

**Oracle® Application Architecture 2.5:
PIP Functional Interoperability Guide**

Release 2.5

Part No. E17430-03

June 2011

ORACLE®

Oracle Application Architecture 2.5: PIP Functional Interoperability Configuration Guide

Part No. E17430-03

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

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

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

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

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

U.S. GOVERNMENT RIGHTS

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

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

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

Contents

| | |
|---|----|
| Contents | 3 |
| Process Integration Pack (PIP) Functional Interoperability Configuration Guide Preface..... | 5 |
| Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide | 5 |
| Oracle Application Integration Architecture - Foundation Pack: Concepts and Technologies Guide | 5 |
| Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide | 6 |
| Oracle Application Integration Architecture Process Integration Packs..... | 7 |
| Additional Resources | 7 |
| Understanding PIP Functional Interoperability | 9 |
| PIP Functional Interoperability Matrix | 10 |
| Chapter 1: Interoperability of Design to Release: Agile - EBS PIP with the Order to Cash PIP | 15 |
| Describing the Product Sync Flows in Design to Release | 15 |
| Describing the Product Sync Flows in Order to Cash | 16 |
| Describing Best Practices for Flows..... | 16 |
| Configuring Routing Rules | 16 |
| Chapter 2: Interoperability of the Customer MDM PIP with the Order to Cash PIP | 17 |
| Describing Customer Create or Update Flows in Siebel CRM | 17 |
| Describing Customer Creates or Updates in Oracle EBS..... | 18 |
| Describing the Merge Process in Interoperability..... | 18 |
| Describing Best Practices for Flows..... | 18 |
| Configuring Routing Rules | 18 |
| Chapter 3: Interoperability of the Product MDM PIP with the Order to Cash PIP | 27 |
| Describing the Product Flows in Order to Cash | 27 |
| Describing the Product Flows in Product MDM..... | 27 |
| Describing Best Practices for Flows..... | 28 |
| Configuring Routing Rules and AIA Configuration Properties | 28 |
| Chapter 4: Interoperability of the Customer MDM PIP with the Comms Order to Bill | 31 |
| Describing Customer Flows in Customer MDM | 31 |
| Describing Customer Flows in Order to Bill | 31 |
| Describing Best Practices for Flows..... | 33 |

| | |
|--|----|
| Using the Merge Process in Customer MDM | 34 |
| Using the Merge Process with Order to Bill | 34 |
| Best Practices for Using the Merge | 34 |
| Configuring Routing Rules | 35 |
| Chapter 5: Interoperability of the Oracle Customer Master Data Management Integration Pack with the Comms Agent Assisted Billing Care PIP | 37 |
| Describing Customer Flows in Customer MDM | 37 |
| Describing Customer Flows in Agent Assisted Billing Care | 37 |
| Best Practices for Flows | 40 |
| Using the Merge Process in Customer MDM | 41 |
| Using the Merge Process with Agent Assisted Billing Care | 41 |
| Best Practices for Using the Merge | 41 |
| Configuring Routing Rules | 42 |
| Chapter 6: Interoperability of the Oracle Product MDM PIP with the Comms Order to Bill | 43 |
| Describing Product Flows in Order to Bill | 43 |
| Describing Product Flows in Product MDM | 43 |
| Creating Metadata in Oracle Product Hub | 43 |
| Creating Products and Discounts in Oracle Product Hub | 44 |
| Creating Bundles and Promotions in Oracle Product Hub | 45 |
| Best Practices for Flows | 45 |
| Changing the System Type for Siebel | 46 |
| Configuring Routing Rules | 46 |
| Manual Redeployment | 48 |
| Index | 49 |

Process Integration Pack (PIP) Functional Interoperability Configuration Guide Preface

This preface discusses:

- Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide
- 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 Process Integration Packs
- Additional resources

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

The *Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide* provides conceptual, setup, and usage information for the following Core Infrastructure Components:

- The Business Service Repository (BSR).
- The Composite Application Validation System (CAVS).
- Error handling and logging.
- The Diagnostics Framework.

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

The 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.
- 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 AIA services to work with the Composite Application Validation System (CAVS).

- Configure Oracle AIA processes to be eligible for error handling and logging.
- Extend enterprise business objects.

In addition, this book provides, Application Integration Architecture naming standards.

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 to rapidly implement a discreet business process.

Additional Resources

The following resources are available:

| Resource | Location |
|--|---|
| Installation Guide | My Oracle Support https://metalink.oracle.com/ |
| Documentation updates | My Oracle Support https://metalink.oracle.com/ |
| Release Notes | Oracle Technology Network http://www.oracle.com/technology/ |
| Known issues, workarounds, and current list of patches | My Oracle Support https://metalink.oracle.com/ |

Understanding PIP Functional Interoperability

This guide describes the functional interoperability of Application Integration Architecture (AIA) process integration packs (PIPs).

A single AIA release contains multiple PIPs, enabling customers to deploy more than one PIP if desired. Functional interoperability is defined as the ability to deploy two or more PIPs on the same SOA instance and for the services provided by the PIPs to work together to fulfill end-to-end business processes. Functionally interoperable PIPs have also been tested together by AIA. PIPs can be interoperable, co-deployable, or co-installable.

For more information, and definitions of these terms, see the AIA PIP Co-Deployment Matrix posted in article ID 881206.1 on My Oracle Support.

Each of the PIPs has been designed to support a specific set of source and target applications. As delivered, the routing rules for each of these PIPs point to specific target applications. However, when a customer deploys more than one PIP, these routing rules must be modified so that the PIPs can meaningfully support interoperability and produce the desired functional outcome.

This document describes the PIPs that can potentially be deployed at the same site. Wherever applicable, recommendations are given for the best practices flow for these PIPs to work together. In addition, routing rule changes that customers need to make to suit the best practices flow are provided.

Customers may still have to make additional changes to the routing rules if they need to interface these PIPs with additional applications that are not supported by the PIPs as delivered.

For more information about using and extending routing rules, see Oracle Enterprise Service Bus Developer's Guide, Creating Routing Services and Routing Rules.

Before embarking on your interoperability project, ensure that you have installed the latest versions and patches for each product.

For more information about versions and patches that must be applied, download the latest version of the AIA 2.5 Installation and Upgrade Guide available on My Oracle Support.

Understanding Install Sequence Dependencies

Depending on the sequence in which PIPs (or patches) are installed, certain artifacts (for example, routing rules, or DVMs) may be overwritten or returned to their original values by the install process.

For more information about install dependencies, download the most recent AIA 2.5 Installation and Upgrade Guide available in article 959726.1 on My Oracle Support.

PIP Functional Interoperability Matrix

Not all of the PIPs that are marked as interoperable in the table below have chapters in this guide. Only those PIPs that require manual changes (like routing rule changes) or that need to follow certain guidelines in order to interoperate have chapters in this guide.

This table provides a list of Oracle AIA 2.5 PIPs along with the PIPs with which they are functionally interoperable:

| Oracle AIA 2.5 PIP | Functionally Interoperable With: |
|---|---|
| Comms Order to Activate: Siebel CRM - OSM | <ul style="list-style-type: none"> Comms Agent Assisted Billing Care: Siebel CRM – BRM Comms Order to Bill: Siebel CRM – BRM Customer MDM: Base Customer MDM: SBL CRM Customer MDM: Comms BRM Product MDM: Base Product MDM: SBL CRM Product MDM: Comms BRM |
| Comms Agent Assisted Billing Care: Siebel CRM – BRM | <ul style="list-style-type: none"> Comms Order to Activate: Siebel CRM - OSM Comms Order to Bill: Siebel CRM – BRM Customer MDM: Base Customer MDM: SBL CRM Customer MDM: Comms BRM Product MDM: Base Product MDM: SBL CRM Product MDM: Comms BRM |
| Comms Order to Bill: Siebel CRM – BRM | <ul style="list-style-type: none"> Comms Order to Activate: Siebel CRM - OSM Comms Agent Assisted Billing Care: Siebel CRM – BRM Customer MDM: Base Customer MDM: SBL CRM Customer MDM: Comms BRM Product MDM: Base Product MDM: SBL CRM Product MDM: Comms BRM |

| Oracle AIA 2.5 PIP | Functionally Interoperable With: |
|--|---|
| Design to Release: Agile – EBS | <ul style="list-style-type: none"> Order to Cash: Siebel CRM – EBS |
| LSP Driver Mgmt: OTM – EBS | <ul style="list-style-type: none"> LSP Financial Mgmt: OTM – EBS LSP Order Mgmt: OTM – EBS – Siebel CRM |
| LSP Financial Mgmt: OTM – EBS | <ul style="list-style-type: none"> LSP Driver Mgmt: OTM – EBS LSP Order Mgmt: OTM – EBS - Siebel CRM |
| LSP Order Mgmt: OTM – EBS – Siebel CRM | <ul style="list-style-type: none"> LSP Driver Mgmt: OTM – EBS LSP Financial Mgmt: OTM – EBS |
| Order to Cash: Siebel CRM – EBS | <ul style="list-style-type: none"> Agile – EBS Customer MDM: Base Customer MDM: SBL CRM Customer MDM: Comms BRM Product MDM: Base Product MDM: SBL CRM Product MDM: Comms BRM |
| Order to Cash: Siebel CRM - SAP | <ul style="list-style-type: none"> Customer MDM: Base Customer MDM: SBL CRM Customer MDM: SAP |
| Customer MDM: Base | <ul style="list-style-type: none"> Customer MDM: SBL CRM Customer MDM: Comms BRM Customer MDM: EBS Customer MDM: SAP Order to Cash: Siebel CRM – EBS Order to Cash: Siebel CRM – SAP Comms Agent Assisted Billing Care: Siebel CRM – BRM Comms Order to Bill: Siebel CRM – BRM Comms Order to Activate: Siebel CRM – OSM |
| Customer MDM: SBL CRM | <ul style="list-style-type: none"> Customer MDM: Base Customer MDM: Comms BRM Customer MDM: EBS Customer MDM: SAP Order to Cash: Siebel CRM – EBS Order to Cash: Siebel CRM – SAP |

| Oracle AIA 2.5 PIP | Functionally Interoperable With: |
|-------------------------|--|
| | <ul style="list-style-type: none"> Comms Agent Assisted Billing Care: Siebel CRM – BRM Comms Order to Bill: Siebel CRM – BRM Comms Order to Activate: Siebel CRM – OSM |
| Customer MDM: EBS | <ul style="list-style-type: none"> Customer MDM: Base Customer MDM: Siebel CRM Customer MDM: Comms BRM Customer MDM: SAP Order to Cash: Siebel CRM – EBS |
| Customer MDM: Comms BRM | <ul style="list-style-type: none"> Customer MDM: Base Customer MDM: SBL CRM Customer MDM: EBS Customer MDM: SAP Comms Agent Assisted Billing Care: Siebel CRM – BRM Comms Order to Bill: Siebel CRM – BRM Comms Order to Activate: Siebel CRM – OSM |
| Product MDM: Base | <ul style="list-style-type: none"> Product MDM: BRM Product MDM: SBL CRM Product MDM: EBS Order to Cash: Siebel CRM – EBS Comms Agent Assisted Billing Care: Siebel CRM – BRM Comms Order to Bill: Siebel CRM – BRM Comms Order to Activate: Siebel CRM – OSM |
| Product MDM: SBL CRM | <ul style="list-style-type: none"> Product MDM: Base Product MDM: BRM Product MDM: EBS Order to Cash: Siebel CRM – EBS Comms Agent Assisted Billing Care: Siebel CRM – BRM Comms Order to Bill: Siebel CRM – BRM Comms Order to Activate: Siebel CRM – OSM |
| Product MDM: EBS | <ul style="list-style-type: none"> Product MDM: Base Product MDM: BRM Product MDM: Siebel CRM |

| Oracle AIA 2.5 PIP | Functionally Interoperable With: |
|------------------------|---|
| | <ul style="list-style-type: none"> Order to Cash: Siebel CRM – EBS |
| Product MDM: Comms BRM | <ul style="list-style-type: none"> Comms Order to Bill: Siebel CRM – BRM Order to Cash: Siebel CRM – EBS Comms Agent Assisted Billing Care: Siebel CRM – BRM Comms Order to Bill: Siebel CRM – BRM Comms Order to Activate: Siebel CRM – OSM |

For more information, see the AIA PIP Co-Deployment Matrix posted in article ID 881206.1 on My Oracle Support.

The matrix is presented as an HTML report that lists all of the PIPs that are available in a release and whether or not they can be safely co-installed and/or co-deployed in a single SOA server instance.

Chapter 1: Interoperability of Design to Release: Agile - EBS PIP with the Order to Cash PIP

This chapter discusses this interoperable pair:

- Agile Product Lifecycle Management Integration Pack for Oracle E-Business Suite: Design to Release (Design to Release: Agile - EBS).
- Siebel CRM Integration Pack for Oracle Order Management: Order to Cash (Order to Cash: Siebel CRM – EBS).

The Design to Release: Agile - EBS/PIM PIP supports the exchange of product information between Agile PLM and Oracle EBS. In the Order to Cash: Siebel CRM – EBS PIP, product information is exchanged between Siebel CRM and Oracle EBS. It is a one way feed from Oracle EBS to Siebel CRM.

For more information, see the Agile Product Lifecycle Management Integration Pack for Oracle E-Business Suite: Design to Release Implementation Guide and the Siebel CRM Integration Pack for Oracle Order Management: Order to Cash Implementation Guide.

Describing the Product Sync Flows in Design to Release

The Design to Release: Agile - EBS/PIM PIP enables and integrates the product development process between Design to Release: Agile - EBS/PIM and Oracle EBS. This PIP supports different scenarios.

Change Order Release: During the product design phase, new products or parts are introduced or existing parts may go through design changes. When the authoring of a part's attributes and design information is complete and is ready for publishing to the manufacturing system, it is released by means of Change Orders. The Change Order Release Process consists of the New Part and Product Release (PREL) and the Product Design Modification flows of Design to Release: Agile - EBS/PIM. The release of a change order in the Design to Release: Agile - EBS/PIM PIP system acts as a trigger for the synchronization of product design information with the ERP system.

For all items pushed to the ERP system, the system verifies whether the items already exist, and whether they have the same revision number as the old item from the Agile system. If the item did not exist in ERP and is being released for the first time from PLM, it is created in ERP. If the item already exists in ERP and the two systems were coordinated concerning item revision, the existing item is updated with new attribute data from Agile.

Describing the Product Sync Flows in Order to Cash

These are the product sync related flows:

- Sync Orderable Product from Oracle EBS to Siebel: When an item (product) is created or updated in Oracle EBS, it is synchronized to Siebel CRM.
- Request BOM Sync from Siebel to Oracle EBS: A product administrator in Siebel CRM requests the details of item relationships for the selected BOM from Oracle EBS to Siebel.

The details of the latest active version of the BOM and associated non-orderable components are synchronized from Oracle EBS to Siebel CRM as result of this request.

Describing Best Practices for Flows

The following are the best practices for product (Item) and BOM synchronization among the various participating applications when Design to Release: Agile - EBS/PIM PIP and the Order to Cash: Siebel CRM – EBS support interoperability:

- Design to Release: Agile - EBS / PIM manages the product life cycle.
- While Design to Release: Agile - EBS / PIM is the system of record for item description, design, specs, and other information, the ERP system has more attributes and placeholders for information than the Design to Release: Agile - EBS / PIM system. Hence, the change order release needs to be updated in the ERP system.
- The update of Item Attribute information from Oracle EBS to Design to Release: Agile - EBS / PIM is done as a batch process.
- Item update from Oracle EBS to Siebel CRM should be done using the Sync Item process.
- If needed, Siebel CRM should request the BOM from Oracle EBS.

Configuring Routing Rules

No changes in the configuration rules are required for interoperability of Design to Release: Agile - EBS / PIM PIP with the Order to Cash: Siebel CRM – EBS PIP.

Chapter 2: Interoperability of the Customer MDM PIP with the Order to Cash PIP

This chapter discusses this interoperable pair and the routing rules that must be changed to enable interoperability:

- Oracle Customer Master Data Management Integration Pack (Customer MDM PIP).
- Siebel CRM Integration Pack for Oracle Order Management: Order to Cash (Order to Cash: Siebel CRM – EBS).

In the Customer MDM PIP, customer information is synchronized from Oracle Customer Hub (OCH) to Siebel CRM and Oracle EBS. In the Order to Cash: Siebel CRM – EBS PIP, customer information is synchronized bi-directionally between Siebel CRM and Oracle EBS.

For more information, see the Oracle Customer Master Data Management Integration Implementation Guide and the Siebel CRM Integration Pack for Oracle Order Management: Order to Cash Implementation Guide.

Describing Customer Create or Update Flows in Siebel CRM

These are the recommendations for Siebel CRM:

- Customers must be created in Siebel CRM using the match-fetch flows. Once created, customers should be routed to Oracle Customer Hub instead of being directly routed to the target applications, in this case, Oracle EBS. Oracle Customer Hub must be configured to publish these created Customers to the target application.
- The order process integration in the Order to Cash: Siebel CRM – EBS PIP has a flow that queries the account associated to the sales order from Siebel CRM. This flow further creates account sites and site-uses in the same business unit as the sales order in Oracle EBS. This customer synchronization flow remains the same and only as part of this flow will new accounts be created in the Oracle EBS fulfillment system.
- When Oracle Customer Hub publishes messages to target systems, if only updates to existing accounts are sent to the Oracle EBS fulfillment system, then a transform that filters only updates can be applied. Applying this filter enables customers to avoid prospect records without any orders from being synced with fulfillment systems. This is documented in the routing rules section.
- There could be a situation where an account may be queued or identified as a duplicate in Oracle Customer Hub which is handled by the synchronize account/merge Oracle Customer Hub ABCS service. However, if an order were submitted using this account, the redundant customer synchronization from order process would synchronize the customer to Oracle EBS

and ensure order submission.

Describing Customer Creates or Updates in Oracle EBS

This is the recommendation for Oracle EBS: when customer records are created or updated in Oracle EBS, they should be routed to Oracle Customer Hub. These transactions should not be synchronized directly to target applications (Siebel CRM). The updated records should be published from Oracle Customer Hub to target applications (Siebel CRM).

Describing the Merge Process in Interoperability

These are the recommendations for using the merge process:

- Account or contact merges can take place only in Oracle Customer Hub.
- Merge messages from Oracle Customer Hub are routed to all target applications registered within Oracle Customer Hub.
- Merges should not be done in any other target applications.
- The Order to Cash: Siebel CRM – EBS PIP functionality of party or account merge in Oracle EBS is not supported in interoperability with Customer MDM PIP merge functionality.

Describing Best Practices for Flows

For Order to Cash and Customer MDM PIP interoperability, these best practices flows are recommended for the customer account synchronization among the participating applications:

- Customer creates or updates in Siebel CRM.
- Customer creates or updates in Oracle EBS.
- Using the merge process.

Configuring Routing Rules

This section describes the routing rule changes that must be made to enable interoperability. There may be install sequence considerations.

For more information about install sequence considerations, see [Understanding Install Sequence Dependencies](#). **For more information** about using and extending routing rules, see Oracle Enterprise Service Bus Developer's Guide, Creating Routing Services and Routing Rules.

For CustomerPartyEBSV2

Note. Every operation should have the routing to the simulator.

The highlighted text represents the change to make to the existing condition.

| Operation | Filter Condition | Service Invoked | Description |
|------------------------|---|--|--|
| QueryCustomerPartyList | {aia:getServiceProperty(' cb; http://xmlns.oracle.com/EnterpriseServices/Core/CustomParty/V2 rcb; CustomerPartyEBSV2','Routing.ActiveRuleset',false())='DEFAULT' and (/ebo:QueryCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not(/ebo:QueryCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and (/ebo:QueryCustomerPartyListEBM/corecom:EBMHeader/corecom:Target/corecom:ApplicationTypeCode='SIEBEL')};{ namespace aia=http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.AIFunctions namespace ebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CustomParty/V2 namespace corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2 } | QueryCustomerPartyListSiebelProvABCSImplV2_1_0::QueryCustomerPartyList | If a routing rule which checks corecom:Target/corecom:ApplicationTypeCode='SIEBEL' & routes to QueryCustomerPartyListSiebelProvABCSImplV2 is not already present, add this O2C routing rule. Used to get the record details from Siebel during order submission flow. |
| SyncCustomerPartyList | {aia:getServiceProperty(' cb; http://xmlns.oracle.com/EnterpriseServices/Core/CustomParty/V2 rcb; CustomerPartyEBSV2','Routing.ActiveRuleset',false())='DEFAULT' and (/ebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not(/ebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and (/ebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Target/corecom:ApplicationTypeCode='UCM' or (not(/ebo:SyncCustomerPartyListEBM/c | SyncCustomerPartyListUCMProvABCSImpl_1_0::SyncCustomerPartyList | Modify MDM routing rule with additional filter condition. Used to route synchronization messages from any target application to ORACLE CUSTOMER HUB when the message is not routed from the Order flow. |

| Operation | Filter Condition | Service Invoked | Description |
|---------------------------|--|---|---|
| | <pre> orecom:EBMHeader/corecom:Target/c orecom:ID/text()) and aia:getSystemType(/ebo:SyncCustome rPartyListEBM/corecom:EBMHeader/c orecom:Sender/corecom:ID)!= 'UCM' and /ebo:SyncCustomerPartyListEBM/core com:EBMHeader/corecom:Sender/cor ecom:CallingServiceName!='Interface CustomerToFulfillmentEBF')));{ namespace corecom=http://xmlns.oracle.com/Ente rpriseObjects/Core/Common/V2 namespace aia=http://www.oracle.com/XSL/Transf orm/java/oracle.apps.aia.core.xpath.AI AFunctions namespace ebo=http://xmlns.oracle.com/Enterpris eObjects/Core/EBO/CustomerParty/V2 } </pre> | | |
| SyncCustome rPartyList | <pre> {aia:getServiceProperty(" cb; http://xml ns.oracle.com/EnterpriseServices/Core /CustomerParty/V2 rcb; CustomerParty EBSV2','Routing.ActiveRuleset',false()) ='DEFAULT' and (/ebo:SyncCustomerPartyListEBM/cor ecom:EBMHeader/corecom:MessageP rocessingInstruction/corecom:Environ mentCode='PRODUCTION' or not(/ebo:SyncCustomerPartyListEBM/c orecom:EBMHeader/corecom:Messag eProcessingInstruction/corecom:Enviro nmentCode/text())) and (/ebo:SyncCustomerPartyListEBM/cor ecom:EBMHeader/corecom:Target/cor ecom:ApplicationTypeCode='EBIZ')));{ namespace aia=http://www.oracle.com/XSL/Transf orm/java/oracle.apps.aia.core.xpath.AI AFunctions namespace corecom=http://xmlns.oracle.com/Ente rpriseObjects/Core/Common/V2 namespace ebo=http://xmlns.oracle.com/Enterpris eObjects/Core/EBO/CustomerParty/V2 } </pre> | SyncCustomerPartyList EbizProvABCImpl_1_0::SyncCustomerPartyL ist | <p>No changes to this MDM routing rule. But add the transformation map (XformSyncCustomerPartyListToUpdateCustomerPartyList.xls) to filter only updates. The file is available at ESB_Projects/CustomerParty/CustomerPartyEBSV2/ from the esb console.</p> <p>Used to route synchronization messages from Oracle Customer Hub to Oracle EBS for already synced records.</p> |
| SyncCustome rPartyList | <pre> {aia:getServiceProperty(" cb; http://xml ns.oracle.com/EnterpriseServices/Core /CustomerParty/V2 rcb; CustomerParty </pre> | SyncCustomerPartyList EbizProvABCImpl_1_0::SyncCustomerPartyL | <p>Add new rule for MDM and O2C interoperability.</p> <p>Used to route</p> |

| Operation | Filter Condition | Service Invoked | Description |
|--------------------------------------|---|--|--|
| | EBSV2','Routing.ActiveRuleset',false()) ='DEFAULT' and (/ebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not (/ebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and (/ebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Target/corecom:ApplicationTypeCode='EBIZ' or (not /ebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Target/corecom:ID/text())) and aia:getSystemType(/ebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Sender/corecom:ID)!='EBIZ' and /ebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Sender/corecom:CallingServiceName='InterfaceCustomerToFulfillmentEBF';{namespace@ebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CustomerParty/V2namespace@aia=http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.AIAFunctions namespace@corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2} | ist | synchronization messages to Oracle EBS from the Siebel Order submission process. |
| ProcessCreditEligibilityVerification | {aia:getServiceProperty(' lcb; http://xmlns.oracle.com/EnterpriseServices/Core/CustomerParty/V2 rcb; CustomerPartyEBSV2','Routing.ActiveRuleset',false())='DEFAULT' and (/ebo:ProcessCreditEligibilityVerificationEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not(/ebo:ProcessCreditEligibilityVerificationEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and (/ebo:ProcessCreditEligibilityVerificationEBM/corecom:EBMHeader/corecom:Target/corecom:ApplicationTypeCode | ProcessCreditEligibilityEbizProvABCSImpl_1_0::ProcessCreditEligibility | Add O2C routing rule. Used to get credit eligibility from Oracle EBS. |

| Operation | Filter Condition | Service Invoked | Description |
|-----------|---|-----------------|-------------|
| | <pre> ='EBIZ' or (not(/ebo:ProcessCreditEligibilityVerificationEBM/corecom:EBMHeader/corecom:Target/corecom:ID/text()) and aia:getSystemType(/ebo:ProcessCreditEligibilityVerificationEBM/corecom:EBMHeader/corecom:Sender/corecom:ID)!='EBIZ')));{ namespace aia=http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.AIAFunctions namespace corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2 namespace ebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CustomerParty/V2 } </pre> | | |

For CustomerPartyResponseEBSV2

| Operation | Filter Condition | Service Invoked | Description |
|-------------------------------|--|--|---|
| SyncCustomerPartyListResponse | <pre> {aia:getServiceProperty('lcb; http://xmlns.oracle.com/EnterpriseServices/Core/CustomerParty/V2 rcb; CustomerPartyResponseEBSV2','Routing.ActiveRuleSet',false())='DEFAULT' and (/ebo:SyncCustomerPartyListResponseEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not(/ebo:SyncCustomerPartyListResponseEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and aia:getSystemType(/ebo:SyncCustomerPartyListResponseEBM/corecom:EBMHeader/corecom:Sender/corecom:ID)!='UCM' and ((/ebo:SyncCustomerPartyListResponseEBM/corecom:EBMHeader/corecom:Sender/corecom:WSAddress/ns3:ReplyTo/ns3:ServiceName != 'InterfaceCustomerToFulfillmentEBF') or not(/ebo:SyncCustomerPartyListResponseEBM/corecom:EBMHeader/corecom: </pre> | SyncCustomerPartyListResponseUCMPProvABCSImpl_1_0::SyncCustomerPartyListResponse | <p>Modify MDM routing rule with additional filter condition.</p> <p>Used to route response messages from target apps to Oracle Customer Hub when the response message is not from the Order flow.</p> |

| Operation | Filter Condition | Service Invoked | Description |
|-------------------------------|--|--|--|
| | <pre> m:Sender/corecom:WSAddress/ns3:ReplyTo/ns3:ServiceName/text())) and (count(/ebo:SyncCustomerPartyListResponseEBM/ebo:DataArea/ebo:SyncCustomerPartyListResponse/corecom:Id entification) > 0));{ namespace corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2 namespace aia=http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.AIFunctions namespace ebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CustomerParty/V2 } </pre> | | |
| SyncCustomerPartyListResponse | <pre> {aia:getServiceProperty(' cb; http://xmlns.oracle.com/EnterpriseServices/Core/CustomerParty/V2 rcb; CustomerPartyResponseEBSV2','Routing.ActiveRuleSet',false())='DEFAULT' and (/ebo:SyncCustomerPartyListResponseEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not(/ebo:SyncCustomerPartyListResponseEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and /ebo:SyncCustomerPartyListResponseEBM/corecom:EBMHeader/corecom:Sender/corecom:WSAddress/ns1:ReplyTo/ns1:ServiceName='InterfaceCustomerToFulfillmentEBF'};{ namespace aia=http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.AIFunctions namespace corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2 namespace ebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CustomerParty/V2 namespace ns1=http://schemas.xmlsoap.org/ws/2003/03/addressing } </pre> | InterfaceCustomerToFulfillmentEBF_1_0::SyncCustomerPartyListResponse | <p>Add O2C routing rule.</p> <p>Used in customer creation flow as part of order submission to route response message to InterfaceCustomerToFullfillment EBF.</p> |

For CustomerPartyOrchestrationEBSV2

| Operation | Filter Condition | Service Invoked | Description |
|--------------------------------|---|---|---|
| InterfaceCustomerToFulfillment | {aia:getServiceProperty(' cb; http://xmlns.oracle.com/EnterpriseServices/Core/ CustomerPartyOrchestrationEBSV2','Routing.ActiveRuleset',false())='DEFAULT' and /corecustparty:ProcessCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not(/corecustparty:ProcessCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text()));} namespace aia=http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.AIFunctions namespace corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2 namespace corecustparty=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CustomerParty/V2 } | InterfaceCustomerToFulfillmentEBF_1_0::InterfaceCustomerToFulfillment | Add O2C routing rule. Used in the order submission flow to route ProcessCustomerPartyListEBM to InterfaceCustomerToFulfillmentEBF. |

For CustomerPartyOrchestrationResponseEBSV2

| Operation | Filter Condition | Service Invoked | Description |
|--|---|--|--|
| InterfaceCustomerToFulfillmentResponse | {aia:getServiceProperty(' cb; http://xmlns.oracle.com/EnterpriseServices/Core/ CustomerPartyOrchestrationResponseEBSV2','Routing.ActiveRuleset',false())='DEFAULT' and (/corecustparty:ProcessCustomerPartyListResponseEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not(/corecustparty:ProcessCustomerPartyListResponseEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text | InterfaceSalesOrderToCustomerEBFV2_1_0::InterfaceCustomerToFulfillmentResponse | Add O2C routing rule. Used in the Order submission flow to route ProcessCustomerPartyListResponseEBM to InterfaceSalesOrderToCustomerEBFV2. This routing rule has been moved from SalesOrderOrchestrationResponseEBSV2 to CustomerPartyOrchestrationResponseEBS. |

| Operation | Filter Condition | Service Invoked | Description |
|-----------|---|-----------------|-------------|
| | t())) and /corecustparty:ProcessCustomerParty ListResponseEBM/corecom:EBMHead er/corecom:Sender/corecom:WSAddre ss/ns3:ReplyTo/ns3:ServiceName='Int erfaceSalesOrderToCustomerEBFV2'; { namespace aia=http://www.oracle.com/XSL/Transf orm/java/oracle.apps.aia.core.xpath.AI AFunctions namespace corecom=http://xmlns.oracle.com/Ente rpriseObjects/Core/Common/V2 namespace ns3=http://schemas.xmlsoap.org/ws/20 03/03/addressing namespace corecustparty=http://xmlns.oracle.com/ EnterpriseObjects/Core/EBO/Custome rParty/V2 } | | |

Chapter 3: Interoperability of the Product MDM PIP with the Order to Cash PIP

This chapter discusses this interoperable pair and the routing rules that must be changed to enable interoperability:

- Oracle Product Master Data Management Integration Pack (Product MDM PIP).
- Siebel CRM Integration Pack for Oracle Order Management: Order to Cash (Order to Cash: Siebel CRM – EBS).

In both the Order to Cash and Product MDM PIPs, product information is synchronized between Siebel CRM and Oracle EBS. The Product MDM PIP also supports the Oracle Product Hub application in addition to Siebel CRM and Oracle EBS.

For more information, see the Oracle Product Master Data Management Integration Pack Implementation Guide and the Siebel CRM Integration Pack for Oracle Order Management: Order to Cash Implementation Guide.

Describing the Product Flows in Order to Cash

The flows provided by the Order to Cash PIP are:

1. Sync Orderable Product from Oracle EBS to Siebel: when an item (product) is created or updated in Oracle EBS, it is synchronized to Siebel CRM.
2. Request Bill of Material (BOM) Sync from Siebel to Oracle EBS: a product administrator in Siebel CRM requests the details of item relationships for the selected BOM from Oracle EBS. As a result of this request, the details of the latest active version of the BOM, as well as associated non-orderable components, are synchronized from Oracle EBS to Siebel CRM.

Describing the Product Flows in Product MDM

The Oracle Product Hub (OPH) is a single consolidated system for defining products (items), Bill of Materials (BOMs, or item structures), and more product related features. The flows provided by the Product MDM PIP are:

1. Synchronize Product Classes from the OPH to Siebel CRM.
2. Synchronize Domain Values from the OPH to Siebel CRM.
3. Synchronize Items and BOM from the OPH to Siebel CRM.
4. Synchronize Items and BOM from the OPH to Oracle E-Business Suite.

Describing Best Practices for Flows

In the Product MDM PIP, product information is synchronized from OPH to Siebel CRM and Oracle EBS. In the Order to Cash PIP, product information is synchronized uni-directionally between Oracle EBS and Siebel CRM.

For more information, see the Oracle Product Master Data Management Integration Pack Implementation Guide and the Siebel CRM Integration Pack for Oracle Order Management: Order to Cash Implementation Guide.

When both Order to Cash and Product MDM PIPs are deployed, the following best practices flows are recommended for the product and bill of materials (BOM) synchronization among the various participating applications.

- Products are created and updated in OPH only so the AIA configuration property 'Product.Source.PIP' will always be MDM.
- Synchronize products from the Product MDM PIP to Siebel CRM and Oracle EBS.
- Sync BOMs from the Product MDM PIP to Siebel CRM and Oracle EBS.
- Request BOM synchronization from Siebel to Oracle EBS.
- Synchronize Product Classes from the Product MDM PIP to Siebel.
- Synchronize Domain Values from the Product MDM PIP to Siebel.

Even though the BOM is synchronized from the Product MDM PIP, this BOM does not contain the Oracle EBS identifiers required to invoke Oracle Product Configurator from Siebel CRM. Thus, the product administrator in Siebel CRM must request the details of the product structure for the selected BOM from Oracle EBS. This synchronization provides the Oracle EBS identifiers in Siebel, so that the Oracle Product Configurator can be successfully invoked.

Configuring Routing Rules and AIA Configuration Properties

This section describes the routing rule and AIA configuration property changes that must be made to enable interoperability. There may be install sequence considerations.

- Set the AIAConfig Property 'PUBLISH_ALL_PRODUCT' for the service 'QueryItemCompositionListEbizProvABCSImpl' to 'Y'. Setting this property allows synchronization of all the Oracle EBS identifiers in Siebel.

For more information about install sequence considerations, see [Understanding Install Sequence Dependencies](#). **For more information** about using and extending routing rules, see Oracle Enterprise Service Bus Developer's Guide, Creating Routing Services and Routing Rules.

Set the following properties in the AIAConfigurationProperties.xml file:

1. Set the AIAConfig property PUBLISH_ALL_PRODUCT for the service QueryItemCompositionListEbizProvABCSImpl to Y. Setting this property allows synchronization of all the Oracle EBS identifiers in Siebel.
2. Set the AIAConfig property Product.Source.PIP for both the InterfaceSyncProductStructureEBF and the SyncProductSiebelProvABCSImpl services to MDM.
3. Set the AIAConfig property COMPLEX_PRODUCT_RELEASE_FLAG, SIMPLE_PRODUCT_RELEASE_FLAG for the service SyncItemCompositionListSiebelProvABCSImpl to Y.

For ItemEBSV2

A new filter condition for the routing rule to the SyncProductSiebelProvABCSImpl needs to be added in the ItemEBSV2 service.

| Operation | Filter Condition | Service Being Invoked | Description of the Routing Rule |
|--------------|--|---|--|
| SyncItemList | <pre>{aia:getServiceProperty(' lcb; http://xmlns.oracle.com/EnterpriseServices/Core/Item/V2 rcb; ItemEBSV2','Routing.ActiveRuleset',false())='DEFAULT' and (count(/ebo:SyncItemListEBM/ebo:DataArea) = 1) and contains(/ebo:SyncItemListEBM/ebo:DataArea[1]/ebo:SyncItemList/corecom:EBMBO:Name, '*')) and (/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not (/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and (xref:lookupXRef('INVENTORY_LOCATION_ID','COMMON',/ebo:SyncItemListEBM/ebo:DataArea[1]/ebo:SyncItemList/corecom:EBMBO:InventoryLocation/corecom:Identification/corecom:BusinessComponentID,'SEBL_01',false())) and (/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Target/corecom:ApplicationType Code='SIEBEL' or not(/ebo:SyncItemListEBM/corecom:EBMHeader/corecom:Target/corecom:ID/text()))}</pre> | SyncProductSiebelProvABCSImpl_1_0::SyncItem | <p>The routing rule in the MDM PIP by default will only route product sync messages created by PIM to Siebel.</p> <p>The additional conditions in the filter documented here, will ensure that “*-items” created in Ebiz, are synced to Siebel since those types of items are mastered in Ebusiness Suite in the Order to Cash PIP scenario.</p> |

| Operation | Filter Condition | Service Being Invoked | Description of the Routing Rule |
|-----------|---|-----------------------|---------------------------------|
| | and (aia:getSystemType(/ebo:SyncItem ListEBM/corecom:EBMHeader/corecom:Sender/corecom:ID) != 'SIEBEL');{ namespace ebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/Item/V2 namespace aia=http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.AIFunctions namespace corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2 namespace corecomEBO=http://xmlns.oracle.com/EnterpriseObjects/Core/CommonEBO/V1 namespace xref=http://www.oracle.com/XSL/Transform/java/oracle.tip.xref.xpath.XrefXPathFunctions} | | |

Chapter 4: Interoperability of the Customer MDM PIP with the Comms Order to Bill

This chapter discusses this interoperable pair:

- Oracle Customer Master Data Management Integration Pack (Customer MDM PIP).
- Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Order to Bill (Comms Order to Bill).

Describing Customer Flows in Customer MDM

Upon creation of a new customer in Siebel, the Oracle Customer MDM PIP sends the customer data to the Oracle Customer Hub. The customer data then goes through a sanitation process within the Oracle Customer Hub where it is cleansed, standardized, matched, and linked to other existing customer entries and possibly enriched. The updated customer data is then returned to Siebel and published to other subscribing applications.

Every time a customer is created in Siebel, that information is published to the Oracle Customer Hub. Even prospects (customers that decide not to go ahead with an order) are published to Oracle Customer Hub. Note however, that prospects are not published to BRM. Prospect data is of no use to a billing system, and its presence creates a performance overhead.

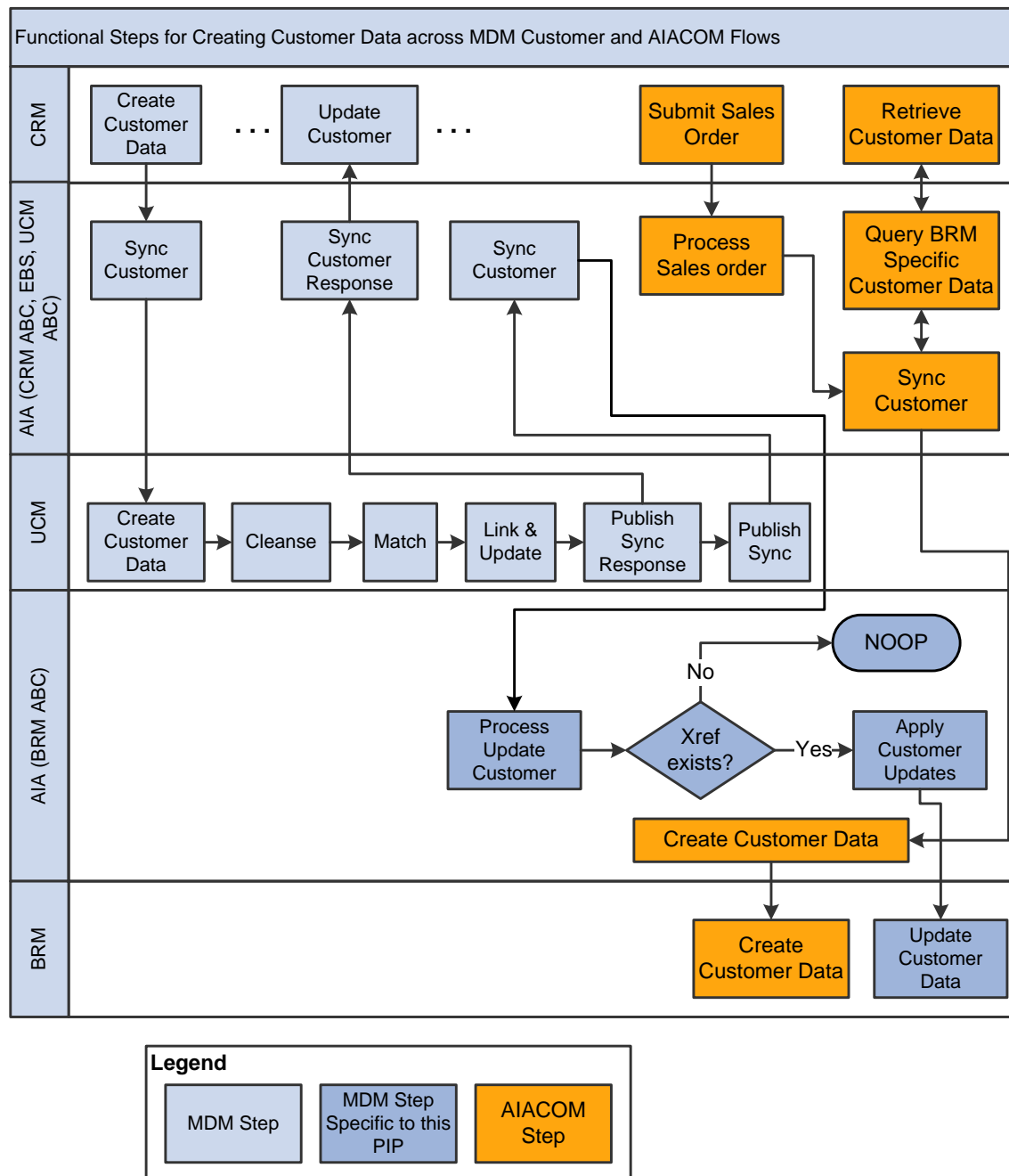
Describing Customer Flows in Order to Bill

In the Order to Bill PIP, upon submission of the order, customer data references are published with the order.

In order to avoid creating prospects in BRM, the creation of customer data in BRM will only take place through an order submission, as part of the regular Order to Bill flow, specifically the Create Customer Data in Billing system flow. In this integration point, AIA detects if the customer is new to BRM (missing cross references) in which case the customer data is queried from Siebel and created in Oracle BRM.

The updates made by Oracle Customer Hub to the customer data through cleansing, standardization, and enrichment will still need to be applied to the data in BRM. When Oracle Customer Hub publishes the cleansed data to other systems, it also publishes the same data to the BRM system. The routing rules check if the published customer data is used in BRM by checking the cross-references. If so, the message is passed to provider system and it applies the data received from the Oracle Customer Hub as an update to the existing customer data, or else it discards it.

This diagram shows the create customer data flows:



Creating customer data across MDM Customer and AIACOM flows

For more information, see the Oracle Customer Master Data Management Integration Implementation Guide and the Oracle Communications Billing and Revenue Management: Order to Bill Implementation Guide.

Describing Best Practices for Flows

These are the recommendations:

- Customers must be created in Siebel CRM using the match-fetch flows. Once created, customers should be routed to Oracle Customer Hub instead of being directly routed to the target applications, in this case, Oracle BRM. The Oracle Customer Hub should be configured to publish the newly created customers to the target application.
- Once created, customer data should be routed to Oracle Customer Hub instead of being directly routed to the target applications. The Oracle Customer Hub should be configured to publish the newly created customers to the target application.
- Publishing from Oracle Customer Hub to BRM must be done in batch. How often the batch needs to be run depends on the specific deployment conditions. It can be set from near-realtime sync (minimum delay in the batch) to a daily batch. Factors that need to be considered in determining the frequency of the batch publishing include: average duration of the order capture process, frequency of bill cycles, performance impact, and possible requirements for having near-realtime updated customer data in the billing system.
- When interoperating with applications that do not support a merge process, we recommend that customers who use the merge in Oracle Customer Hub turn off the publish to the participating applications.

For more information, see Best Practices for Using the Merge, [Disabling the Publish Process](#).

- In order to enable functional interoperability, the AIA configuration property `ProcessUpdateEventsOnly` must be set to “false” when both of these PIPs are installed. The default value for this property is “false” but certain flows require that this property is set to “true.” Customers will have to change this configuration property based on the flows being used:
 - If customers are ONLY using the Create Customer and the Update Customer flows in the MDM Customer PIP, then the `ProcessUpdateEventsOnly` property needs to be “false.”
 - If customers are ONLY using Comms Order to Bill and Comms Agent-Assisted Billing Care, then the value of the flag needs to be set to “true.” This is an optimization guideline, in that in the Comms flows, customer creation is enabled via the order flows. Thus, setting the value to “true” ensures that only update events are propagated outward from the Siebel customer sync requestor service.
 - However, if customers are using BOTH MDM Customer and Order to Bill or Agent-Assisted Billing Care, then the property needs to be “false” to enable the creates from Siebel to be propagated to the Customer Hub.
- There may occur a race condition between an account create or update coming from Siebel CRM to Oracle BRM and the updates coming from OCH to Oracle BRM.
 - If customer data is created in Siebel CRM and published to Oracle BRM, the updates to OCH are applied after the customer data has been created.
 - If customer data is updated in Siebel CRM and published to OCH and Oracle BRM, OCH

also publishes to Oracle BRM after performing cleansing and de-duping. In this case, changes to Oracle BRM are applied by OCH only after the Siebel CRM updates have been applied to OCH.

This race condition is a rare event as it takes place only when the update from OCH reaches Oracle BRM before it reaches Siebel CRM, and before the AIACOM query from Siebel CRM.

For more information about the race condition, see the Oracle Customer Master Data Management Integration - Implementation Guide, Oracle Customer Master Data Management Integration Base Pack, Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle BRM.

Using the Merge Process in Customer MDM

In the Customer MDM PIP, the merge functionality is a core function that allows a data steward to effectively determine a survivor record among several duplicate records, or to create a new record to consolidate several duplicate or similar records. For this process, Oracle Customer Hub publishes the results of the merge by providing the “survivor” record and one or many “victim” records to the source systems. It is up to the source systems to determine whether they can or should consume this message in their own records. Although some participating applications have the ability to request a merge operation, only customer merges executed within Oracle Customer Hub are supported for this release.

Using the Merge Process with Order to Bill

Siebel has the capability and a service to merge records and re-parent the associated entities to reflect the merge performed by Oracle Customer Hub. Unfortunately, BRM does not offer a service to support this feature. Obviously merging accounts or contacts in CRM without merging the corresponding accounts or contacts in BRM is not a feasible option since it will not be possible to retrieve invoice details for victim accounts or submit change orders related to assets parented to victim accounts.

Clients can develop ad-hoc procedures and tools so that administrators can merge customer data in CRM and in BRM to be consistent with the merge data in Oracle Customer Hub. This would be a custom solution not supported by the delivered flows.

Best Practices for Using the Merge

Following are the recommendations for using the merge process:

- Account or contact merges can take place only in Oracle Customer Hub.
- Merge messages from Oracle Customer Hub are routed to all target applications registered within Oracle Customer Hub, including Oracle BRM.
- Merges will not be done in Oracle BRM when receiving a merge message. What will take place is the same behavior we have for a sync message on the survivor: the BRM provider

will check the cross-reference for the survivor customer data and if it exists, will update the survivor customer data.

Disabling the Publish Process

When interoperating with applications that do not support a merge process, we recommend that customers who use the merge in Oracle Customer Hub turn off the publish to the participating applications.

Therefore, for Order to Bill processes that interoperate with MDM for Customer processes, when a merge happens in Oracle Customer Hub, it should not be published to the CRM or BRM instances that participate in the Order to Bill PIP. If the merge is published to Siebel, and the cross-references are updated, then the change order and customer update flows will break.

Perform these steps to turn off the publish after the merge:

1. Query the Oracle Customer Hub Process Merge Request workflow from Siebel tools.
1. Click the Revise button to revise the workflow.
2. Set the value of the process property EnablePubSub = False.
3. Deploy the workflow.

For more information, see the Siebel Book Shelf: Application Development, Siebel Business Process Framework, Workflow Guide.

Configuring Routing Rules

For Customer MDM PIP and Order to Bill interoperability, no routing rules changes or additions are required.

Chapter 5: Interoperability of the Oracle Customer Master Data Management Integration Pack with the Comms Agent Assisted Billing Care PIP

This chapter discusses this interoperable pair:

- Oracle Customer Master Data Management Integration Pack (Customer MDM PIP).
- Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care (Comms Agent Assisted Billing Care).

Describing Customer Flows in Customer MDM

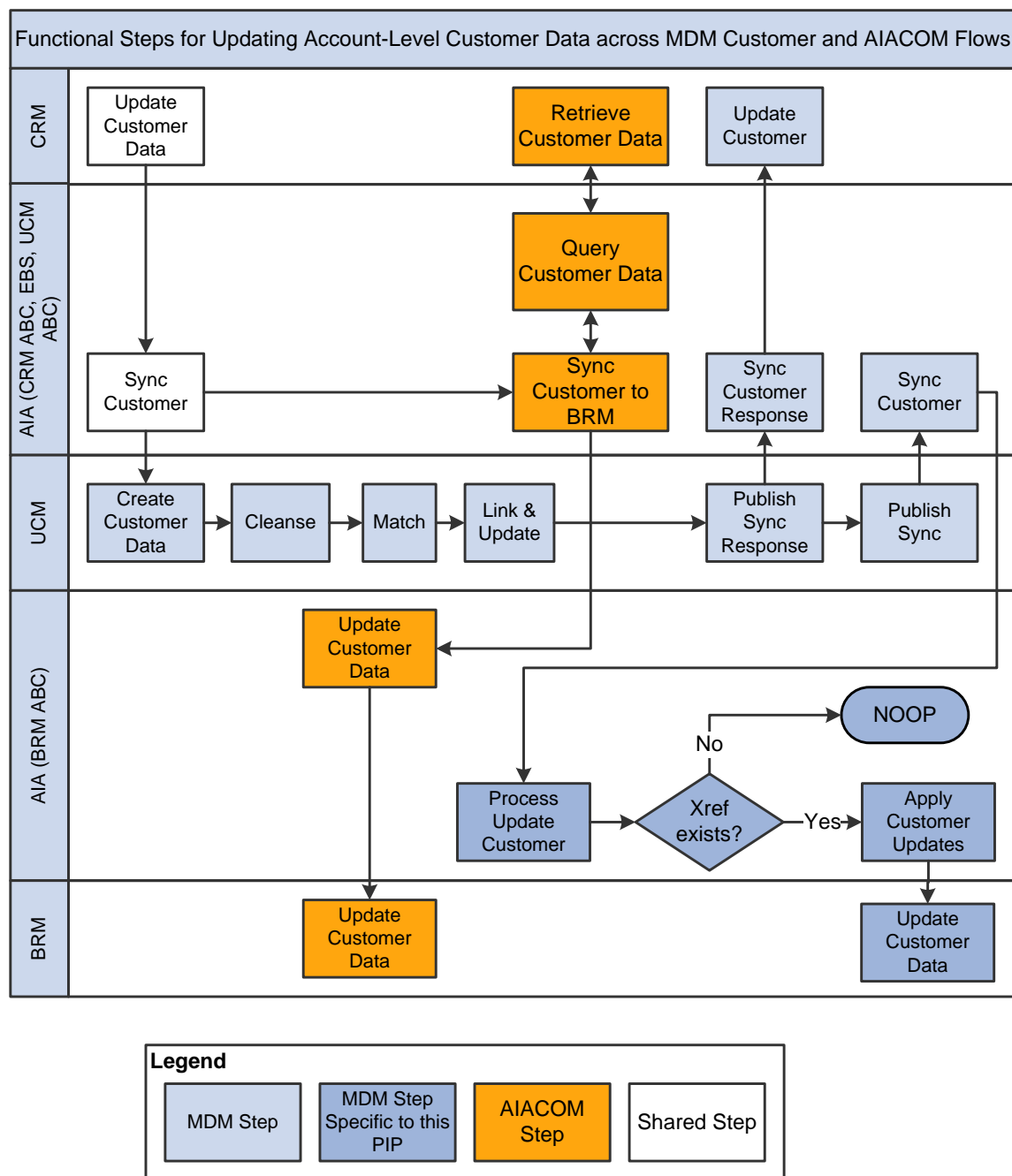
Upon creation of a new customer in Siebel, the Oracle Customer MDM PIP sends the customer data to the Oracle Customer Hub. The customer data then goes through a sanitation process within the Oracle Customer Hub where it is cleansed, standardized, matched, and linked to other existing customer entries and possibly enriched. The updated customer data is then returned to Siebel and published to other subscribing applications.

Describing Customer Flows in Agent Assisted Billing Care

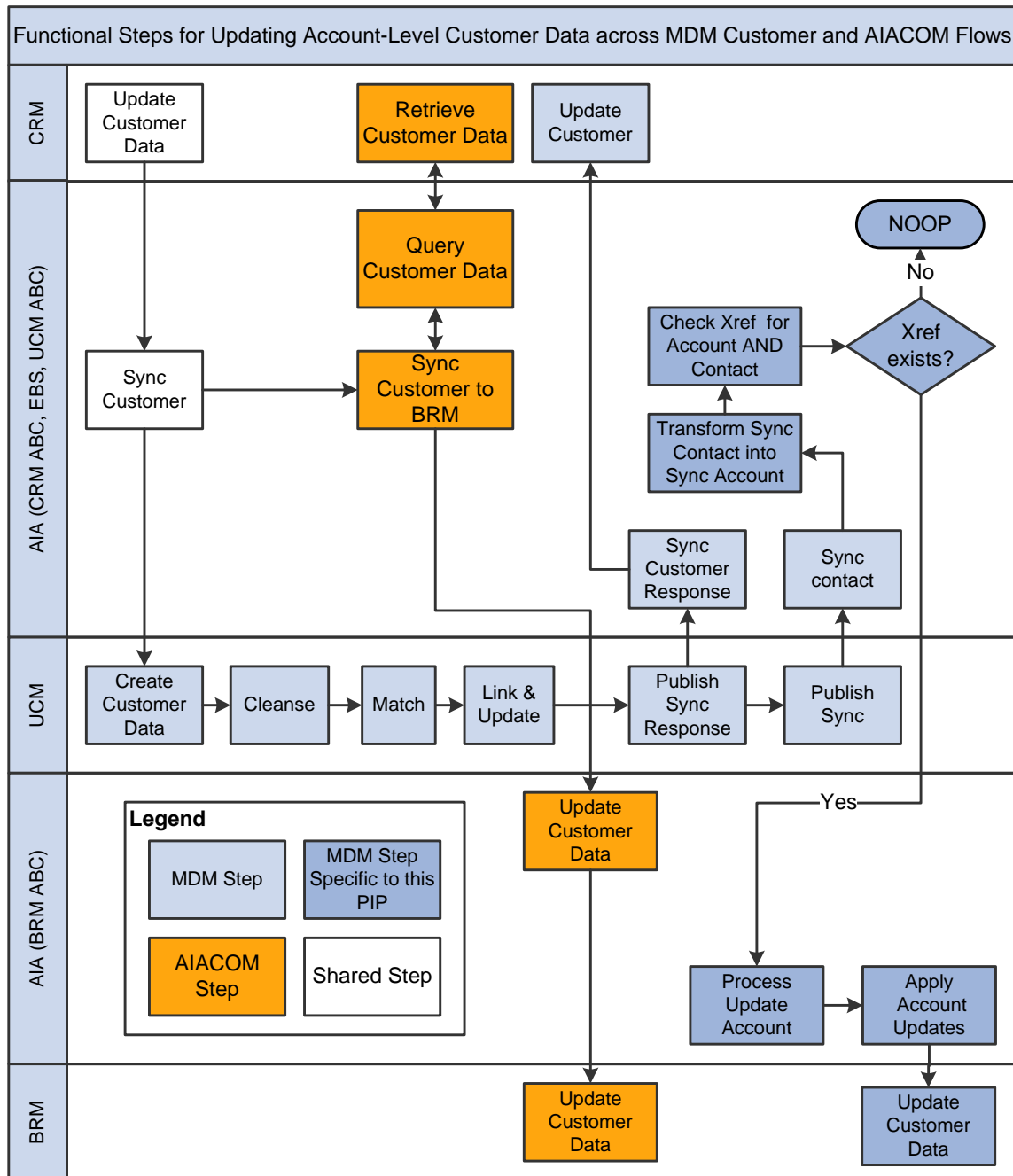
In the Agent Assisted Billing Care PIP, upon an update of the customer data, the customer data is published to AIA. If the customer data is cross-referenced in Oracle BRM, then AIA applies the updates to the customer data in Oracle BRM.

When the two PIPs are used together, the flow in the Agent Assisted Billing Care PIP that updates the customer data from Siebel CRM to Oracle BRM will be performed as usual. In addition, customer data updates will be published as a customer data sync from Oracle Customer Hub to Oracle BRM. This will be done as a batch publish, enabling the delay introduced in the batch will make sure that the updates from Oracle Customer Hub come to BRM after the updates from CRM.

This diagram shows the functional steps when updating account level customer data:



This diagram shows the functional steps when updating contact level customer data:



Updating contact-level customer data across MDM Customer and AIACOM flows

For more information, see the Oracle Customer Master Data Management Integration Pack Implementation Guide and the Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Agent Assisted Billing Care Implementation Guide.

Best Practices for Flows

These are the recommendations:

- The Agent Assisted Billing Care PIP flow for updating the customer data from Siebel CRM to Oracle BRM will not change when used in conjunction to the Customer MDM PIP.
- The sync customer message published by Oracle Customer Hub upon updating customer data will be published to Oracle BRM as well. The routing rules and the Oracle BRM provider will check if a cross-reference for the customer data exists and if so it will use the data published to update the existing customer. If the cross-reference is not found, then the data published by the Oracle Customer Hub will be discarded.
- Publishing from Oracle Customer Hub to BRM will be done in batch. How often the batch need to be run depends on specific deployment conditions. It can be set from near-realtime sync (minimum delay in the batch) to a daily batch. Factors that need to be considered in determining the frequency of the batch publishing include: frequency of bill cycles, performance impact, and possible requirements for having near-realtime updated customer data in the billing system.
- When interoperating with applications that do not support a merge process, we recommend that customers who use the merge in Oracle Customer Hub turn off the publish to the participating applications.

For more information, see Best Practices for Using the Merge, [Disabling the Publish Process](#).

- In order to enable functional interoperability, the AIA configuration property `ProcessUpdateEventsOnly` must be set to “false” when both of these PIPs are installed. The default value for this property is “false” but certain flows require that this property is set to “true.” Customers will have to change this configuration property based on the flows being used:
 - If customers are ONLY using the Create Customer and the Update Customer flows in the MDM Customer PIP, then the `ProcessUpdateEventsOnly` property needs to be “false.”
 - If customers are ONLY using Comms Order to Bill and Comms Agent-Assisted Billing Care, then the value of the flag needs to be set to “true.” This is an optimization guideline, in that in the Comms flows, customer creation is enabled via the order flows. Thus, setting the value to “true” ensures that only update events are propagated outward from the Siebel customer sync requestor service.
 - However, if customers are using BOTH MDM Customer and Order to Bill or Agent-Assisted Billing Care, then the property needs to be “false” to enable the creates from Siebel to be propagated to the Customer Hub.

Using the Merge Process in Customer MDM

In the Customer MDM PIP, the merge functionality is a core function that allows a data steward to effectively determine a survivor record among several duplicate records, or to create a new record to consolidate several duplicate or similar records. For this process, Oracle Customer Hub publishes the results of the merge by providing the “survivor” record and one or many “victim” records to the source systems. It is up to the source systems to determine whether they can or should consume this message in their own records. Although some participating applications have the ability to request a merge operation, only customer merges executed within Oracle Customer Hub are supported for this release.

Using the Merge Process with Agent Assisted Billing Care

Siebel has the capability and a service to merge records and re-parent the associated entities to reflect the merge performed by Oracle Customer Hub. Unfortunately, BRM does not offer a service to support this feature. Obviously merging accounts or contacts in CRM without merging the corresponding accounts or contacts in BRM is not a feasible option since it will not be possible to retrieve invoice details for victim accounts or submit change orders related to assets parented to victim accounts.

Clients can develop ad-hoc procedures and tools so that administrators can merge customer data in CRM and in BRM to be consistent with the merge data in Oracle Customer Hub. This would be a custom solution not supported by the delivered flows.

Best Practices for Using the Merge

Following are the recommendations for using the merge process:

- Account or contact merges can take place only in Oracle Customer Hub.
- Merge messages from Oracle Customer Hub are routed to all target applications registered within Oracle Customer Hub, including Oracle BRM.
- Merges will not be done in Oracle BRM when receiving a merge message. What will take place is the same behavior we have for a sync message on the survivor: the BRM provider will check the cross-reference for the survivor customer data and if it exists, will update the survivor customer data.

Disabling the Publish Process

When interoperating with applications that do not support a merge process, we recommend that customers who use the merge in Oracle Customer Hub turn off the publish to the participating applications.

Therefore, for Agent Assisted Billing Care processes that interoperate with MDM for Customer processes, when a merge happens in Oracle Customer Hub, it should not be published to the CRM or BRM instances that participate in the Agent Assisted Billing Care PIP. If the merge is published to Siebel, and the cross-references are updated, then the billing management and customer update flows will break.

Perform these steps to turn off the publish after the merge:

1. Query the Oracle Customer Hub Process Merge Request workflow from Siebel tools.
2. Click the Revise button to revise the workflow.
3. Set the value of the process property EnablePubSub = False.
4. Deploy the workflow.

For more information, see the Siebel Book Shelf: Application Development, Siebel Business Process Framework, Workflow Guide.

Configuring Routing Rules

For Customer MDM PIP and Agent Assisted Billing Care interoperability, no routing rules changes or additions are required.

Chapter 6: Interoperability of the Oracle Product MDM PIP with the Comms Order to Bill

This chapter discusses this interoperable pair and the routing rules that must be changed to enable interoperability:

- Oracle Product Master Data Management Integration Pack (Product MDM PIP).
- Siebel CRM Integration Pack for Oracle Communications Billing and Revenue Management: Order to Bill (Comms Order to Bill).

Describing Product Flows in Order to Bill

In the Order to Bill PIP, billing products and billing discounts are defined in Oracle Communications Billing and Revenue Management (BRM) and synchronized to Siebel CRM. The billing products and discounts are bundled in Siebel to create service bundles and promotions. These service bundles and promotions are used in the order submission process.

Describing Product Flows in Product MDM

In the Product MDM PIP, billing products and billing discounts are defined in Oracle Product Hub (OPH). These billing products and discounts are bundled to create service bundles and promotions in OPH. These promotions are published to the target applications. The corresponding entities will be created in the target applications. For example, products, discounts, bundles, and promotions are created in Siebel but only billing products and billing discounts are created in BRM.

For more information, see the Oracle Product Master Data Management Integration Pack Implementation Guide and the Oracle Communications Billing and Revenue Management: Order to Bill Implementation Guide.

Creating Metadata in Oracle Product Hub

The Item Catalog categories, attributes, and their associated valuesets are manually created by the product administrator in Oracle Product Hub. These are synchronized as product classes and attributes in Siebel. These are no longer created in Siebel.

Creating Products and Discounts in Oracle Product Hub

Products and discounts are manually created by the product administrator in Oracle Product Hub (OPH). All types of billing products and discounts are supported in OPH. All the fulfillment and communications specific attributes associated with billing products and discounts are set in OPH and synchronized to Siebel. These are synchronized to BRM and to Siebel. Products and discounts are no longer created and published from BRM.

When OPH is used as the product master to define various entities, any product publishing mechanism that exists in other applications must be disabled. For example, in the Order to Bill PIP, BRM is used as the product master and publishes the products and discounts to Siebel.

So in this case, when Oracle Product Hub is used as the product master, the publish mechanism in Oracle BRM must be disabled. To disable the publish, see the BRM documentation.

For more information about disabling the publish, see the BRM documentation, Service Integration Components, Synchronization Queue Data Manager, Installing and configuring the Synchronization Queue DM, Starting and stopping the Synchronization Queue DM.

The pricing associated with billing products is defined in Oracle Product Hub and synchronized to Siebel and BRM.

The following details that are associated with products are now defined in Oracle Product Hub and are synchronized to Siebel. These are no longer defined in Siebel:

1. Price overrides.
2. Discount overrides.
3. Special Rating Products
4. Friends and Family lists.
5. Fulfillment-related information:
 - a. Success dependency.
 - b. Fulfillment Item Code.

The following details that are associated with discounts are now defined in Oracle Product Hub and are synchronized to Siebel.

1. Duration-based discount information.

The following flags that are associated with products, discounts, bundles, and promotions are now set in Oracle Product Hub and are no longer set in Siebel:

1. Orderable flag
2. Track as Asset flag

The following information associated with billing products will still be enriched in BRM:

1. Rate plan selectors for billing products.
2. Pipeline rating for usage based billing products.
3. Pricing for friends and family lists.

Discount models for billing discounts that are associated with billing discounts will still be enriched in BRM.

The following details will still be enriched in Siebel:

1. Matrix discounts.
2. Proration plan for promotions.

Creating Bundles and Promotions in Oracle Product Hub

Bundles (service bundles, nested service bundles, and non service bundles) and promotions are manually created by the product administrator in Oracle Product Hub and synchronized to Siebel. Service bundles and Promotions are no longer created in Siebel.

The service instance-enabled flag that identifies a simple service bundle is set in Oracle Product Hub and synchronized to Siebel. The value is no longer set manually in Siebel.

Note: The flag value must not be changed or reset once the flag is set for a product in Oracle Product Hub, synchronized to target CRM system, and transactions are submitted for processing.

The association of one-time charges (suspend fee, resume fee, cancel fee, and so forth) to the corresponding bundles and account level products must be done in Siebel.

The following promotion-based information is defined in Oracle Product Hub and synchronized to Siebel. These are no longer manually defined during promotion definition in Siebel:

1. Promotion-based discounts.
2. Promotion-based component exclusions.
3. Promotion-based transaction attribute value exclusions.

The following promotion-based information will still be defined in Siebel:

1. Penalty set up for promotion cancellation or upgrade.

Note: There may be other set up steps for participating applications. See the appropriate PIP implementation guide for the participating application set up steps.

Best Practices for Flows

The following scenarios are not tested and are not supported out-of-the-box for interoperability between the Product MDM PIP and the Order to Bill PIP:

1. An implementation where OPH is introduced as the Product Master when there is a single BRM instance and the product synchronization from BRM to Siebel is continued to be used. That is, one set of products are defined in OPH and synched to BRM and Siebel and another set of products are defined in the same BRM instance and are synched to Siebel.

- An implementation where OPH is introduced as the Product Master for Siebel and one of the many BRM instances and the product synchronization from BRM to Siebel is continued to be used. But the BRM instance used to synchronize products and discounts to Siebel is different from the BRM instance that is mastered by OPH.
- An implementation that combines 1 and 2.

Changing the System Type for Siebel

The system type data for Siebel entries in the system definition must be in all upper case. Order to Bill sets this data as mixed case. Product MDM PIP installation changes this data as all upper case. Make sure to verify the system type is upper case for Siebel entries in the system definition.

ORACLE Application Integration Architecture

Home About AIA Logout

Home Service Repository Validation System Setup

Error Notification System Flexfield Configuration PIPs

Setup > System

Application Registry

Internal Id System Code System Type Application Type System Description Contact Name

Search

Search Result

Delete Create Save Cancel

| Select | Internal Id | System Code | System Description | IP Address | URL | System Type | Application Type | Version | Contact Name | Contact Phone |
|-----------------------|-------------------------|--------------------|------------------------|------------------------------|---|---------------|------------------|---------|---------------------|---------------|
| <input type="radio"/> | DIG_SYSTEM | DIAGNOSTICS_SYSTEM | AIA DIAGNOSTICS SYSTEM | 127.0.0.1 | http://diagnostics.aia.com | AIA | DIAGNOSTICS | 1.0 | Diagnostics contact | 1234567890 |
| <input type="radio"/> | AIA_DEMO | AJADEMO | DEMO | 10.0.0.1 | http://aia-demo.com | Demo | SOA-DEMO | 1.0 | AJADEMO Contact | 1234567890 |
| <input type="radio"/> | Portal | PORTAL_01 | Portal Instance 01 | \$PORTAL_HOSTNAME\$ | http://\$PORTAL_HOSTNAME\$ | PORTAL | BILLING | 7.8 | Portal contact | 1234567892 |
| <input type="radio"/> | SEBL_02 | SEBL_02 | Siebel Instance 02 | sd-c78288vce.corp.siebel.com | http://sd-c78288vce.corp.siebel.com | SIEBEL | CRM | 6.0 | Siebel contact | 1234567891 |
| <input type="radio"/> | aa | PBM_01 | PBM Instance 01 | rvs60071rns.us.oracle.com | http://rvs60071rns.us.oracle.com | PBM | PBM | 12.1.1 | PBM contact | 1234567891 |
| <input type="radio"/> | \$(osmcf.internal.id) | OSMCFS_01 | OSMCFS Instance 01 | \$(osmcf.db.host) | http://\$(osmcf.http.host):\$(osmcf.port) | OSMCFS | OSMCFS | 7.0.1 | OSMCFS contact | 1234567891 |
| <input type="radio"/> | \$(osmprov.internal.id) | OSMPROV_01 | OSMPROV Instance 01 | \$(osmprov.db.host) | http://\$(osmprov.http.host):\$(osmprov.port) | OSMPROV | OSMPROV | 7.0.1 | OSMPROV contact | 1234567891 |
| <input type="radio"/> | siebel01 | SEBL_01 | Siebel Instance 01 | sd-cpe1950004.siebel.com | http://sd-c78288vce.corp.siebel.com | Siebel | Siebel | 6.1.1 | Siebel contact | 1234567891 |
| <input type="radio"/> | BRM | BRM_01 | BRM Instance 01 | metrol01.us.oracle.com | http://gll.us.oracle.com:12815 | BRM | BRM | 7.4 | BRM contact | 1234567891 |
| <input type="radio"/> | BRM_02 | BRM_02 | BRM Instance 02 | metrol01.us.oracle.com | http://gll.us.oracle.com:12815 | BRM | BRM | 7.4 | BRM contact | |
| <input type="radio"/> | BRM_03 | BRM_03 | BRM Instance 03 | gll-4.us.oracle.com | http://gll.us.oracle.com:12850 | BRM | BRM | 7.4 | BRM Contact | |

Configuring Routing Rules

This section describes the routing rule changes that must be made to enable interoperability. There may be install sequence considerations.

For more information about install sequence considerations, see [Understanding Install Sequence Dependencies](#). **For more information** about using and extending routing rules, see Oracle Enterprise Service Bus Developer's Guide, Creating Routing Services and Routing Rules.

For CustomerPartyEBSV2

The filter condition text represents the change to make to the existing condition.

| Operation | Filter Condition | Service Invoked | Description |
|------------------------|--|---|---|
| QueryCustomerPartyList | {aia:getServiceProperty('lcb; http://xmlns.oracle.com/EnterpriseServices/Core/CustomParty/V2 rcb; CustomerPartyEBSV2','Routing.ActiveRuleset',false())='DEFAULT' and (/ebo:QueryCustomerPartyListEBM/cor | QueryCustomerPartyListSiebelProvABCSImpV2_1_0::QueryCustomerPartyList | The string 'Siebel' should be changed to 'SIEBEL' in this existing routing rule |

| Operation | Filter Condition | Service Invoked | Description |
|-----------|--|-----------------|-------------|
| | <pre> ecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode='PRODUCTION' or not(/ebo:QueryCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and (/ebo:QueryCustomerPartyListEBM/corecom:EBMHeader/corecom:Target/corecom:ApplicationTypeCode='SIEBEL') };{ namespace aia=http://www.oracle.com/XSL/Transform/java/oracle.apps.aia.core.xpath.AIAFunctions namespace ebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CustomerParty/V2 namespace corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2 } </pre> | | |

For CommunicationCustomerPartyEBSV2

The filter condition text represents the change to make to the existing condition.

| Operation | Filter Condition | Service Invoked | Description |
|-----------------------|--|---|---|
| SyncCustomerPartyList | <pre> count(/ns0:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode[text()='CAVS']) = 0 and count(/ns0:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Sender/corecom:WSAddress/ns4:ReplyTo/ns4:ServiceName[text()='CommsProcessBillingAccountListEBF']) = 0 and /ns0:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Sender/corecom:Application/corecom:ID = 'SIEBEL' and count(/ns0:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Target/corecom:ID)= 0 </pre> | SyncCustomerPartyList BRMCommsJMSProducer::Produce_Message | The string 'Siebel' should be changed to 'SIEBEL' in this existing routing rule |

Manual Redeployment

The Order to Bill PIP and MDM Product PIP can be installed in any sequence. If the MDM Product PIP with its BRM spoke is installed first and then Order to Bill PIP is installed no extra action is required. However if Order to Bill PIP is installed first and then the MDM Product PIP with its BRM spoke is installed then an extra step must be performed before installing MDM PIP.

You must go to `$AIAHOME/config/deployedEBS.properties` file. In this file you see all the EBS services that have been deployed on the server from the respective PIPs.

While installing or deploying an EBS service, AIA deploys it two times. The first time it is installed without routing rules and the second time, with routing rules. As and when an EBS Service is deployed without routing rules an entry is created for the EBS service in the properties file and the value is set to false.

When an EBS service with routing rules is deployed, this property value is set to true. This setting is taken care of by the install scripts.

1. Open this `deployedEBS.properties` file and search for the following entries
 - c. `CommunicationsItemCompositionEBSV1`
 - d. `CommunicationsItemCompositionResponseEBSV1`
 - e. `CommunicationsPriceListEBSV2`
 - f. `CommunicationsPriceListResponseEBSV2`
2. If the entry is set to true then set it to false.

You may also have to manually redeploy all the extra routing rules that you have deployed as part of the implementation of multiple BRM systems.

Index

Comms Agent Assisted Billing Care

- with Customer MDM, 37

Comms Order to Bill

- with Customer MDM, 31

- with Product MDM, 43

Customer MDM

- with Comms Agent Assisted Billing Care, 37

- with Comms Order to Bill, 31

- with Order to Cash, 17

Design to Release

- with Order to Cash, 15

Interoperability

- matrix, 10

- understanding, 9

Order to Cash

- with Customer MDM, 17

- with Design to Release, 15

- with Product MDM, 27

Product MDM

- with Order to Bill, 43

- with Order to Cash, 27