPIMReqABCSImpl depending on which invokes the message.

ClassificationSchemeEBS

In Oracle Product Hub PIP, this service implements the SyncClassificationSchemeList operation. The SyncItemCatalogCategoryPIMReqABCSImpl service invokes SyncClassificationSchemeList operation on the ClassificationSchemeEBS to synchronize ICCs from PIM. The routing rule in the operation routes to the Siebel provider for values sets SyncClassificationSchemeListSiebelProvABCSImpl.

ClassificationSchemeResponseEBS

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

In Oracle Product Hub PIP, this service implements the SyncClassificationListResponse operation. The SyncClassificationListResponse operation on ClassificationSchemeResponseEBS transfers the response message from to SyncItemCatalogCategoryPIMReqABCSImpl.

For more information about EBSs, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer’s Guide*, “Designing and Developing Enterprise Business Services” and *Oracle Application Integration Architecture – Foundation Pack: Concepts and Technologies Guide*, “Understanding Enterprise Business Services.”

CommunicationsPriceListEBSV2

CommunicationsPriceListEBSV2 is implemented as an ESB routing service. It exposes all the enterprise operations that might be performed with a communications pricelist enterprise-business-object.

In Oracle Product Hub PIP, this service implements the SyncPriceListList operation. The PricelistEBSV2 invokes this operation, and it transfers the request to ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl for billing pricelists that need to be synchronized to Siebel.

CommunicationsPriceListResponseEBSV2

CommunicationsPriceListResponseEBSV2 is implemented as an ESB routing service. It exposes all the enterprise operations that might be performed with a communications pricelist enterprise-business-object.

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

For more information about communications-specific services or integrations, see *Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care*, *Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Order to Bill*, or *Oracle Communications Billing and Revenue Management Integration Pack for Oracle E-Business Suite: Revenue Accounting* implementation guides.

CommunicationsItemCompositionEBSV1

The CommunicationsItemCompositionEBSV1 is the enterprise business service that performs the entire item/product related actions like create product/item, update product/item, and synchronize product/item. Based on the routing rules setup, it invokes a provider ABC service.

In Oracle Product Hub PIP, this service implements the SyncItemCompositionList operation. The ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl invokes this operation, and it transfers the request to SyncItemCompositionListSiebelCommsProvABCImpl for Siebel.

CommunicationsItemCompositionResponseEBSV1

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

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

Chapter 2: Oracle Product Master Data Management Integration Base Pack

This chapter provides an overview of the Oracle Product Hub and covers:

- [Overview](#)
- [Synchronization of Items and Bill of Materials from Oracle Product Hub](#)
- [Synchronization of Metadata \(value set and item catalog category\) from Oracle Product Hub](#)
- [Oracle Product Hub Interfaces](#)
- [Oracle Product Hub Integration services](#)
- [Assumptions and Constraints](#)

Overview

The Oracle Product Hub (OPH) or Oracle Product Information Management (PIM) 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 Information Management* documentation.

As part of Oracle Product Hub process integration pack, the process flows from the Oracle Product Hub to the integration options together with the supported features are covered under these synchronization flows:

- [Synchronization of items and bill of materials](#)
- [Synchronization of metadata.](#)

Prerequisites

These are the prerequisites for synchronization of Items:

- Setup cross-references for organization and inventory locations.
- Seed the domain value maps.
- Change configuration properties as needed.

To set these, see [Chapter 6: Implementing the Process Integration Pack for Oracle Product Hub](#).

- ICC used for the items must to be previously synchronized.
- Valueset used for the Items and associated ICC need to be previously synchronized.

For bill of materials synchronization, you must manually setup cross-references between organizations and inventory locations. To set these, see [Chapter 6: Implementing the Process Integration Pack for Oracle Product Hub](#).

Synchronization of Items and Bill of Materials from Oracle Product Hub

These are the features supported in Oracle Product Hub process integration pack for items and bill of materials synchronization from Oracle Product Hub. These features are covered as part of the synchronization flows arranged within the integration options in this document.

Common features supported for the synchronization of items and BOMs

- Support for identification of set of published entities through batch IDs.
- Support for routing through explicit specification of target systems.
- Filtering items and BOMs based on destination systems.
- Multilanguage support for items and BOMs.
- Reporting the status for items synchronization in PIM publication history

Features supported for the synchronization of items and BOMs from Oracle Product Hub (OPH) to Siebel CRM

- Synchronize items.
- Synchronize BOMs.
- Support for workspace management of items and BOMs.
- Synchronize transaction attribute overrides in context of items.
- Synchronize component exclusion in BOMs.
- Synchronize attribute value overrides in BOMs.
- Support for Product Information Management Telco Library.

Features supported for the synchronization of items and BOMs from OPH to Oracle E-Business Suite

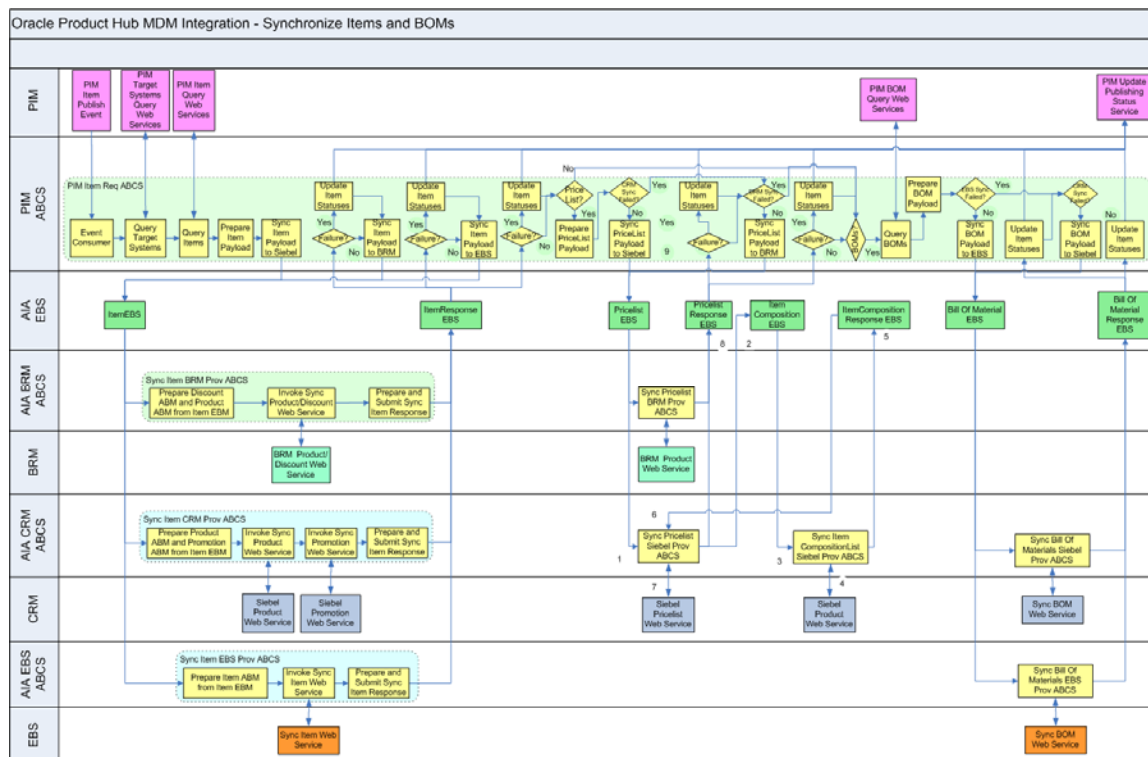
- Synchronize items.
- Synchronize BOMs.

- Synchronize operational attributes.

Features supported for the synchronization of items from OPH (PIM) to Oracle BRM

- Synchronize items (billing products and billing discounts).
- Support for Product Information Management Telco Library.

This graphic shows the process flow from Oracle Product Hub to the participating applications (integration options) for item and BOM synchronization:



Synchronize Items and BOMs from OPH to Siebel CRM and Oracle E-Business Suite

This is the business process flow:

1. Oracle Product Hub (OPH [also known as PIM]) publishes an item publish event to AIA.
2. The event consumer within the PIM ABCS (requestor) receives and processes the event. The event contains the batch ID, which is used to query data from PIM.
3. The PIM requestor ABCS queries the list of target systems from PIM based on the batch ID.
4. The PIM requestor ABCS queries the list of items from PIM 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 destination system.

For each target system, the requestor ABCS invokes ItemEBS which filters the items in the EBM based on the destination system UDA and routes the processing request to the appropriate provider. You do this 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 that contains both items whose entity type is product and items whose 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 you update the product 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 there is a failure, the requestor ABCS updates the publishing status in PIM 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 price list EBS to send pricing information for the items in the batch to Oracle BRM where it is associated to the products by using 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 PIM.
 10. The requestor ABCS invokes 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 to billing events.

12. The item composition EBS invokes the SyncItemCompositionList provider service for creating the event products in Siebel (billing type as Event) when 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 is created for billing events.
14. The price list response EBS is invoked to return a response to the requestor ABCS so that if a failure is to occur, it updates the publishing status in PIM.
15. If BOM was published in the batch, the requestor ABCS queries the BOMs associated to 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 setup 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 PIM for the root items of the BOM published in the batch.

It is assumed that PIM always specifies target systems at the batch level. The GetTargetSystems PIM service returns all the target systems to which the batch should be published. If the 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 there is an error in the item adapter while you are retrieving data from PIM, 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. See: [Setting Configuration Properties](#). Each of these chunks or sub-batches is processed as a batch by itself.

Chunking can be enabled at the item or BOM level, or for both. When chunking is enabled at BOM level, the chunking extracts those many BOMs as the 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. However, the pricelist or BOM is not processed. PIM provides a batch-level update that is called from the SyncItemPIMReqABCSEImpl in case the batch synchronization fails.

Routing Driven Through Explicit Specification of Target Systems by Oracle Product Hub

This feature enables the product administrators to specify the target systems explicitly in the Oracle Product Hub. Some changes that you make in Oracle Product Hub to a product or another entity may apply only to one target system. Explicit specification avoids unnecessary publishing of the payload to those target systems not affected by the changes.

- Enables a more flexible, data-driven, and explicit approach for routing, and by doing this, 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 system is used for routing the payloads to the target applications. AIA first maps the system IDs published by PIM to common system ID and then uses these to route the payload to the target applications.
- Within the published payload, some entities may not apply to some target systems. Each entity in the payload has associated multi-row attribute that is used to identify the subscribing systems for the entity (implemented in PIM as a multi-row 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.

Multilanguage Support 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 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, PIM 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 PIM and therefore, consume only those values that belong to languages supported by the corresponding target applications.
- Some free-text fields on the items are not translatable in the target applications. In this case, although PIM provides all the language codes and the corresponding values, the target application consumes in single default language. This default language supported by the target applications must be configurable.
- For the fields taking values from a valueset whose values are translatable, PIM provides the language independent code and all the values in all the languages. However, AIA only passes the language independent code to the target application.
- For the fields with valueset whose values are not translatable, PIM 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.

Option Class Support in Item Synchronization for Siebel CRM

The item definition in PIM (for product) supports two kinds 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 PIM, 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 an 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 Information Management Implementation Guide*.

Provided here is an illustration of how product and class type relationships are supported in PIM and how the process integration creates them in Siebel.

Root item (Item with a Model type BOM)

Wireless Router (Item)

Bluetooth devices (Item with an Option Class type as BOM and entity type as option group)

Samba Bluetooth Headset

B-Micro Bluetooth Headset

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

This table shows PIM definition of the relationship in the seeded attribute group: version structure associated with the root item. These attributes are component attributes whose values are set in context of the root item.

It is also expected that the Type 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.

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

Relationship Name	Domain Type	Product	Class	Default Cardinality
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.

Provided here is an illustration 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 as follows:

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

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 items that have class relationship with Bluetooth devices must be synchronized to Siebel for the relationship domain to be updated. Updating the Bluetooth devices and synchronizing it 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 component of any root item more than once for any parent-child relationship in PIM.

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

The OPH PIP allows 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 behavior of 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 the Siebel with regard to auto-release.
- As OPH sets the flag at the batch level, Siebel would be unable to use it at the batch level because multiple Siebel services are called for product and discounts, price list items, promotions, and structures. If the flag is set at service invocation level there might be inconsistencies, 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 entity for product and discounts is released in Siebel. Hence, 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 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 to 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 – This default value indicates that the release of the batch should be controlled by the Siebel system parameter.

Note: In this release, when you are using the Siebel system parameter for releasing the workspace, you must ensure that it is consistent with an equivalent system parameter defined in the `AIAConfigurationProperties.xml`.

In this release, the out-of-the-box OPH always publishes batch auto-release flag with a value as *D*.

- Siebel services for creating 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. Else, if the workspace exists and the workspace reuse flag is set to *N*, a new workspace is created whose name is a concatenation of the value in workspace name and a time stamp.

- When the first Siebel service in a batch is invoked, which typically is the service for creating products and discounts:
 - Value that is passed from PIM in the workspace name is used for the Siebel workspace name.
 - Value that is 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:
 - Value that is passed in the batch name is used for the Siebel workspace name.
 - Value that is 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 are invoked and if the batch has succeeded on one or more service, then the creation of product and discounts is invoked with an empty sequence of products and discounts. It has these values for the header attributes:
 - Workspace name is the value that is passed in the batch name.
 - Workspace reuse flag is *Y*.
 - Auto-release flag is *Y*.
- Within Siebel, unpublished workspace is not discarded, as administrators may want to access them for investigation. Periodic cleansing of unpublished workspace is an administrator activity in Siebel.

Note: In this release, the auto-release flag is always passed as *Y* from PIM.

Telco 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 are seeded in OPH. The process integration sets the values of these attributes in the corresponding target applications.

For more information about Telco seeded Library attributes and their corresponding valuesets, see Appendix F: Seeded Item Metadata Libraries, *Oracle Product Information Management Implementation Guide*.

Sellable Product Information Library - Horizontal

- Product Details
 - Definitions: This attribute group is used 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 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 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 simple price on the items. The attribute group is used when 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 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 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 group 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 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 group 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 charge plan associated with the promotion. This attribute group is specifically for the more advanced information about the promotion like adjustments, certain usage plan details, and so on that are associated with the promotion. These are stored as specification group 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 group 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 group 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 group 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 group in the ItemEBO of the canonical layer. These are mapped to the first class attribute of the product entity in Siebel.

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 is not mapped to any field in

the target application, but are used in decision-making or routing or both.

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 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 to the billing products. In billing application, 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 represent the complex rating structure associated with the communications related billing products:
 - **Rate Plan:** The rate plan attributes associated to the billing products are represented in this attribute group.
 - **Tier Group:** This attribute group is used to define multitier rating for the billing products.
 - **Day Time Range:** The attribute group is used to define day and the time when the associated rate data must be applied. The day and time is defined by using calendar dates in this attribute group. This is a [timezone-independent](#) field wherein the time range should not be translated across the systems in different timezones.
 - **Days of the Week Range:** The attribute group is used to define day and time when the associated rate data must be applied. The day and time is defined by specifying the weekdays in a range of calendar dates. This is a [timezone-independent](#) field wherein the time range should not be translated across the systems in different timezones.
 - **Rate Data:** The actual rate or price is grouped under the rate data. Certain discount brackets and proration details that apply to actual charges are defined by 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 to the discount is done by using this attribute group.

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

Communications Product Details Library – Vertical

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

processing.

Component UDA for Item Synchronization

These component UDAs are seeded as part of Telco library, and are synchronized to the target application when a BOM structure is synchronized with the following 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 sub-components of promotions in Siebel.

Version: Structure: The component UDA provides more information on 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 component 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. For information about how to define ICC structure in Siebel, see [Synchronization of Item Catalog Categories](#).

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 Telco 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 applications can perform a *delete all* and *insert all* operations. The interfaces on the target applications must be carefully designed and implemented based on the usage of these 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 group.

Synchronization of 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. This makes it easier to facilitate transactional processes.

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

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

Publication of an item associated to a BOM synchronizes all components for that item and BOM. The message routing rules use domain value mapping setup that lets you 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 when it is created.

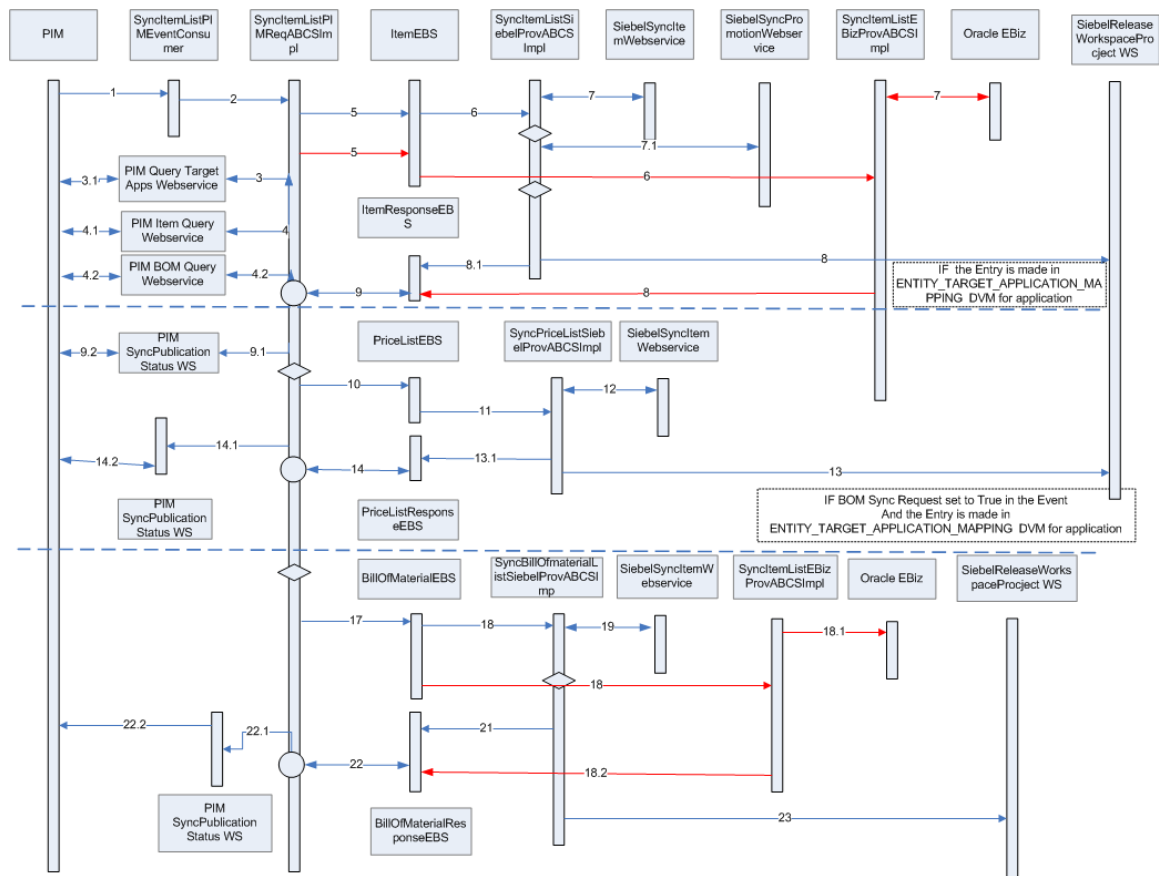
BOM synchronization involves two discrete services, namely, synchronization of component items and synchronization of structures to maximize performance and allow appropriate routing. Oracle E-Business Suite supports creating only future-dated BOM components.

Oracle Product Hub supports seeded operational attributes that control the behavior of the item in various functional Oracle E-Business Suite modules like 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 Information Management, Oracle Inventory User's Guide*.

See: [Synchronization of Items and Bill of Materials](#) for the graphic on item and BOM process flow from Oracle Product Hub to Oracle E-Business Suite.

Integration sequence for Item and BOM is illustrated in this diagram.



Item and BOM Integration Sequence

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

This table describes the steps depicted in the sequence diagram.

Name	Step Description
PIM item event is raised	PIM item events are raised when administrator publishes the items from PIM system. This includes create/update to items or primary BOM associated to items. The event contains a Batch_ID to represent a set of items that are published as a single payload.
SyncItemListItemEventConsumer	SyncItemListItemEventConsumer listens to business event and receives the WF_EVENT_T_msg event payload for item event. SyncItemListItemEventConsumer routes to SyncItemListItemReqABCSImpl with complete event payload.
SyncItemListItemReqABCSImpl	The PIM requester ABC implementation, SyncItemListItemReqABCSImpl, receives the event message from PIM and queries the target systems. It then transforms the event message to the PIM ItemQueryABM and calls the QueryItemListItemAdapter to retrieve the ItemListABM from PIM. The QueryItemListItemAdapter response is then converted into SyncItemListEBM and SyncPriceListEBM and routed to the corresponding EBS service. When the items/price list is synchronized in the participating applications, the status of the

Name	Step Description
	<p>synchronization is sent back to SyncItemPIMReqABCSImpl, which updates back PIM with the status. If the item synchronization is successful, only then the BOM is synchronized for that target.</p> <p>The SyncItemPIMReqABCSImpl invokes QueryBillOfMaterialsListPIMAdapter to retrieve the BillOfMaterialsListABM, transforms the same into the SyncBillOfMaterialsListEBM, and routes it to the corresponding EBS. When the BOM is synchronized in the participating applications, the status of the synchronization is sent back to SyncItemPIMReqABCSImpl, which updates back PIM with the status.</p>
QueryItemPIMAdapter	<p>This adapter service is implemented in BPEL. It is invoked from the SyncItemPIMReqABCSImpl. It takes the batch ID as input and responds back 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 SyncItemPIMReqABCSImpl. It takes the Batch_ID as input and responds back with the BillOfMaterialsListABM payload in that batch.</p>
ItemEBSV2	<p>Invoking ItemEBSV2 with SyncItemPIM operation routes the SyncItemPIMReqABCSImpl to SyncItemPIMReqABCSImpl, SyncItemPIMReqABCSImpl, and SyncItemPIMReqABCSImpl based on target ID.</p>
SyncItemPIMReqABCSImpl	<p>SyncItemPIMReqABCSImpl transforms the SyncItemPIMReqABCSImpl to the appropriate Oracle E-Business Suite item ABM and invokes the appropriate API. The response from the Oracle E-Business Suite API is then transformed into the SyncItemPIMResponseEBM and sent back to ItemResponseEBSV2. In this process, the cross-reference is linked with the Oracle E-Business Suite IDs of the product in the form of InventoryItemID::OrganizationID::OperatingUnitID.</p>
SyncProductSiebelProvABCSImpl	<p>SyncProductSiebelProvABCSImpl transforms SyncItemPIMReqABCSImpl to Siebel product message (without Promotion) and calls Siebel product web-service SWIProductImport to Synchronize the product with all the items in the list and does these in sequence:</p> <ul style="list-style-type: none"> Constructs Siebel promotion message from SyncItemPIMReqABCSImpl. Transforms Siebel promotion message into Siebel promotion ABM. Calls Siebel promotion web service SWIPromotionImport to synchronize the promotions. Transforms the Siebel item and promotion response message into SyncItemPIMResponseEBM and sends it back to ItemResponseEBSV2. <p>In this process, the cross-reference is linked with the Siebel product IDs into ITEM_ITEMID cross-reference. In addition, the cross-reference is linked with the Siebel promotion IDs into PROMOTION_PROMOTIONID cross-reference.</p>
ItemResponseEBSV2	<p>ItemResponseEBSV2 transmits SyncItemPIMResponseEBM message back from the SyncItemPIMReqABCSImpl, SyncItemPIMReqABCSImpl, and</p>

Name	Step Description
	SyncItemListBRMProvABCSImpl to the SyncItemListPIMReqABCS, which then invokes the PIM status update service to update the item publication status in PIM.
PriceListEBSV2	SyncItemListPIMReqABCS converts the PIM item query response ABM into the SyncPriceListEBM based on the specification name Prod_Details_Definitions with UDA Pricing_Code value other than <i>NONE</i> . It then passes the EBM message to the PriceListEBS, which routes the message to ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl.
ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl	ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl calls the Siebel SWIProductImport to synchronize the pricelist in Siebel and then sends the response back to SyncItemPIMReqABCSImpl by using the PriceListResponseEBS.
PriceListResponseEBSV2	PriceListResponseEBS routes the PriceListResponseEBM from ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl to the SyncItemListPIMReqABCS, which then updates the status to PIM.
BillOfMaterialsEBS	SyncItemListPIMReqABCS invokes BillOfMaterialsEBS with 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 routing rule, BillOfMaterialsEBS invokes SyncBillOfMaterialsListEbizProvABCSImpl and SyncBillOfMaterialsListSiebelProvABCSImpl.
SyncBillOfMaterialListSiebelProvABCSImpl	SyncBillOfMaterialListSiebelProvABCSImpl receives the SyncBillOfMaterialEBM from 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 by using the BillOfMaterialListResponseEBS.
BillOfMaterialListResponseEBS	The BillOfMaterialListResponseEBS routes the synchronize BillOfMaterialListResponseEBM from SyncBillOfMaterialListSiebelProvABCSImpl and SyncBillOfMaterialListEbizProvABCSImpl to the SyncItemListPIMReqABCS, which calls the PIM update service to update the status of BOM published in PIM.
SyncBillOfMaterialListEbizProvABCSImpl	It 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 SyncItemListPIMReqABCSImpl.

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

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 and other processes. It allows organizing simple products into structures and selecting the desired structure to publish together 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 common set of products is accurately represented and available across all systems to facilitate transactional processes like 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: [Routing Driven Through Explicit Specification of Target Systems by Oracle Product Hub](#)
- At the batch level, a list of target system is captured to identify the systems that should receive the list of items payload. This list of target system 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 both the data for the root items and 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 a NULL or empty values are nullified. The fields that are not passed in are left as is.
- Within the published payload, some items might not be applicable to some target systems, for example, promotions to Oracle BRM. For each item, the list of items payload also contains multivalue 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 as follows:
 - 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, it 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 where 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.
- Payload for list of items and list of BOMs is a complete payload that contains all attributes and entities even when the publishing takes place to update existing entities.
- The list of BOMs payload contains the structures associated to 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. There is no multi-row, UDA destination systems associated directly with BOMs. All BOMs are sent to all providers that consume BOMs. DVM setting defines if a provider consumes or does not consume BOMs. For a given target system, the synchronization of BOMs does not happen if the synchronization of items fail.
- OPH published messages could be generated from any number of organizations that are established. The integration solution is based on manual pre-synchronization of organizations between OPH, Siebel, and EBS and to create appropriate cross-references. The organization used for publication in OPH is submitted and mapped to the corresponding organization in the target application if any.
- The list of items is published within a single XML message with primary attributes (standard to OPH), operational attribute values, and user-defined attribute. OPH published message must include UDA values in the EBM regardless of whether the downstream application can accommodate consumption of these flexible attributes.

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 option class BOM associated to the root item. The immediate components of the promotion can have BOM of type *option class*. The items or components that are associated to 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) UDA 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

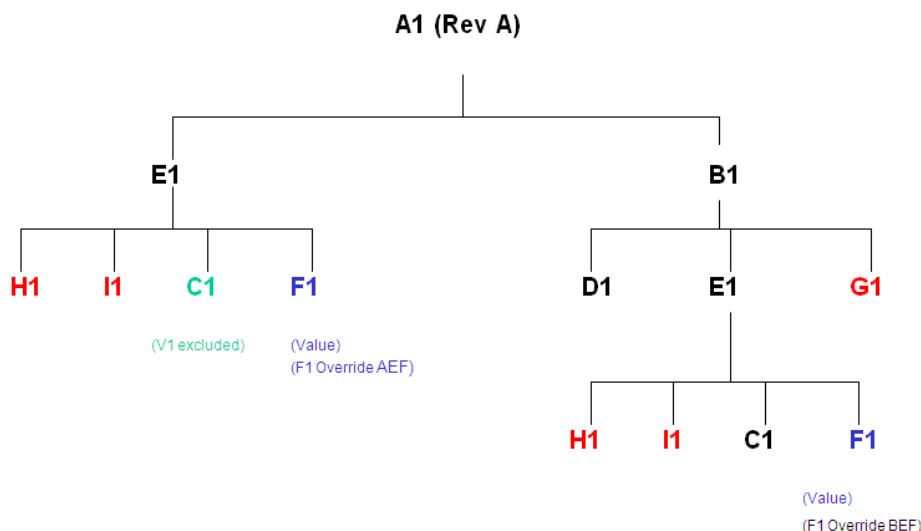
more information.

- 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.
- When 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)
- When 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 partly successful, the status on the promotion item is *failed*.
- Adding/Deleting/Updating the components that are directly associated with the promotions or updating the fields of promotions like promotion name, description, and so on, or updating the promotion related information such as seeded 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, it is expected that the AIA configuration property synchronization flag be set to Y.

Component Exclusions in context of Promotions

When you define a promotion in OPH, some components of the promotion or sub-components under the root item can be excluded.

In this example, 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.



An Example of Component Exclusions

OPH publishes the entire promotion definition with all the components and wherever there are exclusions, 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 under:

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. This 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 two ways:

- Set the minimum cardinality and maximum cardinality of the component UDA to be 0 (zero).
- 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

This example explains the 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 Adapter
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 removing 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 must be set as 0 (zero) in PIM 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 must be set as 0 (zero) in PIM and synchronized to CRM.

Case 3: Exclude component of option class items

The components under an option class item must be excluded in PIM by using the exclusion method that is provided in the product workbench. Default cardinality must be specified in PIM 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 need to be excluded, exclude these from the PIM 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 PIM 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 target application is set to 0 (zero). Therefore, making all the components available as optional items as part of 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 to 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: [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 graphic that shows 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 using these component UDAs:

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

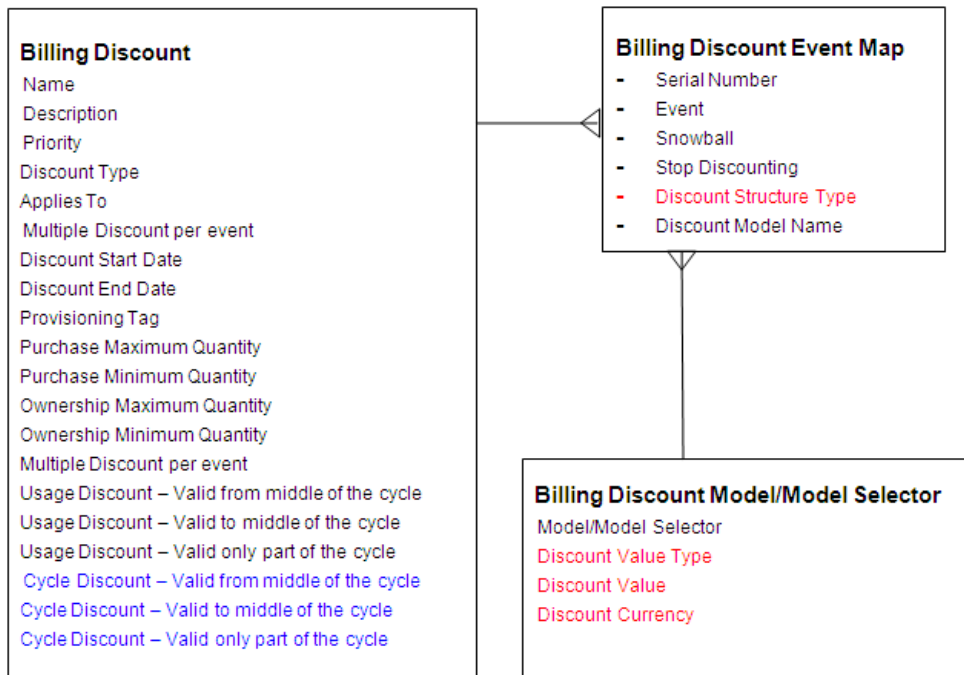
For more information about how to define promotion based discounts, see Siebel CRM documentation

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 discounts 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 that are 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.

This graphic is a conceptual simplification of the Oracle BRM model, which describes a discount and discount model entity and the way they are associated to one another.



Conceptual Model for Discount Entities

The billing-discount-event-map is a conceptual entity that associates the discount to 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 is 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.

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 runtime behavior on when to stop giving the discount if the discount is inactive or canceled. Seeded values are as follows:
 - 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 as follows:
 - 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/model selector associated to 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/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 discount model name is captured in OPH, which has minimal effect 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, there is no cross-validation implemented by OPH between this field and the name of the discount-model entity.

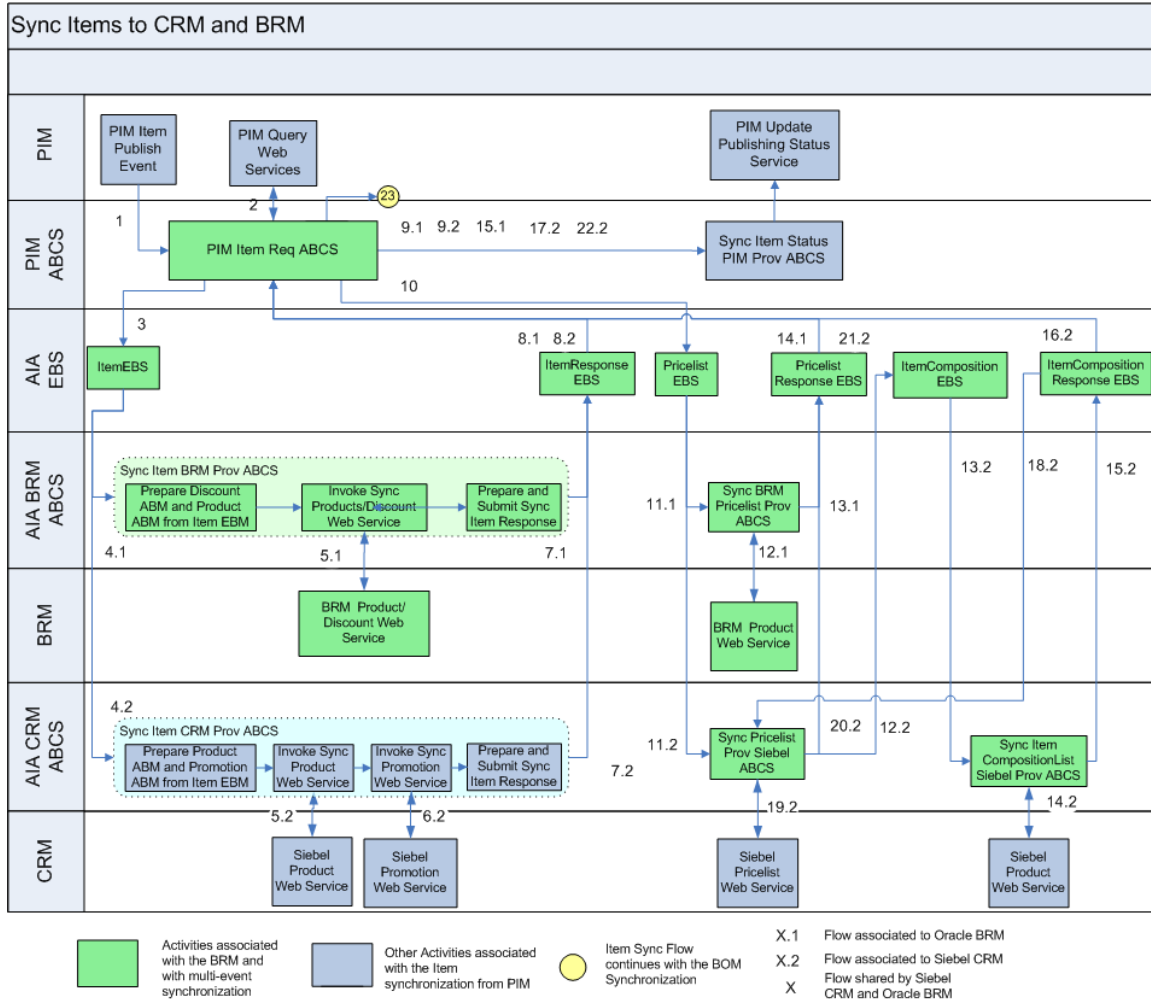
The name for the discount model should be descriptive as the product administrator in OPH can only rely on the name to assign the discount model to a discount. In OPH, the description field in the item can also be used to add more information for the discount model. However, this information limited to OPH and is not visible to BRM administrator as there is no description field for the discount model in the BRM services.

Two alternative approaches are supported:

- You can define a discount model in BRM, and then create in OPH the corresponding item with the *discount model* as value for entity type, making sure that the names match.
- You could first create the discount model in OPH and then enrich in BRM through the synchronize discount flow. Discount models are manually configured in PIM 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 is fully defined in BRM and therefore there is no risk to release a discount with a shell discount model.

This graphic shows the integration flows for items from OPH (PIM) to Siebel CRM and Oracle BRM:



Synchronization of Items to CRM and BRM

The description about the item synchronization to Siebel CRM and Oracle BRM is described here:

Discounts are represented in OPH as items with entity type as discount. The process integration must create product to represent a discount in Siebel and discount entities in BRM for every such item. Discount attributes are captured as UDA in OPH. The discounts are published by OPH together with the other items for the same batch. The item payload typically contains discounts and also products and promotions. Items of type discount model are not published by OPH.

OPH also represents discount models as items with entity type as discount model. They have a unique name and a description. Discount model is manually configured in OPH and is not explicitly synchronized to BRM. The only specific attribute captured in the discount model is the name.

The Oracle E-Business Suite builds one payload for each of the target systems specified by the OPH product administrator. The payload is created by filtering out those items that should not be sent to the target system. The filtering is performed based on the value of the multi-row UDA destination systems. For each item, the requestor ABCS looks for system ID matching the target system ID (for example, CRM01, BRM01, BRM02, and so on), and if found, the value of the associated flag determines whether the item should be processed by that system. Items with flag set to Y are processed and those with the flag set to N are not processed. If not found, it is defaulted to N. You can change the default behavior through extensibility.

- Enterprise business service filters and routes the item payload that contains discounts and products to the target application systems. This includes, Siebel CRM and Oracle BRM. This routing is performed based on the target system list published by OPH.
- Multiple billing system and multiple instances of BRM systems is supported.
- Siebel CRM ABCS processes items of type product or discount together as both are mapped to products in Siebel.
- Oracle BRM ABCS receives the item payload that contains products, discounts, and promotions. Only simple products and discount is synced to Oracle BRM.
- BRM provides one service for the discount synchronization. The services currently implemented by BRM do not preserve local enrichment. However, as the discount header is fully captured in OPH and the discount model entity is not overwritten by the API as we only pass the discount model name, as preserving local enrichment is not necessary for discount synchronization.

Oracle BRM APIs support local enrichment of discount models and implement the local enrichment at the sub-entity level. This means that the BRM APIs do not wipe out the content of existing sub-entities when the parent entities are re-synchronized. In other words, fields that are defined in OPH should not be enriched locally in BRM, as the enrichment is wiped out when an update is published by OPH. There are no rules to restrict this behavior in BRM and it must be enforced by the product administrator as a modeling methodology. We recommend that you preserve local enrichment at the sub-entity level.

OPH always publishes an entire payload including all the fields that are defined in OPH regardless of whether they are modified.

- All discount specific attributes are implemented as UDA by using name-value pairs.

Synchronization of Billing Products from Oracle Product Hub to Oracle Billing and Revenue Management

The following table describes 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 the industry requirements.

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 as item.

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

Note: BRM supports multi-tier rating to be associated with rate plans. The PIM pricing model in the seeded Telco library can be used to define multi-tier rating for an item. The process integration supports multi-tier rating between PIM to 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 PIM is used to define the billing product name and billing product description in BRM, respectively.

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

- Additional Billing Information
- Billing Products Attributes General
- Billing Products Event Map
- Rate Plan
- Rate Data
- Balance Impact
- Tier Group

- Day Time Range
- Days of the Week Range

Multi-Event Product Synchronization from PIM to Siebel.

The products that have multiple charges/events are created as customizable products in Siebel. The events are created as separate products and are added as child components of the main product. The component products represent the events/charges that have the billing type as 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 PIM requestor item ABCS must create the corresponding item and pricelist line cross-references.

For more information about how to create cross-references for multi-event products, see “Appendix A, Oracle Application Integration Architecture for Communications: Maintenance Update Guide”. The document is available on My Oracle Support: <https://support.oracle.com/>.

Methodology to Define Billing Products in PIM

When you define billing products and discounts in PIM, use these guidelines to take advantage of flexibility offered by the process integration:

- The billing products with one usage event are synchronized to Oracle BRM and Siebel CRM.
- The billing product where the usage event is one of the charges in a multi-event case, that usage event is not synchronized to Siebel. As usage events are generally not synchronized to Siebel, the name and/or description of multi-event products should include some user-readable identity of the usage. This enables the product/price administrator to distinguish the synchronized products on the Siebel side.
- The discount value of discount objects are not applied in Siebel. The name and/or description of the discount objects should include the general intent of the discount that can be conveyed on the Siebel order.

Billing Products with Single Rate Plan

- Define the base-product-attribute values and define the rate plan. Ensure that the rate plan name is same as the event name. Else, the BRM service errors out. This indicates that a single rate plan is associated with the charge of the billing product.
- For single rate plans, multi-tier rating can be defined in PIM by using the seeded attribute framework. The process integration synchronizes the multi-tier rating from PIM 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 PIM 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

- For BRM, multiple rate plans can be associated by using rate plan selectors or by using 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/charges.
- The charges for the billing products are defined by using the UDA pricing model supported by PIM. The charges must be created as events in BRM and the corresponding prices must be associated to 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 product with usage ratings 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 PIM, 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 out of the box. You can add more usage type events/charges and define the corresponding the 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

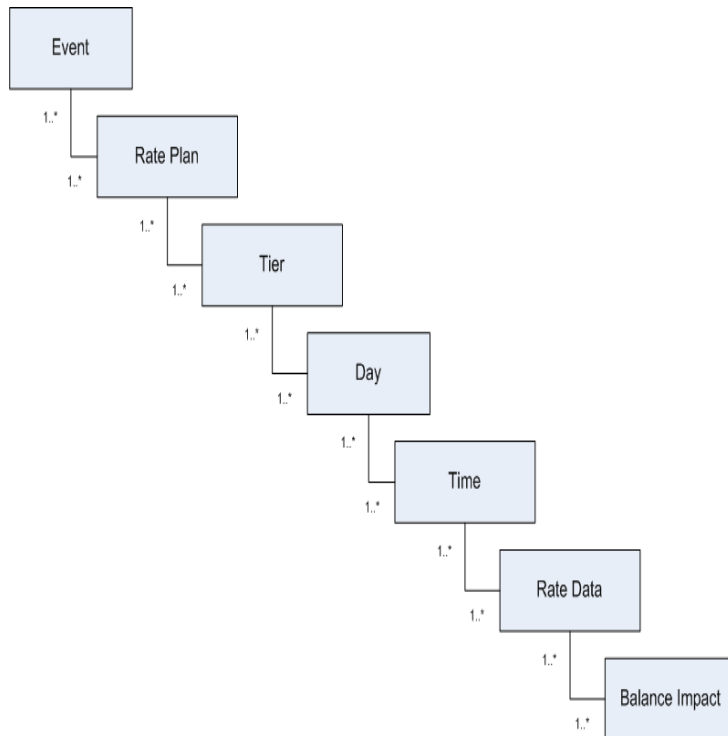
This table summarizes the support offered for pricing in PIM.

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

	Recurring	Single Rate Plan	Complete definition in PIM. The rate plan can also be enriched in BRM. The rate plan name must not be changed in BRM since subsequent updates override the changes.
		Rate Plan Selector	PIM sends the rate-plan-selector name and the rate plans are defined and enriched in BRM.
		Custom event analysis	Complete definition in PIM. The rate plans can also be enriched in BRM. The rate plan names must not be changed in BRM as subsequent updates override the changes.
	Usage	Single Rate Plan	Complete definition in PIM. The rate plan can also be enriched in BRM. The rate plan name must not be changed in BRM as subsequent updates override the changes.
		Rate Plan Selector	PIM sends the rate-plan-selector name and the rate plans are defined and enriched in BRM.
		Custom event analysis	Complete definition in PIM. The rate plans can also be enriched in BRM. The rate plan names must not be changed in BRM as subsequent updates override the changes.
Pipeline Rating	Usage	Pipeline rate plans	PIM sends only the rate plan name. Enrichment of pipeline usage rating is performed in BRM. The enrichment done in BRM is not removed when the updates are synchronized.

Pricing Hierarchy of BRM

This pricing hierarchy graphic shows the real-time rating in Oracle BRM.



The pricing hierarchy can be modeled in PIM by using the seeded Telco 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/charge with real-time single rate plan) in PIM. These have to be synchronized to all the billing instances since these are applied during MACD (move, add, change, delete) orders.

For more information about how to associate one-time charges and penalty charges to the corresponding products in Siebel, see 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 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 non-service bundles, the billing type must not be set.

For more information about definition and methodology for different types of products, see *Oracle Application Integration Architecture for Communications: Maintenance Update Guide*.

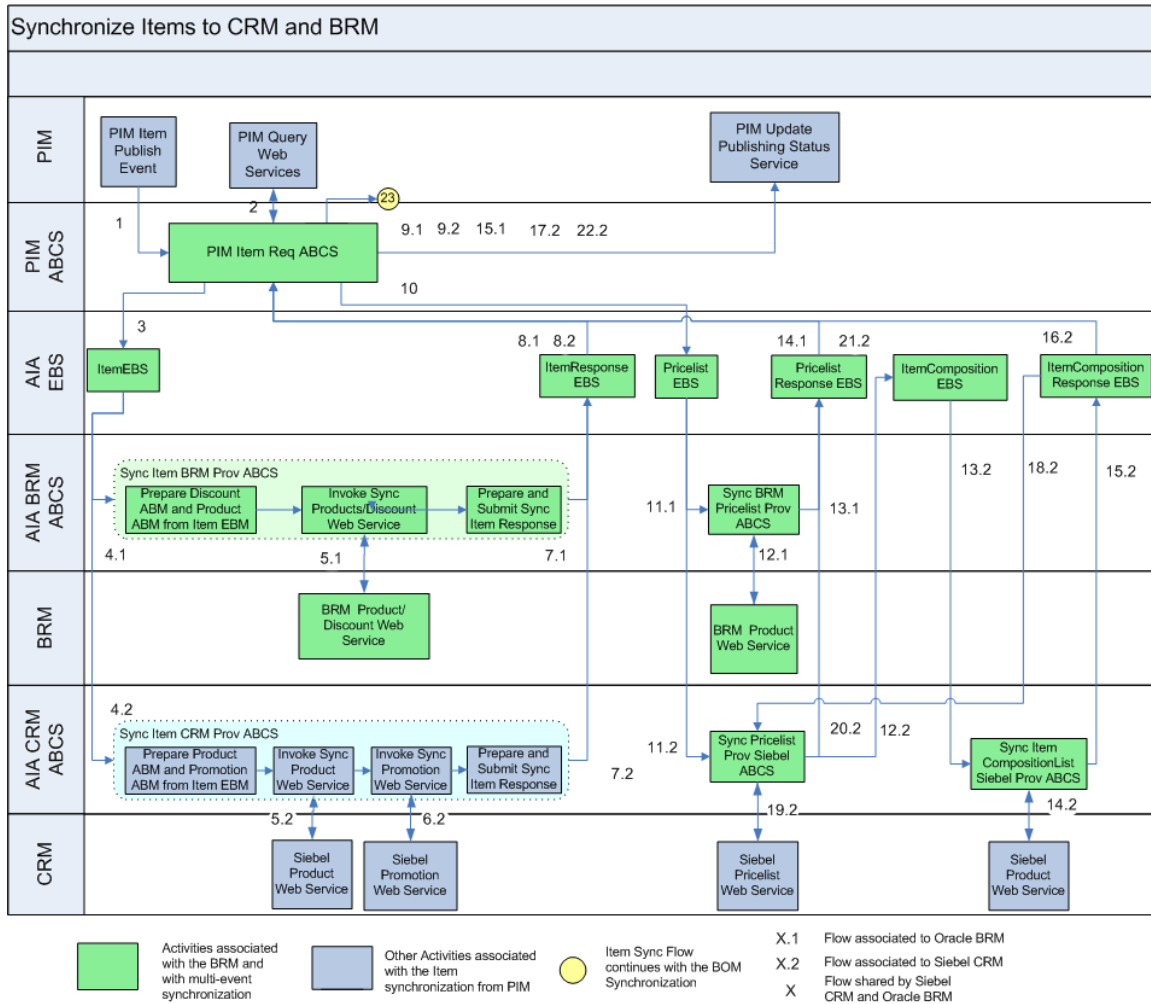
Support for Multiple Rate Tiers and Rate Tier Effectivity.

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 on rate tiers and effectivity, see [Appendix-A](#).

The item representation in OPH (PIM) can be used to model billing products. This graphic illustrates the synchronization of items from OPH to Siebel CRM and Oracle BRM.



Synchronization of Items to Siebel CRM and Oracle BRM

This is the process flow:

1. The PIM item requester ABCS transforms the item definition from PIM and constructs an ItemEBM with a list of items that were published from PIM publication framework. It invokes the ItemEBS to route the ItemEBM to the corresponding provider services.

2. The ItemEBS routes the incoming ItemEBM message to the BRM provider connector service and to the Siebel provider connector service.
3. The BRM service provider transforms the incoming ItemEBM and invokes the OPCODEs to create or update the corresponding product or discount. The entity type identifies the type of item. The item definition has other types of items like promotion, service bundles, non-service bundles, and so on. These have to be ignored in the BRM provider service. Only those items with billing type as subscription, item, and discount are considered for the BRM.
4. The BRM web service returns the status to the connector service that generates the response and invokes the ItemResponseEBS.
5. The ItemResponseEBS returns the response back to the caller, which invokes the update status services. These status services update the publication status of the item in the publication history framework provided by PIM.
6. The PIM item requester service extracts all the pricelist related information and transforms into PriceListEBM. It invokes the PriceListEBS.
7. The PriceListEBS routes the incoming ItemEBM message to the BRM pricelist provider ABCS. The BRM pricelist transforms the incoming ItemEBM and invokes the OPCODEs to update the product with the pricelist information. This adds the events (charges) to the product and associates single rate plans.
8. The BRM web service returns the status to the connector service that generates the response and invokes the PriceListResponseEBS.
9. The PriceListResponseEBS returns the response back to the caller, which invokes the update status services. These status services update the publication status of the item in the publication history framework provided by PIM.
10. The PriceListEBS also routes the incoming ItemEBM message to the Siebel price list provider ABCS. If there are multiple charges associated, the connector service constructs an ItemCompositionEBM and invokes an ItemCompositionEBS, which routes the ItemCompositionEBM to synchronize-item-composition Siebel provider service. The provider service invokes Siebel services that create the corresponding products in Siebel (billing type as event). The services return the status back to the provider service that generates the response and invokes ItemCompositionResponseEBS.
11. The ItemCompositionResponseEBS returns the response back to the caller, Siebel Pricelist provider ABCS. In case there are errors while you create event products, the connector service constructs a PriceListResponseEBM and invokes the PriceListResponseEBS. If there were no errors, the connector service calls the Siebel pricelist web services to update the price for all the products that were created. The Siebel web service returns the status to the caller that constructs the PriceListResponseEBM and invokes the PriceListResponseEBS.
12. The PriceListResponseEBS returns the response back to the caller, which invokes the update status services. These status services update the publication status of the item in the publication history framework provided by PIM. See: [Update Publishing Status in PIM upon Publication of Items](#).

These seeded attribute groups are used to define billing products in PIM.

For more information about Telco seeded Library attributes and their corresponding valuesets, see Appendix F: Seeded Item Metadata Libraries, *Oracle Product Information Management Implementation Guide*.

Attribute Group	Attribute	Valueset	Multi-Row	Functional Description and Handling
Additional Entity details			N	
	Internal Reference Code			Unique product code that identifies the billing product. This is an alphanumeric field.
	Effective Start Date			Effective start date of the billing product or discount
	Effective End Date			Effective end date of the billing product or discount
Additional Billing Information				
	Billing entity type	[Subscription, Service Bundle, Event, Discount, Special Rating, System, Item]		Identifies the type of billing entity. If the entity is not discount, will specify the type of billing product.
	Billing Service type			The service type associated with the product. It represents the service to which the product belongs. Oracle BRM maintains the service types and the values from Oracle BRM must be used to create a static non-versionable valueset in PIM. A DVM needs to be created.
Billing Attributes General			N	
	Priority			This priority sets the order in which the billing application analyzes products during rating. The bigger the number, the higher the priority; zero is the lowest priority you can set.
	Multi-RUM Support			(Optional) Flag to indicate support of ratable usage metrics.
	Provisioning Tag			(Optional) They provide a framework to associate extended rating attributes that can enable rating or discounting to vary for a service. This is an entity in BRM. The product admin in BRM creates the provisioning tag in BRM then the provisioning tag names are used to create a static non-versionable valueset in PIM. The

Attribute Group	Attribute	Valueset	Multi-Row	Functional Description and Handling
				process integration will use the name matching technique to set the value of provisioning tag associated with the billing product.
	Supplier Tax ID			(Optional) This is the company or corporate division that sells the product and is responsible for collecting taxes. This is a user-defined LOV in BRM. When product admin in BRM creates the supplier tax ID in BRM then the user-defined LOV is used to create a static non-versionable valueset in PIM. This field is not mapped to the target BRM application, but is available at the AIA layer.
	Allow partial purchase			(Optional) Flag to indicate whether customers are allowed to purchase a part of the product. If a product gives customers 30 hours of Internet connection for \$10, you can allow customers to purchase half the amount for half the cost. A DVM needs to be created.
	Purchase Max Quantity (Purchase Quantity)			(Optional) Represents the maximum quantity of the product that can be purchased per transaction. Use the quantities specified in this AG to define the quantities for billing products.
	Purchase Min Quantity (Purchase Quantity)			(Optional) Represents the minimum quantity of the product that can be purchased per transaction
	Ownership Max Qty(Ownership Qty)			(Optional) Represents the maximum quantity of the product that can be owned by the customer at any given point of time
	Ownership Min Qty(Ownership Qty)			(Optional) Represents the minimum quantity of the product that can be owned by the customer at any given point of time
Event Map			Y	New Attribute group to be seeded
	Event Name			Represents a name of the billable event, or charges modeled as event in the billing application. A DVM needs to be created.
	Measured by	[Occurrence]		Represents ratable usage metric (RUM). RUM defines the quantity being measured and the units used. A DVM needs to be created.

Attribute Group	Attribute	Valueset	Multi-Row	Functional Description and Handling
	Rate Plan Structure	[Single Rate Plan, Rate Plan Selector, Custom Event Analysis]		This will identify the type of rate plan associated with the product. Only 'Single Rate Plan' is supported by the Integration. A DVM needs to be created..
	Rate Plan ID			The identifier of the rate plan associated with the billable event
	Pricing Commit Type	[Committed, Dynamic]		Field that identifies whether a discount-override or a price-override is defined for the charge. This is for CRM. Ignored in BRM (CRM specific). Use a DVM to map these values.
	Dynamic Discount Method	[Percent, Amount]		Whenever a discount override is applied to the charge, the field identifies type of discount. This is for CRM. Ignored in BRM (CRM Specific) Use a DVM to map these values.
Day Time Range			Y	(Defining the attributes in this attribute group is Optional) New Attribute group to be seeded. This is timezone independent .
	Day Time Range ID			Day time restrictions unique identifier
	Day Range Name			Day time restrictions name
	Day Range Start Date			Start date of the day range
	Day Range End Date			End date of the day range
	Use Time of Day Ranges			Flag to indicate whether time range is defined for the day
	Time of Day Range Name			Time range restrictions name
	Start Time			start time of the time range
	End Time			end time of the time range
Tier Group			Y	(Defining the attributes in this attribute group is Optional) New Attribute group to be seeded
	Tier Name			Name of the Rate Tier
	Tier ID			unique identifier of the rate tier
	Effectivity mode	[absolute, relative]		This attribute is specific to billing applications. This is a decision-making attribute for the time dimension.

Attribute Group	Attribute	Valueset	Multi-Row	Functional Description and Handling
				<p>If the value is <i>Absolute</i> the absolute start and end time must be specified.</p> <p>If the value is <i>Relative</i> then the time-period relative to purchase period date must be specified. The relative period is specified by using a set of UOM values like days, months, year, and so on. See: Appendix A for support for rate tiers and effectivity.</p> <p>A DVM needs to be created.</p>
	Start date time			Start date and time of the tier
	End date time			End date and time of the tier
	Relative Start			Relative start from to product purchase date
	Relative Start UOM	[Second, Minute, Hour, Day]		<p>UOM for the relative start date.</p> <p>A DVM needs to be created.</p>
	Relative End			Relative end when the rate tier is no longer active
	Relative End UOM	[Second, Minute, Hour, Day]		<p>UOM for the relative end date.</p> <p>Reuse the DVM created for Relative Start UOM</p>
	Time Day Restrictions	[No Restrictions, Day Restrictions, Time Restrictions]		<p>Indicates whether Day Time restrictions are associated with the Rate Tier. 1. No Restrictions: The rate tier has no day or time restrictions. If you have a day or time of day range in your rate structure, selecting this removes it. 2. Day Restrictions: Set day of week and month of day restrictions for a rate. 3. Time Restrictions: Set time of day restrictions for a rate: Use name matching to set this value. A DVM needs to be created.</p>
Rate Data			Y	(Defining the attributes in this attribute group is Optional) New Attribute group to be seeded
	Rate Data ID			Rate data unique identifier
	Rate Data Name			Rate data name
	Override Credit Limit	Flag		Flag to identify whether the credit limit is overridden. A credit limit is the maximum amount of a resource, such as currency or hours, that can accumulate in an account before the customer is prevented from using the service
	Quantity	[Continuous,		You can group charges together into quantity

Attribute Group	Attribute	Valueset	Multi-Row	Functional Description and Handling
	discount bracket basis	Rate Dependent, Resource Balance		discount brackets to charge different amounts based on quantity. For example, for a telephony service, you could charge 10 cents a minute for the first 120 minutes and 5 cents per minute after 120 minutes. Specifies how the billing application determines the quantities it uses to select the quantity discount bracket. Use name matching to set this value. No DVM is necessary. A DVM needs to be created.
	Quantity discount bracket basis resources	User-defined Resources (USD, Minutes, Loyalty Points)		If the discount bracket basis is 'resource balance' it identifies the resource. A DVM needs to be created for this field to set the value.
	Purchase proration information	[Do not Charge for the cycle, Calculate Charge based on the amount used, Charge for the entire cycle]		Specifies how to charge customers when they purchase the product in the middle of the accounting cycle. A DVM needs to be created..
	Cancel proration information	[Do not Charge for the cycle, Calculate Charge based on the amount used, Charge for the entire cycle]		Specify how to charge customers when they cancel the product in the middle of the accounting cycle. Reuse the DVM created for purchase proration
Rate Plan			Y	New Attribute group to be seeded
	Currency			Reuse the DVM for Currency
	Rate Plan ID			Rate plan identifier
	Rate Plan Name			(Optional) User-defined name for the rate plan
	Description			(Optional) Rate plan description

Attribute Group	Attribute	Valueset	Multi-Row	Functional Description and Handling
	Tax When	[No Taxes, Event Time, End of Cycle]		(Optional) Identifies when the tax must be applied. A DVM needs to be created.
	Tax Code			(Optional) This is user-defined. Identifies the tax code that is applied.
	Bill in Advance	Flag		(Optional) Flag to indicate whether advance billing is allowed
	Charge cycle fees in advance of billing cycle			(Optional) Field to indicate when the monthly charge is billed in advance of the billing cycle
	UOM for Charge cycle fees in advance	[Days, Month]		(Optional) Unit of measure for the 'Charge cycle fees in advance of billing cycle'. Use/Reuse a DVM to map these values
	Apply cycle fee on Billing Date	Flag		(Optional) Flag to indicate that the monthly charge must be applied on the billing date
Balance Impact			Y	New Attribute group to be seeded
	Sequence			(Optional) Sequence for the rate data
	Rate Plan ID			(Optional) Rate plan unique identifier
	Rate Data ID			(Optional) Rate data identifier
	Tier ID			(Optional) Unique identifier of the rate tier
	Day Time Range ID			(Optional) Day time restrictions unique identifier
	Rate Data Minimum			(Optional) Minimum quantity for which the rate is applied
	Rate Data Maximum			(Optional) Maximum quantity for which the rate is applied
	Resource ID	User-defined valueset		<p>Identifies the resource that is charged for the product. For example, dollars, free minutes, and so on. Only Currency based resources are supported.</p> <p>When product admin in BRM adds the resources in BRM then the same resources are used to create a static non-versionable valueset in PIM and update the corresponding DVM with the values of PIM and BRM</p> <p>A DVM needs to be created.</p>
	GLID	User-defined valueset		ID assigned to record the revenues. When product admin in BRM adds the GLIDs in BRM then the same GLIDs are used to create a static non-versionable valueset in PIM and update the corresponding DVM with the values of PIM and

Attribute Group	Attribute	Valueset	Multi-Row	Functional Description and Handling
				BRM A DVM needs to be created.
	Impact Category	User-defined valueset		(Optional) It provides flexibility to specify one rate that charges different amounts based on an event, service, or account attribute. When product admin in BRM adds the impact categories in BRM then the same impact categories are used to create a static non-versionable valueset in PIM and update the corresponding DVM with the values of PIM and BRM A DVM needs to be created..
	Proratable			It is a Flag. Select to allow the resource to be prorated, for a cycle rate only
	Discountable			It is a Flag. Select to allow the resource to be discounted in the deal that is associated with this product
	Sponsorable			It is a Flag. Select to allow the resource to be sponsored by another account
	Grantable			It is a Flag. Select if the balance impact is granting the resource. This enables you to enter start and end times for the resource balance impact.
	Fixed Amount			Represents the fixed amount/price that is applied on the resource that is associated with the billing product. Optional if scaled amount is specified
	Scaled Amount			Represents the amount that can be credited/debited based on rating of the resource. For example, usage of minutes or internet download limits. Optional if fixed amount is specified. In case of usage charges both scaled and fixed amounts can be specified.
	Units	[None, Second, Minute, Hour, Day, Byte, Kilobyte, Gigabyte]		(Optional) Represents the units for the rating. Use a DVM to map these values A DVM needs to be created.
	Pricelist Name			Name of the price list associated with the price/charge This is for CRM. Ignored in BRM (CRM Specific)

Attribute Group	Attribute	Valueset	Multi-Row	Functional Description and Handling
	Promotional Price			This is for CRM. Ignored in BRM (CRM Specific)
	Service Price Percent			A percentage that is used to mark up the price of the service product. (CRM Specific)
	Service Pricing Method	[List Price, Net Price]		Specify the basis used for calculating the price markup of the service product from the covered product. (CRM Specific) Use a DVM to map these values.
	Start Date			(Optional) Start date for the balance impact
	End Date			(Optional) End date for the balance impact

Timezone Dependent and Timezone Independent Fields

Timezone dependent fields identify absolute time when an event happens or will happen. Such fields need to be translated to the timezone 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 timezone-dependent fields are translated to UTC timezone by PIM requestor. If PIM has captured the time as 10:00 AM PST, this is translated to 18:00 UTC. As always, values are coming in UTC format, the Oracle BRM provider does not have to do further translation.

Note: In Oracle Product Hub PIP, most date/time fields published by PIM are translated by the PIM requestor in UTC format; implying, they are timezone dependent.

Timezone independent fields are used to identify a time that should be used regardless of the timezone 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 different timezone as *evening* would always remain 7 P.M. through 7 A.M regardless of the timezone differences.

The timezone-independent fields are used as is by the PIM requestor. If PIM 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 timezone-independent values and does not translate these to UTC timezone. Instead, such values are converted to other data types (strings/numbers) by the Oracle BRM provider as requested by the Oracle BRM services.

Billing systems depend on these timezone 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 timezone-independent, date/time fields are in these attribute groups:

- Day Time Range
- Days of the Week Range

Publication Status for the Items

See: [Update Publishing Status in PIM upon publication of Items](#)

Update Publishing Status in OPH upon Publication of Items

The update of publish status in OPH (PIM) enables a product administrator to know the results of the previous publish and enables republishing if it is necessary. A publishing status is associated to an item and PIM provides services to update the publication history for items (for its publication to a given system within a batch). PIM 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 after the data is fetched from OPH for processing.
- **Succeeded** – Indicates that the item in a batch is successfully published to the target application(s).
- **Failed** – Indicates that the publication of the item to the target application(s) failed.
- **Ignored** – Indicates that the item was ignored by a specific system as it does not apply to it, for example, promotions for BRM. This is set as a default value on the PIM side API to save processing complexity on the AIA side.

AIA updates the publishing status in PIM for items published within a batch after all the entities in the batch are processed (both items and structures).

When the target system processes the entire payload at one time (entire ListItemEBM or entire ListBOMEbm), the service to update status at the batch level is used; else, 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 entity. When the target system offers services that are more granular than the submitted payload, but less granular than the items (like 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 the error occurred in.

The initial string is a message in English whose purpose is to help the PIM product administrator to 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, there may be multiple interactions with a given system (for items, price list, and BOM). The pre-pended message provides this context and allows 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 could 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 PIM using the responses to item, price list, and BOM synchronizations.

Synchronization of Metadata (value set and item catalog category) from Oracle Product Hub

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

The metadata synchronization supports these features:

- [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](#)

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

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 publication framework.

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 PIM. However, at any given point, there is only one active release version of the ICC.

During the ICC publish, the active release version or the future dated version of the ICC is 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 PIM product administrator.

The publish action invokes the integration process and provides basic information about ICC such as ICC unique identifier. The integration process queries OPH and the OPH interfaces provide complete details of the ICC and all the associated entities.

- The integration supports both create and update operation 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/updating structure/relationship, adding/deleting/updating transaction attributes and structures.

- After 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.
- If a version ICC is republished without any changes from PIM to Siebel CRM, then the updates on the product class does 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 release version or future dated versions can be published from PIM publication framework.
- In this release, these versions cannot be published from OPH:
 - Expired or past versions - effective end dates is earlier than the current date.
 - Draft version - versions that is not yet released.
- There is a provision to revert to the earlier version of the ICC. For this, a new version with effective dates must be created in PIM. There is no effect to the integration with this new version as the integration process handles this as new versions.

Auto Release of Entities

The auto-release provision is provided by PIM. By default, auto-release is enabled. The product administrator in PIM 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 on auto-release, see [Controlling Auto-Release of Entities Published from Oracle Product Hub in the same Batch for Siebel](#).

Synchronization of Relationships/Structure under Item Catalog Categories

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

- The ICC can be defined with items or BOMs as the relationships. When the integration queries PIM for the ICC, the structures that are defined as relationships to the ICC are also returned. The ICC has only references 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; else, the integration process fails.
- The query also returns the component UDA associated with the relationship. These 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 component UDA within the context of product class in Siebel.

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

This is an illustration on how various relationships under ICC is 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 PIM 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 PIM, 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 an UDA associated with the item that identifies the type of relationship.

For more information about Telco seeded Library attributes and their corresponding valuesets, see Appendix F: Seeded Item Metadata Libraries, *Oracle Product Information Management Implementation Guide*.

This is an illustration of how product and class type relationships are supported in PIM 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 as option group)

→Samba Bluetooth Headset

→B-Micro Bluetooth Headset

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

The following table shows the PIM definition of the relationship in the seeded attribute group: Version Structure associated with the Wireless Service ICC. These attributes are component attributes whose values are set in context of the Wireless Service ICC.

Relationship Name	Domain Type	Product	Class	Default Cardinality
Relationship1	Product	Wireless Router		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 Wireless Service ICC. 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.

This is an illustration 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

The structure attributes of the Wireless Service ICC in Siebel are as follows:

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

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.

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: PIM provides a seeded Telco Library that consists of set of predefined attribute groups. They consist 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 PIM. 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 like rules, pricelist lines, and so on. See: [Telco Seeded Library Attributes](#) for the definition of multi-row seeded attribute groups.
- When multi-row attribute groups are published, PIM 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 item synchronization process.)

Synchronization of Customer UDA as part of Item Catalog Categories

- The UDA 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 UDA 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 UDA 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 model 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 to 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 custom defined attribute groups that are defined without a prefix is 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.

Synchronization of Transaction Attributes as part of Item Catalog Categories

- The transaction attributes are defined within the context of the ICC in PIM. 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 values set 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 it in 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/update the context specific values) updates the ICC and a new version is released in PIM. 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 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.

Synchronization of Item Catalog Category Hierarchies

- The publication framework for ICC provide 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 has reference to the parent ICC. The integration process associates the child ICC and its parent in Siebel. Or, whenever a parent ICC is updated, the product administrator must publish the complete hierarchy associated with the ICC.
- In case in which the flag to publish the entire hierarchy (parents or children; or parents and children) is set, the integration process queries PIM for all the ICCs in the hierarchy. PIM 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 PIM and published to downstream applications, the integration updates the existing product classes. In addition, the corresponding hierarchy is updated in Siebel by 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 is inherited 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 same for transaction attributes. The process integration updates the corresponding entities in the downstream applications.

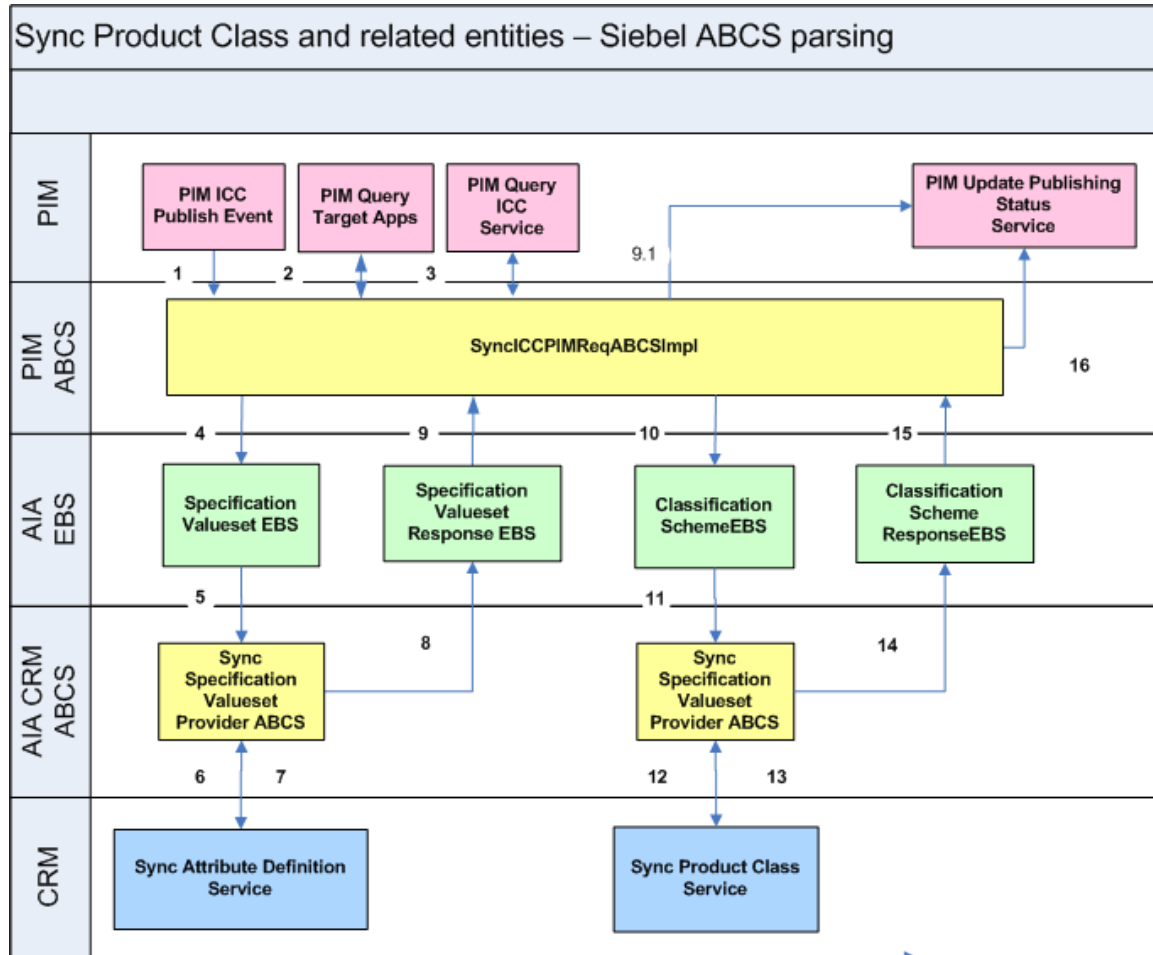
Association of Item Catalog Category to Items

An item catalog category can be associated to the items. The items inherit the static attributes, transaction attributes, and the 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.

Publish Item Catalog Category from PIM to Siebel CRM Implementation Flow

This graphic shows the high-level integration flows for item catalog category (ICC) synchronization.



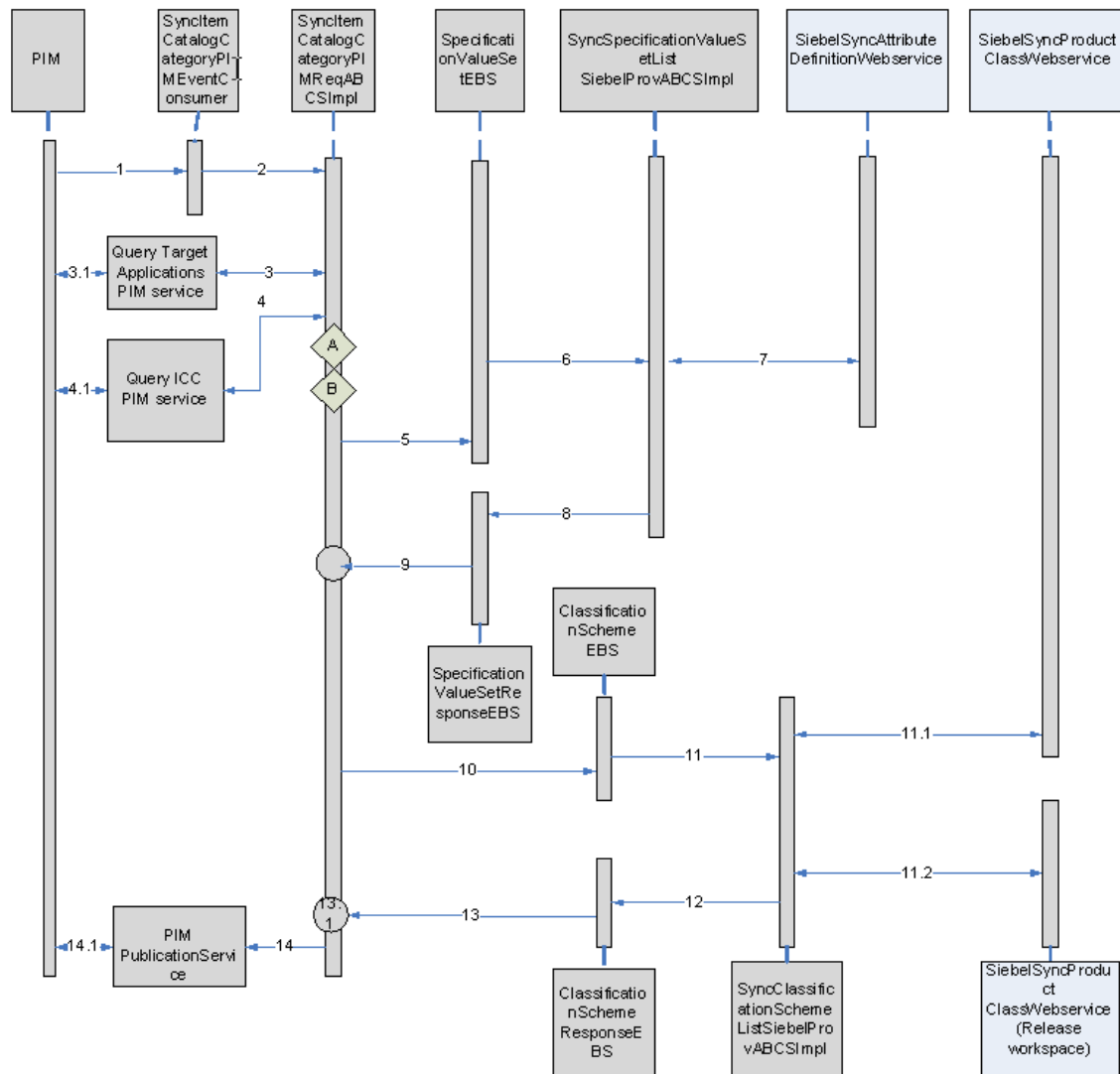
Synchronization of ICC/Product Class created in PIM

These are the synchronization steps:

1. The product administrator in PIM publishes the ICC from publication-framework user interface. The process publishes an event and provides basic information about ICC. The event consumer process invokes the requester ABCS for PIM.
2. The requester ABCS queries the target applications from PIM by calling the get target systems service.
3. After the target systems are retrieved, the requester queries the ICC definition from PIM by calling the PIM Query ICC interface. The PIM 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 to different attributes within the same or different attribute group in an ICC.

4. The PIM requester connector service transforms the response from the PIM 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 success or failure indication.
8. As soon as 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 PIM requester ABCS transforms the valueset response EBM and invokes the PIM service to update the status. The PIM service updates the publication history for the valueset.
10. The PIM 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 its attributes. The product class also has structure/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 PIM requester ABCS transforms the classification scheme response EBM and invokes the PIM service to update the status. The PIM service updates the publication history for the ICC.

This graphic illustrates the services and their interaction.



Synchronization of ICC in PIM

This is the interaction sequence:

Steps	Name	Step Description
1	PIM ICC publish event is raised from PIM System	PIM ICC publish events are raised when product administrator publishes the ICC from PIM system.
2	SyncItemCatalogCategoryPIMEvent Consumer	SyncItemCatalogCategoryPIMEventConsumer listens to business event and receives the WF_EVENT_T_msg event payload. SyncItemCatalogCategoryPIMEventConsumer routes to SyncItemCatalogCategoryPIMReqABCImpl with complete event payload.

Steps	Name	Step Description
3	SyncItemCatalogCategoryPIMReqABCSImpl	<p>The PIM Requestor, SyncItemCatalogCategoryPIMReqABCSImpl, receives the event payload and calls the PIM publication service to get the list of target applications by passing batch ID. It then constructs the query ICC PIM ABM, calls the PIM 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 to T-Attr in PIM and SyncValueSet flag is set to <i>True</i> in AIAConfigurationProperties.xml) and SyncClassificationSchemeListEBM.</p> <p>If valuesets are associated to T-Attr and valueset synchronize property is set to <i>True</i> in the AIAConfigurationProperties.xml, it invokes SpecificationValueSetEBS, passing SyncSpecificationValueSetListEBM to synchronize valuesets in Siebel and other participating applications.</p> <p>As soon as the valueset synchronization is complete, it calls the classification scheme EBS passing SyncClassificationSchemeListEBM to synchronize the ICC.</p> <p>It gets the ICC synchronization response, constructs the PIM publication service ABM, and calls the PIM publication web service.</p>
3.1	PIM 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	PIM Query ICC web service	PIM Query ICC web service receives the batch ID, publishes parent hierarchy and child hierarchy flags, and returns the ICC details associated to that batch.
5	SpecificationValueSetEBS	SpecificationValueSetEBS receives SyncSpecificationValueSetListEBM as an input and routes it to SyncSpecificationValueSetListSiebelProvABCSImpl service to synchronize valuesets.
6	SyncSpecificationValueSetListSiebelProvABCSImpl	SyncSpecificationValueSetListSiebelProvABCSImpl receives SyncSpecificationValueSetListEBM. It transforms the EBM into SyncAttributeDefinition ABM and calls the Siebel valueset-synchronization web service. As soon as the synchronization is complete, it gets the response from the Siebel, 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 an input and routes it to SyncItemCatalogCategoryPIMReqABCSImpl service.

Steps	Name	Step Description
9	ClassificationSchemeEBS	One the valueset synchronization is done successfully in Siebel, SyncItemCatalogCategoryPIMReqABCSImpl calls the ClassificationSchemeEBS passing SyncClassificationSchemeListEBM as its input and routes it to SyncClassificationSchemeListSiebelProvABCSImpl service.
10	SyncClassificationSchemeListSiebelProvABCSImpl	<p>SyncClassificationListSiebelProvABCSImpl receives SyncClassificationSchemeListEBM. It transforms the EBM into SWIClassImportUpsert_Input message and calls the Siebel product-class synchronization web service to synchronize simple product class along with attributes, structure if associated.</p> <p>It then populates the cross-reference with the corresponding Siebel IDs. As soon as it is complete, 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.</p>
11.1	Synchronize Product Class Siebel web service	Synchronize product class Siebel web service synchronizes the product class in Siebel and returns the response msg.
11.2	Siebel workspace release web service	As soon as the product class synchronization is complete, SyncClassificationSchemeListSiebelProvABCSImpl calls the Siebel product class synchronization-web-service again, passing workspace name, workspace refuse flag, and workspace auto-release flag to release the workspace.
12	ClassificationSchemeResponseEBS	ClassificationSchemeResponseEBS gets the SyncClassificationSchemeListResponse message and routes it to SyncItemCatalogCategoryPIMReqABCSImpl.
13	PIM Publication Service	<p>After the PIM Requestor gets the ICC synchronization response message, it calls the PIM ICC publish status web service to update the publication status in PIM. These are the possible status values:</p> <ul style="list-style-type: none"> Failed Succeeded Partial Fail (in case valueset fails)

Synchronization of Valuesets

The valuesets can be published independently or during the ICC publish from the PIM publication framework. The valuesets can be associated with the static UDA, customer UDA, and transaction attributes. For all the seeded valuesets of the Telco 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. See: [Multi Language Support for Valuesets](#).

Synchronization of Valuesets as part of Item Catalog Categories

- Whenever the integration queries OPH for ICC that is published, the valuesets that are associated with the attributes of the ICC are returned.

Note: This includes the non-versionable 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, PIM returns a unique set (union) of valuesets associated with the ICC.

- The integration supports only synchronize operation
- When a versioned valueset is republished to Siebel CRM, the updates on the attribute definition does not create a new version in Siebel CRM. After 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 Telco library.
- Whenever an ICC is published, the publication framework extracts the version of all the valuesets that were associated when the ICC was released. Implied, the version of the valueset that was associated to the ICC when the ICC was created might differ from the active release version of the valueset. Therefore, publishing always collects 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. There is no one-one mapping between the versions of entities in PIM and the downstream applications. OPH provides validation to ensure that only release version of the valueset can be associated to the attribute.

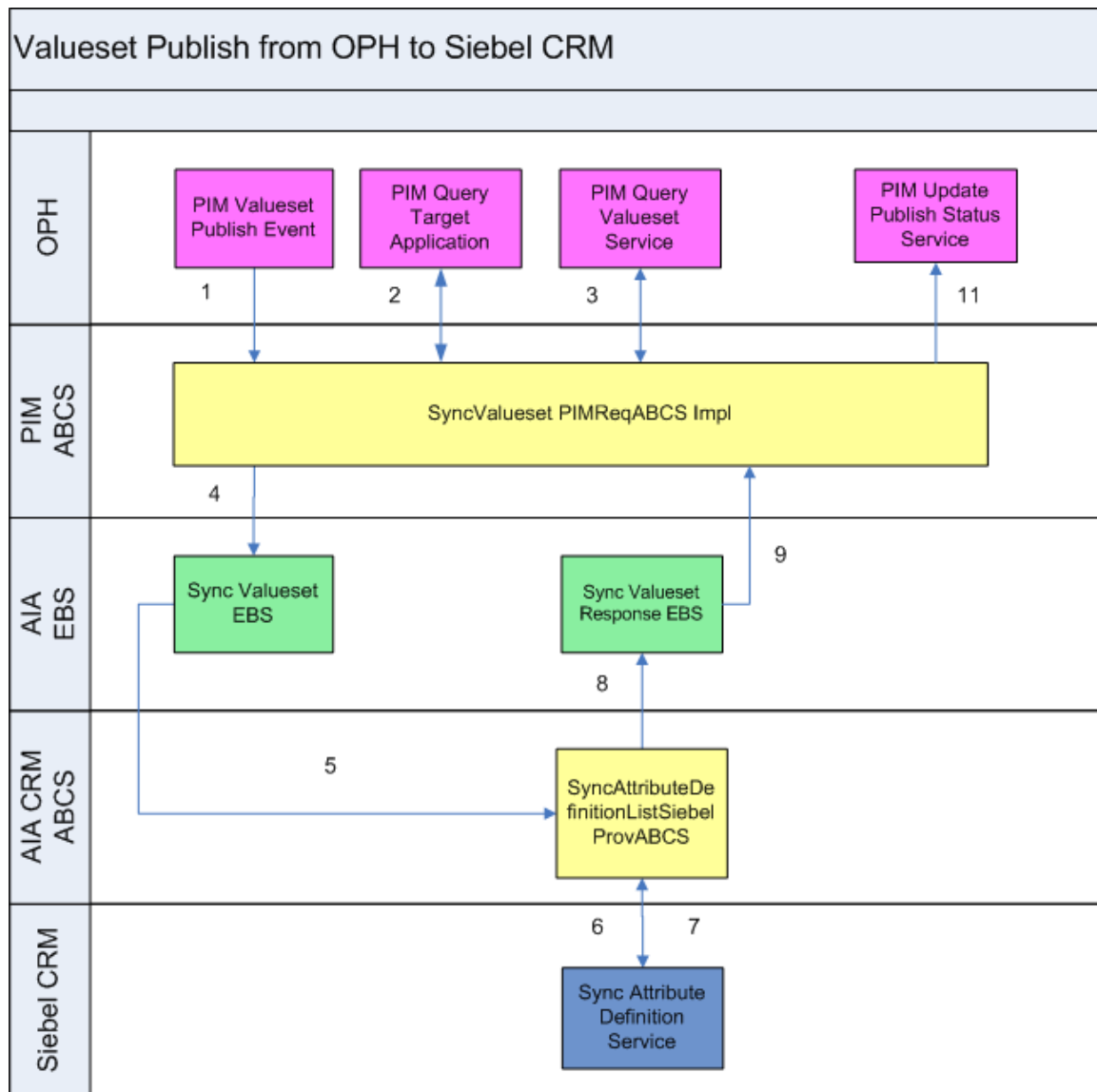
Synchronization of Valuesets Independently

- The publication framework provides separate user interface to search and to publish the valuesets. Multiple versions of the valueset exist in PIM, but at any point, there is only one active release version of a valueset. Only the active release version and the future dated

versions of the valuesets can be published. In PIM, you can publish one or more valuesets per publish process to the downstream participating applications. The publishing of valueset is a manual process performed by the PIM 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.
- When a version valueset is republished to Siebel CRM, the updates on the attribute definition does 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—a version that is not yet released, cannot be published from OPH.
- After the attribute definitions are successfully created in Siebel for versionable valuesets of PIM, 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 (PIM does not support Boolean type. If the valueset name has BOOL suffix, it is treated as Boolean type)
 - INT (PIM does not support Integer type. If the valueset name has INT suffix, it is treated as Integer type)
- Siebel supports all these datatypes and Boolean and Integer datatypes except Datetime.

This graphic illustrates the valueset publish from OPH to Siebel CRM.



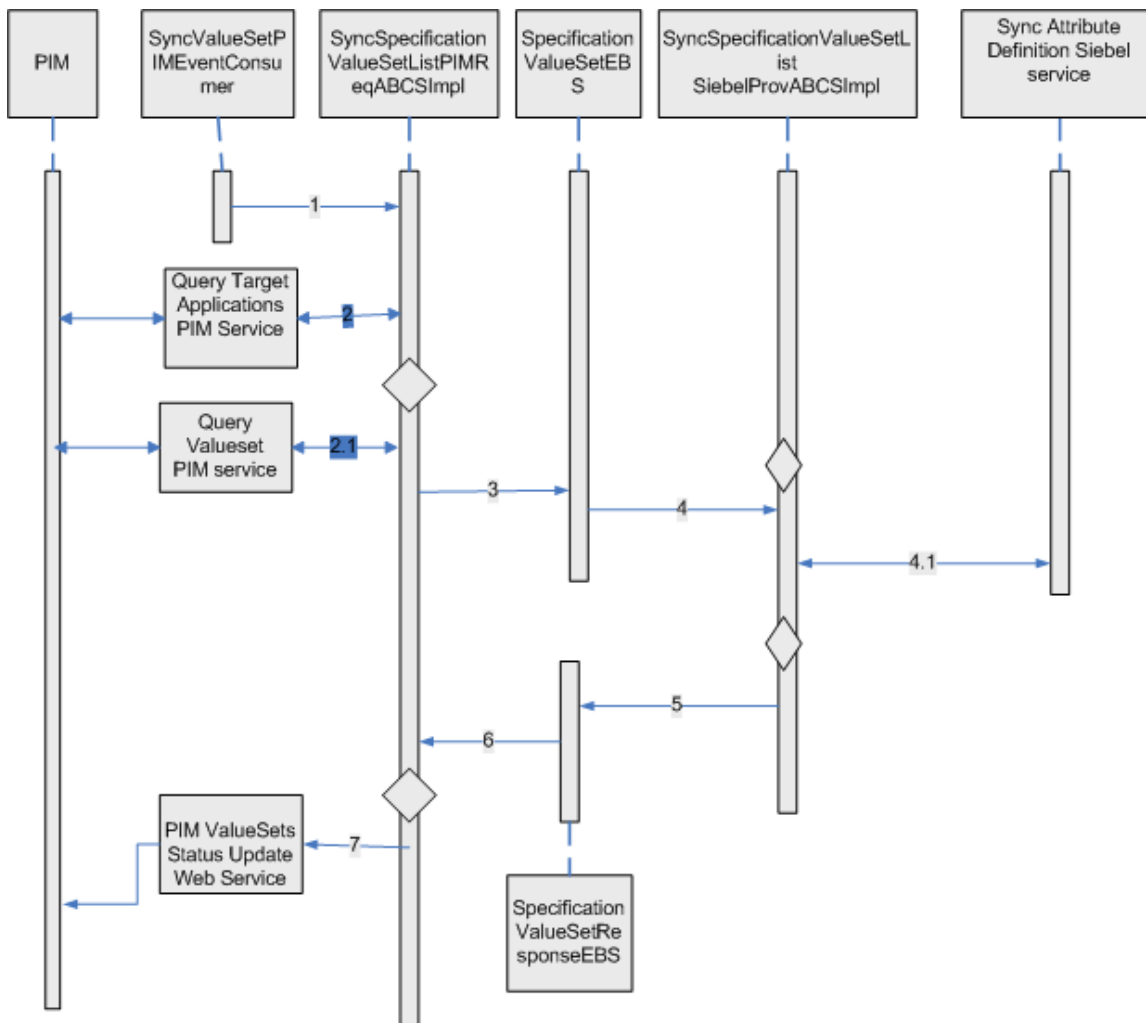
Valueset Publish from OPH to Siebel CRM

These are the steps depicted in the diagram.

1. The product administrator in PIM publishes one or more valuesets from 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 PIM
2. The requester ABCS queries the PIM valueset service.
3. The PIM valueset service returns a single payload with all the valuesets to the PIM requester ABCS.
4. The PIM 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. For *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 PIM provider ABCS.
10. The update valueset publish status PIM provider ABCS invokes the PIM-update-status web service and provides the valuesets and the corresponding statuses. The PIM updates status service updates the publication history to reflect the status of publish.

This graphic illustrates the services and their interactions.



Valueset Publish from OPH to Siebel CRM

This is the interaction flow:

Name	Description
------	-------------

Name	Description
PIM valueset event is raised from PIM System	<p>PIM valueset events are raised when product administrator publishes the valuesets from PIM system.</p> <p>Valuesets can be published in batches.</p>
SyncValueSetPIMEventConsumer	<p>SyncValueSetPIMEventConsumer listens to business event and receives the WF_EVENT_T_msg event payload for valueset event.</p> <p>SyncValueSetPIMEventConsumer routes to SyncSpecificationValueSetListPIMReqABCServiceImpl with complete event payload.</p>
SyncSpecificationValueSetListPIMReqABCServiceImpl	<p>The PIM requester ABC implementation service, SyncSpecificationValueSetListPIMReqABCServiceImpl, takes the batch ID from WF_EVENT_T ABM payload, queries the get target systems service, and calls Query valueset PIM service to get the response.</p> <p>SyncSpecificationValueSetListPIMReqABCServiceImpl service also calls the query target applications PIM service and gets the response back. It counts the target applications from response, and based on the count it constructs the EBM for applications.</p> <p>SyncSpecificationValueSetListPIMReqABCServiceImpl service transforms the Query valueset PIM service response ABM into QuerySpecificationValueSetList EBM and calls the SpecificationValueSetEBS Service.</p>
SpecificationValueSetEBS	<p>SpecificationValueSetEBS Service gets SyncSpecificationValueSetList EBM as its input and routes it to SyncSpecificationValueSetListSiebelProvABCServiceImpl service.</p>
SyncSpecificationValueSetListSiebelProvABCServiceImpl	<p>SyncSpecificationValueSetListSiebelProvABCServiceImpl receives the SyncSpecificationValueSetList EBM message as input.</p> <p>SyncSpecificationValueSetListSiebelProvABCServiceImpl transforms the SyncSpecificationValueSetList EBM message into Siebel request ABM message and calls the synchronize attribute definition Siebel web service.</p> <p>This service also filters the MLS mapping. There is a filter condition, which uses DVM to get the details on how many languages a provider system is supporting. Only those languages are synchronized to provider system, for example, Siebel.</p> <p>Synchronize attribute definition Siebel web service returns the response to SyncSpecificationValueSetListSiebelProvABCServiceImpl service, which has the status details.</p>
SyncSpecificationValueSetListSiebelProvABCServiceImpl	<p>SyncSpecificationValueSetListSiebelProvABCServiceImpl receives the response from synchronize attribute definition Siebel web service and transforms the Siebel response ABM message into SyncSpecificationValueSetListResponseEBM and calls the SpecificationValueSetResponseEBS service.</p> <p>The valueset synchronization is based on all or none basis. If PIM is sending a batch of 10 valuesets and out of which 5 (five) fail, this status is sent back to PIM that batch was failed and the error that comes from provider service is logged in error handling framework.</p>

Name	Description
SpecificationValueSetResponseEBS	SpecificationValueSetResponseEBS routes the SyncSpecificationValueSetListResponseEBM to SyncSpecificationValueSetListPIMReqABCImpl
SyncSpecificationValueSetListPIMReqABCImpl	SyncSpecificationValueSetListPIMReqABCImpl service transforms the SyncSpecificationValueSetListResponseEBM message into PIM valueset status-update web-service input message and calls the PIM value set status-update web service.
PIM value set status update web service	This web service takes input message from SyncSpecificationValueSetListPIMReqABCImpl service and updates the status in PIM as: <ul style="list-style-type: none"> Success Failed

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 ICC one at a time because this is performed design time.

Multi Language Support for Valueset Synchronization

See: [Multilanguage Support for Item Synchronization](#).

Oracle Product Hub Interfaces

The Oracle Product Hub artifacts used by this integration are as follows:

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

Oracle Product Hub Integration Services

These are the services delivered with this integration:

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

You can use the **Integration Scenario Summary** page in the Oracle AIA Console to search for and view the integration scenarios that use a particular ABC service.

For more information, see *Oracle Application Integration Architecture – Foundation Pack: Core Infrastructure Components Guide*, “Using the BSR,” Using the BSR UI to View Integration Scenarios.

SyncItemCatalogCategoryPIMEventConsumer

The SyncItemCatalogCategoryPIMEventConsumer is implemented as a lightweight ESB routing service. The SyncItemCatalogCategoryPIMEventConsumer has an Oracle applications adapter configured to listen for PIM ICC publish business event. There is one service with one operation to read the Oracle PIM message SyncItemCatalogCategoryPIMReqMsg from the Oracle AQ WF_BPEL_Q: SyncItemCatalogCategoryPIMEventConsumer. It is implemented as an ESB process with an Oracle applications adapter for listening to business event.

SyncItemCatalogCategoryPIMReqABCImpl

The SyncItemCatalogCategoryPIMReqABCImpl is a BPEL process that receives the PIM ICC Event message from the SyncItemCatalogCategoryPIMEventConsumer service and is responsible for calling the PIM query participating applications list for the batch and PIM Query ICC web service, based on event payload. It transforms the PIM ICC ABM messages into the appropriate SyncClassificationListEBM and SyncSpecificationValueSetListEBM. It invokes SpecificationValueSetListEBS (SyncSpecificationValueSetList operation) to synchronize valuesets in Siebel and ClassificationEBS to synchronize product class in Siebel (SyncClassificationList Operation).

It is an asynchronous service that accepts the PIM ICC event message as a request and follows all the standards of a requester ABCS.

SyncItemListPIMEventConsumer

The SyncItemListPIMEventConsumer service has an Oracle applications adapter configured to listen for PIM business event. This service has only one operation and that is to read the PIM message SyncItemListReqMsg from the Oracle AQ (advanced queuing system).

SyncItemListPIMEventConsumer is implemented as an enterprise-service-bus process with an Oracle applications adapter for listening to business events.

SyncItemListPIMReqABCImpl

The SyncItemListPIMReqABCImpl is a BPEL process that receives the PIM event message WF_BPEL_EVENT_T from the SyncItemListPIMEventConsumer service. The event payload has the batch ID, workspace name, structure flag, auto-release, and other details as needed by the PIM service to query. It then calls these PIM services with the batch ID:

- PublicationService_GetBatchSystems: This service is used query list of target applications associated with the batch to which the items/pricelists/BOM EBM's have to be published.
- QueryItemListAdapter: To get the details of the items in the batch.
- QueryBillOfMaterialsListAdapter: To get the details of the BOM in the batch.

The resultant PIM item and BOM query response ABM message is then converted into

- SyncItemListEBM
- SyncPriceListEBM
- SyncBillOfMaterialsListEBM

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 there is no pricelist or BOM 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 PIM application with the item response status for every entity in the batch.

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 target system is configured to receive the price list (ENTITY_TARGET_APPLICATION_MAPPING DVM). If there is no BOM 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 PIM application with the pricelist response status.

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 `SyncItemReqABCS` calls the `QueryBillOfMaterialsListPIMAdapter` service. The `SyncBillOfMaterialsListEBM` is sent to the participating applications.

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

QueryItemReqPIMAdapter

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

QueryBillOfMaterialsListPIMAdapter

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

SyncSpecificationValueSetListPIMEventConsumer

The `SyncSpecificationValueSetListPIMEventConsumer` is implemented as a lightweight ESB service. The `SyncSpecificationValueSetListPIMEventConsumer` has an Oracle applications adapter configured to listen for PIM valueset publish business event. There is one service with one operation to read the Oracle PIM message `SyncValueSetReqMsg` from the Oracle AQ `WF_BPEL_Q: SyncSpecificationValueSetListPIMEventConsumer`. It is implemented as an ESB process with an Oracle applications adapter for listening to business event. PIM synchronize value set business event: `oracle.apps.ego.extfwk.publishValueSet`.

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 PIM valueset event message from the SyncValueSetPIMEventConsumer service. This service is responsible for calling the PIM query (participating applications) valueset web service, based on event payload. It transforms the PIM valueset ABM messages into the appropriate SyncSpecificationValueSetListEBM. It invokes SpecificationValueSetEBS (SyncSpecificationValueSetList operation) to synchronize valuesets in Siebel.

This asynchronous service accepts the PIM 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 provider system can support. Once the synchronization is complete, it gets the response message and passes the response to SpecificationValueSetResponseEBS.

Assumptions and Constraints

The assumptions and constraints for Oracle Product Hub PIP include the following:

1. Oracle Product Hub is the master for definition of items, BOMs and metadata. Any changes/enrichment that may 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 after it is released. The parent ICC cannot be changed even to create a new version. This is PIM limitation.
5. The BOM item of type option class and entity type as option group and its component items must have the same item catalog category.
6. Changes to the seeded static attribute groups are not supported. If the customers want to introduce new static attributes, they must define new attribute groups and static attributes. The process integration creates these as flex attributes in Siebel. If new valuesets are introduced in OPH for new UDA, the integration passes the information as freeform text.
7. The items and BOMs in the structure of an ICC must be published before the ICC is published.
8. The same item cannot be added as a component of a BOM Item. This is PIM limitation.
9. The MLS support offered by PIM for the valuesets depend on the validation type associated with the valueset. The validation type independent and translatable is supported for this release.

10. PIM does not support Boolean and Integer data type. 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. However, the values must always be same, and seeded.
11. Publishing alternate catalogs from Oracle Product Hub is not supported by the integration.
12. Variant type attribute group in PIM is not supported by the integration.
13. There is no explicit synchronization flow for discount models from OPH to Oracle BRM.
14. The item catalog categories that are associated with the items must be synchronized before the items are published.
15. If the participating application is 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 before you synchronize items that are associated with any other organization.
16. In case of participating application being Oracle E-Business Suite, it is assumed that the primary BOM associated with the BOM must be synchronized prior to invoking synchronization of alternate BOMs to Oracle E-Business Suite.
17. Items whose BOM item type is option class and the entity type is option group, the user item type must be set to option choice.
18. Every operating unit in PIM (also known as 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 is defined as 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, see [Setting Up Cross-References](#) for establishing cross-references. (Applicable only if O2C PIP is being planned or implemented.)
19. 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 to a single operating unit can be synchronized to Siebel CRM. (Applicable only if O2C PIP is being planned or implemented.)
20. The process integration ignores the custom attributes that are added to the structure header though Oracle Product Hub publishes them whenever the corresponding item is published.
21. Oracle Product Hub produces multiple BOM revisions. It can also 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.
22. Oracle Product Hub must be deployed as a separate instance than Oracle E-Business Suite.
23. Synchronization of user-defined attributes (UDAs) to descriptive flexible fields in Oracle E-Business Suite is not supported as part of this release. Therefore, an out-of-the-box solution does not map UDAs for Oracle E-Business Suite provider service.
24. For Oracle E-Business Suite, account IDs used for the integration remain synchronized for PIM and Oracle E-Business Suite. The locator code and sub-inventory need to be manually set up in Oracle E-Business Suite.

25. 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 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.
26. Multiple rate plans using rate plan selectors are not supported in PIM. Implying, for real-time rating either single rate plan or multiple single rate plans must be associated to one-time, recurring and usage charges in PIM and the rate plans for the rate plan selectors must be enriched in Oracle BRM.
27. The usage-based products are defined in PIM as separate products and are not defined as a part of multi-event product. The pipeline rating for usage products is defined and associated in Oracle BRM.
28. The usage based rating using pipeline rate plans cannot be defined in PIM. Therefore, they are not supported by the integration. They must be defined in Oracle BRM.
29. The enrichment for one-time and cycle charges must not change the rate plan names in Oracle BRM.
30. The process integration does not support multi-tier pricing between PIM and Siebel.
31. The discount models are associated with the items that represent the discounts in PIM. The complete definition of the discount model is performed in Oracle BRM. The discount model name must not be updated in BRM, since the subsequent updates will override the value.
32. PIM does not support adding the same item as component of BOM more than once. Siebel and process integration does support these. This is a PIM limitation. If you require it, then you must use the operation sequence support that is provided by PIM
33. For Siebel CRM, these entities and associated subcomponents need to be locally enriched:
- Eligibility rules.
 - Proration plans for items of entity type as promotion.
 - Compatibility rule matrix.
 - Matrix discounts.
 - Volume Discounts.
 - Attribute adjustments.

For more information, refer *Known Issues and Workarounds* on My Oracle Support (<https://support.oracle.com/>).

Chapter 3: Oracle Product Master Data Management Integration Option for Siebel CRM

This chapter provides an overview of the Siebel CRM process flows and covers:

- [Overview](#)
- [Siebel CRM interfaces](#)
- [Siebel CRM integration services](#)

Overview

Siebel CRM is available as an installation option in the process integration pack for Oracle Product Hub. There is no synchronization flow from Siebel CRM back to the Oracle Product Hub. The flows from OPH to Siebel CRM and the features applicable to those flows are as follows:

- [Synchronization of Items and Bill of Materials](#)
- Synchronization of Metadata, that consists of:
 - [Synchronization of Item Catalog Categories](#)
 - [Synchronization of Valuesets](#)

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

Siebel CRM Interfaces

Siebel CRM interfaces that are used by the integration are as follows:

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 Oracle Product Hub PIP.

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

Siebel CRM Integration Services

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

- [SyncProductSiebelProvABCImpl](#)
- [SyncBillOfMaterialsListSiebelProvABCImpl](#)
- [SyncClassificationSchemeListSiebelProvABCImpl](#)
- [SyncSpecificationValueSetListSiebelProvABCImpl](#)
- [ProductOptimizedSyncPriceListLineListSiebelProvABCImpl](#)
- [SyncItemCompositionListSiebelCommsProvABCImpl](#)

SyncProductSiebelProvABCImpl

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

For type promotion, it transforms this message into a Siebel 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. For 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.

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 PIM, the provider services has two distinct behavior:

- 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 PIM are set as relationship domain for the class type relationship.

SyncClassificationSchemeListSiebelProvABCImpl

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

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

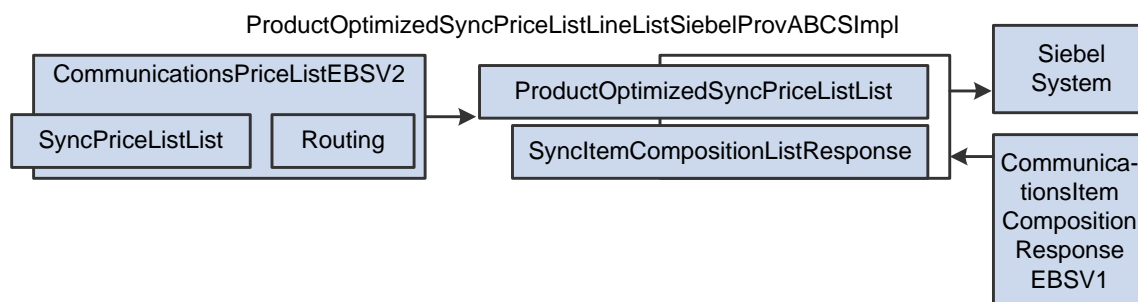
SyncSpecificationValueSetListSiebelProvABCImpl

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

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

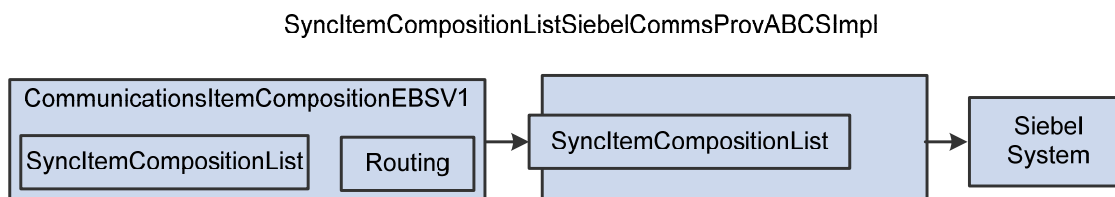
ProductOptimizedSyncPriceListListSiebelCommsProvABCImpl

The ProductOptimizedSyncPriceListLineListSiebelProvABCImpl service transforms the PriceList EBM into a Siebel price list message and then calls the Siebel pricelist web service on operation InsertOrUpdate. The ProductOptimizedSyncPriceListLineListSiebelProvABCImpl transforms the PriceList EBM into a Siebel product message and then calls the Siebel product web service on operation Product_spclImport_spclUpdate. The Siebel web service completes the request and returns a response message. ProductOptimizedSyncPriceListLineListSiebelProvABCImpl then transforms the Siebel response message into the PriceList response EBM and sends it back to PriceListEBS.



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.



For more information see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer’s Guide*, “Designing and Constructing ABC Services” and *Oracle Application Integration Architecture - Foundation Pack: Concepts and Technologies Guide*, “Understanding ABC Services”

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

This chapter provides an overview of the Oracle E-Business Suite process flows and covers:

- [Overview](#)
- [Oracle E-Business Suite interfaces](#)
- [Oracle E-Business Suite integration services](#)
- [Set Up Users and Roles in Oracle E-Business Suite Instance](#)

Overview

Oracle E-Business Suite is available as an option while you install the process integration pack for Oracle Product Hub. The flows available in OPH PIP when Oracle E-Business Suite is selected as a participating application are covered under [Synchronizing Items and BOMs from Oracle Product Hub to Oracle E-Business Suite](#).

There is no flow from Oracle E-Business Suite back to Oracle Product Hub.

For more information about Oracle E-Business Suite, see Oracle E-Business Suite product documentation.

Oracle E-Business Suite Interfaces

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

Item Synchronization: Inbound to EBS Web Services

- Service name: SyncItemLEbizAdapter
 - Operation name: SyncItemLEbizAdapter
 - 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 EBS web services, see the *E-Business Suite references: E-Business Suite Electronic Technical Reference Manual* (eTRM) located on My Oracle Support under the E-Business Suite Information Center, Oracle Integration Repository located at <http://irep.oracle.com>, Oracle Applications Release 11.5.10+ Online Documentation Library, located on the Oracle Technology Network (<http://www.oracle.com/technology/documentation/applications.html>).

Oracle E-Business Suite Integration Services

The following is a list of integration services that are required for Oracle E-Business Suite to integrate with Oracle Product Hub.

- [SyncItemListEbizProvABCImpl](#)
- [SyncBillOfMaterialsListEbizProvABCImpl](#)
- [SyncItemListEbizAdapter](#)
- [SyncBillOfMaterialsListEbizAdapter](#)
- [TransformAppContextEbizService](#)
- [QueryResponsibilityEbizAdapter](#)

SyncItemListEbizProvABCImpl

This single operation service accepts a SyncItemListEBM product message as a request. This service transforms the SyncItemListEBM product message into Oracle E-Business Suite application business message (ABM) format and executes Oracle E-Business Suite 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.

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.

SyncItemPIMReqABCImpl invokes SyncBillOfMaterialsListEbizProvABCImpl through BillOfMaterialEBS. The response is sent back to SyncItemPIMReqABCImpl 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 Oracle applications adapter for synchronizing the BOM.

SyncItemListEbizAdapter

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

SyncBillOfMaterialsListEbizAdapter

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

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.

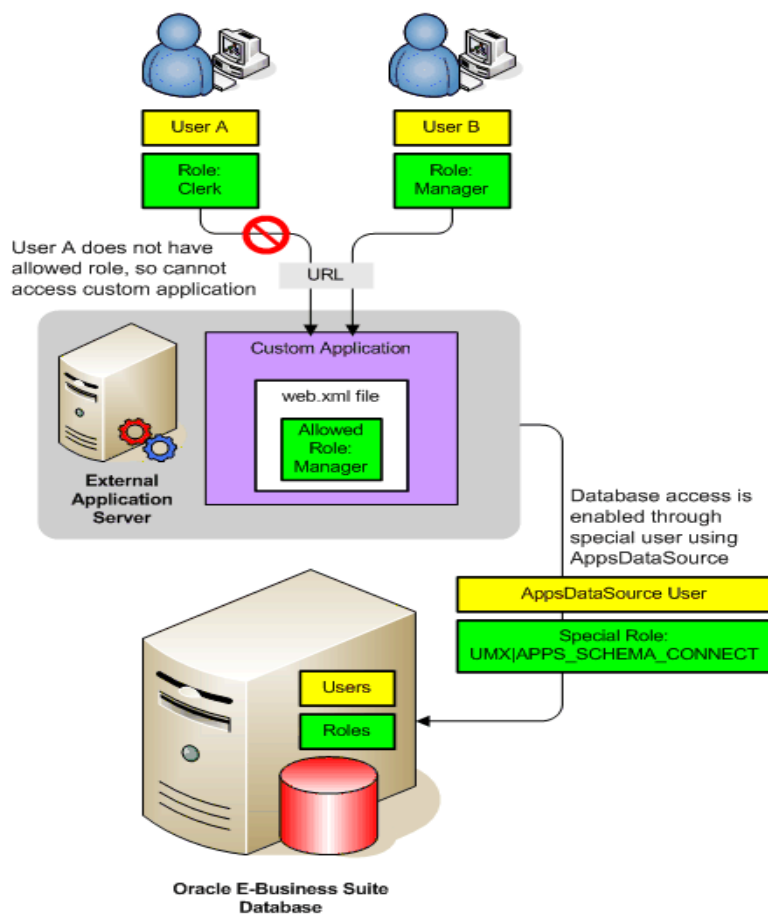
QueryResponsibilityEbizAdapter

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.

Set Up Users and Roles in Oracle E-Business Suite Instance

On the Oracle E-Business Suite instance, for every `<role-name>UMX|DEMO</role-name>` section defined in the `web.xml` in the following step, define a role with the same code using the User Management responsibility in Oracle E-Business Suite and assign the role(s) to the intended Oracle E-Business Suite user(s).

The following diagram shows how users and roles are used in the JAAS and AppsDataSource setups:



1. There are two different users, A (with Clerk role) and B (with Manager role), accessing a protected custom application (through a URL) on an external application server.
2. The custom application has a `web.xml` file that allows access for the Manager role as part of the JAAS setup.
3. User A does not have the Manager role, so is not allowed access to the custom application.

4. The external application server has an AppsDataSource set up to allow access to the Oracle E-Business Suite database using the dedicated AppsDataSource user that has the special UMX|APPS_SCHEMA_CONNECT role assigned to the dedicated user.
5. A repository of users and roles resides inside the Oracle E-Business Suite database.

Chapter 5: Oracle Product Master Data Management Integration Option for Oracle Communications Billing and Revenue Management

This chapter provides an overview of the Oracle BRM process flows and covers:

- [Overview](#)
- [Oracle BRM interfaces](#)
- [Oracle BRM Integration Services](#)

Overview

Oracle BRM is installed differently than other participating applications covered in this process integration pack for Oracle Product Hub. While Siebel CRM and Oracle E-Business Suite are installed through Oracle Universal Installer (OUI), Oracle BRM is installed as a patch.

For more information about how to install Oracle BRM, see *Oracle Product Master Data Management Integration Option for Oracle Communications Billing and Revenue Management 2.5: Installation Guide*

The flows available in OPH PIP when Oracle BRM is selected as a participating application are covered under [Synchronization of Items and Bill of Materials from Oracle Product Hub](#).

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

There is no flow from Oracle BRM back to Oracle Product Hub.

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.

Oracle BRM Integration Services

The following is a list of integration services that are required for Oracle BRM to integrate with Oracle Product Hub.

- SyncItemListBRMProvABCImpl
- SyncPriceListListBRMProvABCImpl

SyncItemListBRMProvABCImpl

This single operation service receives SyncItemListEBM as a request from ItemEBS. This input is transformed to BRM Pricing ABM to invoke BRM PCM_OP_PRICE_SET_PRICE_LIST web service. When the web service returns back to the BPEL with BRM IDs, cross-reference values are updated with the BRM identifiers.

SyncPriceListListBRMProvABCImpl

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.

Chapter 6: Implementing the Process Integration Pack for Oracle Product Hub

This chapter provides an overview of the implementation activities for Oracle Product Hub PIP and covers:

- [Adding a New Spoke Instance](#)
- [Adding an Integration Option or Target Instance for MDM Integration](#)
- [Extracting Organizations and Inventory Locations Data to Setup Cross-References](#)
- [Setting up Cross-References](#)
- [Identifying Cross-References](#)
- [Describing Domain Value Maps](#)
- [Handling Errors](#)
- [Viewing EBO Implementation Maps \(EIMs\)](#)
- [Setting Configuration Properties](#)
- [Setting up the Tata for Seeded Attributes of PIM Telco Library](#)

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

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. Log on 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:

1. Name : JD Edwards

2. 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. For 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. Log on 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: The following table lists sample DVMs.

COMMON	PIM_01
--------	--------

EBIZ_01 EBSSS	EBSSS
SEBL_01 Siebel	Siebel
BRM_01 BRM	BRM
JDE_01 JDE	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

The following table lists examples for System Name and 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.xp
ath.

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.

Adding an Integration Option or Target Instance for MDM Integration

To add a new integration option or target instance:

1. Add the instance name to the DefineSourceSystem by using Product Information Management Data Librarian responsibility.
2. Add the same entry on Destination_System valueset.
3. Add the entry in TARGET_ID DVM with COMMON value pointing to system defined in business service repository (BSR) system with PIM_01 value pointing to system defined in PIM system.
4. Add the entry in ENTITY_TO_TARGET_APPLICATION DVM for processing PriceList and BOM.
5. Add the routing rules to the new instances.
6. Run the scripts to create the columns that correspond to the instances in the cross-reference tables.

Extracting Organizations and Inventory Locations Data to Setup Cross-References

Inventory organizations can be widely 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 are classified as inventory organization to which product can be associated.

Other aspects are as follows:

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

Note: Creating operating units or inventory organizations is beyond the scope of this document. See Oracle E-Business guides for more information.

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 performed to support order capture in Siebel, we 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: [Mapping Siebel CRM](#).

If Oracle E-Business Suite is one of the participating applications in the PIP Implementation and O2C is part of the same, you first need to perform these steps. After these steps are complete, you need to implement a similar organization hierarchy setup in PIM application. If O2C is not in the set of PIPs this step is not required.

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

1. Log on to Oracle E-Business Suite database.
2. Run an 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 by using the XREF_TABLE_NAME: INVENTORY_LOCATION_ID.

To extract the operating unit details in Oracle E-Business Suite:

1. Log on to Oracle E-Business Suite database.
2. Run an 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 '
```

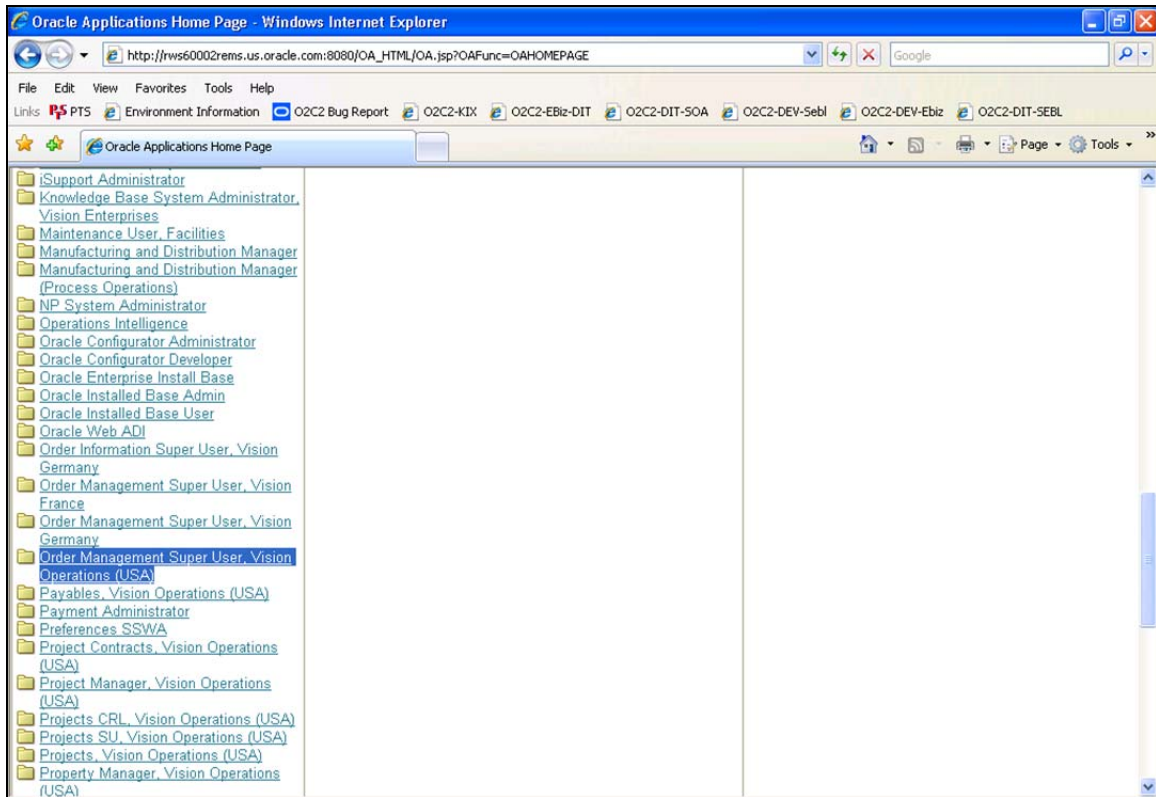
3. [Identify the Item Validation Organization for Oracle E-Business Suite](#) for the operating unit that you want.

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 on to Oracle E-Business Suite.

This page appears.

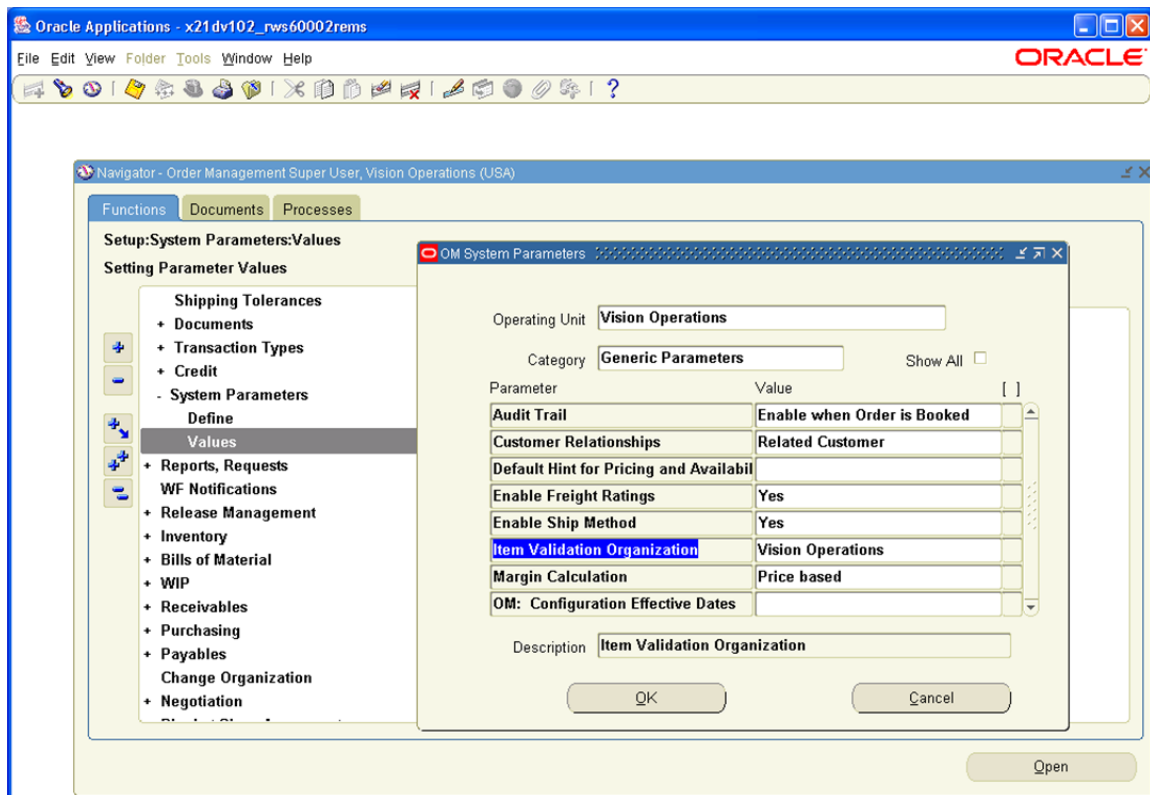


Oracle applications home page example

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).



System parameters

4. Follow these steps by 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. Note the value that appears in **Item Validation Organization**. (Examples: Vision Operations, Vision Germany, and Vision France). Mapping Oracle Product Hub Organizations

Extract Inventory Location and Operating Unit IDs in Oracle Product Hub

To extract inventory location IDs:

1. Log on to Oracle Product Hub database.
2. Run an 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 operating unit IDs:

1. Log on to Oracle Product Hub database.
2. Run an 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. 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 that you want.

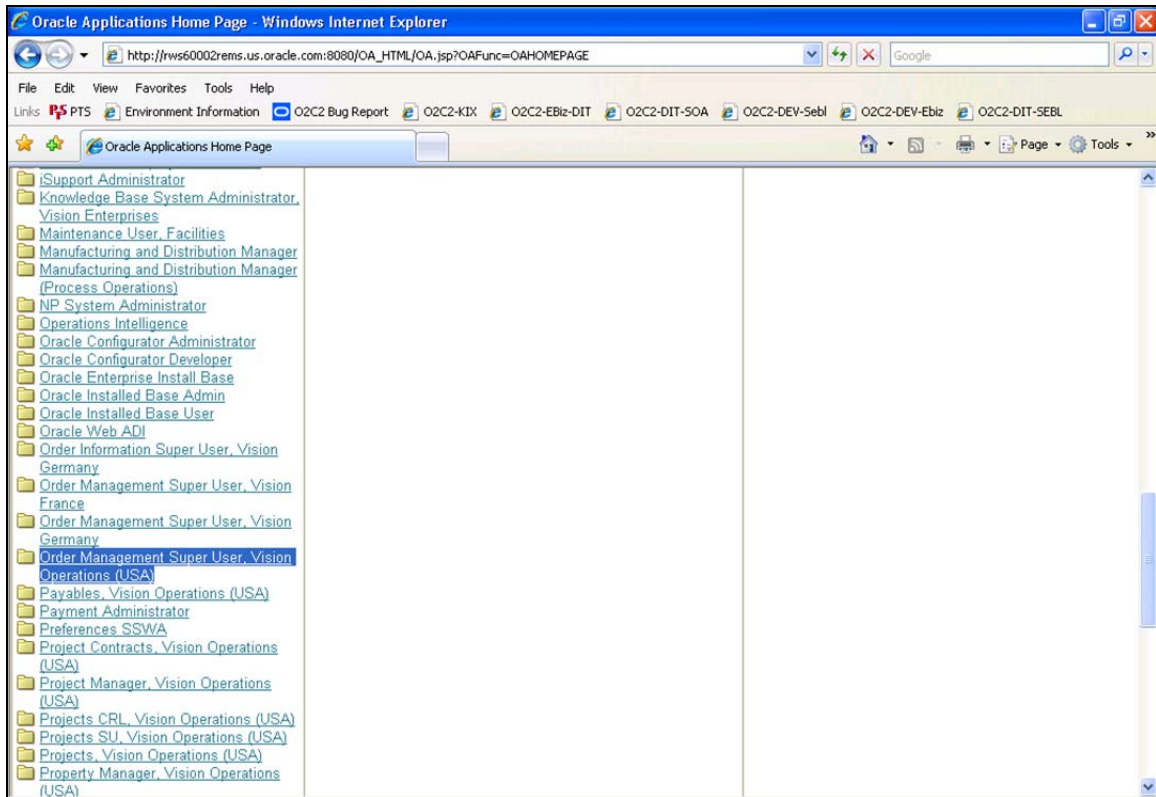
Identify Item Validation Organization for Oracle Product Hub.

To identify the item validation organization from the names of the organizations you noted in [Extract Inventory Organization and Operating Unit IDs in 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 on to Oracle Product Hub.

This page appears.

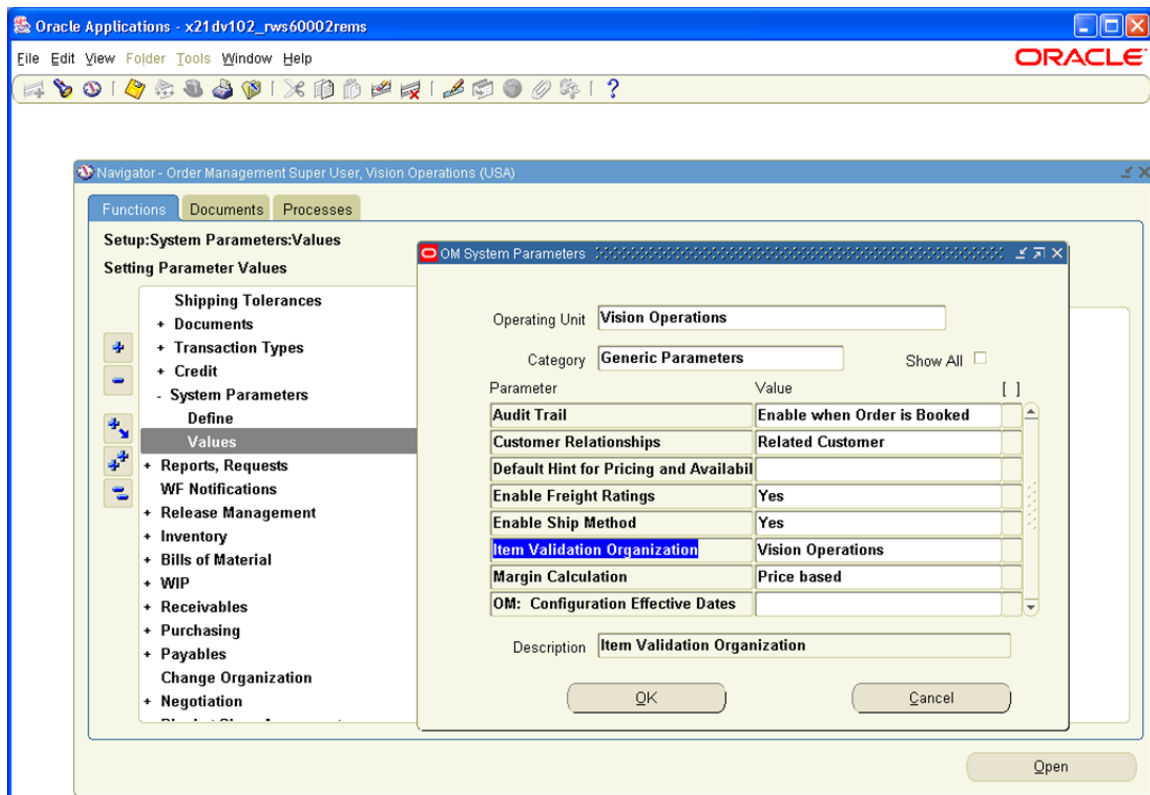


Oracle applications home page example

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)**.



System parameters

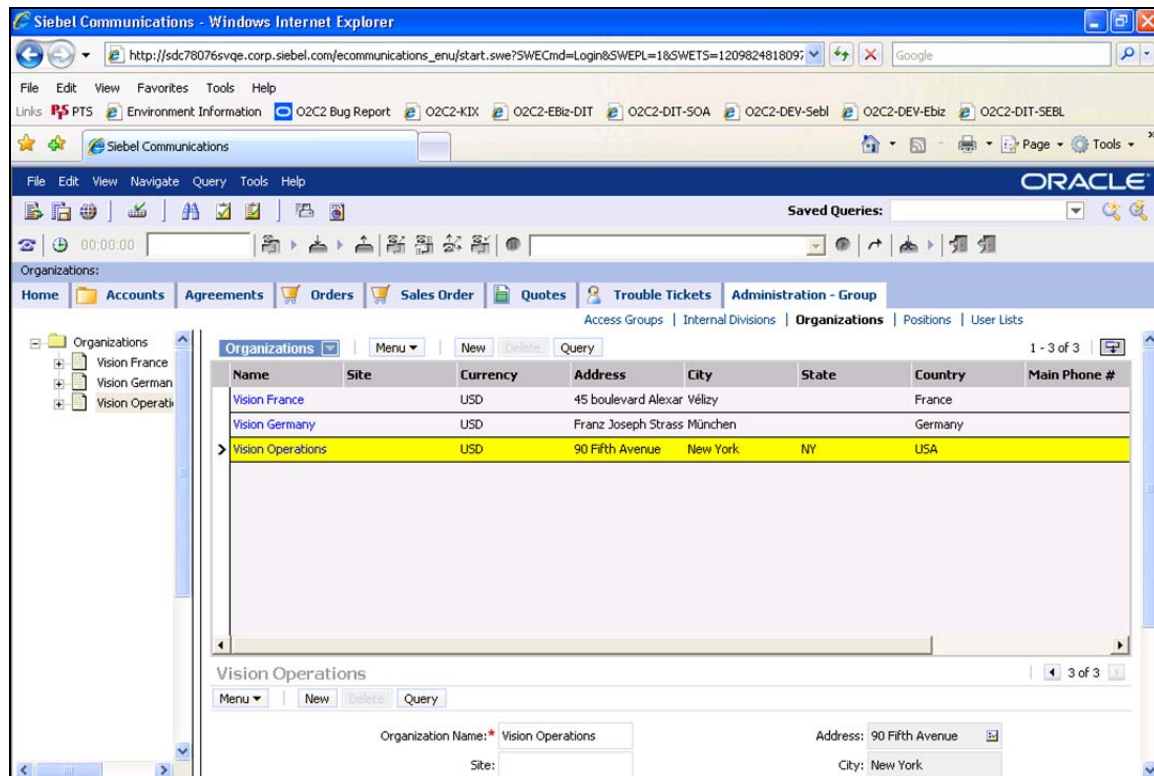
4. Follow these steps by using the **System Parameters** screen for each of the order management responsibilities associated with various operating units that are of interest to you:
 - a. Select **Values** from **Setting Parameter Values**.
 - b. Enter **Vision Operations** in **Operating Unit** and **Generic Parameters** in **Category**.
 - c. Note down the value that appears in **Item validation organization**. (Examples: Vision Operations, Vision Germany, and Vision France).

Mapping Siebel CRM

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

1. Log on to Siebel Applications.
2. Click **Site Map**.
3. Select **Administration-Groups, Organizations**.

For the Oracle E-Business Suite operating units that were identified in [Identify the Item Validation Organization for Oracle E-Business Suite](#), create those in Siebel CRM. Here is a screen shot:



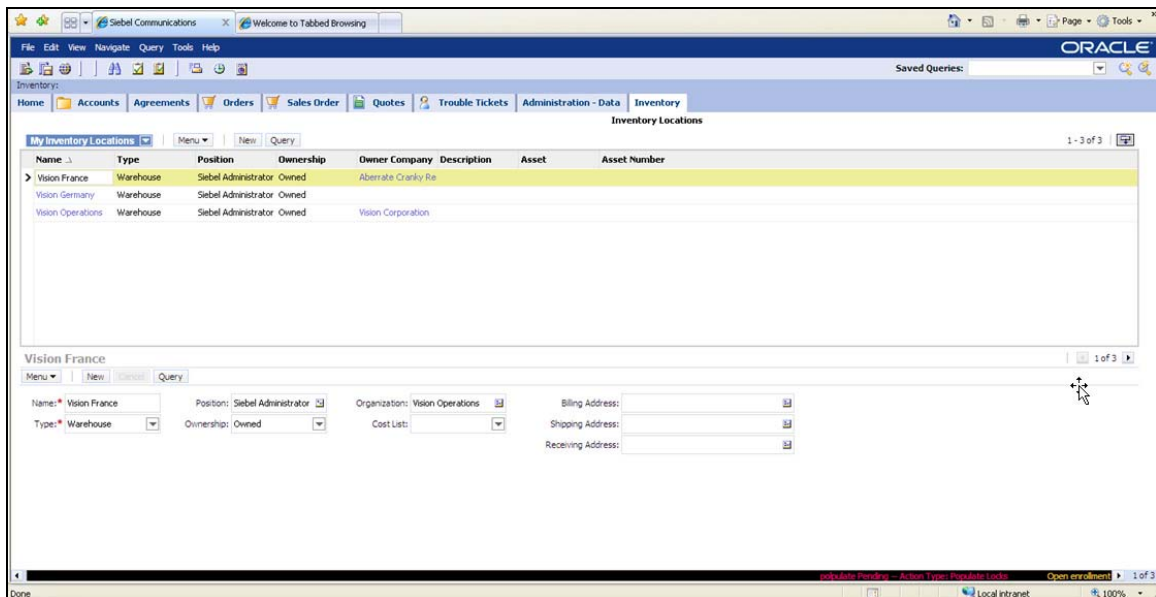
Administration – Group tab in Siebel

Creating Inventory Locations in Siebel

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

1. Log on to Siebel applications.
2. Click **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 the Inventory Location screen:



Inventory tab in Siebel

Extract Business Unit and Inventory Location IDs in Siebel CRM

To extract inventory location details for Siebel CRM:

1. Log on 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%')
```

Setting Up Oracle BRM

These are the setups required for Oracle BRM:

- Install and configure BRM JCA adapter.

For more information about how to install and set up the BRM JCA adapter, refer 'Installing the BRM JCA Resource Adapter' and 'Service Integration Components, JCA Resource Adapter' in 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, refer 'About creating a price list, About setting up a price list' in Oracle BRM documentation.

- Set up pricing for friends and family functionality.

For more information about how to set up pricing, refer 'Setting Up Pricing and Rating, Working with extended rating attributes' in Oracle BRM documentation,

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 the following:

Populating Cross-References

Note: If an organization is not defined as item validation organization, there should be no cross-reference value for the Siebel column for such inventory organizations.

To populate the cross-references:

1. Create **Organization.xml** and **Inventory_location_id.xml** by using the sample provided here. Update the XML files with the IDs from Siebel CRM, Oracle E-Business Suite, and Oracle Product Hub.

Sample **Organization.xml**:

```
<xref xmlns="http://xmlns.oracle.com/xref">
  <table name="ORGANIZATION_ID">
    <columns>
      <column name="EBIZ_01"/>
      <column name="SEBL_01"/>
    </columns>
    <rows>
      <row>
        <cell colName="EBIZ_01">204</cell>
        <cell colName="SEBL_01">88-25CHZ</cell>
        <cell colName="PIM_01">204</cell>
      </row>
      <row>
        <cell colName="EBIZ_01">888</cell>
        <cell colName="SEBL_01">88-25CIA</cell>
        <cell colName="PIM_01">888</cell>
      </row>
    </rows>
  </table>
</xref>
```

```

                <row>
                    <cell colName="EBIZ_01">911</cell>
                    <cell colName="SEBL_01">88-25CI5</cell>
                    <cell colName="PIM_01">911</cell>
                </row>
            </rows>
        </table>
    </xref>

```

Sample INVENTORY_location_id.xml:

```

<xref xmlns="http://xmlns.oracle.com/xref">
    <table name="INVENTORY_LOCATION_ID">
        <columns>
            <column name="EBIZ_01"/>
            <column name="SEBL_01"/>
        </columns>
        <rows>
            <row>
                <cell colName="EBIZ_01">204</cell>
                <cell colName="SEBL_01">88-5I54T</cell>
                <cell colName="PIM_01">204</cell>
            </row>
            <row>
                <cell colName="EBIZ_01">888</cell>
                <cell colName="SEBL_01">88-29CMT</cell>
                <cell colName="PIM_01">888</cell>
            </row>
            <row>
                <cell colName="EBIZ_01">911</cell>
                <cell colName="SEBL_01">88-30GX4</cell>
                <cell colName="PIM_01">911</cell>
            </row>
        </rows>
    </table>
</xref>

```

2. Run the **xrefimport** tool to import the cross-references.
3. Copy the files to a **temp** directory on the SOA server.
4. Telnet to the SOA Server and change directory to the **xrefimport** tool home

```
-bash-3.00$ cd $SOA_HOME/integration/esb/bin
```

5. Set these environment variables.

```

-bash-3.00$ export OC4J_USERNAME=oc4jadmin
-bash-3.00$ export OC4J_PASSWORD=welcome1
-bash-3.00$ export DB_USER=aia
-bash-3.00$ export DB_PASSWORD=aia
-bash-3.00$ export
DB_URL="jdbc:oracle:thin:@adc60119fems.us.oracle.com:1549:o2c2sysa"

```

6. Run the import for organization cross-reference by using this command:

```

bash-3.00$. xrefimport.sh -file ~/orginvsetup/organization.xml -
generate COMMON

```

7. Run the import for inventory location cross-reference by using this command:

```
bash-3.00$. xrefimport.sh -file
~/orginvsetup/Inventory_location_id.xml -generate COMMON
```

Validating Cross-References

To validate cross-references:

1. Log on 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.

```
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'))
```

3. Replace the value for the inventory locations and replace INVENTORY_LOCATION_ID to ORGANIZATION_ID for the operating units selected earlier. (The number of operating units and inventory locations depends on your set up.)

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. SyncItemCompositionListSiebelCommsProvABCSImpl
 - b. ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl

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

Identifying 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 other users can refer to it.

For more information about cross-references, see *Oracle Application Integration Architecture – Foundation Pack - Integration Developer's Guide* and *Oracle Cross Reference User Guide*.

These are the cross-references for Oracle Product Hub PIP:

For more information about cross-references for communications related products, see *Appendix A, Oracle Application Integration Architecture for Communications: Maintenance Update Guide*.

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.
SIEBELPRODUCTEVENTXREF	ITEM_ID_COMMON,LINEPRICECODE	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.

Describing 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, though administrators can add 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 you run any initial loads or initiate any incremental transactional flows.

These are the domain value mappings (DVMs) for the Oracle Product Hub PIP:

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_BOOLEAN	EBIZ_01,COMMON	DVM to convert Boolean value to number.
DATA_TYPE_CODE	EBIZ_01,COMMON, SEBL_01	It specifies the data type of the valueset
DAYS_OF_WEEK	EBIZ_01,COMMON, SEBL_01, PIM_01, JDEE1_01, P6EPPM_01	Indicates the days of the week as Monday, Tuesday, and so on.
ENTITY_TO_TARGET_APPLICATION	TARGET_ID, RECEIVE_PRICELIST_FLAG , RECEIVE_BOM_FLAG	List of entities 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_CURRENTFUTUREPAST_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,	Cross-reference value: InventoryPlanningCode.

DVM Type	DVM Column Name	Description
CODE	PIM_01,JDEE1_01	
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.
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 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 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.

DVM Type	DVM Column Name	Description
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_OVERRIDETYPECODE	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, CRM_01	Cross-reference value: product status.
RATEPLAN_TYPECODE	COMMON, SEBL_01, PIM_01	Indicates the type of rate plan associated with the charge of the product, for example, Single Rate Plan, Rate Plan Selector, Custom event analysis, and so on.
RATEPLAN_UNITCODE	COMMON, PIM_01	Indicates the rate plan unit code.
STRUCTURE_NAME_ALTERNATE_BOM	EBIZ_01,COMMON,PIM_01	Mapping between structure name and alternate BOM.
TARGET_ID	COMMON, UCM_01, PIM_01	It specifies the common IDs of the systems to which synchronization should be done. This DVM is used to map system defined in PIM for Publication and system defined in BSR.
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.

DVM Type	DVM Column Name	Description
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_PRIMARYCLASSIFICATIONCODE	COMMON, EBIZ_01	Used for mapping item catalog group code.
ITEM_BILLING_TYPE_CODE	EBIZ_01, COMMON, SEBL_01, PIM_01, AGILE_01, BRM_01	This DVM maps the Billing Type of a product, for example, subscription, item, system, and so on.
APPLIES_TO	PIM_01, COMMON, BRM_01, SEBL_01	This DVM is used 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.
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, CRM_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_PURCHASEPRORATIONCODE	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_CANCELCATIONPRO	PIM_01, COMMON, BRM_01,	Maps rate data cancelation proration code.

DVM Type	DVM Column Name	Description
RATIONCODE	SEBL_01	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_IMPACTCLASSIFICATIONCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the impact category associated with the rate, for example, Default, Premium, Standard, Free, and so on.
PLANIMPACT_UNITCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the scaled unit codes, for example, Second, Minute, Hour, and so on.
RATEDATA_DISCOUNTBRACKETCODE	PIM_01, COMMON, BRM_01, SEBL_01	Indicates the discount brackets to charge different amounts based on quantity, for example, Continuous, Rate Dependent, Resource, and so on.
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_TAXAPPLICABILITYCODE	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.

DVM Type	DVM Column Name	Description
ITEM_PROMOTION_DURATION	COMMON, SEBL_01, PIM_01	Indicates Promotion Duration. For example: New Duration.
PRICE_ADJUSTMENT_TYPE	COMMON, SEBL_01, PIM_01	Indicated Price Adjustment in Promotion Ex. Price Override, Discount Amount.

The considerations for setting up DVMs include the following:

- **STRUCTURE_NAME_ALTERNATE_BOM:** This is used to map the STRUCTURE_NAME in PIM 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 that is defined in Oracle Product Hub (PIM).

For every structure name that is defined in PIM, 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 PIM, 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.

- An entry should be made in this DVM even for primary BOM.

Handling Errors

Based on the roles that are defined for the services, email notifications are sent if there is an error. There are no AIA specific errors thrown by the process integration for product management services.

Based on the roles that are defined for the services, email notifications are sent if a service errors out.

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 Application Integration Architecture – Foundation Pack: Core Infrastructure Components Guide*, “Setting Up and Using Error Handling and Logging.”

Viewing EBO Implementation Maps (EIMs)

For more information about how services are mapped, see the My Oracle Support document: EBO Implementation Maps (EIMs) 881022.1.

Setting Configuration Properties

Set these properties in the **AIAConfigurationProperties.xml** file. The file is located in **<aia.home>/config/**.

Note: Whenever the **AIAConfigurationProperties.xml** file is updated, the file must be reloaded for updates to be reflected in the applications or services that use the updated properties. Click **Reload** on the **Configuration** page in the Oracle AIA Console to perform this reload. Alternatively, you can perform the reload by rebooting the server.

For more information about configuration, see the *Oracle Application Integration Architecture Core Components Guide, "Working with the BSR," Loading Oracle AIA Configuration Properties File Updates*.

Settings for SyncItemLEbizProvABCSImpl service

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 will select the default system ID from this configuration property.
Routing.SyncItemLEbizAdapter.CAVS.EndpointURI		This property sets the endpoint URL for the CAVS simulator.
Routing.SyncItemLEbizAdapter.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SyncItemLEbizAdapter.EBIZ_01.EndpointURI		Endpoint URI of the Oracle E-Business Suite adapter
Routing.ItemResponseEBSV2.SyncItemLEbizResponse.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.SyncItemLEbizResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.ItemResponseEBSV2.SyncItemLEbizResponse.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.

System Property	Value/Default Value	Description
ABCSExtension.PreInvokeSyncItemListABM	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.PostXformABMtoEBM	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

Settings for SyncProductSiebelProvABCServiceImpl service

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 will select 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.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.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.

Property Name	Value/Default Value	Description
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.SyncItemResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.ItemResponseEBSV2.SyncItemResponse.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.SyncItemResponse.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.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.PostXformABMtoABMEBM	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.PreInvokeSWIPPromotionIntegrationABM	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.PostInvokeSWIPPromotionIntegrationABM	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.

Property Name	Value/Default Value	Description
Product.Source.PIP	MDM/O2C	This property is updated during PIP installation. For MDM PIP, 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.
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.
LANGUAGE_CODES	Valid language code of the language in which the Description will be synced to Siebel	This value determines the language in which the Description will be synchronized from OPH to Siebel. You must make sure that only valid language codes as configured within the PIM/OPH system are used. If invalid language codes are specified, the description field will not be synchronized from PIM to Siebel. Also, make sure that the LANGUAGE_CODES property set within QueryItemListAdapter contains the language Code within the comma-separated list.

Settings for SyncBillOfMaterialsListEbizProvABCImpl service

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 will select 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.
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

Property Name	Value/Default Value	Description
		point is to be made, depending on whether it is true or false.

Settings for SyncBillOfMaterialsListSiebelProvABCSImpl service

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.SWProductIntegration.SEBL_01.EndpointURI		Siebel web service end-point location. This is a SOAP endpoint URL.
Routing.SWProductIntegration.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SWProductIntegration.CAVS.EndpointURI		This property sets the endpoint URL for the CAVS simulator.
Routing.SWPromotionIntegration.SEBL_01.EndpointURI		Siebel web service end-point location. This is a SOAP endpoint URL.
Routing.SWPromotionIntegration.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.SWPromotionIntegration.CAVS.EndpointURI		This property sets the endpoint URL for the CAVS simulator.
Routing.BillOfMaterialsResponseEB S.SyncBillOfMaterialsListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
Routing.BillOfMaterialsResponseEB S.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.BillOfMaterialsResponseEB S.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

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.
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>

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 will select 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.

Settings for SyncItemListItemReqABCImpl service

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 will select the default system ID from this configuration property.
Routing.QueryItemListItemAdapter.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
Routing.QueryItemListItemAdapter.PIM_01.EndpointURI		PIM web service end-point location. This is a SOAP endpoint URL.
Routing.QueryItemListItemAdapter.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		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		PIM web service end-point location. This is a SOAP endpoint URL.

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.PIM_01.EndpointURI		PIM web service end-point location. This is a SOAP endpoint URL.
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

System Property	Value/Default Value	Description
		true or false.
ABCSExtension. PreInvokePublicationServiceItemLi tEBM	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. PreInvokePriceListEBSSyncItemLi EBM	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. PreInvokePublicationServicePriceli stEBM	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. PreXformABMtoEBMIBillOfMaterials ListABM	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.PreInvokeBillOfMat erialsListEBSSyncBOMListEBM	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.PostXformEBMtoAB MBOMListEBM	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.PreInvokePublicati onServiceBOMListEBM	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_T ARGETS		This property controls number of parallel targets to be processed for Item when synchronizing

System Property	Value/Default Value	Description
		items to multiple targets.
NUMBER_OF_PARALLEL_BOM_TARGETS		This property controls number of parallel targets to be processed for BOMs to multiple targets.

Settings for SyncItemListBRMProvABCSImpl service

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.BRMPRICINGService_ptt.RouteToCAVS	false/true	Determines whether to route the endpoint URL to either the end application service or CAVS for simulating the service.
Routing.BRMPRICINGService_ptt.CAVS.EndpointURI	NA	Sets the endpoint URL for the CAVS simulator.
Routing.BRMPRICINGService_ptt.BRM_01.EndpointURI	eis/BRM1	Sets the endpoint URL for the target application.
Routing.BRMPRICINGService_ptt.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the environment code element to the value PRODUCTION.
ABCSExtension.PreInvokeABS	false/true	Used as an extension point before application business service is invoked. This property determines whether to make invocation of service at the extension point, depending on 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.

Settings for SyncPriceListListBRMProvABCImpl service

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 select 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.BRMPRICINGService_ptt.RouteToCAVS	NA	Determines whether to route the EndpointURI to either the end application service or CAVS for simulating the service.
Routing.BRMPRICINGService_ptt.CAVS.EndpointURI	NA	Sets the EndpointURI for the CAVS simulator
Routing.BRMPRICINGService_ptt.BRM_01.EndpointURI	eis/BRM1	Sets the EndpointURI for the target application
Routing.BRMPRICINGService_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.
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.

Settings for QueryBillOfMaterialsListPIMAdapter service

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

System Property	Value/Default Value	Description
		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.Endpoint URI		This property sets the endpoint URL for the CAVS simulator.
Routing.ItemService.PIM_01.EndpointURI		PIM web service end-point location. This is a SOAP endpoint URL.

Settings for QueryItemListPIMAdapter service

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)
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.Endpoint URI		This property sets the endpoint URL for the CAVS simulator.
Routing.ItemService.PIM_01.EndpointURI		PIM web service end-point location. This is a SOAP endpoint URL.
LANGUAGE_CODES	Comma separated list of valid language codes	This value determines the list of languages that will be synced from OPH to the target system(s). Typically, language codes are a comma separated list of values. You must make sure that only the valid language codes as configured within the PIM/OPH system are used. If invalid language codes are specified, the description and long description fields will not be synchronized from PIM to the target system(s)

Settings for SyncSpecificationValueSetListPIMReqABCImpl service

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.PreInvokeEBSSpecificationValueSetListEBM	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.PostInvokeEBSSpecificationValueSetListEBM	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.
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		PIM web service end-point location. This is a SOAP endpoint URL.
Routing.ValueSetService.PIM_01.EndpointURI		PIM web service end-point location. This is a

System Property	Value/Default Value	Description
pointURI		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.

Settings for SyncItemCatalogCategoryPIMReqABCImpl service

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.
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		PIM web service end-point location. This is a SOAP endpoint URL.
Routing.PublicationService.PIM_01.EndpointURI		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.

System Property	Value/Default Value	Description
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.
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

Settings for SyncClassificationSchemeListSiebelProvABCSImpl service

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 will select 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

System Property	Value/Default Value	Description
		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.
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

Settings for SyncSpecificationValueSetListSiebelProvABCImpl service

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 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.

System Property	Value/Default Value	Description
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.SWIAAttributeImport.RouteToCAVS	false/true	Determines whether the endpoint URL should be routed to either the end application service or CAVS for simulating the service.
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.

System Property	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	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.

Settings for ProductOptimizedSyncPriceListListSiebelCommsProvABCSImpl service

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	No default value.	Siebel PriceList web service end point location. This is a SOAP end point URL. If

System Property	Value/Default Value	Description
e.EndpointURI		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.
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.

Settings for SyncItemCompositionListSiebelCommsProvABCSImpl service

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.

System Property	Value/Default Value	Description
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.
Routing.SiebelProductService.SEBL_01.EndpointURI	http://\$<http.hostname>:\$<http.port>/eai_enus/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/asyncresponsesimulator	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

System Property	Value/Default Value	Description
		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 EBMto 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).

Settings for SyncItemListBRMProvABCSImpl service

Property Name	Property Value	Description
Default.SystemID	BRM_01	It is the responsibility of the application to send the SystemID from which the request is being sent. If any requestor application fails to send this, AIA will select the default SystemID from this config property.
ABCSExtension.PreProcessABM false	false/true	This property is used as an extension point before the ABM is transformed to EBM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false
ABCSExtension.PreProcessEBM	false/true	This property is used as an extension point after the ABM is transformed to EBM. It determines invocation of service at the extension point is to be made or not, depending on whether it is true

		or false
Transformation.EnableExtensions	false/true	Property used for determining enabling XSL 2.0 extensions in the Transformations.
Routing.ItemEBSV2.SyncItemList.RouteToCAVS	false/true	Determines whether the EndpointURI should be routed to either the end application service or CAVS for simulating the service.
Routing.ItemEBSV2.SyncItemList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION/CAVS	Sets the EnvironmentCode element to the a value PRODUCTION
Routing.ItemEBSV2.SyncItemList.CAVS.EndpointURI		This property sets the EndpointURI for the CAVS simulator

Settings for SyncPriceListListBRMProvABCSImpl service

Property Name	Property Value	Description
Default.SystemID	BRM_01	It is the responsibility of the application to send 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.PreProcessEBM	false/true	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not, depending on whether it is true or false
ABCSExtension.PreProcessABM	false/true	This property is used as an extension point After the EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false
ABCSExtension.PostProcessABM	false/true	This property is used as an extension point after the response ABM is received from the provider and before it is transformed to response EBM. It determines invocation of service at the extension point is to be made or not, depending on whether it is true or false
ABCSExtension.PostProcessEBM	false/true	This property is used as an extension point after the response ABM is received from the provider and before it is transformed to response EBM. It determines invocation of service at the extension point is to be made or not, depending on whether it is true or false
Transformation.EnableExtensions	false/true	Property used for determining enabling XSL 2.0 extensions in the Transformations.
Routing.BRMProduct.RouteToCAVS	false/true	Determines whether the EndpointURI should be routed to either the end application service or

Property Name	Property Value	Description
		CAVS for simulating the service.
Routing. BRMProduct.CAVS.EndpointURI		This property sets the EndpointURI for the CAVS simulator
Routing. BRMProduct.BRM_01.EndpointURI		This property sets the EndpointURI for the target application
Routing. BRMProduct.MessageProcessingInstr uction.EnvironmentCode	PRODUCTION	Sets the EnvironmentCode element to the a value PRODUCTION

Setting Up the Data for Seeded Attributes of PIM Telco Library

These entities have to be set up in target applications, and the corresponding attribute valueset must be updated with the values in PIM.

Attribute Group Name	Attribute Name
Product Details: Definitions	Product Line (Siebel)
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)

The following table lists the attributes and respective DVMs in which the values for these attributes must be added.

Attribute Name	DVM
GLID	GL_CODE

Provisioning Tag	PROVISIONING_TAG
Impact Category	PRICE_IMPACTCLASSIFICATIONCODE

Setting up List of Values in Siebel

Once the OPH PIP 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 PIM to Siebel that are associated with multiple inventory organizations that in turn are associated with the same operating unit?**

No. Uniqueness of a product in Siebel is based on a combination of product, business unit, and vendor. Therefore, if such synchronization happens, the key does not remain unique.

- Q **Why setup only the cross-references for inventory location associated with item validation organization for Siebel?**

Item synchronization to Siebel is based on the requirement that Siebel is able to use that item as part of order capture in the O2C PIP. In Oracle E-Business Suite, order can refer to products that are part of item validation organization only. Therefore, it is required to populate the cross-references for inventory location only with item validation organization.

- Q **How to extend the Siebel provider service to consume UDA published by Oracle E-Business Suite?**

For SyncProductSiebelProvABCSImpl and SyncBillOfMaterialsListSiebelProvABCSImpl, set the flag enableCustomExtensions in XformSyncItemEBMTToSiebelProdABM_Custom_UDA.xsl or XformSyncBillOfMaterialsListEBMTToSiebelProdABM_Custom_UDA.xsl 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 BOM. Sales BOM needs to be identify by Sales as type, if this is change the routing rules needs to be changed to reflect the new value, since this value determines if the product get routed to Siebel or not.

- Q **Does the service support sharing of inventory organization across multiple operating units?**

No. This release of the PIP 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?**

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?**

See: [Setting Configuration Properties](#).

- Q **When invoking PIM service, the PIM errors out with a message “No Access to Method [Method Name]”.**

To resolve this, here is a workaround:

- a. Remove the existing grants to service methods on **Irep**.
 - b. Open **Functional Administrator**, **Core Services**, **Caching Framework**, Select **Total Cache Components**, Select **All**.
 - c. Click **Clear all cache**.
 - d. Re-create grants.
 - e. Click **Redeploy** on **Irep** service page to redeploy the service.
- Q **When invoking PIM services multiple times, the subsequent invocation may take long time and the service request ends with a SocketTimeout error**

To resolve this, here is a workaround:

- a. Open **Integrated SOA Gateway**, Open **SOA Monitor**.
- b. Click **Purge**.
- c. Click **Turn off Web Service Monitor**.

Appendix A

This table depicts the support for rate tiers and effectivity.

	Rate Defined in PIM?	Rate Synchro nized to BRM?	Rate synchro nized to	Revision Defined in PIM?	Revision synchro nized to BRM?	Revision synchro nized to Siebel?	Effectivit y Defined in PIM? (absolute)	Effectivit y synchro nized to BRM?	Absolute Effectivit y synchro nized to	Relative Effectivit y synchro nized to Siebel?
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	No *2	No	No	Yes	No	Yes	Yes	No	Yes	N/A
Product	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	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 absolute)	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 absolute)	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, refer *Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Order to Bill 2.5 - 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.

Appendix B: User Defined Attributes Framework

This appendix includes the following sections:

- [Extending for Custom UDAs](#)
- [Implementation of UDA Transformation](#)
- [Performance Optimization Using Chunking](#)

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'](#).

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.

Mapping Artifacts

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

Mapping Operational Attributes to ItemEBM

Artifacts:

- XformItemResponsePIMABMTToSyncItemListEBM.xml
- XformItemListResponsePIMABMTToSyncItemListEBM_Custom.xml

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

Mapping Seeded Set of UDAs for Communications library to EBM

Artifacts:

- XformUDAAGInterfaceCommsPIMBOMRespABMTToSyncBOMEBM.xml

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

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

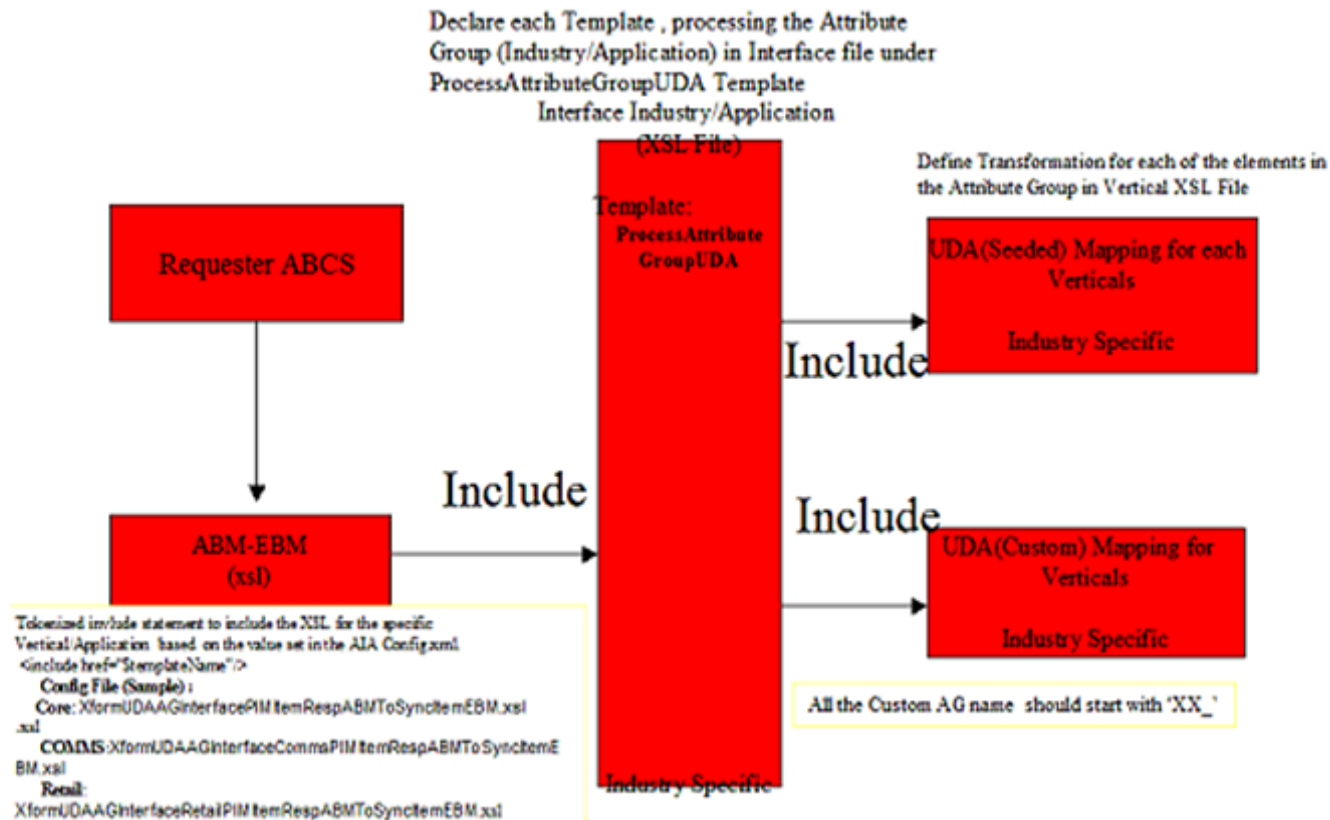
The mappings for UDAs are defined in the following xsl for each of the attributes within the attribute group:

- 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.

Implementation of UDA Transformation

The following figure illustrates the UDAs to first class attributes (FCA) transformation in ItemEBM:



Adding UDA to an FCA in Item EBM

Note: We do not recommend you 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. The following table shows an example of a UDA Attribute Group (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. The following figure provides an example about how to define transformations for each of the elements in the attribute group in the XformUDACommsPIMItemRespABMToSyncItemEBM.xsl.

```

<xsl:template name=" ItemManufactureAttrGen"
<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>

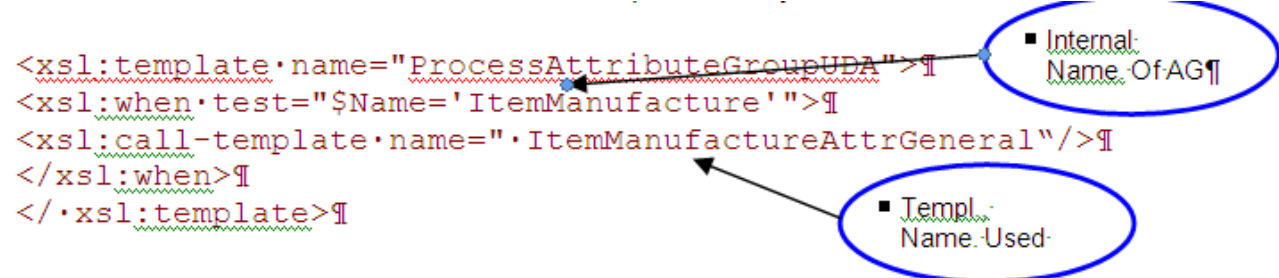
```

Templ.
Name. Ref

Internal name
of the
Attribute.

Code Sample

3. Declare each template, which processes the attribute group in interface file, under ProcessAttributeGroupUDA template. The following figure shows a code sample.



```

<xsl:template name="ProcessAttributeGroupUDA">␣
<xsl:when test="$Name='ItemManufacture'">␣
<xsl:call-template name="ItemManufactureAttrGeneral"/>␣
</xsl:when>␣
</xsl:template>␣

```

Internal Name Of AG

Templ. Name Used

Code Sample

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 (XformRestructureSyncItemEBM.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.

```

<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>
      </xsl:for-each>
    </ns0:SyncItemListEBM>
  </template>

```

Code Sample

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

Mapping Custom UDAs to EBM

Note: We recommend you to apply this 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.

Artifacts (Requester ABCS)

The following artifacts can be found under PIPS\Core\PIM\RequestorABCS\SyncItemListItemReqABCSImpl\bpel where the integrations have been downloaded and are installed:

- XformUDAAGInterfaceCommsPIMBOMRespABMToSyncBOMEBM.xml
- XformUDAAGInterfaceCommsPIMItemRespABMToSyncItemListItemEBM.xml
- XformUDACommsPIMBOMRespABMToSyncBOMEBM_Custom.xml
- XformUDACommsPIMItemRespABMToSyncItemListItemEBM_Custom.xml
- XformRestructureSyncBillOfMaterialsListEBM.xml
- XformRestructureSyncItemListItemEBM.xml

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.

- XformSyncItemListItemEBMToSiebelProdABM.xml
- XformUDAInterfaceCommsSiebelPromotionEBMToABM.xml
- XformUDACommsSiebelProductEBMToABM.xml
- XformUDACommsSiebelProductEBMToABM_Custom.xml

- XformRestructureProductABM.xsl
- XformSyncItemEBMTToSiebelPromotionABM.xsl
- XformUDAInterfaceCommsSiebelProductEBMTToABM.xsl
- XformUDACommsSiebelPromotionEBMTToABM.xsl
- XformUDACommsSiebelPromotionEBMTToABM_Custom.xsl
- XformRestructurePromotionABM.xsl

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.

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 QueryItemPIMAdapter. 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

Appendix C: Cross References for the Process Integration for Product Management

This appendix describes the cross references used in the Process Integration for Product Management. This includes the following sections:

- Cross Reference Values
- Integration Solution Cross References
- Product Synchronization Flow
- Discount Synchronization Flow

Cross Reference Values

The values below denote the entries made to the cross reference (xref) table and what they mean.

- **ITEM_ID:** Cross-references the BRM (Portal) ProductID and the Siebel ProductID.

COMMON: Auto generated GUID.

PORTAL_01: POID of BRM Product ABM.

SEBL_01: ProductID of Siebel Product ABM.

- **PRICELINE_ID:** Cross-references the BRM (Portal) Product ID to Siebel PriceLineID. Also links to the COMMON of ITEM_ID.

COMMON: Auto generated GUID.

PORTAL_01: POID of BRM Product ABM.

SEBL_01: Siebel PriceListItemID for the main product.

ITEM_ID_COMMON: From ITEM_ID.COMMON.

- **PRICELINETYPE_ID**: Cross-references BRM (Portal) Product's Events to Siebel PriceLineID. Also links to the COMMON of ITEM_ID.

COMMON: Auto generated GUID.

PORTAL_01: POID of BRM Product ABM + Event Name.

SEBL_01: Siebel PriceListItemID for the event product.

ITEM_ID_COMMON: From ITEM_ID.COMMON.

- **SIEBELPRODUCTEVENTXREF**: Cross -references BRM (Portal) Product's Event that is Associated with the main product in Siebel. This xref is used only in the Product Optimized Sync.

ITEM_ID_COMMON: From ITEM_ID.COMMON

LINEPRICETYPECODE: PRICELINETYPE_ID.COMMON

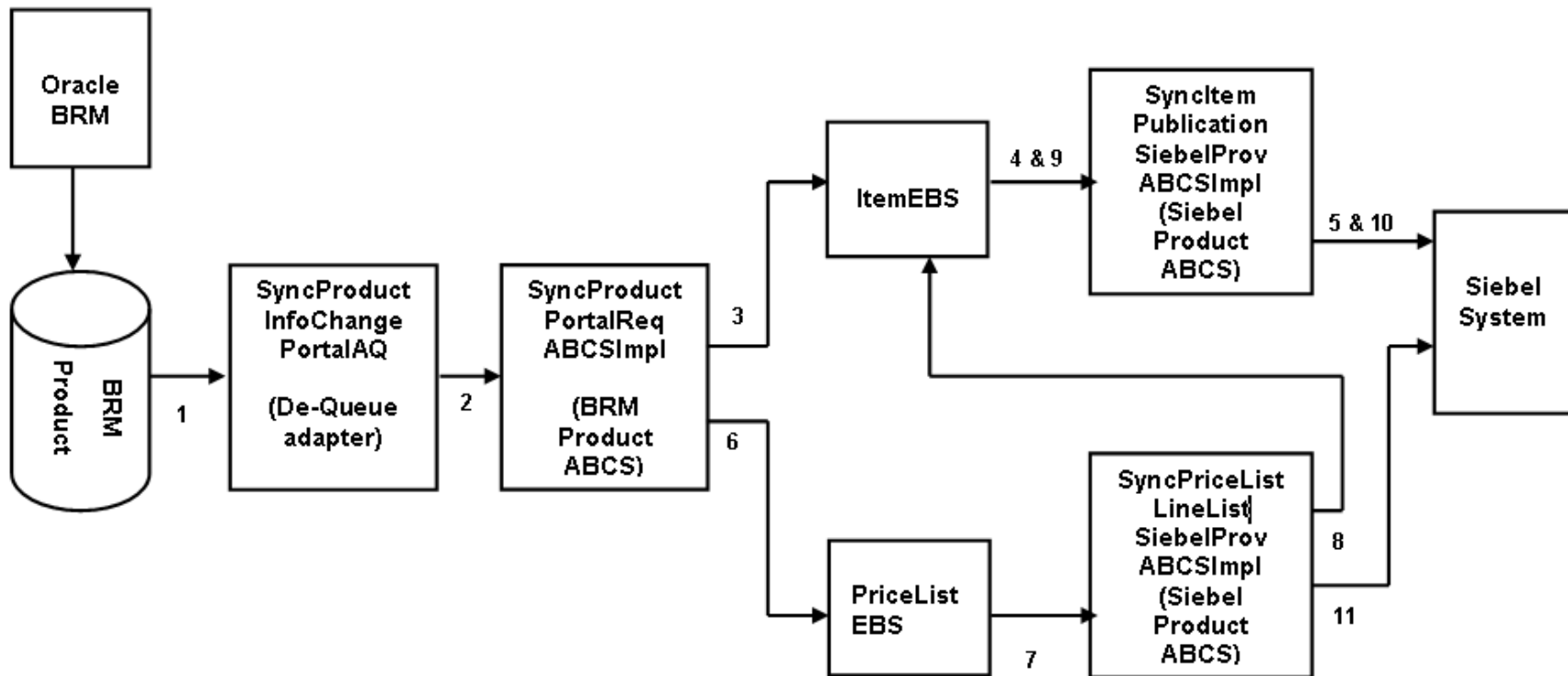
Integration Solution Cross References

The following table lists the cross references:

Operation	Entity	Siebel CRM ID	Oracle BRM ID
Inserts/Refers	ITEM_ID	Product Id	Product Id
Inserts/Refers	PRICELINE_ID (Main Products only)	Price Line Id to Common ITEM_ID of main product	Product Id
Inserts/Refers	PRICELINETYPE_ID (For Event/special type products)	Price Line Id and to Common ITEM_ID (both only if exists, may not be in case of optimized mode)	Generated Product Id for Event products (ProductIDEvent Name)
Inserts/Refers	SIEBELPRODUCTEVENTXREF (Used in Optimization mode only)	Common ITEM_ID for the parent product to Common PRICELINETYPE_ID for event product	

Product Synchronization Flow

This diagram shows the product synchronization flow.



Product synchronization flow

1. Before the call 3 which SyncProductPortalReqABCSImpl makes to ItemEBS, the following entries are made in to XREF_DATA table: ITEM_ID, COMMON, POID of BRM product.

XREF_TABLE_NAME	XREF_COLUMN_NAME
ITEM_ID	COMMON
ITEM_ID	<<POID of BRM product>>

2. During the response back from Siebel to SyncItemPublicationSiebelProvABCSImpl an entry is made to the XREF_DATA table with following value: ProductID of Siebel Product.

XREF_TABLE_NAME	XREF_COLUMN_NAME
ITEM_ID	<<ProductID of Siebel product>>

3. Before the call 6 from SyncProductPortalReqABCSImpl to PriceListEBS is made, the following entries are made in to XREF_DATA table: PRICELINE_ID, COMMON, POID of BRM product, PRICELINETYPE_ID.

XREF_TABLE_NAME	XREF_COLUMN_NAME
PRICELINE_ID	COMMON
PRICELINE_ID	POID of BRM product
PRICELINE_TYPE	COMMON
PRICELINE_TYPE	POID of BRM product

4. During the response from SyncItemPublicationSiebelProvABCSImpl for the call 8, following entries are made to XREF_DATA table: ITEM_ID, COMMON, ProductID of Siebel Product

XREF_TABLE_NAME	XREF_COLUMN_NAME
ITEM_ID	COMMON
ITEM_ID	<<POID of Siebel product>>

Note: For Simple Products' sync this call is not made since the main product is already synced as an Item

5. Before the call 11 from SyncPriceListLineListSiebelProvABCSImpl to Siebel System is made, following entries are made to the XREF_DATA table: PRICELINETYPE_ID, COMMON, ITEM_ID_COMMON, PRICELINE_ID.

XREF_TABLE_NAME	XREF_COLUMN_NAME
PRICELINETYPE_ID	COMMON
PRICELINETYPE_ID	ITEM_ID_COMMON
PRICELINE_ID	COMMON

PRICELINE_ID	ITEM_ID_COMMON
PRICELINE_ID	ITEM_ID_COMMON

6. During the response back from Siebel System, SyncPriceListLineListSiebelProvABCSImpl entries are made to the XREF_DATA table with following values: Siebel PriceListItemID for the main product and PriceListItemID for the event product.

XREF_TABLE_NAME	XREF_COLUMN_NAME
PRICELINE_ID	<<ProductID of Siebel product>>
PRICELINETYPE_ID	<<ProductID of Siebel product>>

Note: For Simple Products' sync this call is not made since the main product is already synced as an Item

XREF_TABLE_NAME	XREF_COLUMN_NAME
PRICELINE_ID	<<ProductID of Siebel product>>

In the case of the Optimized Product Synchronization flow, the SyncPriceListLineListSiebelProvABCSImpl is replaced with the ProductOptimizedSyncPriceListLineListSiebelProvABCSImpl by changing the routing rule in PriceListEBS. In this case, steps 7, 8, and 9 are applicable instead of steps 5 and 6.

7. Before the call 8 from ProductOptimizedSyncPriceListLineListSiebelProvABCSImpl to ItemEBS is made, the following entries are made in to XREF_DATA table: SIEBELPRODUCTEVENTXREF, ITEM_ID_COMMON, LINEPRICETYPECODE

XREF_TABLE_NAME	XREF_COLUMN_NAME
SIEBELPRODUCTEVENTXREF	ITEM_ID_COMMON
SIEBELPRODUCTEVENTXREF	LINEPRICETYPECODE

8. Before the call 11 from ProductOptimizedSyncPriceListLineListSiebelProvABCSImpl to Siebel System is made, following entries are made to the XREF_DATA table: PRICELINE_ID, COMMON, ITEM_ID_COMMON, PRICELINETYPE_ID.

XREF_TABLE_NAME	XREF_COLUMN_NAME
PRICELINE_ID	COMMON
PRICELINE_ID	ITEM_ID_COMMON
PRICELINETYPE_ID	COMMON
PRICELINETYPE_ID	ITEM_ID_COMMON

9. During the response back from Siebel System, ProductOptimizedSyncPriceListLineListSiebelProvABCSImpl entries are made to the XREF_DATA table with following values: Siebel PriceListItemID for the main product and PriceListItemID for the event product.

XREF_TABLE_NAME	XREF_COLUMN_NAME
PRICELINE_ID	<<ProductID of Siebel product>>
PRICELINETYPE_ID	<<ProductID of Siebel product>>

Examples

There are two examples, one for a simple product synchronization, and one for a complex product synchronization.

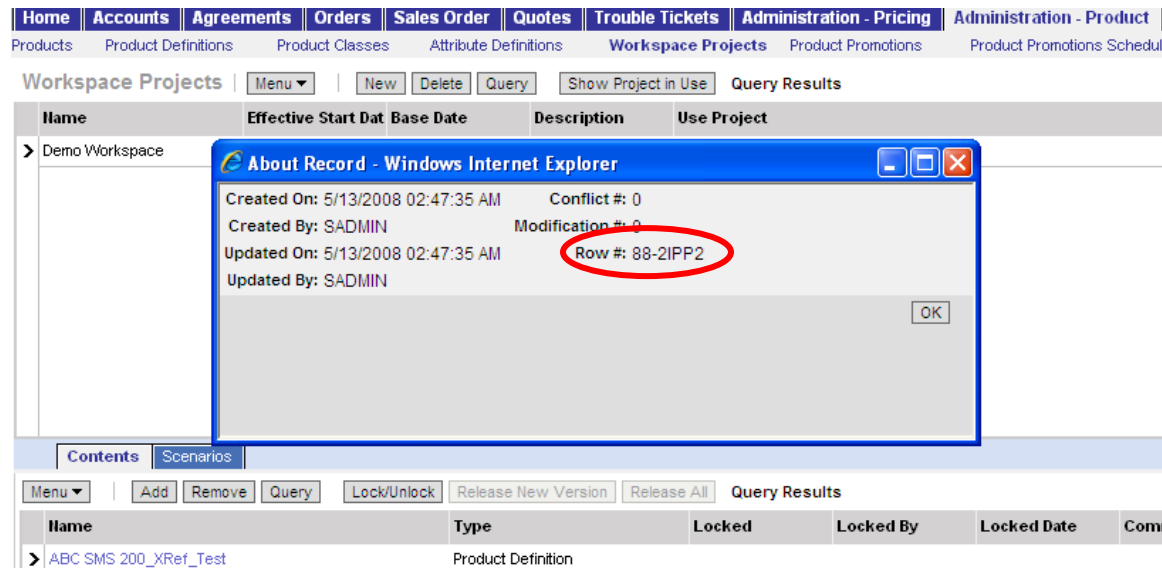
Example 1

Let us consider an actual scenario where a simple product is being synchronized from BRM to CRM.

1. Creating products in BRM to be synchronized with Siebel:



2. Verifying the synchronized records in Siebel:



3. The snapshots below show how the data is entered into the xref table and corresponds with points 1 to 8 described above:

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ID	PORTAL_01	<<ROWNUM_1>>	<<PRT_PROD_01>>
2	ITEM_ID	COMMON	<<ROWNUM_1>>	<<COMMON_PROD_01>>

Table corresponding to point 1

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ID	SEBL_01	<<ROWNUM_1>>	<<CRM_PROD_01>>

Table corresponding to point 2

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	PRICELINE_ID	PORTAL_01	<<ROWNUM_2>>	<<PRT_PROD_01>>
2	PRICELINE_ID	COMMON	<<ROWNUM_2>>	<<COMMON_PRICE_ID1>>
3	PRICELINETYPE_ID	PORTAL_01	<<ROWNUM_3>>	<<PRT_PROD_01_EVENT1>>
4	PRICELINETYPE_ID	COMMON	<<ROWNUM_3>>	<<COMMON_PRICETYPE_ID1>>

Table corresponding to point 3

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	PRICELINE_ID	ITEM_ID_COMMON	<<ROWNUM_2>>	<<COMMON_PROD_01>>
2	PRICELINE_ID	SEBL_01	<<ROWNUM_2>>	<<CRM_PRICE_01>>

Table corresponding to point 5

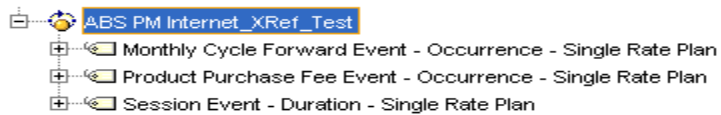
Table showing complete entry for the product sync

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ID	PORTAL_01	<<ROWNUM_1>>	<<PRT_PROD_01>>
2	ITEM_ID	COMMON	<<ROWNUM_1>>	<<COMMON_PROD_01>>
3	ITEM_ID	SEBL_01	<<ROWNUM_1>>	<<CRM_PROD_01>>
4	PRICELINE_ID	PORTAL_01	<<ROWNUM_2>>	<<PRT_PROD_01>>
5	PRICELINE_ID	COMMON	<<ROWNUM_2>>	<<COMMON_PRICE_ID1>>
6	PRICELINETYPE_ID	PORTAL_01	<<ROWNUM_3>>	<<PRT_PROD_01_EVENT1>>
7	PRICELINETYPE_ID	COMMON	<<ROWNUM_3>>	<<COMMON_PRICETYPE_ID1>>
8	PRICELINE_ID	ITEM_ID_COMMON	<<ROWNUM_2>>	<<COMMON_PROD_01>>
9	PRICELINE_ID	SEBL_01	<<ROWNUM_2>>	<<CRM_PRICE_01>>

Example 2

Now let us consider the scenario where a Complex product is being synced from BRM to CRM.

1. Creating products in BRM to be synchronized with Siebel:



2. Verifying the synchronized records in Siebel:

Products | Product Definitions | Product Classes | Attribute Definitions | Workspace Projects | Product Promotions | Product Promotions Schedule | Eligibility and Compatibility Matrices | Product Catalog | Product Lines | Product Features | External Products

Products | Menu | New | Delete | Query | Release

Name	Service Type	Description	Product Line	Locked By	Check Eligibility	Inclusive Eligibility	Revision	Status	Compo
> ABS PM Internet_XRef_Test		ABS PM Internet.							
ABS PM Internet_XRef_Test-CFM		Monthly Cycle Forward Event							
ABS PM Internet_XRef_Test-PURCHASE		Product Purchase Fee Event							

About Record - Windows Internet Explorer

Created On: 3/11/2008 09:53:19 PM Conflict #: 0
 Created By: SADMIN Modification #: 0
 Updated On: 3/11/2008 09:53:30 PM Row #: 88-26VP7
 Updated By: SADMIN
 Last Updated Source: Object Manager - Default
 Last Updated On: 3/11/2008 09:53:31 PM

About Record - Windows Internet Explorer

Created On: 3/11/2008 09:53:25 PM Conflict #: 0
 Created By: SADMIN Modification #: 0
 Updated On: 3/11/2008 09:53:25 PM Row #: 88-26VQ2
 Updated By: SADMIN
 Last Updated Source: Object Manager - Default
 Last Updated On: 3/11/2008 09:53:26 PM

About Record - Windows Internet Explorer

Created On: 3/11/2008 09:53:24 PM Conflict #: 0
 Created By: SADMIN Modification #: 0
 Updated On: 3/11/2008 09:53:24 PM Row #: 88-26VPR
 Updated By: SADMIN
 Last Updated Source: Object Manager - Default
 Last Updated On: 3/11/2008 09:53:25 PM

3. The snapshots below show how the data is entered into the xref table and corresponds with points 1 to 8 described above:

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ID	PORTAL_01	<<ROWNUM_1>>	<<PRT_PROD_01>>
2	ITEM_ID	COMMON	<<ROWNUM_1>>	<<COMMON_PROD_01>>

Table corresponding to point 1

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ID	SEBL_01	<<ROWNUM_1>>	<<CRM_PROD_01>>

Table corresponding to point 2

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	PRICELINE_ID	PORTAL_01	<<ROWNUM_2>>	<<PRT_PROD_01>>
2	PRICELINE_ID	COMMON	<<ROWNUM_2>>	<<COMMON_PRICE_01>>
3	PRICELINETYPE_ID	PORTAL_01	<<ROWNUM_3>>	<<PRT_PROD_01_EVENT1>>
4	PRICELINETYPE_ID	COMMON	<<ROWNUM_3>>	<<COMMON_PRICETYPE_01>>
5	PRICELINETYPE_ID	PORTAL_01	<<ROWNUM_4>>	<<PRT_PROD_01_EVENT2>>
6	PRICELINETYPE_ID	COMMON	<<ROWNUM_4>>	<<COMMON_PRICETYPE_02>>

Table corresponding to point 3

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ID	COMMON	<<ROWNUM_5>>	<<COMMON_PRICETYPE_01>>
2	ITEM_ID	SEBL_01	<<ROWNUM_5>>	<<CRM_PROD_02>>
3	ITEM_ID	COMMON	<<ROWNUM_6>>	<<COMMON_PRICETYPE_02>>
4	ITEM_ID	SEBL_01	<<ROWNUM_6>>	<<CRM_PROD_03>>

Table corresponding to point 4

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	PRICELINETYPE_ID	ITEM_ID_COMMON	<<ROWNUM_3>>	<<COMMON_PRICETYPE_01>>
2	PRICELINETYPE_ID	ITEM_ID_COMMON	<<ROWNUM_4>>	<<COMMON_PRICETYPE_02>>

3	PRICELINE_ID	ITEM_ID_COMMON	<<ROWNUM_2>>	<<COMMON_PROD_01>>
---	--------------	----------------	--------------	--------------------

Table corresponding to point 5

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	PRICELINETYPE_ID	SEBL_01	<<ROWNUM_3>>	<<CRM_ITEM_PRICE_01>>
2	PRICELINETYPE_ID	SEBL_01	<<ROWNUM_4>>	<<CRM_ITEM_PRICE_02>>
3	PRICELINE_ID	SEBL_01	<<ROWNUM_2>>	<<CRM_MAIN_PRICE>>

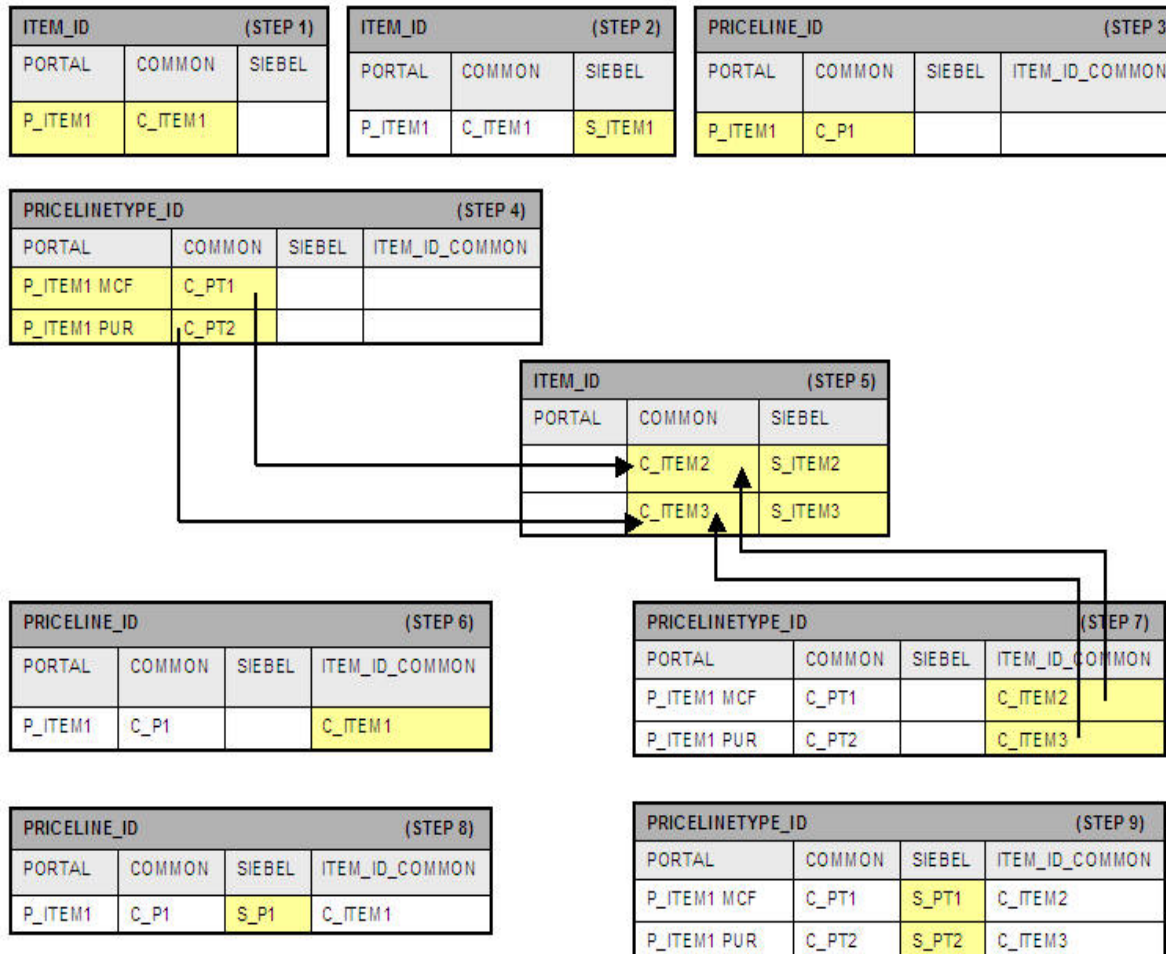
Table corresponding to point 6

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
1	ITEM_ID	PORTAL_01	<<ROWNUM_1>>	<<PRT_PROD_01>>
2	ITEM_ID	COMMON	<<ROWNUM_1>>	<<COMMON_PROD_01>>
3	ITEM_ID	SEBL_01	<<ROWNUM_1>>	<<CRM_PROD_01>>
4	PRICELINE_ID	PORTAL_01	<<ROWNUM_2>>	<<PRT_PROD_01>>
5	PRICELINE_ID	COMMON	<<ROWNUM_2>>	<<COMMON_PRICE_01>>
6	PRICELINETYPE_ID	PORTAL_01	<<ROWNUM_3>>	<<PRT_PROD_01_EVENT1>>
7	PRICELINETYPE_ID	COMMON	<<ROWNUM_3>>	<<COMMON_PRICETYPE_01>>
8	PRICELINETYPE_ID	PORTAL_01	<<ROWNUM_4>>	<<PRT_PROD_01_EVENT2>>
9	PRICELINETYPE_ID	COMMON	<<ROWNUM_4>>	<<COMMON_PRICETYPE_02>>

RESULTS:				
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE
10	ITEM_ID	COMMON	<<ROWNUM_5>>	<<COMMON_PRICETYPE_01>>
11	ITEM_ID	SEBL_01	<<ROWNUM_5>>	<<CRM_PROD_02>>
12	ITEM_ID	COMMON	<<ROWNUM_6>>	<<COMMON_PRICETYPE_02>>
13	ITEM_ID	SEBL_01	<<ROWNUM_6>>	<<CRM_PROD_03>>
14	PRICELINETYPE_ID	ITEM_ID _COMMON	<<ROWNUM_3>>	<<COMMON_PRICETYPE_01>>
15	PRICELINETYPE_ID	ITEM_ID _COMMON	<<ROWNUM_4>>	<<COMMON_PRICETYPE_02>>
16	PRICELINE_ID	ITEM_ID _COMMON	<<ROWNUM_2>>	<<COMMON_PROD_01>>
17	PRICELINETYPE_ID	SEBL_01	<<ROWNUM_3>>	<<CRM_ITEM_PRICE_01>>
18	PRICELINETYPE_ID	SEBL_01	<<ROWNUM_4>>	<<CRM_ITEM_PRICE_02>>
19	PRICELINE_ID	SEBL_01	<<ROWNUM_2>>	<<CRM_MAIN_PRICE>>

Table showing complete entry for the product sync

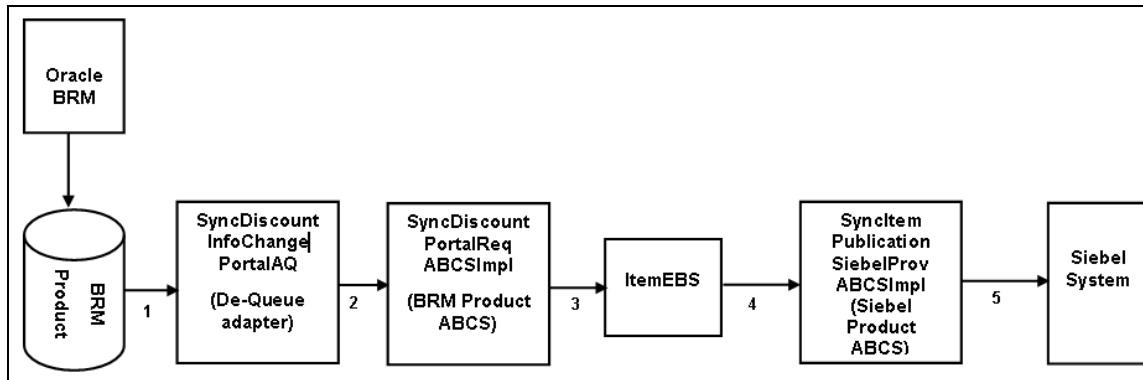
The figure below shows a bird's eye view of how the mappings are maintained in the xref table:



Cross reference table

Discount Synchronization Flow

This diagram shows the discount synchronization flow. Each step is described in detail.



Discount synchronization flow

Example

- Before the call 3 which SyncDiscountPortalReqABCSImpl makes to ItemEBS, the following entries are made in to XREF_DATA table: ITEM_ID, COMMON, POID of BRM product.

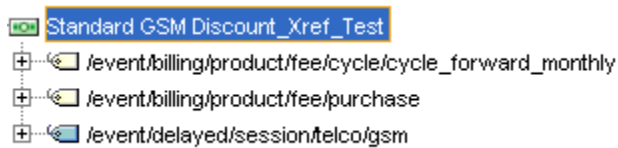
XREF_TABLE_NAME	XREF_COLUMN_NAME
ITEM_ID	COMMON
ITEM_ID	<<POID of BRM product>>

- During the response back from Siebel system, SyncItemPublicationSiebelProvABCSImpl an entry is made to the XREF_DATA table with following value: ProductID of Siebel Product.

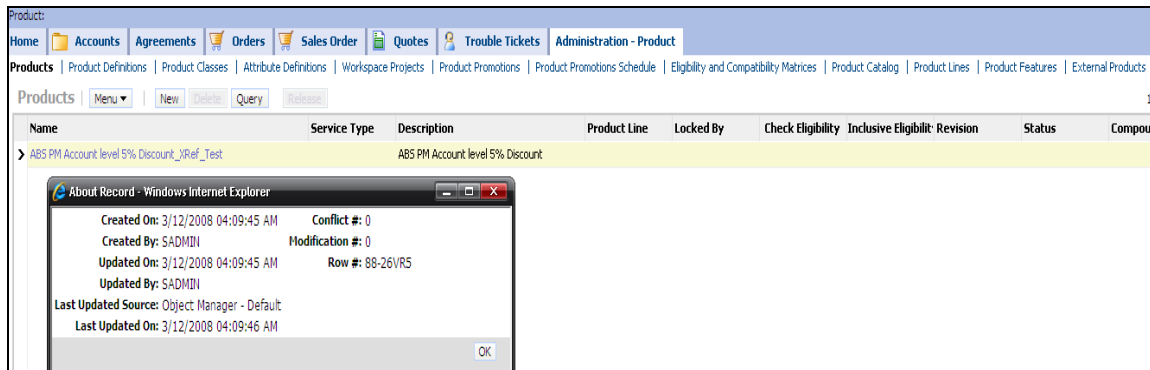
XREF_TABLE_NAME	XREF_COLUMN_NAME
ITEM_ID	<<ProductID of Siebel Product>>

Now let us consider an actual scenario where a discount is being synced from BRM to CRM.

- Creating Discount in BRM to be synced with Siebel:



- Verifying the synced records in Siebel:



- The snapshots below shows how data is entered into xref table corresponding to the points 1 and 2 explained above.

Results:							
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE	IS_DELETED	LAST_MODIFIED	LAST_ACCESSED
1	ITEM_ID	PORTAL_01	2E60E990F02D11DCBFCAF1F293F06D61	0.0.0.1 /discount 600048 0	N	1205323775657	1205323775657
2	ITEM_ID	COMMON	2E60E990F02D11DCBFCAF1F293F06D61	2d313734373134383431383534303233	N	1205323775657	1205323775657

Figure corresponding to point 1

Results:							
	XREF_TABLE_NAME	XREF_COLUMN_NAME	ROW_NUMBER	VALUE	IS_DELETED	LAST_MODIFIED	LAST_ACCESSED
1	ITEM_ID	SEBL_02	2E60E990F02D11DCBFCAF1F293F06D61	88-26VR5	N	1205323788212	1205323788212

Figure corresponding to point 2

Index

- Adding Target Instance, 95
- AIA Integration Services, 7
- Associating ICC, 65
- Assumptions and Constraints, 81
- Auto Release of Entities, 60
- Auto-Release of Entities, 22
- Billing Products with Rate Plan Selector, 44
- Billing Products with Single Rate Plan, 44
- Billing Products with Usage Rating, 45
- BillOfMaterialsEBO, 6
- BillOfMaterialsEBS, 8
- BillOfMaterialsReponseEBS, 9
- Business Unit and Inventory Location IDs in Siebel, 102
- ClassificationSchemeEBO, 6
- ClassificationSchemeEBS, 10
- ClassificationSchemeResponseEBS, 10
- Communications Product Details Library, 26
- Communications Services Billing Library, 25
- Communications/ItemCompositionEBO, 7
- Communications/PriceListEBO, 7
- CommunicationsItemCompositionEBSV1, 11
- CommunicationsItemCompositionResponseEBSV1, 11
- CommunicationsPriceListEBSV2, 10
- CommunicationsPriceListResponseEBSV2, 10
- Component Exclusions in context of Promotions, 34
- Component Overrides, 37
- Component UDA for Item Synchronization, 26
- Core AIA Components, 6
- Defining Billing Products, 44
- Domain Value Maps, 106
- Exclusions within Option Class BOMs, 35
- Extracting Organizations and Inventory Locations, 96
- Frequently Asked Questions, 135
- Handling Errors, 112
- Identifying Cross-References, 105
- Implementing the Process Integration Pack, 95
- Inventory Location and Operating Unit IDs in Oracle Product Hub, 99
- Item Validation Organization for Oracle E-Business Suite, 97
- Item Validation Organization for Oracle Product Hub, 99
- ItemEBO, 6
- ItemEBSV2, 8
- ItemResponseEBSV2, 8
- JCA Adapter
 - Install and Configure, 102
- Mapping Siebel CRM, 101
- Mapping Siebel CRM organizations, 98
- MDM PIP
 - Process Flows, 4
- Multi-Event Product Synchronization, 44
- Multilanguage Support, 18
- One-Time and Penalty Charges, 47
- OPH Integration Services, 78

- Option Class, 19
- Oracle BRM Integration Services, 94
- Oracle BRM Interfaces, 93
- Oracle E-Business Suite Integration Services, 90
- Oracle E-Business Suite Interfaces, 89
- Oracle Product Hub Interfaces, 76
- Organization and Operating Unit IDs in Oracle E-Business Suite, 96
- Overview
 - Base Pack, 13
 - MDM PIP, 1
 - Oracle BRM, 93
 - Oracle E-Business Suite, 89
 - Siebel CRM, 85
- Populating Cross-References, 103
- Prerequisites
 - Base Pack, 13
- PriceListEBO, 6
- PriceListEBSV2, 9
- PriceListResponseEBSV2, 9
- Product Management Library, 25
- ProductOptimizedSyncPriceListListSiebelCommsProvABCServiceImpl, 88
- Promotion Based Discounts, 38
- Publication Status, 57
- Publishing ICC, 65
- QueryBillOfMaterialsListPIMAdapter, 80
- QueryItemListPIMAdapter, 80
- QueryResponsibilityEbizAdapter, 91
- Rate tiers and effectivity, 137
- Sellable Product Information Library, 23
- Setting Configuration Properties, 112
- Setting Up Cross-References, 103
- Setting Up Oracle BRM, 102
- Setting up the Data for Seeded Attributes, 134
- Settings for
 - ProductOptimizedSyncPriceListListSiebelCommsProvABCServiceImpl service, 130
- Settings for
 - QueryBillOfMaterialsListPIMAdapter service, 124
- Settings for QueryItemListPIMAdapter service, 124
- Settings for
 - SyncBillOfMaterialsListEbizProvABCServiceImpl service, 116
- Settings for
 - SyncBillOfMaterialsListSiebelProvABCServiceImpl service, 117
- Settings for
 - SyncClassificationSchemeListSiebelProvABCServiceImpl service, 127
- Settings for
 - SyncItemCatalogCategoryPIMReqABCServiceImpl service, 126
- Settings for
 - SyncItemCompositionListSiebelCommsProvABCServiceImpl service, 131
- Settings for
 - SyncItemListBRMProvABCServiceImpl service, 122
- Settings for SyncItemListEbizProvABCServiceImpl service, 112
- Settings for SyncItemListPIMReqABCServiceImpl service, 119
- Settings for
 - SyncProductSiebelProvABCServiceImpl service, 113
- Settings for
 - SyncSpecificationValueSetListPIMReqABCServiceImpl service, 125
- Settings for
 - SyncSpecificationValueSetListSiebelProvABCServiceImpl service, 128
- Siebel CRM Integration Services, 86

- Siebel CRM Interfaces, 85
- SpecificationValueSetEBO, 6
- SpecificationValueSetEBS, 9
- SpecificationValueSetResponseEBS, 9
- Support for Multi-Row Attribute Groups, 27
- SyncBillOfMaterialsListEbizAdapter, 91
- SyncBillOfMaterialsListEbizProvABCSImpl, 90
- SyncBillOfMaterialsListEBM, 6
- SyncBillOfMaterialsListResponseEBM, 6
- SyncBillOfMaterialsListSiebelProvABCSImpl, 87
- SyncClassificationSchemeListEBM, 6
- SyncClassificationSchemeListResponseEBM, 7
- SyncClassificationSchemeListSiebelProvABCSImpl, 87
- Synchronization of Attribute Groups, 62
- Synchronization of Billing Products, 42
- Synchronization of Customer UDA, 63
- Synchronization of Discounts and Discount Models, 38
- Synchronization of ICC, 59
- Synchronization of ICC Hierarchies, 64
- Synchronization of Items and BOMs, 14, 28
- Synchronization of Metadata, 58
- Synchronization of Products and Discounts and Associated Structures, 31
- Synchronization of Promotions, 33
- Synchronization of Relationships/Structure, 60
- Synchronization of Transaction Attributes, 64
- Synchronization of Valuesets, 70
- SyncItemCatalogCategoryPIMEventConsumer, 78
- SyncItemCatalogCategoryPIMReqABCSImpl, 78
- SyncItemCompositionListEBM, 7
- SyncItemCompositionListResponseEBM, 7
- SyncItemCompositionListSiebelCommsProvABCSImpl, 88
- SyncItemListBRMProvABCSImpl, 94
- SyncItemListEbizAdapter, 91
- SyncItemListEbizProvABCSImpl, 90
- SyncItemListEBM, 6
- SyncItemListPIMEventConsumer, 79
- SyncItemListPIMReqABCSImpl, 79
- SyncItemListResponseEBM, 6
- SyncPriceListEBM, 6
- SyncPriceListListBRMProvABCSImpl, 94
- SyncPriceListResponseEBM, 6
- SyncProductSiebelProvABCSImpl, 86
- SyncSpecificationValueSetListEBM, 6
- SyncSpecificationValueSetListPIMEventConsumer, 80
- SyncSpecificationValueSetListPIMReqABCSImpl, 81
- SyncSpecificationValueSetListResponseEBM, 6
- SyncSpecificationValueSetListSiebelProvABCSImpl, 87
- Target System Routing, 17
- Telco Seeded Attributes, 23
- Terminology, 2
- Timezone Dependent and Independent Fields, 56
- Transaction Attribute Value Exclusions, 37
- Transaction Attributes, 19
- TransformAppContextEbizService, 91
- Validating Cross-References, 105
- Viewing EBO Implementation Maps, 112