Oracle Customer Data Management Integration Base Pack

Implementation Guide

Release 2.5

E17414-03

January 2012

ORACLE

Oracle Customer Data Management Integration Base Pack Implementation Guide, Release 2.5

E17414-03

Primary Author: Oracle Corporation

Contents

Preface 1

Oracle Application Integration Architecture Foundation Pack Concepts and Technologies Guide 1

Oracle Application Integration Architecture - Foundation Pack: Integration Developer's Guide . 2

Oracle Application Integration Architecture - Foundation Pack: Core Infrastructure Components Guide 2

Oracle Application Integration Architecture Process Integration Packs 3

Additional Resources 3

Chapter 1: Understanding the Oracle Customer Master Data Management Integration 5

Overview 5

Terminology 6

High-Level Integration Flows 7

Match and Fetch Organizations or Persons 8

Synchronizing Organizations or Persons 8

Merge Flows 9

Data Enrichment Flow 9

Core AIA Components 10

AIA Integration Services 10

CustomerPartyEBSV2 11

CustomerPartyResponseEBSV2 11

CustomerPartyOrchestrationEBSV2 12

CustomerPartyOrchestrationResponseEBSV2 12

ProcessCustomerPartyEBS 12

CommunicationsCustomerPartyEBSV2 12

CommunicationsCustomerPartyResponseEBSV2 12

FetchCustomerPartyEBF 13

CommsProcessFulfillmentOrderBillingAccountListEBF 13

CommsProcessBillingAccountListEBF 14

InterfaceContactToAccountEBF 15

Chapter 2: Oracle Customer Master Data Management Integration Base Pack 17

Overview 17

Prerequisites ... 17
Process Flows ... 18
Match and Fetch Organizations and Persons between Oracle Customer Hub and Participating Applications ... 18
Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle E-Business Suite ... 18
Synchronization of Organizations and Persons from Oracle Customer Hub to Siebel CRM 23
Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle BRM 28
Synchronization of Organizations and Persons from Oracle Customer Hub to SAP ... 37
Synchronization of Account Contacts from Oracle Customer Hub to SAP ... 39
Data enrichment flow ... 42
Merge Flow from Oracle Customer Hub to Siebel CRM ... 42
Merge Flow from Oracle Customer Hub to Oracle E-Business Suite ... 46
Merge Flow from Oracle Customer Hub to Oracle BRM ... 46
OCH Integration Services ... 48
SyncOrganizationUCMJMSProducer ... 48
SyncOrganizationUCMJMSConsumer ... 49
SyncPersonUCMJMSProducer ... 49
SyncPersonUCMJMSConsumer ... 49
QueryCustomerPartyListUCMProvABCSImpl ... 49
QueryCustomerPartyUCMProvABCSImpl ... 49
SyncOrganizationUCMReqABCSImpl ... 49
SyncCustomerPartyListResponseUCMProvABCSImpl ... 50
SyncPersonUCMReqABCSImpl ... 50
ProcessPersonUCMReqABCSImpl ... 50
SyncCustomerPartyListUCMProvABCSImpl ... 50
Oracle Customer Hub Interfaces ... 50
Inbound Web Services: Match and Fetch ... 51
Inbound Web Services: Organization and Person Synchronization ... 51
Outbound Web Services: Organization and Person Synchronization ... 51
Inbound Web Services: Account Contact Synchronization ... 51
Outbound Web Services: Account Contact Synchronization ... 52
Assumptions and Constraints ... 52
Chapter 3: Oracle Customer Master Data Management Integration Option for Siebel CRM ... 59
Overview ... 59

Prerequisites ... 59
Process Flows ... 60
Match and Fetch between Siebel CRM and Oracle Customer Hub ... 60
Synchronizing Organizations and Persons from Siebel CRM to Oracle Customer Hub ... 64
Siebel CRM Integration Services ... 69
SyncAccountSiebelAggregatorAdapter ... 70
SyncContactSiebelAggregatorAdapter ... 70
SyncAddressSiebelAggregatorAdapter ... 70
SyncCustomerSiebelEventAggregator ... 70
SyncAccountSiebelReqABCSImpl ... 71
SyncContactSiebelReqABCSImpl ... 71
SyncCustomerPartyListSiebelProvABCSImpl ... 71
SyncCustomerPartyListResponseSiebelProvABCSImpl ... 71
SyncBPSiebelAggregatorAdapter ... 71
SyncAcctSiebelAggrEventConsumer ... 71
SyncContSiebelAggrEventConsumer ... 72
Siebel CRM Interfaces ... 72
Inbound Web Services: Organization and Person Synchronization ... 72
Outbound Web Services: Organization and Person Synchronization ... 72
Inbound Web Services: Match and Fetch ... 73
Outbound Web Services: Match and Fetch ... 73
Chapter 4: Oracle Customer Master Data Management Integration Option for Oracle E-Business Suite ... 75
Overview ... 75
Prerequisites ... 75
Process Flows ... 76
Match and Fetch between Oracle E-Business Suite and Oracle Customer Hub ... 76
Synchronization of Organizations and Persons from Oracle E-Business Suite to Oracle Customer Hub ... 80
Oracle E-Business Suite Integration Services ... 85
SyncCustomerPartyListEbizProvABCSImpl ... 85
QueryPersonAccountEbizUpdateAdapter ... 86
QueryCustomerPartyListEbizCreateAdapter ... 86
QueryCustomerPartyListEbizUpdateAdapter ... 86
SyncPersonAccountEbizReqABCSImpl ... 86

SyncPersonAccountEbizEventConsumer 86
SyncCustomerPartyListEbizEventConsumer 86
SyncCustomerPartyListResponseEbizProvABCSImpl 87
SyncCustomerPartyListEbizAdapter 87
SyncCustomerPartyListPersonEbizAdapter 87
SyncCustomerPartyListEbizReqABCSImpl 87
QueryPersonAccountEbizCreateAdapter 87
Oracle E-Business Suite Interfaces 88
Inbound Web Services: Fetch 88
Outbound from EBS/Composite App Event Interfaces: Match and Fetch 88
Inbound Web Services: Person or Organization Synchronization 88
Outbound from EBS Event Interfaces: Organization or Person Synchronization 88
Chapter 5: Oracle Customer Master Data Management Integration Option for Oracle Communications Billing and Revenue Management 91
Overview 91
Oracle BRM Interfaces 91
Oracle BRM Integration Services 91
CommunicationsCustomerPartyEBSV2 92
SyncCustomerPartyListBRMCommsProvABCSImpl 92
CommunicationsCustomerPartyResponseEBSV2 92
SyncCustomerPartyListBRMCommsJMSProducer 92
SyncCustomerPartyListBRM_01CommsJMSConsumer 92
Chapter 6: Oracle Customer Master Data Management Integration Option for SAP 93
Overview 93
Prerequisites 93
Process Flows 93
Synchronizing Organizations and Persons from SAP to OCH 93
SAP Interfaces 96
Inbound Web Services: Organization and Person Synchronization 96
Outbound Web Services: Organization and Person Synchronization 96
Inbound Web Services: Account Contact Synchronization 96
Outbound Web Services: Account Contact Synchronization 96
SAP Integration Services 97
SyncCustomerSAPReqABCSImpl 97
SyncCustomerPartyListSAPProvABCSImpl 97

SyncCustomerPartyListResponseSAPProvABCSImpl 97
SyncCustomerSAPReqABCSImpl 97
Chapter 7: Implementing the Process Integration Pack for Oracle Customer Hub 99
Configuring Siebel CRM 99
Setting Up Organizations 101
Oracle Customer Hub 101
Obtaining Oracle E-Business Suite Operating Unit IDs 102
Obtaining SAP Sales Areas 102
Mapping Siebel CRM Organizations 103
Setting Up Cross References 103
Identifying Siebel Row IDs 103
Identifying Oracle E-Business Suite Entities 104
Identifying SAP Entities 104
Identifying Oracle Customer Hub Row IDs 104
Populating Cross-References 105
Validating Cross-References 106
Identifying Cross-References 106
Describing Domain Value Maps 108
Creating Oracle E-Business Suite System Profiles 110
Creating SAP System Profiles 110
Scheduling Concurrent Processes 111
Oracle AS Adapter 111
Enabling SAP Events 111
Configuring Sequencing for the Synchronize Flows from Siebel CRM to Oracle Customer Hub 112
Handling Errors 113
Using the Composite Application Framework 113
Viewing EBO Implementation Maps (EIMs) 113
Setting Configuration Properties 114
Chapter 8: Customer Data Enrichment 141
Overview 141
Prerequisites 141
Process Flow 142
Data Enrichment Integration Services 144
Cross-References 145

Domain Value Maps 145
Handling Errors 146
Viewing EBO Implementation Maps (EIMs) 146
Setting Configuration Properties 146
Index 149

Preface

This preface discusses:

- 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 Oracle Customer Master Data Management Integration

This chapter provides an overview of the Oracle Customer Hub Process Integration pack and covers:

- High-Level Integration Flows
- Core AIA Components
- AIA Integration Services

Overview

The Oracle Customer Hub PIP solution provides an enterprise-level customer master integrated solution for publishing customer data updates from OCH to the participating applications covered as part of the OCH PIP.

You have the option to select one or more participating applications when installing Oracle Customer Hub PIP. Records are persisted in the local applications and are then synchronized to Oracle Customer Hub in realtime or periodic, batch-enabled mode so that the participating applications have the most up-to-date definition of the customer and related entities.

Oracle Customer Hub PIP provides a partial flow to a third-party customer-data enrichment provider that cleanses, recognizes, enriches, and protects Oracle Customer Hub contact records. This flow ensures that the customer-related entities that are mastered and published to subscribing systems truly represent the enriched, single, conformed view of the customer.

The integration flows delivered with this PIP are an effort to harness these powerful features as part of a pre-built, integrated solution between the OCH and the participating applications.

For more information about installing and selecting the participating applications (integration options) based on your business need, refer *AIA Oracle Application Integration Architecture 2.5: Installation and Upgrade Guide*.

Note: Please read the chapters pertaining to each integration option for details about the supported process flows, integration services, and interfaces pertaining to that option (application).

Terminology

The Oracle Customer Hub PIP can be thought of as a collection of core services used to synchronize customer (and prospect) entities across the participating applications.

The term *customer* as referenced throughout this document describes an entity (organization or person) with which the company providing services may have or may enter into a financial relationship.

Here are some definitions that are helpful when reading this implementation guide:

Term	Definition
Party	Siebel CRM: High-level entity that includes persons, accounts, business units, and households. No functional usage of party in Siebel. Oracle E-Business Suite: based on Trading Community Architecture (TCA), a party is defined as any individual or organization with which the implementing organization can do business. A party in Oracle TCA could be a customer, (in case a selling relationship has been established) or vendor, employee, or relationship, all defined by a "Party Type." The same party can have multiple such roles within EBS.
Contact	Siebel CRM: A contact describes an individual (person) with whom a company expects to do business. Contacts can be related to an account (organization customer) or another contact (individual customer). Contacts can be associated to a primary account and additional non-primary accounts. Oracle E-Business Suite: A contact describes a specific relationship between two parties. The relationship between organization and a person party, which is also called an organization contact. Can also define the relationship between a party and an account (role type = contact in hz_cust_acct_roles).
Account	Siebel CRM: An account is an organization (or organization subtype) that can be sold or serviced. An account can be associated to one or many contacts or persons but only one contact can be primary. Oracle E-Business Suite: An account describes the specific attributes of a party that are relevant to the financial relationship among parties. An account in Oracle E-Business Suite cannot exist by itself without a party. An account in EBS can be associated with an individual (person) or a company (organization) but cannot in it and of itself be an individual or organization.
Customer	Siebel CRM: A customer can be an account (org customer) or a contact (person customer). Both entities can be associated to an order. Oracle E-Business Suite: A customer is a combination of party and account and represents a person or organization with which the implementing organization has established a financial relationship. All relevant information of the specific financial relationship with a party is modeled in the account layer entities in Oracle, whereas all the base information like name, address information, contact points are modeled in the party layer entities. Siebel account = Oracle E-Business Suite organization party + Oracle E-Business Suite account Siebel contact = Oracle E-Business Suite person party + Oracle E-Business Suite account This integration pack supports the notion of a "customer" in both B2B and B2C environment.

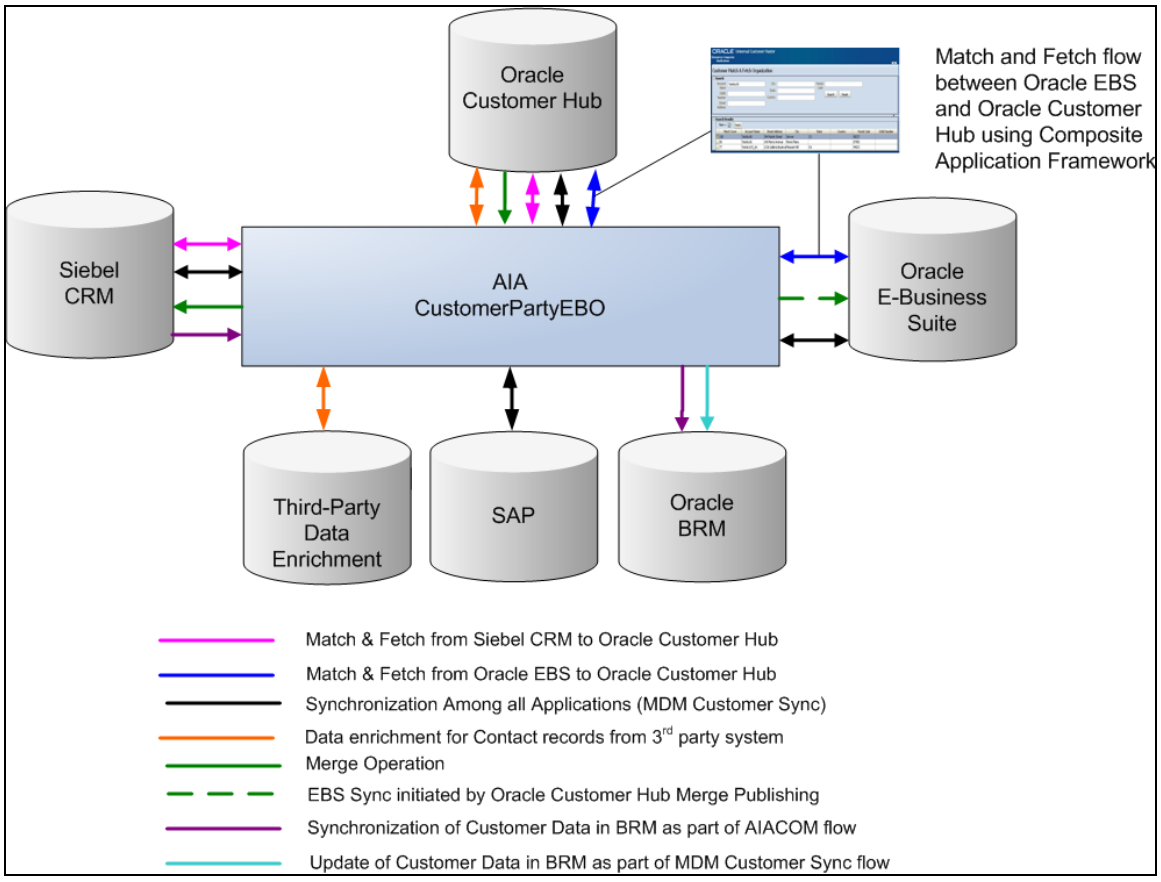
Note: To standardize terminology, we use the organization and person naming conventions throughout the documentation.

This convention is consistent with the entity terms used in the CustomerPartyEBO as well as the terminology in the Oracle E-Business Suite TCA model, but departs from the terminology used in Siebel CRM and Oracle Customer Hub. The Oracle Customer Hub web services that have been developed to support this PIP adopt the organization and person terminology for the service and schema definitions, even though the underlying integration objects being referenced continue to use the legacy terms of account and contact respectively.

Documentation	Siebel CRM	Oracle Customer Hub	Oracle E-Business Suite	AIA CustomerPartyEBO	Oracle BRM	SAP
Organization	Account	Account	Organization	Organization	Account	Account
Person	Contact	Contact	Person	Person	Contact	Account

High-Level Integration Flows

Note: This diagram depicts the flows among all the participating applications; however, the customers have the flexibility to select the participating applications based on the business needs.



Oracle Customer Hub PIP high-level integration flows

At a high-level, the Oracle Customer Hub (OCH) process integration pack supports these process flows using the AIA architectural framework:

- Match and Fetch Organizations or Persons
- Synchronizing Organizations or Persons
- Merge Flows
- Data Enrichment Flow

Match and Fetch Organizations or Persons

The match functionality enables users to enter a complete or partial customer record in a source application and send the record to the Oracle Customer Hub so that matching logic can be executed against the record. Oracle Customer Hub returns a list of potential match candidates that are presented to the source application prior to a customer record being committed to the source application database. The match customer service invokes the fuzzy-matching capabilities of Oracle Customer Hub so that an all-inclusive list of candidates is returned. The fuzzy-matching service either supplements or replaces a normal search query executed against the local database to locate a particular customer account or contact.

The fetch process enables users to select a specific record from the list of candidates returned by Oracle Customer Hub in the match process and send a fetch back to Oracle Customer Hub to retrieve and return the full customer profile of the selected record. Oracle Customer Hub returns the latest information for that customer record which either creates a new record in the source application or updates an existing record in the source application.

The process flows for match and fetch covered under the OCH PIP are described in the chapters pertaining to the components that are involved in the match and fetch flows.

- See: Chapter 3: Oracle Customer Master Data Management Integration Option for Siebel CRM for match and fetch between OCH and Siebel CRM.
- See: Chapter 4: Oracle Customer Master Data Management Integration Option for Oracle E-Business Suite for match and fetch between OCH and Oracle E-Business Suite.

Synchronizing Organizations or Persons

These flows enable you to synchronize new or updated records from a particular source to one or many target systems. The source can be the Oracle Customer Hub application that invokes the synchronization process to create or update records in the participating applications, or the source can be one of the participating applications that invoke the synchronization process to create or update records in Oracle Customer Hub. The synchronization represents a single service to perform create or an update call depending on the existence of the customer in the source and target applications.

These integration flows can be triggered manually or as part of an automated process, and is leveraged as part of the match/fetch process.

The synchronization flows for organization and persons covered under the OCH PIP are described in the chapters pertaining to the components that are involved in the synchronization flows.

- See: Chapter 2: Oracle Customer Master Data Management Integration Base Pack for synchronization flows from OCH to the participating applications.

- See: Chapter 3: Oracle Customer Master Data Management Integration Option for Siebel CRM for synchronization flows from Siebel CRM to OCH.
- See: Chapter 4: Oracle Customer Master Data Management Integration Option for Oracle E-Business Suite for synchronization flows from Oracle E-Business Suite to OCH.
- See: Chapter 5: Oracle Customer Master Data Management Integration Option for Oracle Communications Billing and Revenue Management for synchronization flows from Oracle BRM to OCH.
- See: Chapter 6: Oracle Customer Master Data Management Integration Option for SAP for synchronization flows from SAP to OCH.

Merge Flows

A core feature in Oracle Customer Hub is the ability for a data steward to merge customer records. When the data steward merges duplicate account or contact records based on data quality and data cleansing functionality applied to the records, OCH publishes a merge message, which contains the survivor record (to be updated) and the victim record (to be deleted). This merge by data steward invokes functionality in OCH to perform merge of customer victim record(s) into customer surviving record and update the OCH cross-referencing information and source history tables. The merge messages must include the victim record and the surviving record profile.

Merging account or contact records in OCH can be triggered manually or automatically.

The merge customer request published by OCH is conditionally or optionally adopted by the subscribing applications. While most applications typically subscribe to a merge message from OCH, not all subscribing applications may have the ability to implement the merge of survivor and victims records. Without this ability to implement an application merge, there is a significant risk that records remain out of sync. This risk may amount to updates to victim records in applications that do not subscribe to the merge message and may cause updates to victim records in OCH.

By default, the Customer MDM PIP merges the customer data in Siebel when a merge is published by the Oracle Customer Hub; however, you can turn off this default behavior should you have a business need when interoperating with applications that do not support a merge process.

To turn off merge publishing by Oracle Customer Hub, set the process property EnablePubSub to *False* for the UCM Process Merge Request workflow.

For more information about the workflow, refer to *Siebel Book Shelf: Siebel Business Process Framework: Workflow Guide*.

The merge flows from the OCH to the participating applications as part of the OCH PIP are described in Chapter 2: Oracle Customer Master Data Management Integration Base Pack.

Data Enrichment Flow

Data enrichment represents a spoke service to the CustomerPartyEBO to make a request for clean or enriched information from third-party data provider services. This third-party provider cleanses, recognizes, enriches, and protects Oracle Customer Hub contact records.

Data enrichment is described in Chapter 8: Customer Data Enrichment.

Note. Please read the chapters pertaining to each participating application for details about the supported process flows, integration services, and interfaces pertaining to that application.

Core AIA Components

These are the core AIA components as part of OCH PIP:

- CustomerPartyEBO
- QueryCustomerPartyEBM
- QueryCustomerPartyResponseEBM
- QueryCustomerPartyListEBM
- QueryCustomerPartyListResponseEBM
- SyncCustomerPartyListEBM
- SyncCustomerPartyListResponseEBM
- ProcessCustomerPartyListEBM

The core EBO and EBM XSD files can be located by EBO within this parent folder: 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/.

For detailed documentation of individual EBOs, click the **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 utilize 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 instance, to add new data elements. These extensions are protected and remain intact after 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:

- CustomerPartyEBSV2

- CustomerPartyResponseEBSV2
- CustomerPartyOrchestrationEBSV2
- CustomerPartyOrchestrationReponseEBSV2
- ProcessCustomerPartyEBS
- CommunicationsCustomerPartyEBSV2
- CommunicationsCustomerPartyResponseEBSV2
- FetchCustomerPartyEBF
- CommsProcessFulfillmentOrderBillingAccountListEBF
- CommsProcessBillingAccountListEBF
- InterfaceContactToAccountEBF

CustomerPartyEBSV2

The match and fetch flows use some of the same services as the synchronization flows.

The CustomerPartyEBSV2 is implemented as a lightweight ESB routing service that exposes all of the enterprise operations that can be performed with a CustomerParty enterprise object. All of the Oracle Customer Hub PIP integration flows make use of the operations provided by this enterprise business service.

This service is deployed as part of the MDM installation. A few routing rules are added to route the messages as part of the Oracle BRM option for OCH. These routing rules route the messages from Siebel and OCH to Oracle BRM, as appropriate.

Oracle Customer Hub PIP synchronization flows use the SyncCustomerPartyList operation of CustomerPartyEBSV2.

CustomerPartyResponseEBSV2

The match and fetch flows use some of the same services as the synchronization flows.

The CustomerPartyResponseEBSV2 is implemented as a lightweight ESB routing service that exposes all of the enterprise response operations that can be performed with a CustomerParty enterprise object. All of the Oracle Customer Hub PIP integration flows make use of the response operations provided by this enterprise business service.

Oracle Customer Hub PIP synchronization flows use the SyncCustomerPartyListResponse operation of CustomerPartyResponseEBSV2.

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"

CustomerPartyOrchestrationEBSV2

The CustomerPartyOrchestrationEBSV2 is the front-end interface to the FetchCustomerPartyEBF. AIA guidelines suggest that EBFs be called only by other EBFs or an EBS. These operations are used by this service:

- AsyncFetchCustomerParty
- FetchCustomerParty

CustomerPartyOrchestrationResponseEBSV2

The CustomerPartyOrchestrationResponseEBSV2 exposes the callback operation, which is invoked by the FetchCustomerPartyEBF after the synchronization has completed. This service routes back to the original caller instance. This service uses AsyncFetchCustomerPartyResponse operation.

ProcessCustomerPartyEBS

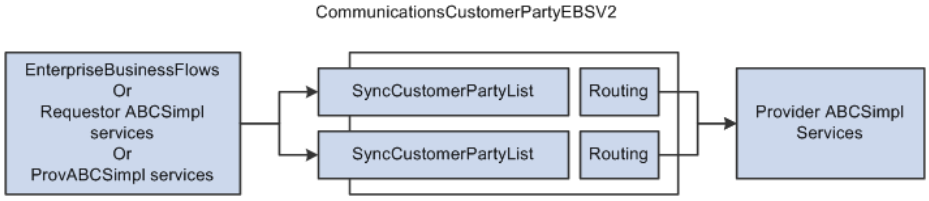
This ESB routing service routes the ProcessCustomerPartyListEBM to the third-party provider and passes the response EBM back to the Oracle Customer Hub requester.

CommunicationsCustomerPartyEBSV2

The CommunicationsCustomerPartyEBSV2 enterprise business service exposes all of the enterprise operations that can be performed with a CustomerParty enterprise object for Communication industry. The integration uses these operations provided by the CommunicationsCustomerPartyEBSV2:

- QueryCustomerPartyList
- SyncCustomerPartyList

This diagram illustrates the relationship of CommunicationsCustomerPartyEBSV2 with the other services in the integration flow:



CommunicationsCustomerPartyEBSV2

CommunicationsCustomerPartyResponseEBSV2

CommunicationsCustomerPartyResponseEBSV2 exposes all of the enterprise response operations that can be performed with a CustomerParty enterprise object. All of the customer management integration flows use the operations provided by this enterprise business service.

This diagram illustrates the relationship of CommunicationsCustomerPartyResponseEBSV2 with the other services in the integration flow:

CommunicationsCustomerPartyResponseEBSV2

For more information about communications-specific services or integrations, refer *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.

FetchCustomerPartyEBF

The FetchCustomerPartyEBF orchestrates the query request to the customer master provider (Oracle Customer Hub) and the subsequent synchronization of that customer to the requesting system (Oracle E-Business Suite and Siebel CRM). This EBF supports both asynchronous and synchronous interaction patterns. The asynchronous pattern is used by Siebel requests with a callback using the CustomerPartyResponse EBS. The synchronous pattern is used by Oracle E-Business Suite for its request and response interaction.

For more information about EBFs, see Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide, "Designing and Constructing Enterprise Business Flows" and Oracle Application Integration Architecture – Foundation Pack: Concepts and Technologies Guide, "Enterprise Business Flow Processes."

CommsProcessFulfillmentOrderBillingAccountListEBF

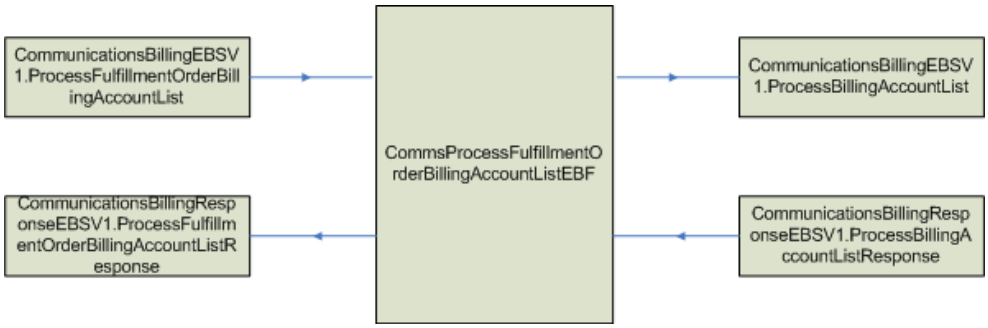
This Enterprise Business Flow (EBF) extracts the CustomerData from OrderEBM. The process loops through every order line and extracts any customer account or billing profile that it encounters.

This service has two operations: The first operation accepts the ProcessFulfillmentOrderBillingAccountListEBM and is used by the process to order data, and the other operation is used by the process to send the response back to the calling process (using the ProcessFulfillmentOrderBillingAccountListEBM).

The transformations include:

- ProcessFulfillmentOrderBillingAccountList to ResponseEBM.xsl
- ProcessFulfillmentOrderBillingAccountListEBM to ProcessBillingAccountListEBM.xsl

This diagram illustrates the relationship of CommsProcessFulfillmentOrderBillingAccountListEBF with the other services in the integration flow:



CommsProcessFulfillmentOrderBillingAccountListEBF

The CommsProcessFulfillmentOrderBillingAccountListEBF enterprise business flow is implemented as an asynchronous delayed response BPEL process.

For more information about communications-specific services or integrations, refer *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.

CommsProcessBillingAccountListEBF

This EBF service creates or synchronizes all the customer accounts and billing profiles in an appropriate billing system. The Order Processing integration flow invokes this service with a list of Customer Account IDs, Billing Profile IDs, and the Target System ID. When the process is complete, a response is sent back to the order flow confirming that all accounts have been set up in the target billing system, and the order processing can continue.

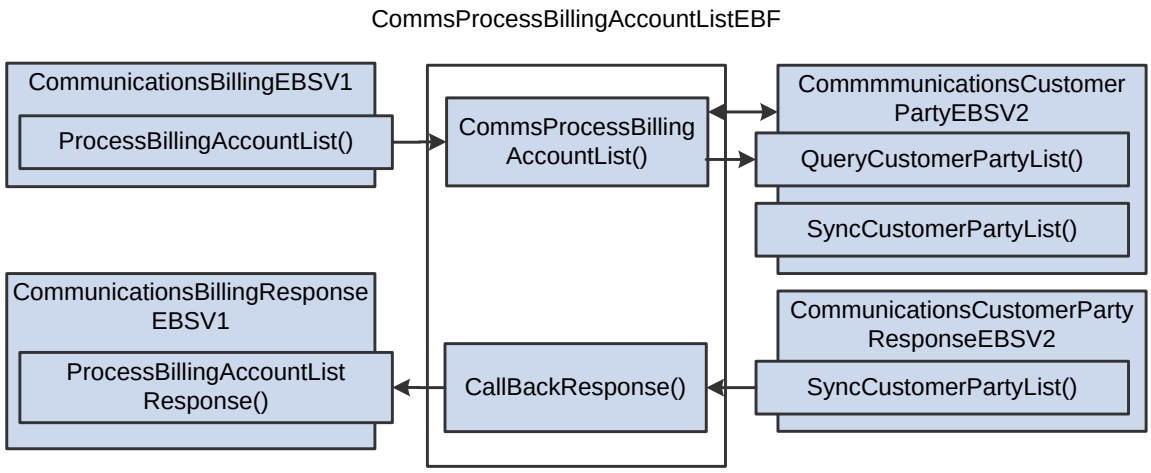
This service provides two operations. The first operation accepts the ProcessBillingAccountListEBM, and is used by the process to receive the customer data to be synchronized. The other operation is used by the process to send the response back to the calling process (using the SyncCustomerPartyListResponseEBM).

The data area of the message contains one or more customer account IDs. For each account, one or more bill profile IDs must be synchronized to the target billing system. The customer data indicates both the hierarchical and the paying relationships between the accounts.

This service creates or synchronizes one or more customers (identified by ID only) and their billing profiles to a particular target billing system (identified in the EBM header). Therefore, the responsibilities of this service include:

- To determine whether the customer already exists and is up to date in the target billing system. If so, it optimizes and does not try to create or synchronize.
- If needed, retrieve the customer data from the appropriate Siebel CRM system using the provided IDs.
- To optimize, if possible, the number and size of queries back into Siebel CRM for the customer data.
- To create or to update the customers and billing profiles in the target billing system, reflecting the customer hierarchy, and paying relationships among the customers.

This graphic illustrates the relationship of the CommsProcessBillingAccountListEBF with the other services in the integration flow:



CommsProcessBillingAccountListEBF

InterfaceContactToAccountEBF

InterfaceContactToAccountEBF is an asynchronous BPEL process, which routes the account message from the CustomerPartyEBSV2 to CommunicationsCustomerPartyEBSV2 when a contact is updated in OCH. This service sends the account details from the payload (one message at a time) to EBS services, which in turn synchronizes it to Oracle BRM. Upon receiving the account list, InterfaceContactToAccountEBF service performs these validations on the accounts:

- It checks whether the account is for an update in BRM.
- It ensures that the associated contact for the account is the primary contact for the account.

If both these validations pass, CommunicationsCustomerPartyEBSV2 routes the message for further processing.

Chapter 2: Oracle Customer Master Data Management Integration Base Pack

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

- Process Flows
- OCH Integration Services
- Oracle Customer Hub Interfaces
- Assumptions and Constraints

Overview

Oracle Customer Hub (also called Siebel UCM) enables organizations to centralize information from heterogeneous systems, creating a single view of customer information that can be leveraged across all functional departments and analytical systems. Oracle Customer Hub provides a single, consolidated system of record for the definition of customer-related entities such as contacts, addresses, and accounts as well as the relationships within these entities. Oracle Customer Hub can also serve as the system of entry for a Data Steward to update customer records directly in Oracle Customer Hub and publish this customer definition to all subscribing applications.

For more information about Oracle Customer Hub, see Siebel UCM product documentation.

Prerequisites

For Match and Fetch flows:

- All participating applications must be set up and working properly.

  On the Oracle Customer Hub side, this includes Data Quality services even when provided by a third-party application.

- For the match/fetch integration flow between Oracle E-Business Suite and Oracle Customer Hub, you must install ADF runtime.

For more information about installing the ADF runtime, see the My Oracle Support note 746109.1: *Deploying the Universal Customer Master Match/Fetch Composite User Interface*.

For synchronization flows:

- SAP system must be registered in OCH system registration to provide create and update privileges to SAP.

- OCH configuration determines the method of publishing as real-time or batch.
- Cross-references for organizations are manually set up.

For merge flows:

- The prerequisite is that accounts/contacts have been synchronized between Oracle Customer Hub and Siebel CRM/Oracle E-Business Suite prior to publishing the merge from Oracle Customer Hub.

Process Flows

These are the process flows from Oracle Customer Hub (OCH):

- Match and Fetch Organizations and Persons between Oracle Customer Hub and Participating Applications
- Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle E-Business Suite
- Synchronization of Organizations and Persons from Oracle Customer Hub to Siebel CRM
- Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle BRM
- Synchronization of Organizations and Persons from Oracle Customer Hub to SAP
- Synchronization of Account Contacts from Oracle Customer Hub to SAP
- Data enrichment flow
- Merge Flow from Oracle Customer Hub to Siebel CRM
- Merge Flow from Oracle Customer Hub to Oracle E-Business Suite
- Merge Flow from Oracle Customer Hub to Oracle BRM

Match and Fetch Organizations and Persons between Oracle Customer Hub and Participating Applications

The match and fetch flow between Oracle Customer Hub and a participating application have been covered in the chapter specific to the participation application involved in the flow.

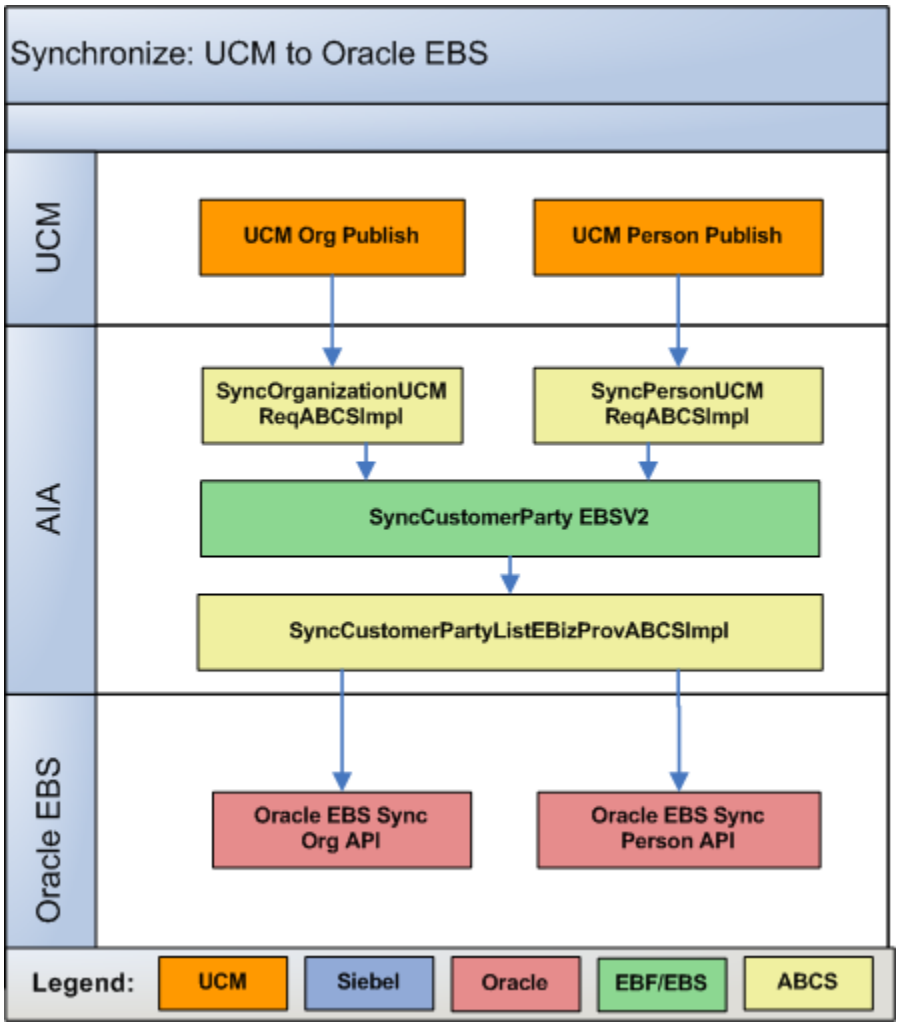
- See: Match and Fetch between Siebel CRM and Oracle Customer Hub for match and fetch between Oracle Customer Hub and Siebel CRM.
- See: Match and Fetch between Oracle E-Business Suite and Oracle Customer Hub for match and fetch between Oracle Customer Hub and Oracle E-Business Suite.

Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle E-Business Suite

This integration flow synchronizes organization and person parties and customer accounts from Oracle Customer Hub to Oracle E-Business Suite. The single synchronize service that is invoked publishes a message using AIA to the subscribing Oracle E-Business Suite application(s).

This integration flow can be triggered manually or as part of an automated process, and is leveraged as part of the match/fetch process.

This activity diagram illustrates the synchronization of organizations and persons from Oracle Customer Hub to Oracle E-Business Suite:



Synchronizing organizations and persons from Oracle Customer Hub to Oracle E-Business Suite

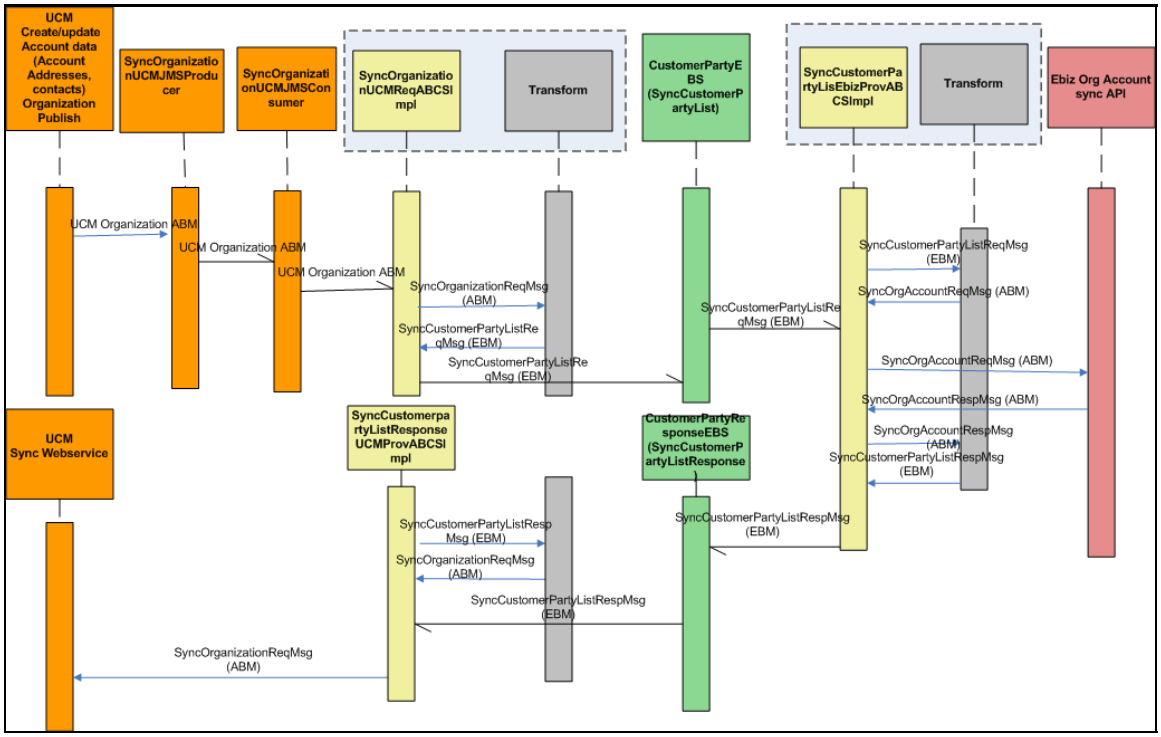
Synchronizing Organizations from Oracle Customer Hub to Oracle E-Business Suite

This integration flow synchronizes new accounts or updates from Oracle Customer Hub to Oracle E-Business Suite. This integration flow uses these interfaces:

- SyncOrganizationUCMJMSProducer
- SyncOrganizationUCMJMSConsumer
- SyncOrganizationUCMReqABCSImpl
- CustomerPartyEBSV2

- SyncCustomerPartyListEbizProvABCSImpl
- CustomerPartyResponseEBSV2
- SyncCustomerPartyListResponseUCMProvABCSImpl

This sequence diagram illustrates the integration flow:



Synchronizing organizations from Oracle Customer Hub to Oracle E-Business Suite

When you initiate this process, these events occur:

1. Oracle Customer Hub publishes messages that initiate the SyncOrganizationUCMJMSProducer, with the UCMAccountABM message.
2. The SyncAccountUCMJMSProducer queues the Oracle Customer Hub Account ABM message onto a JMS queue.
3. The message in the queue is picked up by the SyncAccountUCMJMSConsumer service, which then routes the message to the SyncAccountUCMReqABCSImpl.
4. The Oracle Customer Hub Requester ABC implementation, SyncAccountUCMReqABCSImpl, transforms the Oracle Customer Hub ABM to the SyncCustomerPartyListEBM and invokes the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

   When there are multiple accounts in the message, the message has to be transformed into several EBM messages with accounts with the same target or one EBM per account with all its targets. In either case, there are multiple calls to the EBS from the SyncAccountUCMReqABCSImpl.

5. Invoking the CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Oracle E-Business Suite provider ABC implementation service, SyncCustomerPartyListEbizProvABCSImpl.

6. The Oracle E-Business Suite provider ABC implementation service, SyncCustomerPartyListEbizProvABCSImpl, transforms the SyncCustomerPartyListEBM into the appropriate Oracle E-Business Suite organization account ABM and invokes the Oracle E-Business Suite API.
7. The response from the Oracle E-Business Suite API is then transformed into the SyncCustomerPartyListResponseEBM in the process of which, the cross references tables are populated with the Oracle E-Business Suite identifier values if not already present.

   The response is also sent to the CustomerPartyResponseEBSV2, which routes the message to the SyncCustomerPartyListResponseUCMProvABCSImpl to set the Oracle E-Business Suite ID within Oracle Customer Hub.

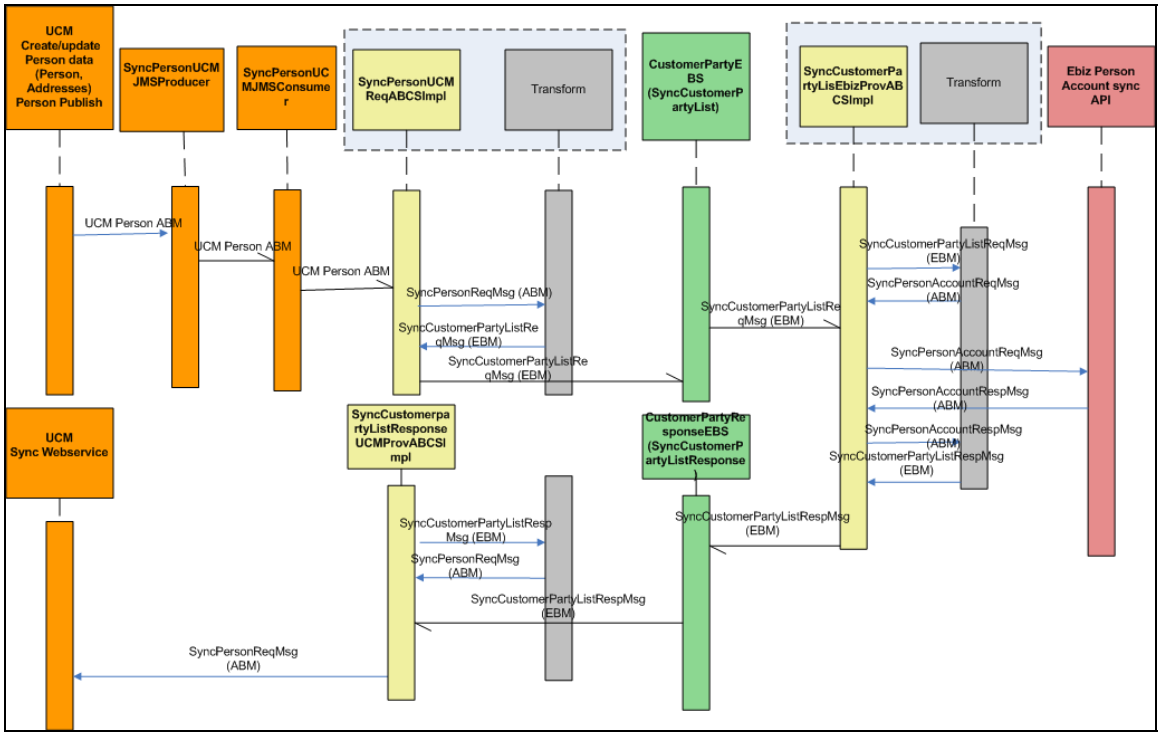
8. The SyncCustomerPartyListResponseUCMProvABCSImpl transforms the SyncCustomerPartyListResponseEBM into the Oracle Customer Hub organization ABM and synchronizes the external ID into Oracle Customer Hub.

Synchronizing Persons from Oracle Customer Hub to Oracle E-Business Suite

This integration flow is used to synchronize new persons or updates from Oracle Customer Hub to Oracle E-Business Suite. This integration flow uses these interfaces:

- SyncPersonUCMJMSProducer
- SyncPersonUCMJMSConsumer
- SyncPersonUCMReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListEbizProvABCSImpl
- CustomerPartyResponseEBSV2
- SyncCustomerPartyListRsponseUCMProvABCSImpl

This sequence diagram illustrates the integration flow:



Synchronizing persons from Oracle Customer Hub to Oracle E-Business Suite

When you initiate this process, these events occur:

1. Oracle Customer Hub publishes messages that initiate the SyncPersonUCMJMSProducer, with the UCMPersonABM message.
2. The SyncPersonUCMJMSProducer queues the Oracle Customer Hub person ABM message onto a JMS queue.
3. The message in the queue is picked up by the SyncPersonUCMJMSConsumer service, which then routes the message to the SyncPersonUCMReqABCSImpl.
4. The Oracle Customer Hub requester ABC implementation, SyncPersonUCMReqABCSImpl, transforms the Oracle Customer Hub ABM into the SyncCustomerPartyListEBM and invokes the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

   When there are multiple persons in the message, the message has to be transformed into several EBM messages with persons with the same target or one EBM per person with all its targets. In either case, there are multiple calls to the EBS from the SyncPersonUCMReqABCSImpl.

5. Invoking the CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Oracle E-Business Suite provider ABC implementation service, SyncCustomerPartyListEbizProvABCSImpl.
6. The Oracle E-Business Suite provider ABC implementation service, SyncCustomerPartyListEbizProvABCSImpl, transforms the SyncCustomerPartyListEBM to the appropriate Oracle E-Business Suite person account ABM and invokes the Oracle E-Business Suite API.

7. The response from the Oracle E-Business Suite API is then transformed into the SyncCustomerPartyListResponseEBM in the process of which the cross references tables are populated with the Oracle E-Business Suite identifier values if they are not already present.

   The response is also sent to the CustomerPartyResponseEBSV2, which routes the message to the SyncCustomerPartyListResponseUCMProvABCSImpl to set the Oracle E-Business Suite ID within Oracle Customer Hub.

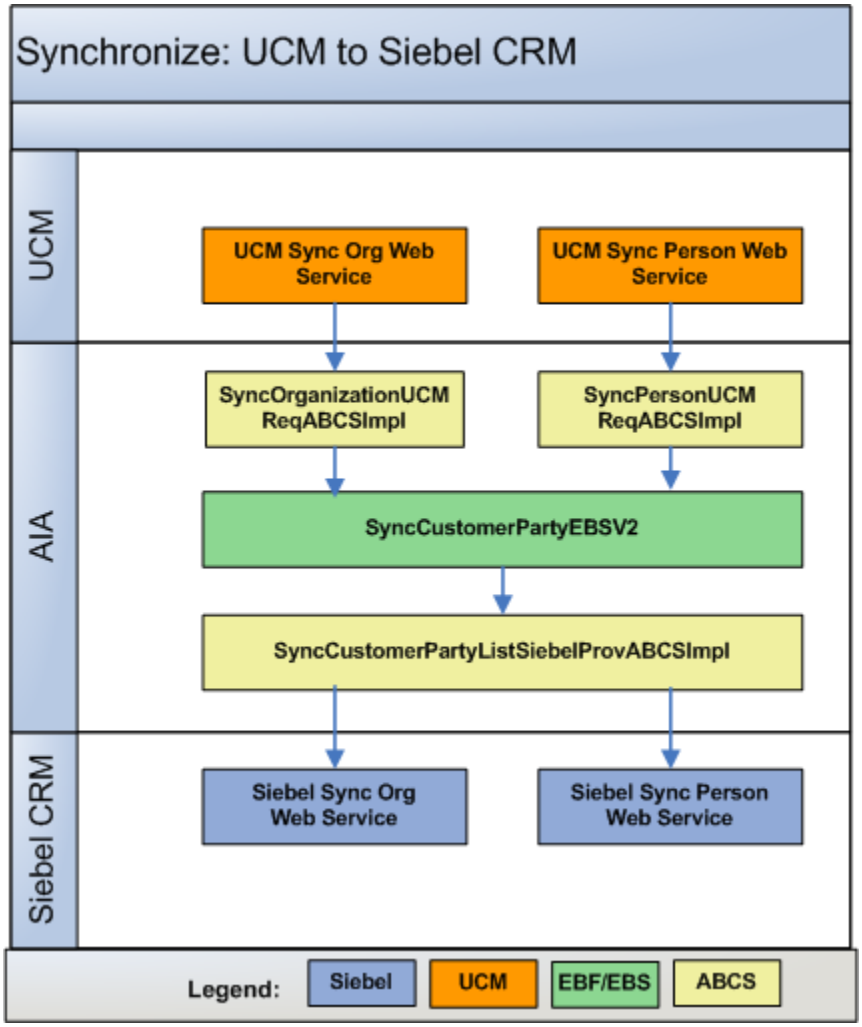
8. The SyncCustomerPartyListResponseUCMProvABCSImpl transforms the SyncCustomerPartyListResponseEBM into the Oracle Customer Hub person ABM synchronizes the external ID into Oracle Customer Hub.

Synchronization of Organizations and Persons from Oracle Customer Hub to Siebel CRM

This integration flow enables the synchronization of accounts and contacts from Oracle Customer Hub to Siebel CRM. This integration flow can be triggered manually or as part of an automated process.

Siebel CRM is expected to subscribe to all synchronization messages that Oracle Customer Hub produces.

This activity diagram illustrates the synchronization of organizations and persons from Oracle Customer Hub to Siebel CRM:



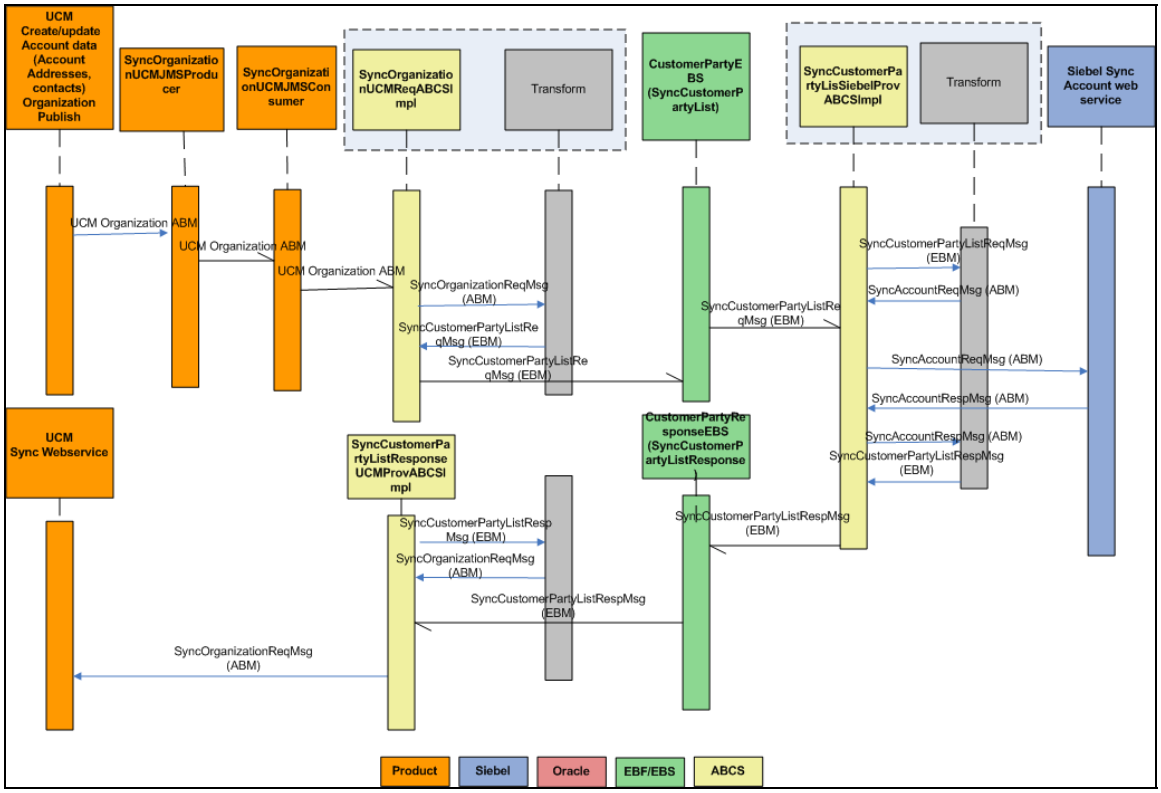
Synchronizing organizations and persons from Oracle Customer Hub to Siebel CRM

Synchronizing Organizations from Oracle Customer Hub to Siebel CRM

This integration flow is used to synchronize new account or updates from Oracle Customer Hub to Siebel. This integration flow uses these interfaces:

- SyncOrganizationUCMJMSProducer
- SyncOrganizationUCMJMSConsumer
- SyncOrganizationUCMReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListSiebelProvABCSImpl
- CustomerPartyResponseEBSV2
- SyncCustomerPartyListResponseUCMProvABCSImpl

This sequence diagram illustrates the integration flow:



Synchronizing organizations from Oracle Customer Hub to Siebel CRM

When you initiate this process, these events occur:

1. Oracle Customer Hub publishes messages that initiate the SyncOrganizationUCMJMSProducer, with the UCMAccountABM message.
2. The SyncAccountUCMJMSProducer queues the Oracle Customer Hub account ABM message onto a JMS queue.
3. The message in the queue is picked up by the SyncAccountUCMJMSConsumer service, which then routes the message to the SyncAccountUCMReqABCSImpl.
4. The Oracle Customer Hub requester ABC implementation, SyncAccountUCMReqABCSImpl, transforms the Oracle Customer Hub ABM into the SyncCustomerPartyListEBM and invokes the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

   When there are multiple accounts in the message, the message has to be transformed into several EBM messages for accounts with the same target or one EBM per account with all its targets. In either case, there are multiple calls to the EBS from the SyncAccountUCMReqABCSImpl.

5. Invoking the CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Siebel provider ABC implementation service, SyncCustomerPartyListSiebelProvABCSImpl.
6. The Siebel provider ABC implementation service, SyncCustomerPartyListSiebelProvABCSImpl, transforms the SyncCustomerPartyListEBM into the appropriate Siebel account ABM and invokes the Siebel web service.

7. The response from the Siebel web service is then transformed into the SyncCustomerPartyListResponseEBM in the process of which, the cross references tables are populated with the Siebel identifier values if not already present.

   The response is also sent to the CustomerPartyResponseEBSV2, which routes the message to the SyncCustomerPartyListResponseUCMProvABCSImpl to set the Siebel ID within Oracle Customer Hub.

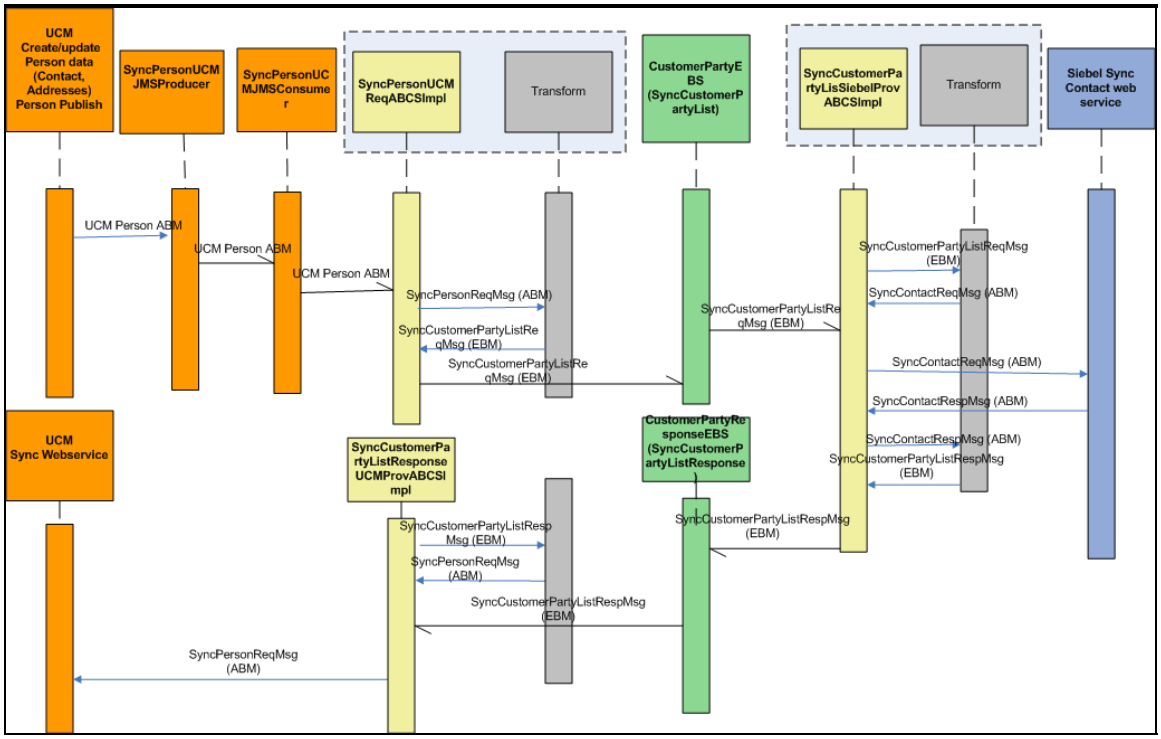
8. The SyncCustomerPartyListResponseUCMProvABCSImpl transforms the SyncCustomerPartyListResponseEBM into the Oracle Customer Hub organization ABM synchronizes the Siebel ID into Oracle Customer Hub.

Synchronizing Persons from Oracle Customer Hub to Siebel CRM

This integration flow is used to synchronize new persons or updates from Oracle Customer Hub to Siebel CRM. This integration flow uses these interfaces:

- SyncPersonUCMJMSProducer
- SyncPersonUCMJMSConsumer
- SyncPersonUCMReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListSiebelProvABCSImpl
- CustomerPartyResponseEBSV2
- SyncCustomerPartyListRsponseUCMProvABCSImpl

This sequence diagram illustrates the integration flow:



Synchronizing persons from Oracle Customer Hub to Siebel CRM

When you initiate this process, these events occur:

1. Oracle Customer Hub publishes messages that initiate the SyncPersonUCMJMSProducer, with the UCMPersonABM message.
2. The SyncPersonUCMJMSProducer queues the Oracle Customer Hub person ABM message onto a JMS queue.
3. The message in the queue is picked up by the SyncPersonUCMJMSConsumer service, which then routes the message to the SyncPersonUCMReqABCSImpl.
4. The Oracle Customer Hub requester ABC implementation, SyncPersonUCMReqABCSImpl, transforms the Oracle Customer Hub ABM into the SyncCustomerPartyListEBM and invokes the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

   When there are multiple persons in the message, the message has to be transformed into several EBM messages with persons with the same target or one EBM per person with all its targets. In either case, there are multiple calls to the EBS from the SyncPersonUCMReqABCSImpl.

5. Invoking the CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Siebel provider ABC implementation service, SyncCustomerPartyListSiebelProvABCSImpl.
6. The Siebel provider ABC implementation service, SyncCustomerPartyListSiebelProvABCSImpl, transforms the SyncCustomerPartyListEBM into the appropriate Siebel contact ABM and invokes the Siebel web service.

7. The response from the Siebel web service is then transformed into the SyncCustomerPartyListResponseEBM in the process of which, the cross references tables are populated with the Siebel identifier values if not already present.

   The response is also sent to the CustomerPartyResponseEBSV2, which routes the message to the SyncCustomerPartyListResponseUCMProvABCSImpl to set the Siebel ID within Oracle Customer Hub.

8. The SyncCustomerPartyListResponseUCMProvABCSImpl transforms the SyncCustomerPartyListResponseEBM into the Oracle Customer Hub person ABM and synchronizes the Siebel ID into Oracle Customer Hub.

Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle BRM

The synchronization of organizations (accounts) and persons (contacts) from OCH to Oracle BRM enforces that the customer data in Oracle BRM is consistent with the version of the data from the OCH. Billing systems contain considerable amount of customer data, which is used to bill customers and prepare billing invoices for customers. The inconsistency in data can translate to loss of revenues and result in higher maintenance costs and poor customer satisfaction.

Oracle Customer Hub publishes account updates to Billing and Revenue Management (Oracle BRM) under these circumstances:

- New customer data is created in Siebel CRM as part of order submission flow. OCH receives the data from Siebel CRM, cleanses it, and publishes it to the subscribing applications.

  In this case, Oracle BRM receives the data from Siebel CRM through order and an update from OCH.

- Customer data is updated in OCH because of an update in Siebel CRM. OCH receives the data from Siebel CRM, cleanses it, and publishes it to Oracle BRM.

- Customer data is updated in OCH because of an update in an application other than Siebel CRM.

  OCH receives the data, cleanses it, and publishes it to the subscribing applications including Oracle BRM.

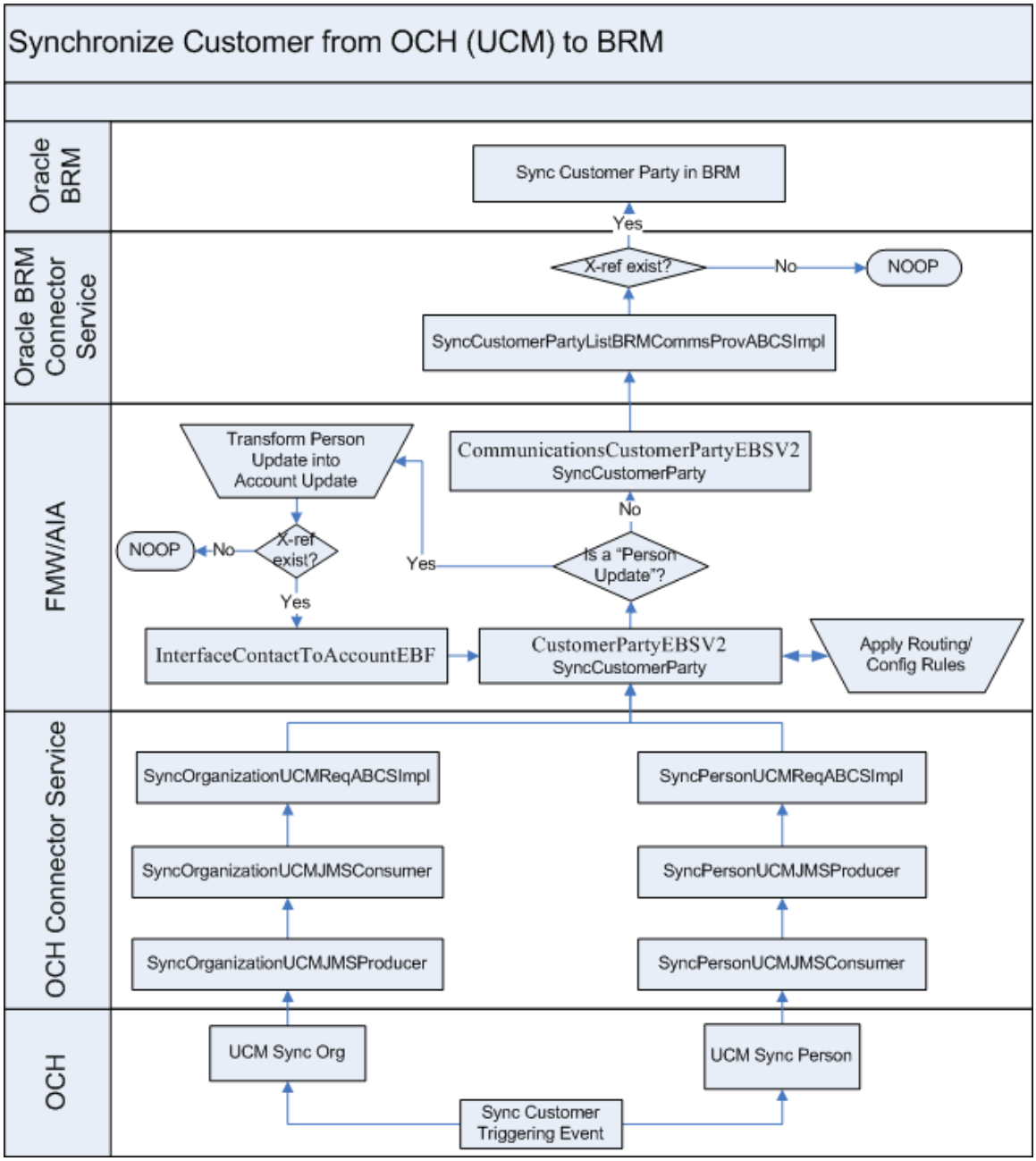
- Customer data is updated in OCH due to a merge either because of link and update or because of data steward's action.

  See: Merge flow from Oracle Customer Hub to Oracle BRM

The synchronization flow between Oracle Customer Hub and Oracle BRM is unidirectional where the customer data updates are published from Oracle Customer Hub to Oracle BRM.

Note: For OCH to Oracle BRM synchronization flow, publishing is always done in batch.

This graphic illustrates the process flow for customer data updates published from Oracle Customer Hub to Oracle BRM.



Synchronize customer from OCH to Oracle BRM

The flow from OCH to Oracle BRM enforces that the creation of new customer data can take place only as part of an order submission flow, because it is not recommended to have the prospects created in a billing system.

For more information about the Order Submission Flow, refer to the *Siebel CRM Integration Pack for Oracle Communications BRM: Order to Bill Implementation Guide*.

The synchronization process in OCH updates organization (party or customer) account information and their related entities, such as addresses and related contacts/accounts in Oracle BRM. It also updates person (party or customer) account information and their related entities, such as addresses, and related accounts in Oracle BRM. Oracle BRM can optionally subscribe to this synchronization operation depending on the configuration.

The synchronize customer request by OCH:

- Ensures that Oracle BRM has most current customer records consistent with the golden version of the customer data owned by Oracle Customer Hub (OCH)
- Enables invocation of single operation to update organization and person parties and customer accounts with the changes applied by Oracle Customer Hub (OCH) in one or more Oracle BRM applications.

The preset routing ensures that create messages are not propagated to Oracle BRM to avoid creating prospects in Oracle BRM as creation of customer data in Oracle BRM should take place only as part of the order submission flow.

OCH invokes a single synchronization service that publishes a message using AIA to the subscribing applications.

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. The dependency between the two updates (the update from OCH should be applied only after the create/update action from Siebel CRM has been performed) is not enforced.

However, if a race condition does take place, the cleansed data from OCH may be overwritten by the original data in Siebel CRM. Therefore, to fix this rare situation, the OCH administrator performs a periodic synchronization where the data that was not properly synchronized due to race condition is re-synchronized. See: Periodic Cron-Job for Re-Syncing Customer Data from OCH to Oracle BRM.

Oracle Customer Hub updates can be published to Oracle BRM during customer data creation or customer data updates.

OCH Publishes Updates to BRM for New Customer Data

The flow associated with the organization and person updates from OCH to Oracle BRM is composed of two parts for synchronization request:

Note: The synchronization message from OCH is always at the account level.

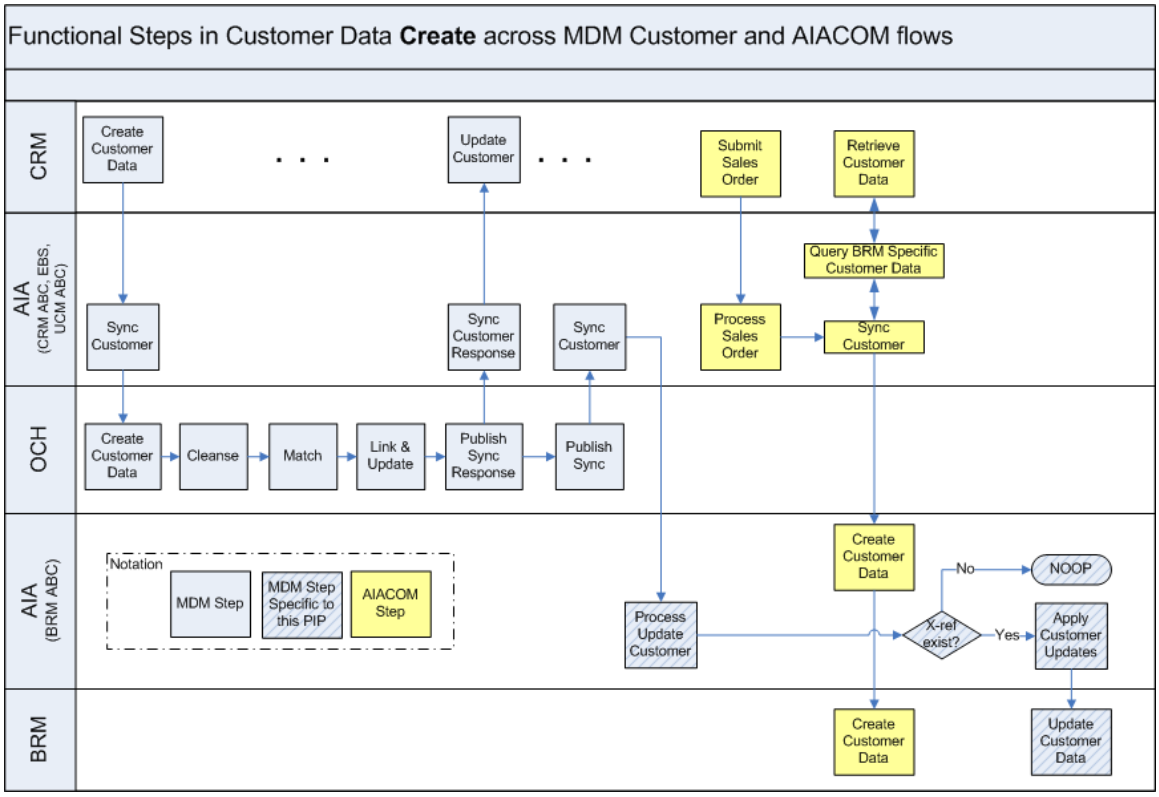
- OCH publishes customer data updates to AIA: This spoke is common to the standard customer data synchronization and reused from the horizontal integration.

The AIA CustomerPartyEBS routes the updates to the subscribing Oracle BRM instances. This AIA EBS needs to be configured to route a request for customer data update to Oracle BRM when customer-data synchronization message is received.

- AIA applies Customer Data updates to Oracle BRM: This spoke is a modification of the existing Oracle BRM customer data update.

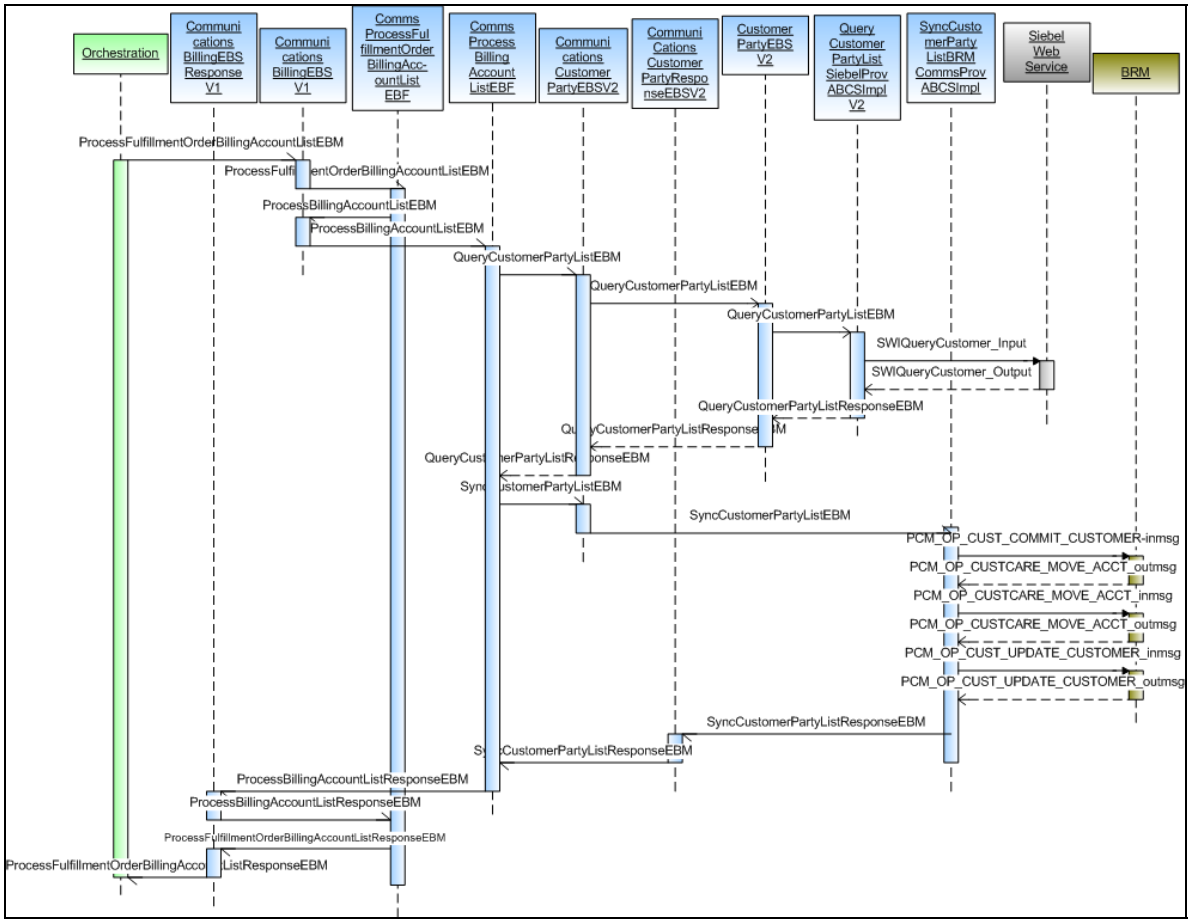
  In Oracle BRM SyncCustomerPartyListBRMCommsProvABCS, the AIA checks that a cross-reference for the entity exists. If the cross-reference exists, the changes are applied. Else, they are discarded as the customer data was not created in the Oracle BRM instance.

This graphic illustrates the functional steps when customer data is created in Siebel CRM (as part of agent-assisted billing care flow) and OCH:



Creating customer data (Oracle BRM and OCH)

This graphic illustrated the customer-account flow process from Siebel CRM to Oracle BRM.



Create, synchronize customer account

This is the process flow:

1. In Siebel CRM, when you create a sales order for an account and submit the order, this action triggers the order flow, which calls the orchestration engine.

   This orchestration engine initiates the order flow.

2. The orchestration engine calls the CommsProcessFulfillmentOrderBillingAccountListEBF, which takes the order EBM and extracts the relevant customer data to create a customer EBM (ProcessBillingAccountListEBM).

   The EBF then calls the CommsProcessBillingAccountListEBF for further customer processing.

3. The enterprise business function, CommsProcessBillingAccountListEBF, invokes enterprise business service, CommunicationsCustomerPartyEBSV2, with operation QueryCustomerPartyList and message QueryCustomerPartyListEBM.

   This is used to fetch the account data from the Siebel CRM before creating the account in Oracle BRM.

4. Invoking CommunicationsCustomerPartyEBSV2 with operation 'QueryCustomerPartyList' routes the QueryCustomerPartyListEBM to the core CustomerPartyEBSV2.QueryCustomerPartyList port.

   This calls the Siebel Provider ABC implementation service, QueryCustomerPartyListSiebelProvABCSImplV2, (a core service implemented by OCH PIP). This service transforms the QueryCustomerPartyListEBM to the Siebel specific ABM and invokes Siebel's SWI_Customer_Party_Service.

5. The Query Web Service queries the Siebel CRM database and fetches the account data, which is then sent back as a response using the same Siebel ABM.
6. The response message is then transformed to the response EBM, QueryCustomerPartyListResponseEBM, by the Query Provider ABC implementation service. This response is sent back to the CommsProcessBillingAccountListEBF service through the CustomerPartyEBSV2 and CommunicationsCustomerPartyEBSV2.
7. The CommsProcessBillingAccountListEBF instantiates the SyncCustomerPartyListBRMCommsProvABCSImpl.

   This service invokes the PCM_OP_CUST_COMMIT_CUSTOMER opcode, if it is a new account to be synced. For an already existing account, the PCM_OP_CUST_UPDATE_CUSTOMER or PCM_OP_CUSTCARE_MOVE_ACCT opcode is called, as required.

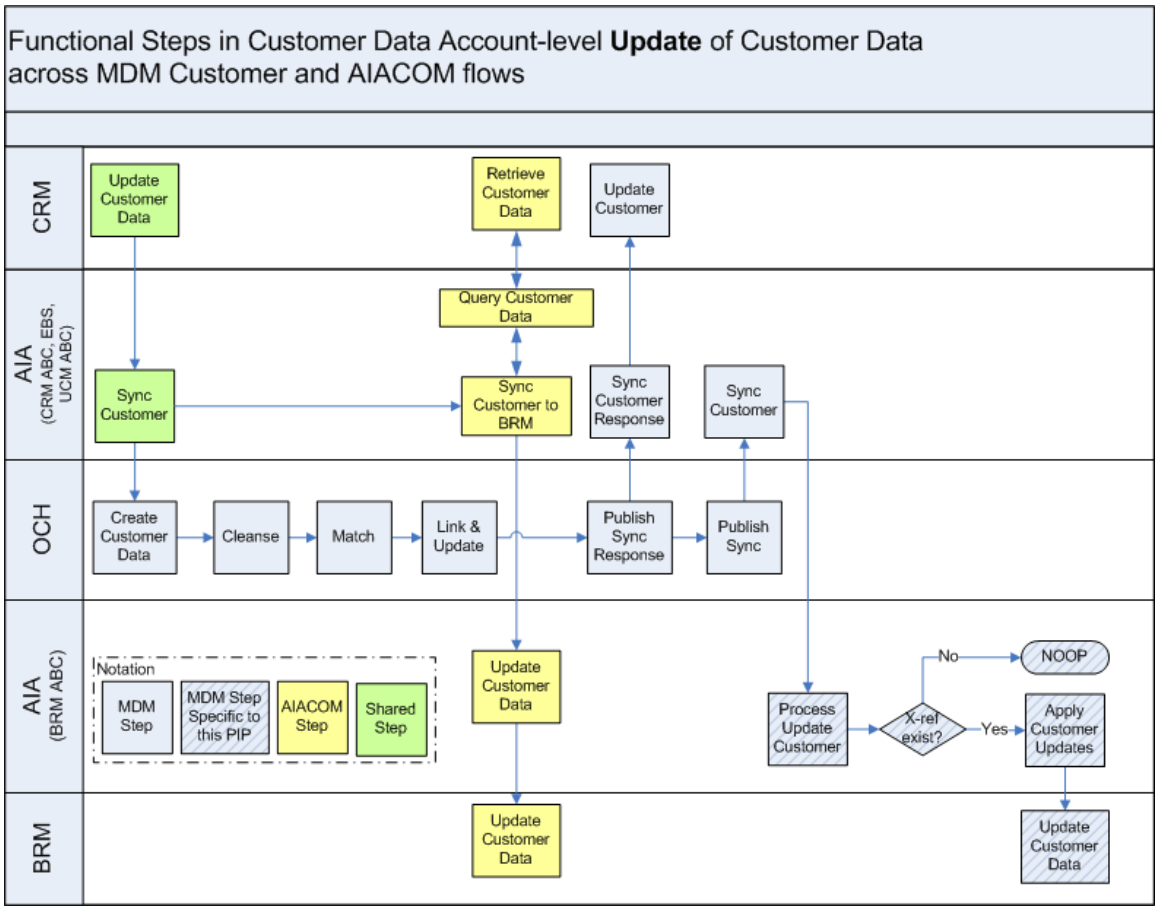
8. If the accounts are successfully synced or updates to an existing account are successfully done, the appropriate response is sent back to the CommsProcessFulfillmentOrderBillingAccountListEBF as an asynchronous delayed response.

OCH Publishes Updates to BRM for Updates to Existing Customer Data

OCH publishing updates to Oracle BRM for the changes made to existing data can be differentiated as updates made to an account (organization) and updates made to a contact (person).

For the updates made to an account, the process flow is the same as described in OCH Publishes Updates to BRM for New Customer Data, where OCH publishes an account synchronization message that is routed as an account update to Oracle BRM.

This graphic illustrates the interactions among the systems when customer data is updated in Siebel CRM (as part of agent-assisted billing care flow) and OCH at account level.



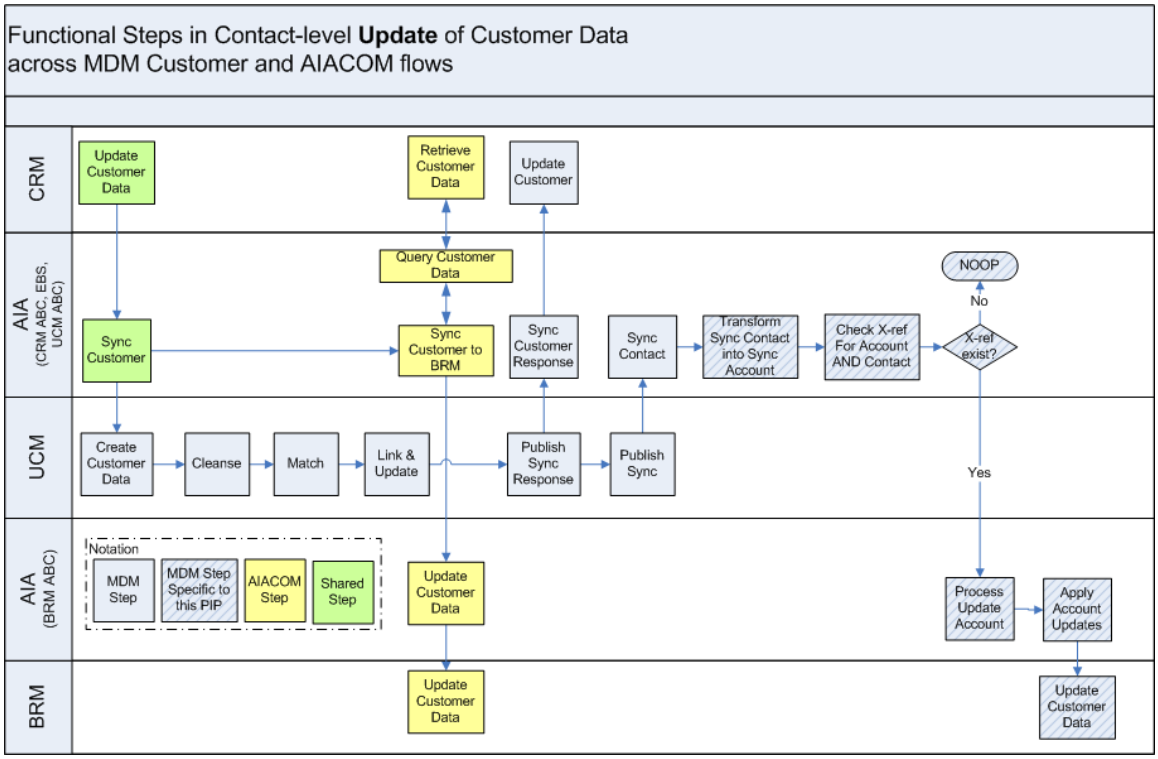
Updating customer data (Oracle BRM and OCH)

For the updates affecting only the person data, OCH publishes a contact update message. If the contact update is an update to the primary contact for an existing account in Oracle BRM, the Oracle BRM data is updated as well.

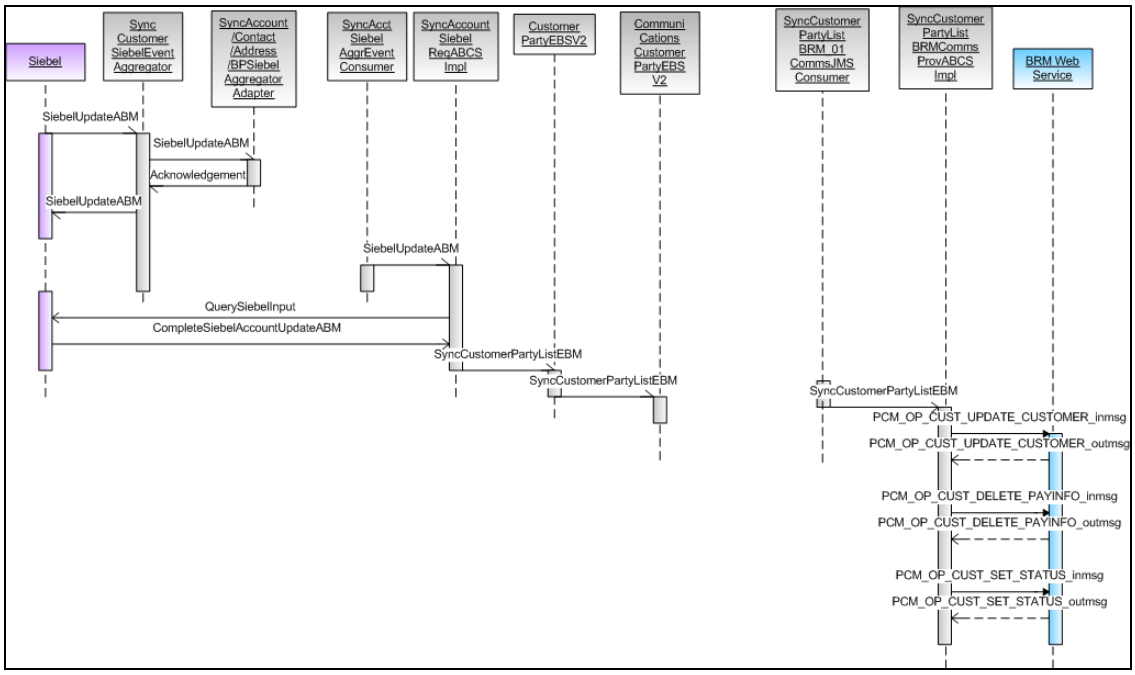
The synchronization message from OCH to BRM is always at the account level as the BRM provider expects an account message. Even when a new contact is created, it is done in the context of an account creation. If the contact is changed in Siebel, Siebel sends the contact message along with a list of impacted accounts, that is, even though a contact is changed it is really an account message that is passed to BRM provider.

In the event the contact data is changed in OCH, OCH publishes a person message (and not account message) which needs to be converted into an account message before sending to Oracle BRM. For this to happen, a transformation takes place in the routing rule that converts the person message into account message and routes it to InterfaceContactToAccountEBF, which checks whether both the account and the contact exist in the cross-reference. If both of them exist in the cross-reference, this EBF sends the account message for processing to Oracle BRM.

This graphic illustrates the interaction between the systems when customer data is updated at the contact level:



This graphic illustrates the customer update process initiated in Siebel CRM when an account is updated, and the change propagated to the Oracle BRM.



Update customer account

This is the process flow:

1. When an account attribute is updated in Siebel CRM such as address, contact or billing profile, it causes the Siebel CRM workflow to invoke the SyncCustomerSiebelEventAggregator with the SiebelUpdateABM message containing details of the account that has been updated.

   Corresponding to the kind of update, there can be four kinds of Siebel messages: ListOfSWICustomerIO, ListOfSWIBillingProfileIO, ListOfSWIContactIO, and ListOfSWIAddressIO. The SyncCustomerSiebelEventAggregator then calls a database adapter that executes a PL/SQL script which extracts and stores the relevant IDs (like account, contact, and billprofile) in a database table AIA_AGGREGATED_ENTITIES.

2. The IDs in the database table are stored in such a way that the hierarchy of IDs is maintained.

   For example, BillingProfileID remains a child of some account ID. The account ID along with its entire child IDs is picked up from the database table by the SyncAcctSiebelEventAggrConsumer process.

3. The consumer process then calls the SyncAccountSiebelReqABCSImpl process. This process takes all the IDs, constructs a Siebel Query Input ABM, and calls the Siebel Query Web service to get the entire account data from Siebel.

   After getting the data, it is transformed into the SyncCustomerPartyListEBM, and CustomerPartyEBSV2.SyncCustomerPartyList operation is called.

4. The CustomerPartyEBSV2.SyncCustomerPartyList operation calls the CommunicationsCustomerPartyEBSV2.SyncCustomerPartyList operation.
5. The operation CommunicationsCustomerPartyEBSV2.SyncCustomerPartyList puts the message in the topic named CPARTY_SYNC_TOPIC
6. Depending on the instances of Oracle BRM or any other billing system, consumers can be defined that have subscribed to CPARTY_SYNC_TOPIC.

   One such consumer for the default implementation is available by the name SyncCustomerPartyListBRM_01CommsJMSConsumer. The SyncCustomerPartyListBRM_01CommsJMSConsumer, which listens to CPARTY_SYNC_TOPIC for messages picks up the arriving message and passes it on to the process SyncCustomerPartyListBRMCommsProvABCSImpl after duly checking if the message should go to the ensuing provider ABCS or not, and correspondingly stamping the target ID.

7. The SyncCustomerPartyListBRMCommsProvABCSImpl process then calls the PCM_OP_CUST_UPDATE_CUSTOMER opcode or PCM_OP_CUST_DELETE_PAYINFO, as required, to synchronize the updated data to Oracle BRM.

Periodic Cron-Job for Re-Syncing Customer Data from OCH to Oracle BRM

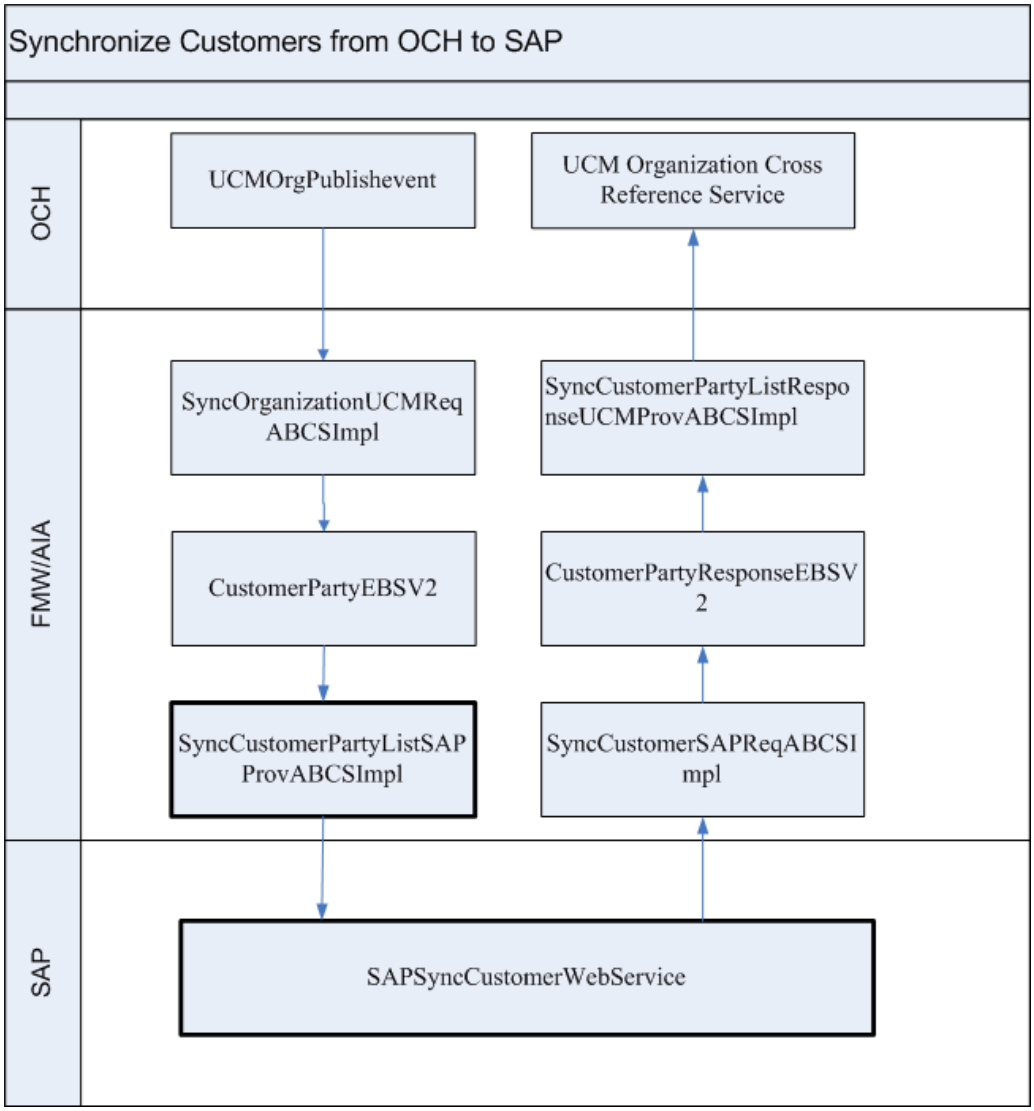
Periodic cron-job ensures that all cleansed data that may not have been synced from OCH to Oracle BRM because of race conditions is periodically synced. It also ensures that Oracle BRM customer data is updated with the cleansed and trusted version from the OCH even in the rare event that the batch update could not be applied.

The data steward configures OCH to publish periodic synchronization to target applications. During this periodic synchronization, all customer data (account and contact) that are created and updated during the last (periodic) synchronization are published to the subscribing applications.

Synchronization of Organizations and Persons from Oracle Customer Hub to SAP

This synchronization ensures that SAP has the most current customer record to execute application specific processes and applications. This synchronization invokes a single operation to create or update organization and person parties and customer accounts in SAP applications. The synchronization service that publishes (using AIA) message to the subscribing SAP application may be triggered manually or as part of an automated process.

This graphic illustrates the synchronization process from Oracle Customer Hub to SAP.



Synchronize customers from OCH to SAP

When a customer is created or updated in OCH, the customer data is sent to SAP through the AIA layer and that customer is created in SAP. SAP in turn sends back the SAP ID to OCH, which is stored in OCH.

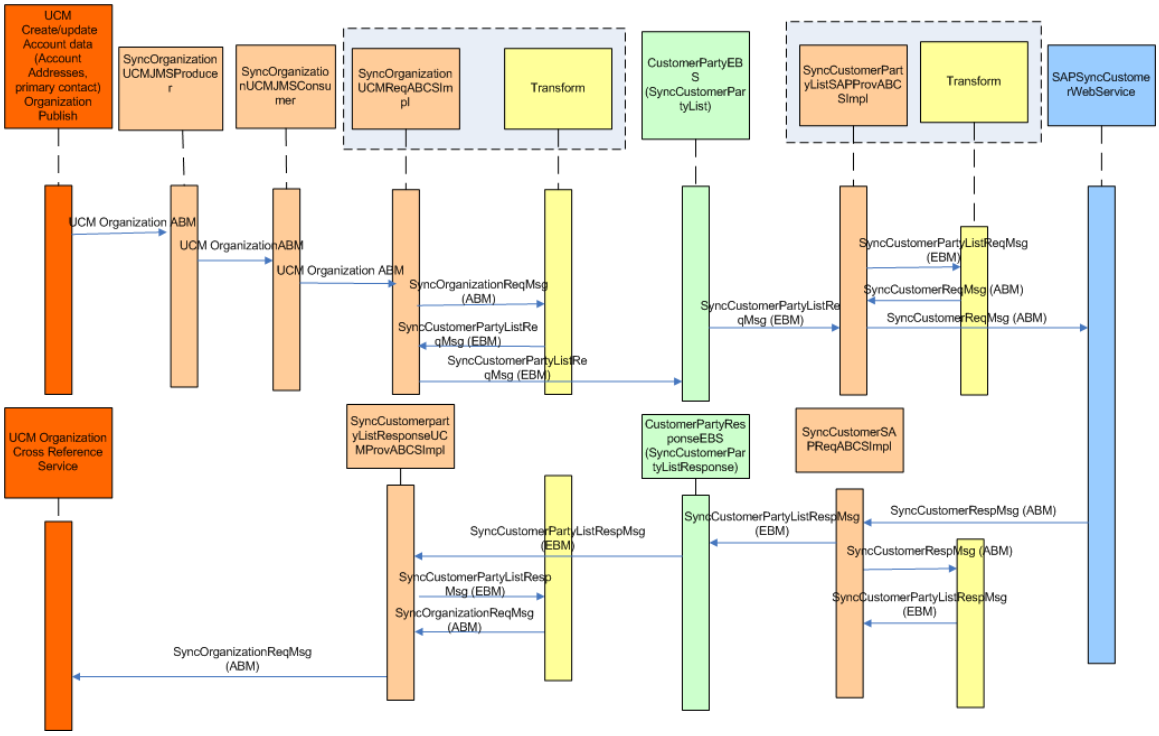
Note: SAP does not differentiate between person and organization type of customers; therefore, it is recommended that new (organization or person) customers be created as accounts in OCH.

For OCH to SAP person synchronization, person-type customers are created as account/contact, which are similar to organization-type customers. In OCH, creating individual customers as accounts is recommended.

OCH	SAP
Organization create/update (Mapped to 'Accounts' entity in OCH)	Mapped to Customer in SAP
Contact details for the Organization (Mapped to Account -> Contact in OCH)	Mapped to Customer -> Contact details in SAP
Person create/update (Mapped to 'Accounts' entity in OCH)	Mapped to Customer in SAP
Contact detail for the Person (Mapped to Account -> Contact in OCH)	Mapped to Customer -> Contact details in SAP

As SAP maintains only one address per customer, SAP uses only the primary address sent from OCH during the organization and person synchronization.

This graphic illustrates the synchronization process flows from Siebel UCM (OCH) to SAP.



OCH to SAP process integration flow

This integration flow synchronizes new accounts or updates and uses these interfaces:

- SyncOrganizationUCMJMSProducer
- SyncOrganizationUCMJMSConsumer
- SyncOrganizationUCMReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListSAPProvABCSImpl
- SyncCustomerSAPReqABCSImpl
- CustomerPartyEBSResponseV2
- SyncCustomerPartyListResponseUCMProvABCSImpl

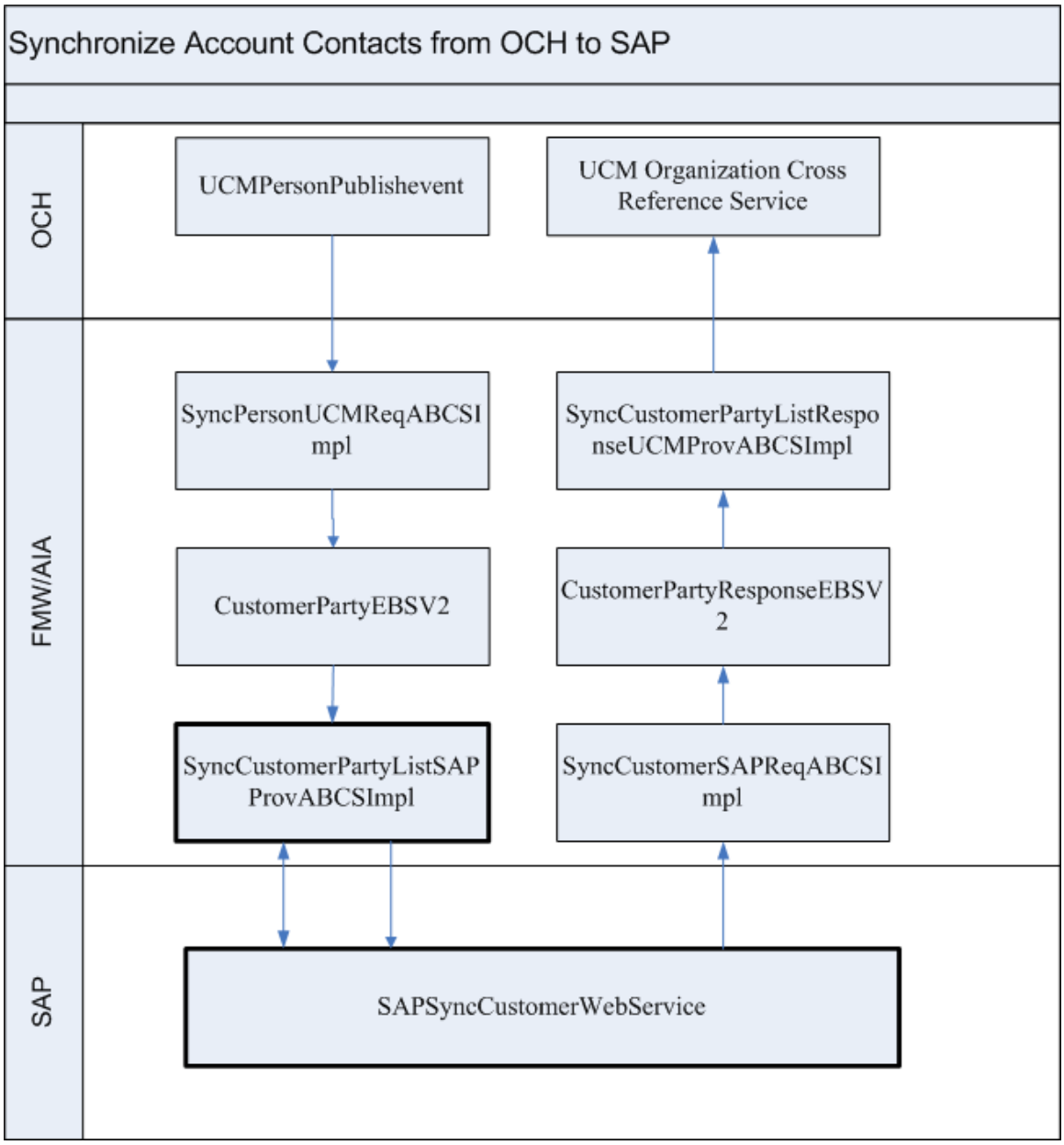
The steps in the sequence diagram are described here:

1. OCH publishes account messages either real time or by bulk mode. This initiates the SyncOrganizationUCMJMSProducer with the UCMOrganizationABM message.
2. SyncOrganizationUCMJMSProducer queues the UCMOrganizationABM message onto a JMS queue.
3. The message in the queue is picked up by the SyncOrganizationUCMJMSConsumer service which then routes the message to SyncOrganizationUCMReqABCSImpl
4. The OCH Requester ABCS implementation, SyncOrganizationUCMReqABCSImpl, transforms the OCH ABM to the SyncCustomerPartyListEBM and invokes the 'SyncCustomerPartyList' operation of the CustomerPartyEBSV2.
5. Invoking CustomerPartyEBSV2 with the 'SyncCustomerPartyList' operation routes the SyncCustomerPartyListEBM to the SAP provider ABCS Implementation service, SyncCustomerPartyListSAPProvABCSImpl.
6. SyncCustomerPartyListSAPProvABCSImpl transforms the SyncCustomerPartyListEBM to the appropriate SAP Customer ABM and invokes the SAP IDOC.
7. The SyncCustomerSAPReqABCSImpl transforms the response SAP IDOC to the SyncCustomerPartyListResponseEBM, in the process of which the cross references tables are populated with the SAP identifier values if not already present. And the response is also sent to CustomerPartyResponseEBSV2 which routes the message to SyncCustomerPartyListResponseUCMProvABCSImpl to set the SAP id within OCH.
8. SyncCustomerPartyListResponseUCMProvABCSImpl transforms SyncCustomerPartyListResponseEBM to UCM Organization ABM and synchronizes the external ID into OCH.

Synchronization of Account Contacts from Oracle Customer Hub to SAP

This integration flow synchronizes account contacts from OCH to SAP. The single synchronize service that is invoked publishes a message using AIA to the subscribing SAP application(s). This integration flow takes places when a non- primary contact is added/updated to an existing account or a primary contact of an account is updated.

This activity diagram illustrates the synchronization of account contacts from OCH to SAP:



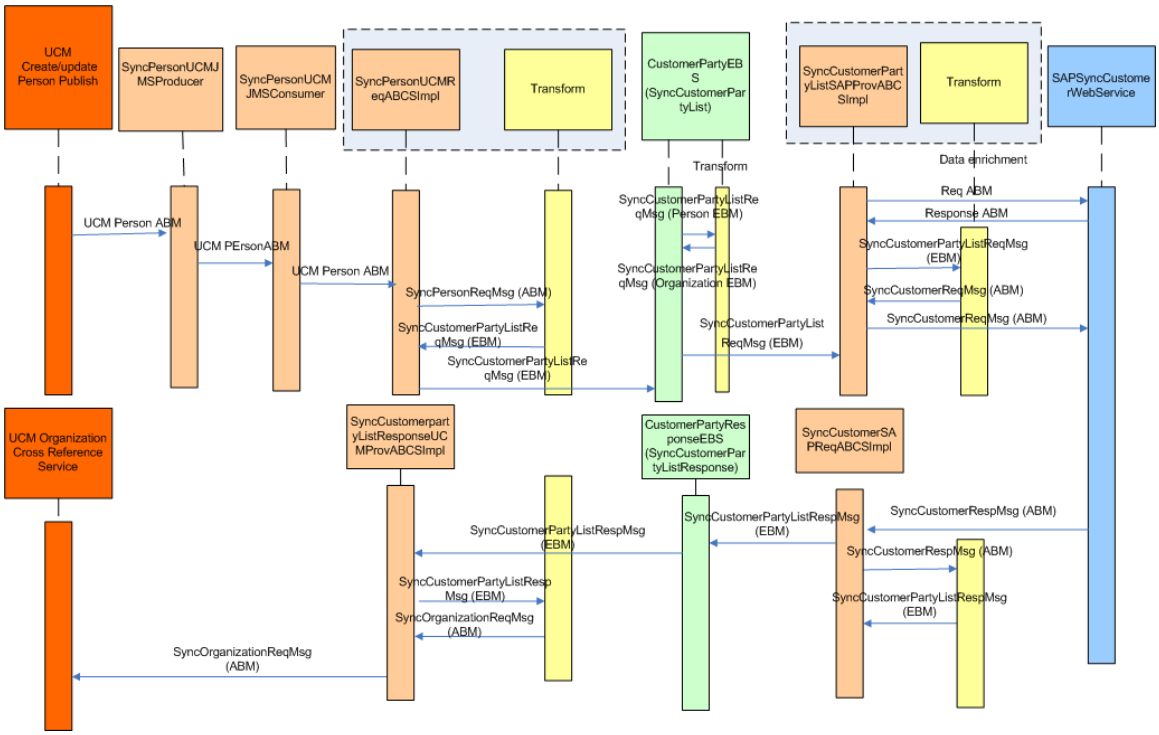
Synchronizing Account contacts from OCH to SAP

This integration flow uses these interfaces:

- SyncPersonUCMJMSProducer
- SyncPersonUCMJMSConsumer
- SyncPersonUCMReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListSAPProvABCSImpl
- SyncCustomerSAPReqABCSImpl
- CustomerPartyEBSResponseV2

- SynchCustomerPartyListResponseUCMProvABCSImpl

This sequence diagram illustrates the integration flow:



Synchronizing Account contacts from OCH to SAP

The following events occur when the process is initiated:

1. OCH publishes person messages either real time or by bulk mode. This initiates the SyncPersonUCMJMSProducer with the UCMPersonABM message.
2. SyncPersonUCMJMSProducer queues the UCMPersonABM message onto a JMS queue.
3. The message in the queue is picked up by the SyncPersonUCMJMSConsumer service which then routes the message to SyncPersonUCMReqABCSImpl
4. The OCH Requestor ABCS implementation, SyncPersonUCMReqABCSImpl, transforms the UCMPersonABM to the SyncCustomerPartyListEBM and invokes the ‘SyncCustomerPartyList’ operation of the CustomerPartyEBSV2.
5. Invoking CustomerPartyEBSV2 with the ‘SyncCustomerPartyList’ operation transforms and routes the SyncCustomerPartyListEBM to the SAP provider ABCS Implementation service, SyncCustomerPartyListSAPProvABCSImpl.
6. SyncCustomerPartyListSAPProvABCSImpl enriches and transforms the SyncCustomerPartyListEBM to the appropriate SAP Customer ABM and invokes the SAP IDOC.
7. The SyncCustomerSAPReqABCSImpl transforms the response SAP IDOC to the SyncCustomerPartyListResponseEBM, in the process of which the cross references tables are populated with the SAP identifier values if not already present. And the response is also sent to CustomerPartyResponseEBSV2 which routes the message to SyncCustomerPartyListResponseUCMProvABCSImpl to set the SAP ID within OCH.

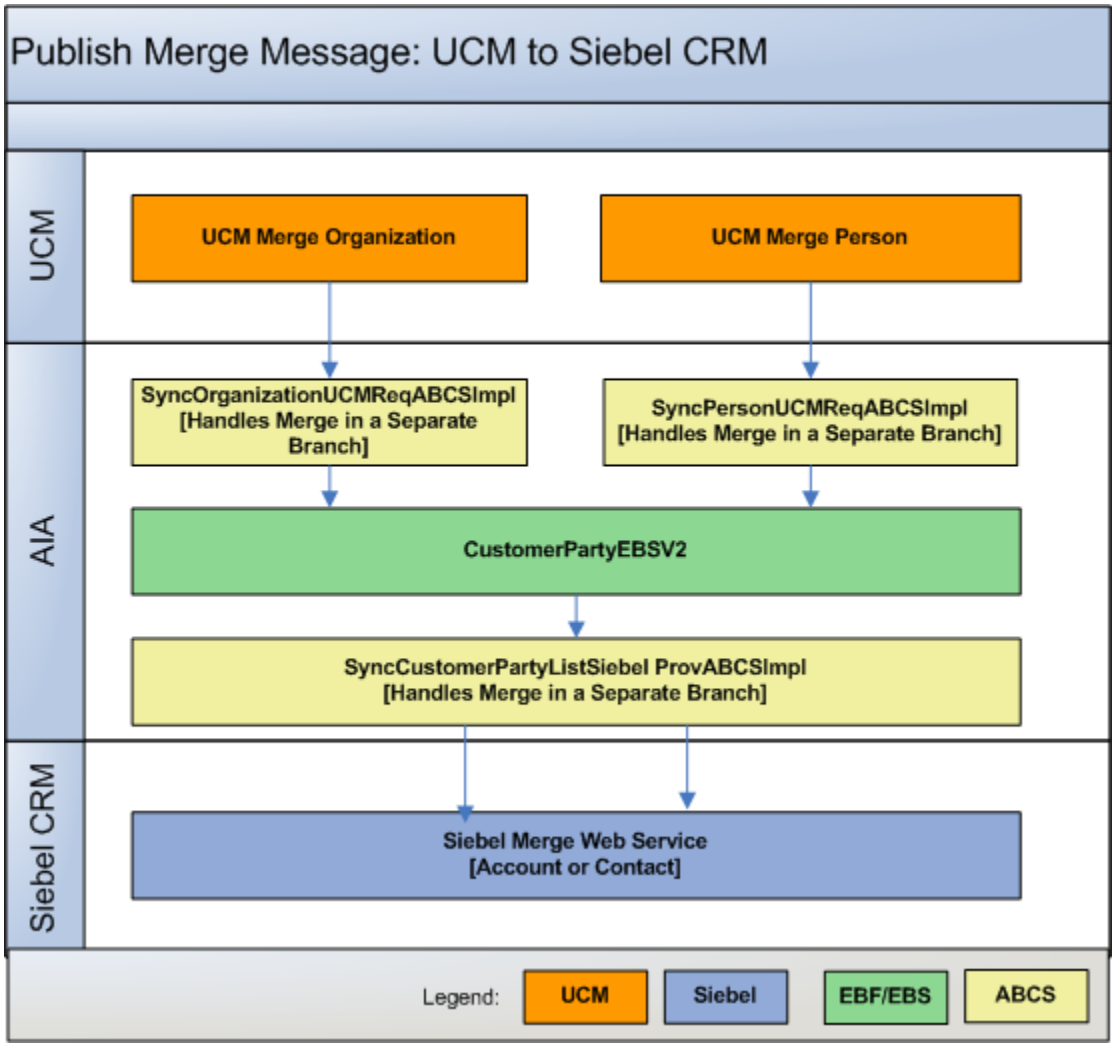
8. SyncCustomerPartyListResponseUCMProvABCSImpl transforms SyncCustomerPartyListResponseEBM to OCH Organization ABM and synchronizes the external ID into OCH.

Data enrichment flow

See: Chapter 8: Customer Data Enrichment.

Merge Flow from Oracle Customer Hub to Siebel CRM

This activity diagram illustrates the merge from Oracle Customer Hub to Siebel CRM integration flow:



Merging organizations and persons from Oracle Customer Hub to Siebel CRM

By default, the Customer MDM PIP merges the customer data in Siebel when a merge is published by the Oracle Customer Hub; however, you can turn off this default behavior should you have a business need when interoperating with applications that do not support a merge process.

To turn off merge publishing by Oracle Customer Hub, set the process property EnablePubSub to *False* for the UCM Process Merge Request workflow.

For more information about the workflow, refer to *Siebel Book Shelf: Siebel Business Process Framework: Workflow Guide*.

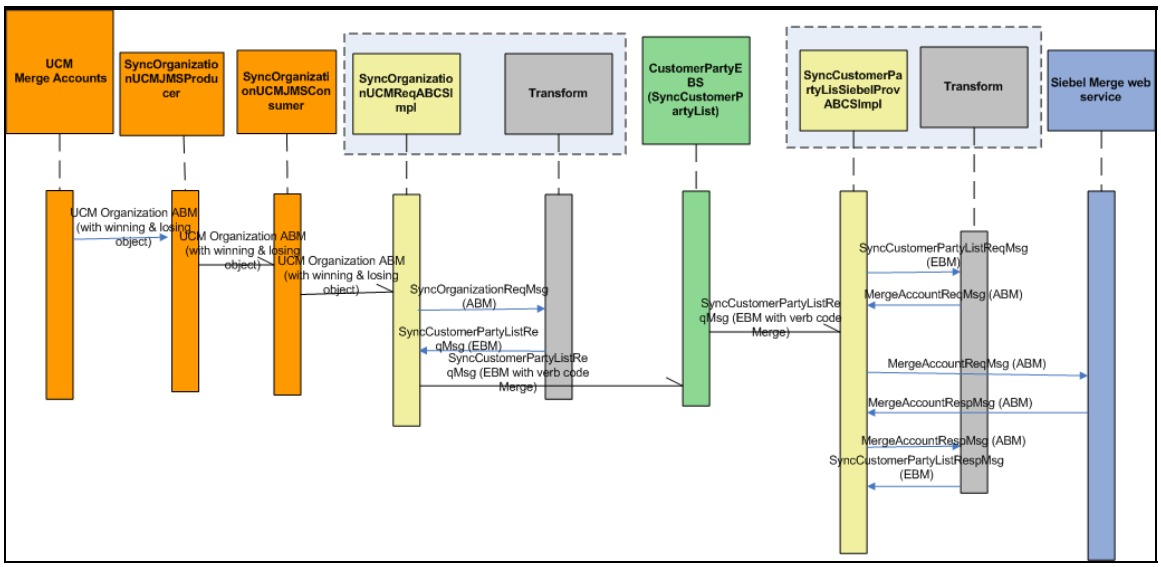
Merging Organizations from Oracle Customer Hub to Siebel

This integration flow uses these interfaces:

- SyncOrganizationUCMJMSProducer
- SyncOrganizationUCMJMSConsumer
- SyncOrganizationUCMReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListSiebelProvABCSImpl

Note that these services are the same services used by the synchronization flows. These same Siebel ABC services deal with different transformations and ABMs to correspond to two different web services in Siebel.

This sequence diagram illustrates the integration flow:



Merging organizations from Oracle Customer Hub to Siebel CRM

When you initiate this process, these events occur:

1. Oracle Customer Hub publishes merge account messages in real time when two accounts are merged in the Existing Duplicates screen in Oracle Customer Hub.

   This initiates the SyncOrganizationUCMJMSProducer, with the UCMAccountABM message that contains both the survivor and the victim object.

2. The SyncOrganizationUCMJMSProducer queues the Oracle Customer Hub Account ABM message onto a JMS queue.
3. The message in the queue is picked up by the SyncOrganizationUCMJMSConsumer service, which then routes the message to the SyncOrganizationUCMReqABCSImpl.
4. The Oracle Customer Hub requester ABC implementation, SyncOrganizationUCMReqABCSImpl, transforms the Oracle Customer Hub ABM with both the survivor and the victim object to the SyncCustomerPartyListEBM with the verb code 'Merge' and invokes the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

   When there are multiple target systems in the message, the message has to be transformed into several EBM messages, one for each target, and there are multiple calls to the EBS from the SyncOrganizationUCMReqABCSImpl.
5. Invoking the CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Siebel provider ABC implementation service, SyncCustomerPartyListSiebelProvABCSImpl.
6. The Siebel provider ABC implementation service, SyncCustomerPartyListSiebelProvABCSImpl identifies the verb code 'Merge' and transforms the SyncCustomerPartyListEBM to the appropriate Siebel merge ABM and invokes the Siebel web service for merging accounts.

   The Siebel Merge web service performs a merge of the victim account to the survivor account and re-parents all the transactions to the survivor record within Siebel. The response from Siebel is then transformed to mark the corresponding cross-reference entries for deletion.

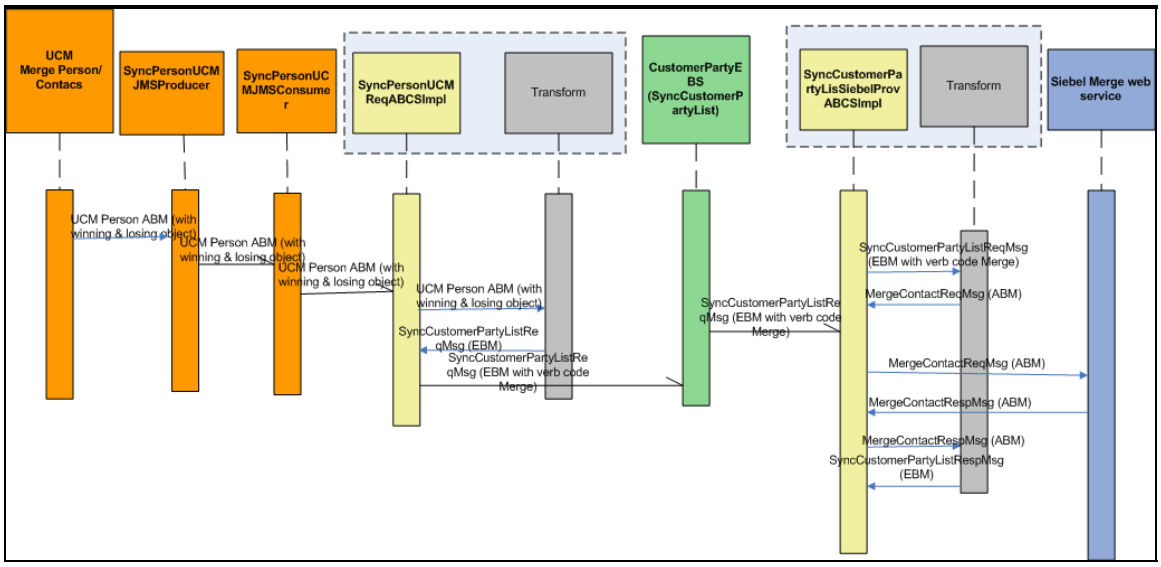
Merging Persons from Oracle Customer Hub to Siebel CRM

This integration flow uses these interfaces:

- SyncPersonUCMJMSProducer
- SyncPersonUCMJMSConsumer
- SyncPersonUCMReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListSiebelProvABCSImpl

Note that these services are the same services used by the synchronization flows. These same Siebel ABC services deal with different transformations and ABMs to correspond to two different web services in Siebel.

This sequence diagram illustrates the integration flow:



Merging persons from Oracle Customer Hub to Siebel CRM

When you initiate this process, these events occur:

1. Oracle Customer Hub publishes a person message with the survivor and the victim object in real time when two contacts are merged in the Existing Duplicates screen in Oracle Customer Hub.

   This initiates the SyncPersonUCMJMSProducer, with the UCMPersonABM message.

2. The SyncPersonUCMJMSProducer queues the Oracle Customer Hub Person ABM message onto a JMS queue.

3. The message in the queue is picked up by the SyncPersonUCMJMSConsumer service, which then routes the message to the SyncPersonUCMReqABCSImpl.

4. The Oracle Customer Hub requester ABC implementation, SyncPersonUCMReqABCSImpl, transforms the Oracle Customer Hub ABM with the survivor and victim object into the SyncCustomerPartyListEBM with the verb code Merge and invokes the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

5. When there are multiple targets in the message, the message has to be transformed into several EBM messages, one for each target and there are multiple calls to the EBS from the SyncPersonUCMReqABCSImpl.

6. Invoking the CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Siebel provider ABC implementation service, SyncCustomerPartyListSiebelProvABCSImpl.

7. The Siebel provider ABC implementation service, SyncCustomerPartyListSiebelProvABCSImpl, identifies the verb code Merge, and transforms the SyncCustomerPartyListEBM to the appropriate Siebel merge ABM. It then invokes the Siebel merge web service.

   The Siebel Merge web service performs a merge of the victim contact to the survivor contact and re-parents all the transactions to the survivor record within Siebel. The response is then transformed to mark the corresponding entries in the cross reference for deletion.

Merge Flow from Oracle Customer Hub to Oracle E-Business Suite

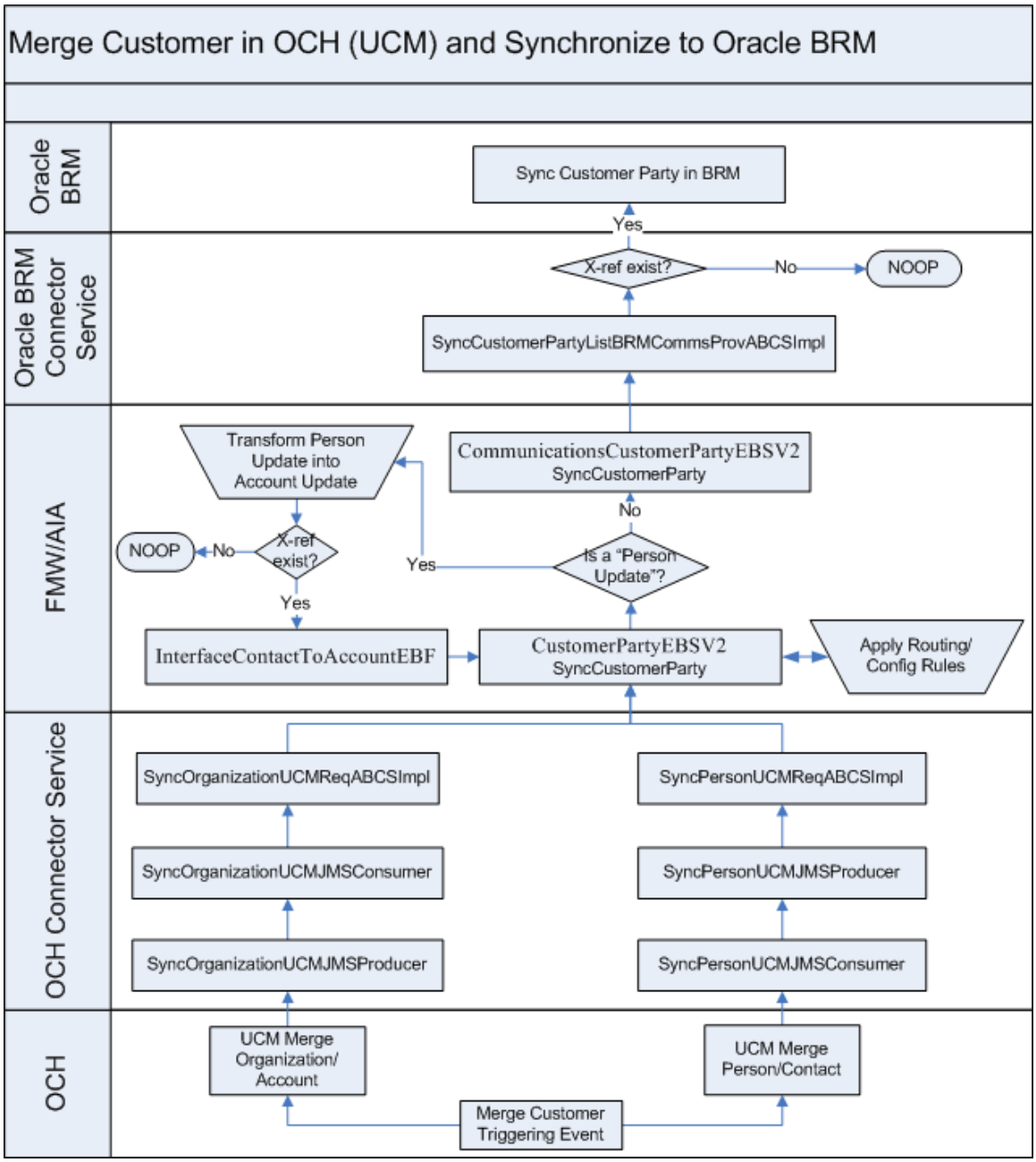
The Oracle E-Business Suite ABCS consumes the merge message, but updates only the survivor record with the latest details while ignoring the request to delete the victim records. This includes the creation of new child entries like addresses and contacts that is similar to the synchronization flow from Oracle Customer Hub to Oracle E-Business Suite. Out-of-the-box invocation of merge within Oracle E-Business Suite is not supported as part of this flow.

A merge in Oracle E-Business Suite involves running concurrent jobs, which is not supported.

Merge Flow from Oracle Customer Hub to Oracle BRM

The merge customer request by OCH enables Oracle BRM to update customer data with the changes published by OCH. Oracle BRM ABCS updates the survivor record with the latest details while ignoring the request to delete the victim records.

This diagram illustrates the merge flow from OCH to Oracle BRM.



Merge flow from OCH to Oracle BRM

The flow associated with the organization and person updates from OCH to Oracle BRM is composed of two spokes for merge request:

- OCH publishes customer data updates to AIA: This spoke is common to the standard customer data merge and reused from the horizontal integration.

  The AIA SyncCustomerPartyEBS routes the updates to the subscribing Oracle BRM instances. This AIA EBS needs to be configured to route a request for customer data update to Oracle BRM when a customer data merge message is received.

- AIA applies customer data updates to Oracle BRM: This spoke is a modification of the existing Oracle BRM customer data update.

  In Oracle BRM SyncCustomerPartyListBRMCommsProvABCS, the AIA checks that a cross-reference for the entity exists. If the cross-reference exists, the changes are applied. Else, they are discarded as the customer data was not created in the Oracle BRM instance.

Some of the subscribing applications such as Siebel CRM may be able to merge records and re-parent the associated entities to reflect the merge performed by OCH, but Oracle BRM does not offer a service to support this feature. Therefore, merging account or contacts in Siebel CRM without merging the corresponding accounts or contacts in Oracle BRM is not a feasible option. The reason being, it would not be possible to retrieve invoice details for victim accounts or submit change orders related to assets parented to victim accounts.

A merge operation in OCH only triggers an update of the survivor account in both Siebel CRM and Oracle BRM. However, customers may customize their procedures and tools so that an administrator can merge customer data in Siebel CRM and in Oracle BRM to be consistent with the merge data in OCH. This is a suggested custom solution and is not part of the flows covered in this process integration pack for OCH.

OCH Integration Services

These are the integration services:

- SyncOrganizationUCMJMSProducer
- SyncOrganizationUCMJMSConsumer
- SyncPersonUCMJMSProducer
- SyncPersonUCMJMSConsumer
- QueryCustomerPartyListUCMProvABCSImpl
- QueryCustomerPartyUCMProvABCSImpl
- SyncOrganizationUCMReqABCSImpl
- SyncCustomerPartyListResponseUCMProvABCSImpl
- SyncPersonUCMReqABCSImpl
- ProcessPersonUCMReqABCSImpl
- SyncCustomerPartyListUCMProvABCSImpl

SyncOrganizationUCMJMSProducer

The SyncOrganizationUCMJMSProducer is implemented as a BPEL process since it involves JMS Header manipulations that cannot be done in ESB. This service is responsible for queueing the Oracle Customer Hub Organization payload into the JMS queue. When queueing, a unique JMSCorrelationID is populated in the JMS Header of each message. This ID is used when dequeueing the message in the SyncOrganizationUCMJMSConsumer. This operation is defined in this service: SyncOrganization.

SyncOrganizationUCMJMSConsumer

The SyncOrganizationUCMJMSConsumer service is implemented as an ESB process with a JMS adapter and routing services. This service is responsible for dequeueing the Oracle Customer Hub Organization message from the JMS queue and invoking the requestor ABCS, SyncOrganizationUCMReqABCSImpl. The dequeue operation is done depending on the JMSCorrelationID that is populated when the message is enqueued.

SyncPersonUCMJMSProducer

The SyncPersonUCMJMSProducer service is implemented as a BPEL process since it involves JMS Header manipulations that cannot be done in ESB. This service is responsible for enqueueing the Oracle Customer Hub Person payload into the JMS queue. When queueing, a unique JMSCorrelationID is populated in the JMS Header of each message. This ID is used when dequeueing the message in the SyncPersonUCMJMSConsumer. This operation is defined in SyncPerson service.

SyncPersonUCMJMSConsumer

The SyncPersonUCMJMSConsumer is implemented as an ESB process with a JMS adapter and routing services. This service is responsible for dequeueing the Oracle Customer Hub Person message from the JMS queue and invoking the requestor ABCS, SyncPersonUCMReqABCSImpl. The dequeue operation is done depending on the JMSCorrelationID that is populated when the message is enqueued.

QueryCustomerPartyListUCMProvABCSImpl

The QueryCustomerPartyListUCMProvABCSImpl provider ABC service is used by the match flow to query both organization and person information from Oracle Customer Hub. It returns a list of organization or person candidates that matches the sent criteria. This service has one synchronous request/reply operation: QueryCustomerPartyList.

QueryCustomerPartyUCMProvABCSImpl

The QueryCustomerPartyUCMProvABCSImpl provider service is used as part of the fetch flow for both organization and person requests. It accepts an organization or person common ID as the request criteria and returns the full organization or person data record from Oracle Customer Hub. This service has one synchronous request/reply operation: QueryCustomerParty.

SyncOrganizationUCMReqABCSImpl

The SyncOrganizationUCMReqABCSImpl service is responsible for transforming the Oracle Customer Hub message into the SyncCustomerPartyList EBM format and invoking the SyncCustomerPartyList operation of the CustomerPartyEBSV2. This operation is defined in the service: SyncOrganization.

SyncCustomerPartyListResponseUCMProvABCSImpl

The SyncCustomerPartyListResponseUCMProvABCSImpl is implemented in BPEL and is invoked by the SyncCustomerPartyListResponse operation of the CustomerPartyResponseEBSV2, passing in the SyncCustomerPartyListResponseEBM payload. This service transforms the EBM into an Oracle Customer Hub ABM and invokes the Oracle Customer Hub web service interface to synchronize the external system IDs.

SyncPersonUCMReqABCSImpl

The SyncPersonUCMReqABCSImpl service is responsible for transforming the Oracle Customer Hub message into the SyncCustomerPartyList EBM format and invoking the SyncCustomerPartyList operation of the CustomerPartyEBSV2. This operation is defined in the service: SyncPerson.

ProcessPersonUCMReqABCSImpl

Oracle Customer Hub ABCS Connector service initiated from Oracle Customer Hub for data enrichment. This synchronous service maps the Oracle Customer Hub ABM to ProcessCustomerPartyListEBM and invokes the ProcessCustomerPartyEBS. After getting a response, maps the enriched response EBM back to Oracle Customer Hub ABM.

SyncCustomerPartyListUCMProvABCSImpl

The SyncCustomerPartyListUCMProvABCSImpl service is implemented in BPEL and is invoked by the SyncCustomerPartyList operation of the CustomerPartyEBSV2, passing in the synchronize CustomerPartyListEBM payload. This service transforms the EBM into a OCH ABM and invokes the OCH web service interface. It also captures the response from the OCH Application, transforms the OCH Response ABM to the SyncCustomerPArtyListResponseEBM payload, and invokes the SyncCustomerPartyListResponse operation of the CustomerPartyResponseEBSV2.

Oracle Customer Hub Interfaces

The Inbound/Outbound services, as applicable, are listed below:

- Inbound Web Services: Match and Fetch
- Inbound Web Services: Organization and Person Synchronization
- Outbound Web Services: Organization and Person Synchronization
- Inbound Web Services: Account Contact Synchronization
- Outbound Web Services: Account Contact Synchronization

Inbound Web Services: Match and Fetch

- Service: OrganizationMatchService

  Operation: OrganizationMatch

- Service: PersonMatchWebService

  Operation: PersonMatch

- OrganizationService
- PersonService

For more information about Oracle Customer Hub (Siebel UCM) web services, see *Siebel CRM Web Services Reference Addendum for Siebel UCM* on My Oracle Support 3.

Inbound Web Services: Organization and Person Synchronization

Name	Schema
Service:OrganizationService Operation: createOrganization	ListOfSwiOrganizationIO.xsd
Service: PersonService Operation: createPerson	ListOfSwiPersonIO.xsd
OrganizationCrossReferenceService	ListOfSwiOrganizationPublishIO.xsd

Outbound Web Services: Organization and Person Synchronization

Name	Schema
SyncOrganizationUCMJMSProducer	ListOfSwiOrganizationPublishIO.xsd
SyncPersonUCMJMSProducer	ListOfSwiPersonPublishIO.xsd

Inbound Web Services: Account Contact Synchronization

Name	Schema
OrganizationCrossReferenceService	ListOfSwiOrganizationPublishIO.xsd

Outbound Web Services: Account Contact Synchronization

Name	Schema
SyncPersonUCMJMSProducer	ListOfSwiPersonPublishIO.xsd

Assumptions and Constraints

These are the assumptions and constraints:

1. Oracle Customer Hub PIP supports these application and middleware components:
   - Fusion Middleware
   - Foundation Pack
   - Oracle Customer Hub (Siebel UCM)
   - Siebel CRM
   - Oracle E-Business Suite
   - Oracle BRM
   - SAP ECC

For more information about installing and selecting the participating applications (integration options) based on your business need, refer *AIA Oracle Application Integration Architecture 2.5: Installation and Upgrade Guide*.

Note: Any use of OCH PIP as part of a vertical/industry-specific initiative may require the OCH PIP to be certified on different combinations of applications, middleware, and AIA foundation pack objects.

2. The synchronization of delete actions on accounts, addresses, or contacts is not supported in this PIP.
3. Synchronization of assets to OCH is not in scope for this PIP
4. Synchronization of account Hierarchies to OCH is not in scope for this PIP
5. Synchronization of bill Info data to OCH is not in scope for this PIP
6. Synchronization of OCH Households is not in scope for this PIP
7. Synchronization of Financial Accounts to OCH is not in scope for this PIP.
8. Synchronization of Households to OCH is not in scope for this PIP.
9. Synchronization of Billing Profiles to OCH is not in scope for this PIP.
10. AIA cross-references are created for any entity that is created across applications using AIA. These cross-references are used for different type of entities including accounts, contacts, addresses, orders, service requests, trouble tickets, products, and so on.

11. AIA cross-references are created as part of an integration flow when an application requests AIA services to create corresponding entities in one or more other applications.
12. AIA cross-references are used to propagate any change or update to an entity from one application to the corresponding entities in other applications.
13. OCH must be configured to auto-trigger web services for bi-directional create/update of customers with a participating application.
14. The significant differences between the Oracle Customer Hub, Siebel CRM, and EBO data models with respect to modeling Customer entities necessitate that the solution include built-in mapping and configuration settings.

    These settings ensure that the out-of-the-box functionality and feature set of the solution are maximized and that the solution can be deployed independently or in conjunction with other PIP solutions. For more information about configuration settings, see Handling Errors.

15. Oracle Customer Hub must be deployed as a separate instance from Siebel CRM.
16. Siebel CRM, Oracle E-Business Suite, and SAP must be set up as Registered Systems in Oracle Customer Hub to enable local cross-referencing, survivorship rules, and the ability to manage CRUD privileges on each participating application.

    This setup is part of the standard Oracle Customer Hub configuration setup.

17. All organization and person messages published from Oracle Customer Hub are consumed by Oracle E-Business Suite out-of-the-box.
18. The functional impact is that a customer account is established in Oracle E-Business Suite in advance of a known financial relationship with the party, which may be desired from a business perspective; however, this functionality may undermine the inherent definition of a customer account under the Oracle E-Business Suite Trading Community Architecture (TCA) model. If this functionality is not required and that the records should be consumed by Oracle E-Business Suite only in the case of a financial transaction, either the routing rules need to be modified so that messages are not published to Oracle E-Business Suite or the source system registrations within Oracle Customer Hub need to be updated so that Oracle E-Business Suite is not included for published messages.
19. The match and fetch integration flow from Oracle E-Business Suite to Oracle Customer Hub uses a Composite Application Framework user interface.

    The Composite Application Framework supports FMW 11g; therefore, a separate 11g application server must be installed and supported in order for this specific feature to be implemented.

20. During a synchronization, if Oracle Customer Hub publishes an existing record for an incoming new record which is identified as a duplicate, the existing record is synchronized to the target applications whereas the incoming duplicate record is left as-is due to pending transactions.

    In this case, the duplicate records have to be analyzed by an administrator and manually merged.

21. Oracle Customer Hub cross-references are created by Oracle Customer Hub when Customer entities (accounts and contacts) are published from applications to Oracle Customer Hub and identified by Oracle Customer Hub as referring to the same party.

Oracle Customer Hub cross-references can link records across applications (for example, ACME Account in CRM and ACME Account in Oracle BRM) as well as within one single application (for example, ACME Account and ACME Inc. Account in CRM).

22. Oracle Customer Hub cross-references may also be created for applications that are not integrated through AIA.
23. 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.
24. While Oracle Customer Hub supports the internal unmerge of previously merged entities, no publishing takes place from Oracle Customer Hub as a result of an unmerge.

    This implies that if unmerge is used within Oracle Customer Hub, the corresponding records are not automatically updated in Siebel CRM/Oracle E-Business Suite. This can lead to inconsistencies in customer data between Oracle Customer Hub and Siebel CRM/Oracle E-Business Suite unless a manual process is adopted for updating survivors and re-creating victims in Siebel CRM/Oracle E-Business Suite or the integration solution is extended during deployment to support unmerge publishing so that in Siebel CRM/Oracle E-Business Suite survivors are updated and victims re-created automatically.

25. Siebel CRM accounts should not be merged when Oracle BRM is present in the deployment. Merging Siebel CRM accounts creates issues if the corresponding accounts are not merged in Oracle BRM.

    If the merger of survivor and victims does not take place in Oracle BRM, but it is performed in CRM, it is not possible to retrieve billing information for the victims. There can also be issues with assets that have been re-parented in Siebel CRM but not in Oracle BRM. A change order for a merged account may not be able to find the assets in the Oracle BRM, as they are still associated to the victim accounts.

26. The match process involves the passing of search criteria to the Oracle Customer Hub application to generate a list of possible candidates, and it is likely that these candidates have multiple addresses and phone numbers as part of their profile.

    In the Siebel-enabled user interface, only the primary address and phone number fields are presented as part of each candidate profile. This compressed view of displayed values enhances the user experience without sacrificing the ability for the user to make an informed decision about the candidate to be fetched. The Composite Application Framework user interface displays all of the addresses that are associated to the candidates.

27. When an organization or contact is synchronized with Oracle Customer Hub, the related organization is not fetched from the AIA cross-reference and passed into Oracle Customer Hub.

    In Oracle Customer Hub, accounts are supposed to be across business units, we call the Oracle Customer Hub web service without any organization IDs. However, within Oracle Customer Hub, the business unit is defaulted, as it is a required column. Therefore, we are supporting only one business unit for organization/contact synchronization into Oracle Customer Hub. However, if Oracle Customer Hub publishes records with multiple business units, those related organizations, if set up in the cross-reference and target applications are associated and supported as part of the integration flow.

28. OCH does not maintain cross-references for Oracle BRM customer data, because:
    - Oracle BRM does not publish customer data update/create to OCH. This is not in scope

of this process integration pack for OCH.

- Customer data is not created in Oracle BRM as result of publishing from Oracle BRM.
- AIA maintains cross-references for the accounts and contact information created in Oracle BRM.

29. Contact/Person update in OCH (Siebel UCM) does not propagate the changes to Oracle BRM.
30. Publishing of OCH unmerge service is not supported by this PIP
31. For Oracle BRM, Siebel CRM is the accounts master where accounts are created and updated.
32. Status updates are not propagated from OCH to Oracle BRM. In addition, status updates are not propagated by the AIACOM PIPs from CRM to Oracle BRM. The reason being that an account could become inactive in CRM while there are still pending charges that need to be calculated and applied to it by the billing system; therefore, it cannot be marked as inactive in Oracle BRM. Similarly, updates coming from OCH should not change the account status in Oracle BRM.
33. Billing profiles from the Siebel CRM system are synchronized at the most to one billing system. Customer accounts and billing profiles are first synchronized to the billing system during order processing and not prior.

    Once synchronized to a particular billing system, a customer account is kept synchronized using real time integration flows. The customer account synchronization that occurs during order processing assumes that if an account has already been created in the billing system, it is current and up-to-date.

34. Oracle BRM account merge is not in the scope for this OCH PIP.
35. Siebel CRM account hierarchy is not synchronized to the billing system.

    Instead, the billing account /service account relationship on a Siebel CRM order line is sent to the billing system as parent account and child account, respectively.

36. Creation of customer data in Oracle BRM takes place only as part of the order-submission flow part of the Order to Bill PIP.

    This enforces that prospects accounts are not created in Oracle BRM.

37. Initial Load from Oracle BRM/Siebel CRM/Oracle E-Business Suite into OCH is not in the scope.

    Initial loads are typically specific to particular deployment and require a customized solution.

38. OCH has capabilities to master non-customers (partners, prospects, suppliers, and so on), generally as an input to campaign management processes, partner loyalty/relationship programs, or other processes.

    Although this particular PIP assumes that OCH acts as a hub for mastering customers defined as entities with which a selling relationship has been established, the integration flows modeled as part of this solution do not require that the entities being mastered are necessarily customers.

39. SAP does not differentiate between person and organization type of customers. Hence, both person customer as well as organization type of customers must be created as ‘Accounts’ in OCH which are then mapped to ‘Customers’ in SAP.

40. During OCH to SAP synchronization, if OCH does not send organization, AIA populates default organization from the configuration properties and synchronizes with SAP.
41. Customer data enrichment is done one-person-record at a time and only for the primary address.
42. Initial Bulk load of customer data between SAP and OCH is out of scope.
43. In SAP there is only one address per customer; therefore, during SAP to OCH synchronization, SAP address is mapped to primary address in OCH. In addition, during OCH to SAP synchronization, SAP receives only primary address.
44. SAP sales area related attributes like sales organization, distribution channel and division are mapped to business unit/organization of OCH.
45. Every unique combination of the three-part SAP sales area key, there would be a corresponding business unit in OCH – Initial setup of sales area and corresponding OCH organization must be created or loaded by the users prior to using the PIP.
46. Account group field is mandatory in SAP but does not have exact mapping in OCH; therefore, AIA would populate a default value of *0001* in the configuration properties and send to SAP.
47. Language field is mandatory in SAP, but does not have exact mapping in OCH; therefore, AIA would populate a default value of *EN* (English) in the configuration properties and send to SAP.
48. Customer pricing procedure is mandatory in SAP, but does not have exact mapping in OCH; therefore, AIA would populate a default value of *1(one)* in the configuration properties and send to SAP.
49. Shipping conditions is mandatory in SAP, but does not have exact mapping in OCH; therefore, AIA would populate a default value of *02* in the configuration properties and send to SAP.
50. Tax classification is mandatory in SAP, but does not have exact mapping in OCH; therefore, AIA would populate a default value of *1* (one) in the configuration properties and send to SAP.
51. Sold-to, bill-to, ship-to and payer, which are required for customer creation in SAP are automatically populated to customer number in SAP.
52. Company code in SAP does not have exact mapping in OCH and it cannot be defaulted. Hence, OCH Synchronization is done without company code data in this PIP.

    Even though the customer master can be created without company code data, it affects further SAP functionality like invoicing and financial posting; therefore, company code data is must and should be populated, maintained separately in SAP using standard SAP transaction.

53. Transportation zones are country specific and it should be configured during the SAP implementation; therefore, there is no single default value available from vanilla SAP for all country codes.

    As there is no exact mapping in OCH, transportation zone can be defaulted to *0000000001*, with the prerequisite that *Transportation zone 0000000001* needs to be configured in SAP system for all the country codes before synchronization starts.

54. OCH to SAP job title synchronization is based on the JOB_TITLE DVM. If the OCH job title is not available in DVM, a blank value is sent to SAP.

55. Since IDOC is asynchronous and it does not return a response for a customer created or updated in SAP, in order to populate cross reference tables we are going with change pointers on SAP which trigger the response immediately once a customer is created or updated. This response is sent to the requestor service which receives cross references common values in 'NAME2', 'ORT02' and 'PARAU' IDOC fields from the Inbound IDOC. The field 'DATLT' in IDOC is used to capture the target system id in response flow. The field DATLT in IDOC is used to capture the target system ID in response flow. Hence, these fields should not be used or modified.
56. In order to differentiate outbound IDOC's and inbound IDOC's in AIA , whenever a customer is created or updated in SAP, a field 'KATR1'with value "B1" is sent along with the outbound IDOC.
57. Currently PIP handles five tax categories using TAX_CATEGORY DVM (for a given sales organization and country we may have a single/multiple tax categories in SAP), it can be extended by minimal code change to support any number of tax categories.
58. SAP does not support single contact being referenced by multiple customers. Hence, any contact being added to the accounts in OCH has to be a new contact and not an already existing one.

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

This chapter provides an overview of the process flows and discusses:

- Process Flows
- Siebel CRM Integration Services
- Siebel CRM Interfaces

Overview

These are the flows when Siebel CRM option is chosen when installing Oracle Customer Hub PIP:

- Match and Fetch between Siebel CRM and Oracle Customer Hub.
- Synchronizing Organizations and Persons from Siebel CRM to Oracle Customer Hub.

For process flows from Oracle Customer Hub to Siebel CRM, see Chapter 2: Oracle Customer Master Data Management Integration Base Pack.

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

Prerequisites

These are the prerequisites:

- Person requires First Name and Last Name.
- Organization requires Account Name.
- The business units being used must be seeded in all applications and in cross-references.
- The prerequisite is that accounts/contacts have been synchronized between Oracle Customer Hub and Siebel CRM prior to publishing the merge from the Oracle Customer Hub.

Process Flows

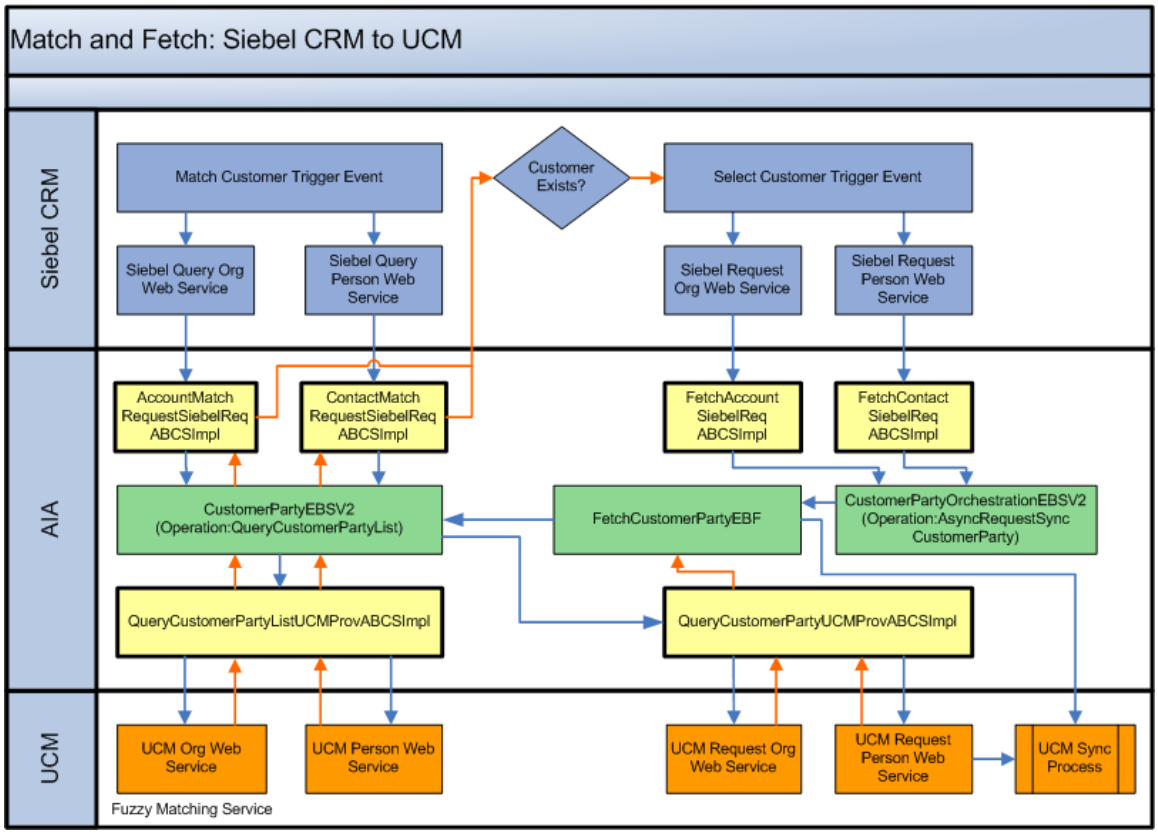
These are the process flows from Siebel CRM component.

- Match and Fetch between Siebel CRM and Oracle Customer Hub
- Synchronizing Organizations and Persons from Siebel CRM to Oracle Customer Hub

Match and Fetch between Siebel CRM and Oracle Customer Hub

This integration flow matches organizations (accounts) or persons (contacts) between Siebel CRM and OCH. This flow is initiated by Siebel CRM to generate a list of potential candidates from OCH. From the list presented, a specific organization or person is requested from OCH, which returns the full profile of the selected record. The full profile from OCH is then sent to Siebel at which point, the organization (account) or person (contact) is synchronized.

This graphic illustrates the match and fetch flow between Siebel CRM and Oracle Customer Hub:



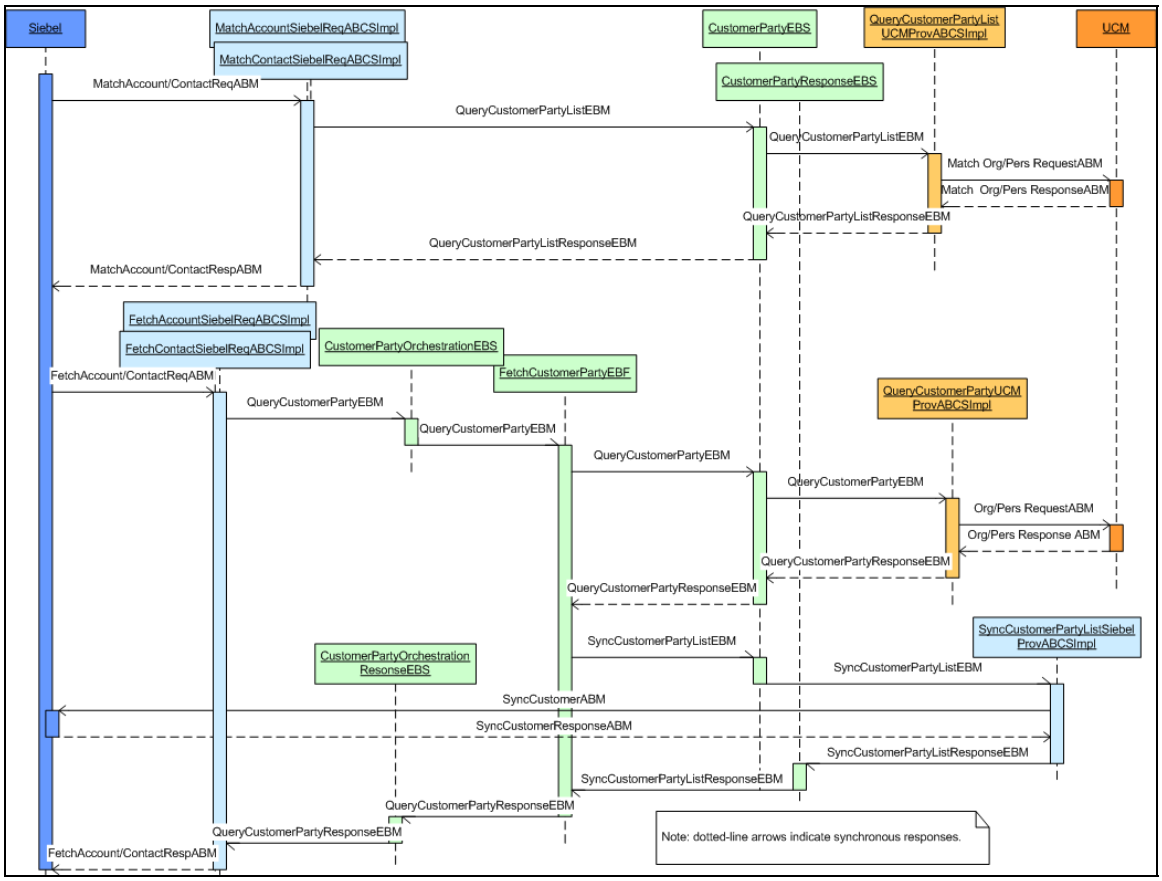
Match and fetch between Siebel CRM and Oracle Customer Hub

This integration flow uses these interfaces:

- MatchAccountSiebelReqABCSImpl
- MatchContactSiebelReqABCSImpl
- CustomerPartyEBSV2
- CustomerPartyResponseEBSV2

- QueryCustomerPartyListUCMProvABCSImpl
- FetchAccountSiebelReqABCSImpl
- FetchContactSiebelReqABCSImpl
- CustomerPartyOrchestrationEBSV2
- FetchCustomerPartyEBF
- QueryCustomerPartyUCMProvABCSImpl
- SyncCustomerPartyListSiebelProvABCSImpl
- CustomerPartyOrchestrationResponseEBSV2

This sequence diagram illustrates the integration flow:



Match and fetch between Siebel CRM to Oracle Customer Hub

When you initiate this process, these events occur:

1. In Siebel CRM, a CSR in a call center receives a call from a representative who may be a customer or an individual.

The CSR enters an organization or an individual's information, such as name and zip code, and clicks on the Match button from the Call Center Siebel application. This triggers the match flow where the Siebel application synchronously invokes the MatchAccountSiebelReqABCSImpl connector service if the caller was from an organization. The MatchContactSiebelReqABCSImpl connector service is synchronously invoked if the caller is an individual.

2. The MatchAccountSiebelReqABCSImpl or MatchContactSiebelReqABCSImpl connector service receives the Siebel Account or Contact Match requestor ABM and transforms it into the QueryCustomerPartyListEBM with an organization or person customer-party query criteria.

   It then synchronously invokes the QueryCustomerPartyList operation of the CustomerPartyEBSV2 enterprise business service passing in the QueryCustomerPartyListEBM.

3. The CustomerPartyEBSV2 routes the QueryCustomerPartyListEBM to the QueryCustomerPartyListUCMProvABCSImpl provider service, which is synchronously invoked.
4. The QueryCustomerPartyListUCMProvABCSImpl determines whether the request is for an organization or a person.

   It transforms the QueryCustomerPartyListEBM, which contains the match criteria to the Oracle Customer Hub specific organization or person Match requestor ABM. It invokes either the Oracle Customer Hub OrganizationMatchService or PersonMatchService.

5. The Oracle Customer Hub OrganizationMatchService or PersonMatchService queries the Oracle Customer Hub database, and based on the match criteria, returns the organization or person Match response ABM, which contains a list of the organization or person candidates back to the QueryCustomerPartyListUCMProvABCSImpl that it synchronously invoked.
6. The QueryCustomerPartyListUCMProvABCSImpl receives the organization or person Match response ABM and transforms it into the QueryCustomerPartyListResponseEBM.

   During the transformation, the common IDs are generated and cross-reference database is populated where necessary. The response message is sent back to the CustomerPartyEBSV2.

7. The CustomerPartyEBSV2 receives the QueryCustomerPartyListResponseEBM response from the QueryCustomerPartyListUCMProvABCSImpl and passes the message back to the MatchAccountSiebelReqABCSImpl or the MatchContactSiebelReqABCSImpl depending on which invoked it.
8. The MatchAccountSiebelReqABCSImpl or the MatchContactSiebelReqABCSImpl transforms the response to the Siebel specific account or contact Match requestor ABM and sends a list of candidate customer parties that meet the criteria to the Siebel CRM application.
9. The CSR reviews the list of customer parties displayed, and picks the exact account from the list by clicking on the Fetch button in the application.

   This triggers the Fetch flow, where the Siebel application synchronously invokes the FetchAccountSiebelReqABCSImpl connector service if the caller was from an organization. The FetchContactSiebelReqABCSImpl connector service is invoked synchronously when the caller is an individual.

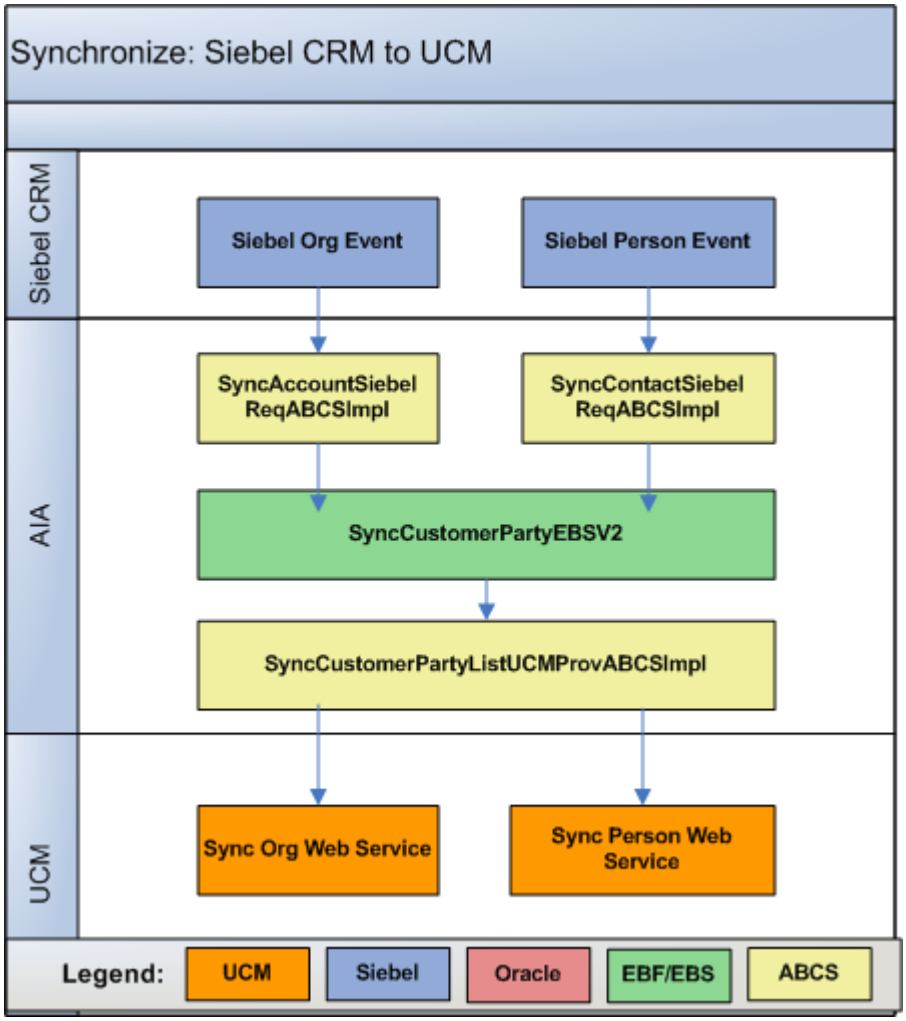
10. The FetchAccountSiebelReqABCSImpl or FetchContactSiebelReqABCSImpl connector service receives the Siebel account or contact Fetch requestor ABM and transforms it into the QueryCustomerPartyEBM with an organization or person customer-party query criteria. It then asynchronously invokes the AsyncFetchCustomerParty operation of the CustomerPartyOrchestrationEBSV2 enterprise business service, and passes in the QueryCustomerPartyEBM, which contains the account's identifier (the common ID).
11. The CustomerPartyOrchestrationEBSV2 is a routing service that asynchronously invokes the AsyncFetchCustomerParty operation of the FetchCustomerPartyEBF, and passes the QueryCustomerPartyEBM.
12. The FetchCustomerPartyEBF receives the QueryCustomerPartyEBM and synchronously invokes the QueryCustomerParty operation of the CustomerPartyEBSV2.
13. The CustomerPartyEBSV2 routes the QueryCustomerPartyEBM to the QueryCustomerPartyUCMProvABCSImpl provider service, which is synchronously invoked.
14. The QueryCustomerPartyUCMProvABCSImpl determines whether the request is for an organization or a person. It transforms the QueryCustomerPartyEBM, which contains the organization or person ID to the Oracle Customer Hub-specific organization or person fetch requestor ABM. It invokes either the Oracle Customer Hub OrganizationFetchService or PersonFetchService.
15. The Oracle Customer Hub OrganizationFetchService or PersonFetchService queries the Oracle Customer Hub database based on the organization or person ID. It returns the organization or person fetch response ABM, which contains the full organization or person data record from Oracle Customer Hub back to the QueryCustomerPartyUCMProvABCSImpl that synchronously invoked it.
16. The QueryCustomerPartyUCMProvABCSImpl transforms the fetch response ABM into QueryCustomerPartyResponseEBM. The response message is sent back to the CustomerPartyEBSV2.
17. The CustomerPartyEBSV2 receives the QueryCustomerPartyResponseEBM response from the QueryCustomerPartyUCMProvABCSImpl and then passes the message back to the FetchCustomerPartyEBF that invoked it.
18. The FetchCustomerPartyEBF receives the QueryCustomerPartyResponseEBM and transforms it into the SyncCustomerPartyListEBM, which asynchronously invokes the SyncCustomerPartyList operation of CustomerPartyEBSV2.
19. The SyncCustomerPartyList operation of CustomerPartyEBSV2 asynchronously routes the SyncCustomerPartyListEBM message to the SyncCustomerPartyListSiebelProvABCSImpl.
20. The SyncCustomerPartyListSiebelProvABCSImpl receives the SyncCustomerPartyListEBM, and then determines whether an account or contact synchronization is necessary. It transforms the SyncCustomerPartyListEBM into Siebel account or contact SyncCustomerABM and invokes the appropriate Siebel synchronization service.
21. The Siebel synchronization service creates the account or contact record and returns the SyncCustomerResponseABM, which contains the details of the record added.
22. The SyncCustomerPartyListSiebelProvABCSImpl receives the SyncCustomerResponseABM and transforms it into the SyncCustomerPartyListResponseEBM. As part of the transformation, the Siebel ID is added to the cross-reference table. It then invokes the SyncCustomerPartyListResponse operation of the CustomerPartyResponseEBSV2, and passes the SyncCustomerPartyListResponseEBM.

23. The FetchCustomerPartyEBF receives the SyncCustomerPartyListResponseEBM callback from CustomerPartyResponseEBSV2, and transforms the SyncCustomerPartyListResponseEBM into the QueryCustomerPartyResponseEBM. It then asynchronously invokes the AsyncFetchCustomerPartyResponse operation of CustomerPartyOrchestrationResponseEBSV2, and passes the QueryCustomerPartyResponseEBM.
24. The FetchAccountSiebelReqABCSImpl or FetchContactSiebelReqABCSImpl connector service receives the QueryCustomerPartyResponseEBM callback from CustomerPartyOrchestrationResponseEBSV2 and transforms the response into an account or contact fetch response ABM. It sends the response back to the Siebel application where the new account or contact information can be viewed.

Synchronizing Organizations and Persons from Siebel CRM to Oracle Customer Hub

This feature synchronizes customer-related entity information from one or several instances of Siebel CRM to Oracle Customer Hub. Oracle Customer Hub consumes the message using AIA and after applying configurable data quality and data cleansing processes to the record, determines whether to update an existing record or to create a new customer record based on survivorship and update precedence rules in Oracle Customer Hub.

This activity diagram illustrates the synchronization of organizations and persons from Siebel CRM to Oracle Customer Hub:



Synchronizing organizations and persons from Siebel CRM to Oracle Customer Hub

For this integration flow:

- The shared data model between Siebel and Oracle Customer Hub ensures that mappings are consistent among account, contact, and address entities.
- The creation of a new account (org customer) or contact (person customer) in Siebel CRM initiates a process to synchronize information to Oracle Customer Hub. New accounts and contacts should be customers but can be non-customers such as prospects.

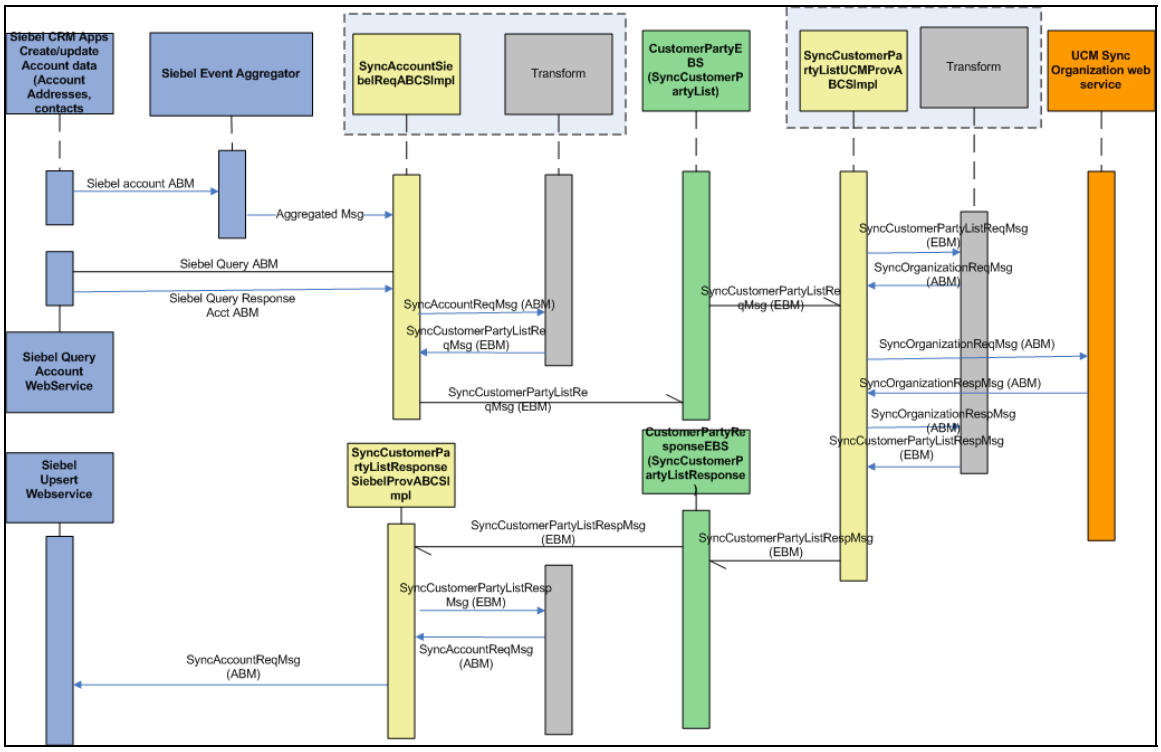
Synchronizing Organizations from Siebel CRM to Oracle Customer Hub

This integration flow synchronizes new accounts or updates from Siebel CRM to Oracle Customer Hub. This integration flow uses these interfaces:

- SyncCustomerSiebelEventAggregator
- SyncAcctSiebelAggrEventConsumer

- SyncAccountSiebelReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListUCMProvABCSImpl
- CustomerPartyResponseEBSV2
- SyncCustomerPartyListResponseSiebelProvABCSImpl
- SyncAccountSiebelAggregatorAdapter
- SyncAddressSiebelAggregatorAdapter
- SyncContactSiebelAggregatorAdapter

This sequence diagram illustrates the integration flow:



Synchronizing organizations from Siebel CRM to Oracle Customer Hub

When you initiate this process, these events occur:

1. Siebel account events are raised when a new account is created or updated in Siebel CRM. This includes creates or updates to addresses and contacts.

   This initiates the SyncCustomerSiebelEventAggregator, with the SiebelAccountABM message.

2. The SyncCustomerSiebelEventAggregator invokes one of three database adapters, depending on the type of event raised: account, contact, or address event.

The database adapters each invoke a PL/SQ procedure that performs the aggregation in a database table. The SyncAcctSiebelAggrEventConsumer listens on the aggregator table and polls the data. Based on the Polling Interval set, the records in the table are polled and the consumer invokes the SyncAccountSiebelReqABCSImpl service.

The polling interval for the account-aggregator-consumer-service is set to 90 seconds by default. To change this value:

a. Go to the Oracle ESB Console on the SOA server.

b. Select SyncAcctSiebelAggrEventConsumer service under AIASystem, Siebel, ABCS.

c. In the Properties tab select '+' button in the properties table. This displays the currently set polling interval.

d. Change the value of the polling interval as required.

e. Click Apply to save the changes.

f. Restart the SOA server for the changes to take effect.

3. The Siebel requester ABC implementation, SyncAccountSiebelReqABCSImpl, transforms the aggregated message into the Siebel Query Request ABM and invokes the Siebel Query web service to fetch the details of the Account (including account, address, and contact details).

4. The QueryResponse ABM is then transformed into the SyncCustomerPartyListEBM and the SyncCustomerPartyList operation of the CustomerPartyEBSV2 is invoked.

5. Invoking CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Oracle Customer Hub provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl.

6. The Oracle Customer Hub provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl, transforms the SyncCustomerPartyListEBM into the appropriate Oracle Customer Hub account ABM and invokes the Oracle Customer Hub API.

7. The response from the Oracle Customer Hub API is an integration object that has the account data if it was successfully created or updated in Oracle Customer Hub, which is then transformed into the SyncCustomerPartyListResponseEBM, in the process of which the cross-references tables are populated with the Oracle Customer Hub identifier values.

   The response is sent to the CustomerPartyResponseEBSV2 if the response comes back from Oracle Customer Hub. If the Oracle Customer Hub ABM message was put in a queue, then the response IO status has to be checked and the cross-reference entry for the Oracle Customer Hub ID must be created at a later time when the publish from Oracle Customer Hub happens for this account using integration IDs.

8. The SyncCustomerPartyListResponseEBM is transformed into the Siebel account ABM and the Siebel account web service is invoked.

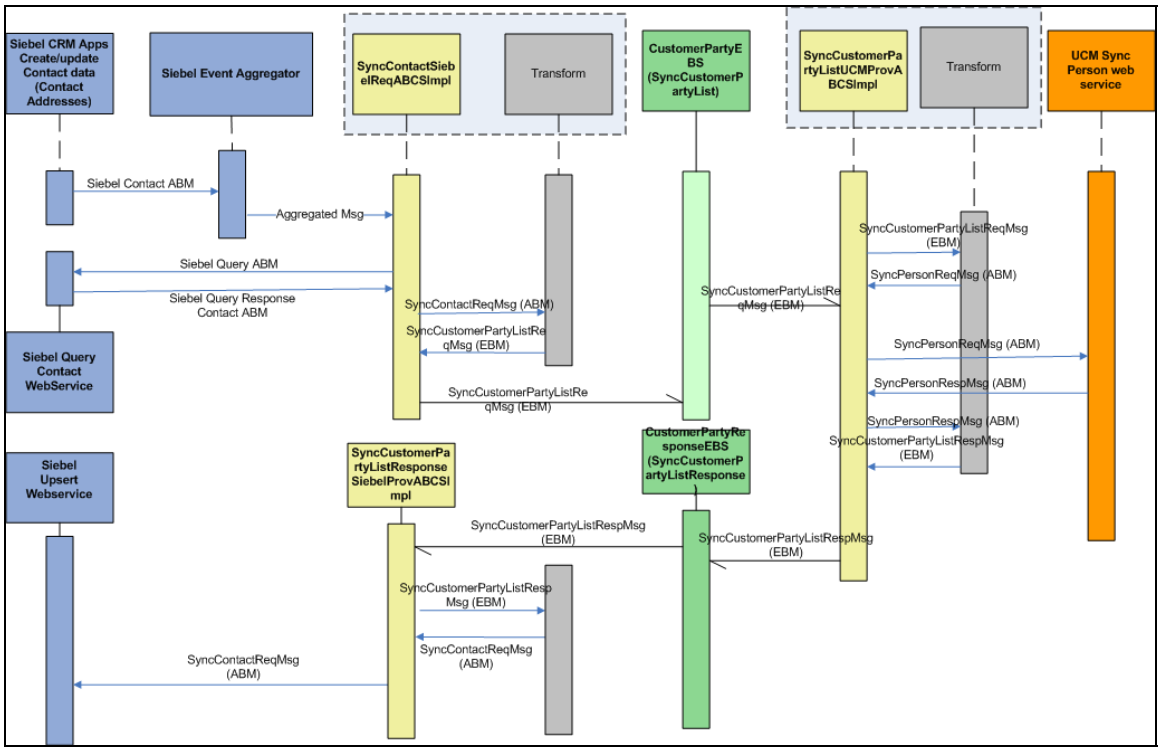
Synchronizing Persons from Siebel CRM to Oracle Customer Hub

This integration flow is used to synchronize new person or updates from Siebel to Oracle Customer Hub. This integration flow uses these interfaces:

- SyncCustomerSiebelEventAggregator

- SyncContSiebelAggrEventConsumer
- SyncContactSiebelReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListUCMProvABCSImpl
- CustomerPartyResponseEBSV2
- SyncCustomerPartyListResponseSiebelProvABCSImpl
- SyncAddressSiebelAggregatorAdapter
- SyncContactSiebelAggregatorAdapter

This sequence diagram illustrates the integration flow:



Synchronizing persons from Siebel CRM to Oracle Customer Hub

When you initiate this process, these events occur:

1. Siebel contact events are raised when a new account is created or updated in Siebel CRM. This includes creates or updates to addresses and contacts.

   This initiates the SyncCustomerSiebelEventAggregator with the SiebelContactABM message.

2. The SyncCustomerSiebelEventAggregator invokes one of two database adapters, depending on the type of event raised: contact or address event.

   The database adapters each invoke a PL/SQ procedure that performs the aggregation in a database table. The SyncContSiebelAggrEventConsumer listens on the aggregator table and polls the data. Based on the polling interval set, the records in the table are polled and the consumer invokes the SyncAccountSiebelReqABCSImpl service.

The polling interval for the contact-aggregator-consumer-service is set to 60 seconds by default. To change this value:

a. Go to the Oracle ESB Console on the SOA server.

b. Select SyncContSiebelAggrEventConsumer service under AIASystem, Siebel, ABCS.

c. In the **Properties** tab select '+' button in the properties table.

This displays the currently set polling interval.

d. Change the value of the polling interval as required.

e. Click **Apply** to save the changes.

f. Restart the SOA server for the changes to take effect.

The Siebel requester ABC implementation, SyncContactSiebelReqABCSImpl, transforms the aggregated message to the Siebel Query request ABM and invokes the Siebel Query web service to fetch the details of the account (including personal address and contact details). The QueryResponse ABM is then transformed into the SyncCustomerPartyListEBM and the SyncCustomerPartyList operation of the CustomerPartyEBSV2 is invoked.

3. Invoking the CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Oracle Customer Hub provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl.
4. The Oracle Customer Hub provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl, transforms the SyncCustomerPartyListEBM into the appropriate Oracle Customer Hub person ABM and invokes the Oracle Customer Hub API.
5. The response from the Oracle Customer Hub API is an integration object that has the person data if it was successfully created or updated in Oracle Customer Hub and is then transformed into the SyncCustomerPartyListResponseEBM in the process of which, the cross-references tables are populated with the Oracle Customer Hub identifier values.

   The response is sent to the CustomerPartyResponseEBSV2 if the response comes back from Oracle Customer Hub. If the Oracle Customer Hub ABM message was put in a queue, then the response IO status has to be checked and the cross-reference entry for the Oracle Customer Hub ID must be created at a later time when the publish from Oracle Customer Hub happens for this person using integration IDs.

6. The SyncCustomerPartyListResponseEBM is transformed into the Siebel contact ABM and the Siebel contact web service is invoked.

Siebel CRM Integration Services

These are the integration services for Siebel CRM:

- SyncAccountSiebelAggregatorAdapter
- SyncContactSiebelAggregatorAdapter
- SyncAddressSiebelAggregatorAdapter
- SyncCustomerSiebelEventAggregator

- SyncAccountSiebelReqABCSImpl
- SyncContactSiebelReqABCSImpl
- SyncCustomerPartyListSiebelProvABCSImpl
- SyncCustomerPartyListResponseSiebelProvABCSImpl
- SyncBPSiebelAggregatorAdapter
- SyncAcctSiebelAggrEventConsumer
- SyncContSiebelAggrEventConsumer

SyncAccountSiebelAggregatorAdapter

The SyncAccountSiebelAggregatorAdapter is implemented as an ESB process with a database adapter and routing services. This service aggregates the account events generated in the Siebel application when an account is created or updated. This service invokes a PL/SQL procedure, AIA_AGGREGATOR_PUB.SIEBEL_AGGREGATE_ACCOUNT, which does the actual aggregation in the AIA aggregator table.

SyncContactSiebelAggregatorAdapter

The SyncContactSiebelAggregatorAdapter is implemented as an ESB process with a database adapter and routing services. This service aggregates the account events generated in the Siebel application when a contact is created or updated. This service invokes a PL/SQL procedure, AIA_AGGREGATOR_PUB.SIEBEL_AGGREGATE_CONTACT, which does the actual aggregation in the AIA aggregator table.

SyncAddressSiebelAggregatorAdapter

The SyncAddressSiebelAggregatorAdapter is implemented as an ESB process with a database adapter and routing services. This service aggregates the account events generated in the Siebel application when an account's address is created or updated. This service invokes a PL/SQL procedure, AIA_AGGREGATOR_PUB.SIEBEL_AGGREGATE_ADDRESS, which does the actual aggregation in the AIA aggregator table.

SyncCustomerSiebelEventAggregator

The SyncCustomerSiebelEventAggregator is implemented as a BPEL process since it involves JMS Header manipulations that cannot be done in ESB. This service has three operations, one for each kind of event raised by the Siebel application when an account, contact, or address is created or updated, or both. Each operation invokes the corresponding aggregator adapter service, which does the aggregation of the event in the AIA Aggregator table. These operations are defined in this service:

- aggregateaccountevent
- aggregatecontactevent

- aggregateaddressevent

SyncAccountSiebelReqABCSImpl

The SyncAccountSiebelReqABCSImpl service is responsible for transforming the Siebel message into the SyncCustomerPartyList EBM format and invoking the SyncCustomerPartyList operation of the CustomerPartyEBSV2. This operation is defined in SyncAccount service.

SyncContactSiebelReqABCSImpl

The SyncContactSiebelReqABCSImpl service is responsible for transforming the Siebel message into the SyncCustomerPartyList EBM format and invoking the SyncCustomerPartyList operation of the CustomerPartyEBSV2. This operation is defined in SyncContact service.

SyncCustomerPartyListSiebelProvABCSImpl

The SyncCustomerPartyListSiebelProvABCSImpl service is implemented in BPEL and is invoked by the SyncCustomerPartyList operation of the CustomerPartyEBSV2, passing in the Synchronize CustomerPartyListEBM payload. This service transforms the EBM into a Siebel ABM and invokes the Siebel web service interface.

SyncCustomerPartyListResponseSiebelProvABCSImpl

The SyncCustomerPartyListResponseSiebelProvABCSImpl service is implemented in BPEL and is invoked by the SyncCustomerPartyListResponse operation of the CustomerPartyResponseEBSV2, passing in the SyncCustomerPartyListResponseEBM payload. This service transforms the EBM into a Siebel ABM and invokes the Siebel web service interface to synchronize the response.

SyncBPSiebelAggregatorAdapter

The SyncBPSiebelAggregatorAdapter is implemented as an ESB process with a database adapter and routing services. This service aggregates the account events generated in the Siebel application when a Billing Profile is created or updated. This service invokes a PL/SQL procedure, AIA_AGGREGATOR_PUB.SIEBEL_AGGREGATE_BP, which does the actual aggregation in the AIA aggregator table.

SyncAcctSiebelAggrEventConsumer

The SyncAcctSiebelAggrEventConsumer service is responsible for dequeueing the aggregated Siebel Account identifiers from the AIA Aggregator table and invoking the appropriate operation of the Requestor ABCS, SyncAccountSiebelReqABCSImpl.

SyncContSiebelAggrEventConsumer

The SyncContSiebelAggrEventConsumer service is responsible for dequeueing the aggregated Siebel Contact identifiers from the AIA Aggregator table and invoking the appropriate operation of the Requestor ABCS, SyncContactSiebelReqABCSImpl.

For more information about the Event Aggregation programming model, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide*, "Describing the Event Aggregation Programming Model."

Siebel CRM Interfaces

These are the Siebel interfaces.

Inbound Web Services: Organization and Person Synchronization

Name	Schema/element
SWICustomerPartyServices.wsdl Operation: SWICustomerPartyInsertOrUpdate	SWICustomerPartyIO.xsd (ListOfSwicustomerpartyio)
SWIContactIO.wsdl Operation: SWIContactServicesInsertOrUpdate	SWIContactIO.xsd (ListOfSwicontactio)
For the merge: WSDL name: SWIMergeService.wsdl Operation: SWIMergeServicesMerge	SWIMergeObjectIO.xsd (ListOfSwimergeobjectio)

Outbound Web Services: Organization and Person Synchronization

Name	Schema
SyncCustomerSiebelEventAggregator (operation: aggregateaccountevent)	SWICustomerPartyIO.xsd (ListOfSwicustomerpartyio)
SyncCustomerSiebelEventAggregator (operation: aggregatecontactevent)	SWIContactIO.xsd (ListOfSwicontactio)
SyncCustomerSiebelEventAggregator (operation: aggregateaddressevent)	SWIAddressIO.xsd (ListOfSwiaddressio)

For more information about Siebel CRM web services, navigate to the documentation library for Siebel applications on Oracle Technology Network and see *Siebel CRM Web Services Reference.*

Inbound Web Services: Match and Fetch

These are the inbound web services for Match and Fetch:

Account Synchronization

- Service: SWICustomerParty
- Operation: SWICustomerPartyInsertOrUpdate

Contact Synchronization

- Service: SWIContactServices
- Operation: SWIContactServicesInsertOrUpdate

Outbound Web Services: Match and Fetch

Name	Description
MatchAccountSiebelReqABCSImpl	The MatchAccountSiebelReqABCSImpl service is synchronously invoked by Siebel CRM when a user enters match criteria in its Account form and triggers the match flow. The match flow returns a list of potential candidates that fits the criteria. This service supports this operation: MatchAccount.
MatchContactSiebelReqABCSImpl	The MatchContactSiebelReqABCSImpl service is synchronously invoked by Siebel CRM when a user enters match criteria in its Contact form and triggers a match flow. The match flow returns a list of potential candidates that fits the criteria. This service supports this operation: MatchContact.
FetchAccountSiebelReqABCSImpl	The FetchAccountSiebelReqABCSImpl service is synchronously invoked by Siebel CRM after the user picks an account from the list of candidates as the best match and requests that it be synchronized to Siebel. This service supports these operations: • FetchAccount • AsyncFetchCustomerPartyResponse
FetchContactSiebelReqABCSImpl	The FetchContactSiebelReqABCSImpl service is synchronously invoked by Siebel CRM after the user

Name	Description
	picks a contact from the list of candidates as the best match and requests that it be synchronized to Siebel. This service supports these operations: • FetchContact • AsyncFetchCustomerPartyResponse
SyncCustomerPartyListSiebelProvABCSImpl	The match and fetch flows use some of the same services as the synchronization flows.

For more information about Siebel CRM web services, navigate to the documentation library for Siebel applications on Oracle Technology Network and see *Siebel CRM Web Services Reference.*

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

This chapter provides an overview of Oracle E-Business Suite (Oracle E-Business Suite) and discusses:

- Process flows from Oracle E-Business Suite.
- Oracle E-Business Suite Interfaces
- Oracle E-Business Suite Integration Services

Overview

These are the process flows when Oracle E-Business Suite is chosen as a participating application when installing Oracle Customer Hub:

- Match and Fetch between Oracle E-Business Suite and Oracle Customer Hub.
- Synchronization of Organizations and Persons from Oracle E-Business Suite to Oracle Customer Hub.

For process flows from Oracle Customer Hub to Oracle E-Business Suite, see Chapter 2: Oracle Customer Master Data Management Integration Base Pack.

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

Prerequisites

- For the match/fetch integration flow between Oracle E-Business Suite and Oracle Customer Hub, you must install ADF runtime.

For more information about installing the ADF runtime, see *My Oracle Support note 746109.1: Deploying the Universal Customer Master Match/Fetch Composite User Interface*.

- Address is required in Siebel CRM in order to create an account in Oracle E-Business Suite.
- Accounts/contacts have been synchronized between Oracle Customer Hub and Oracle E-Business Suite prior to publishing the merge from the Oracle Customer Hub.

Process Flows

These are the process flows from Oracle E-Business Suite.

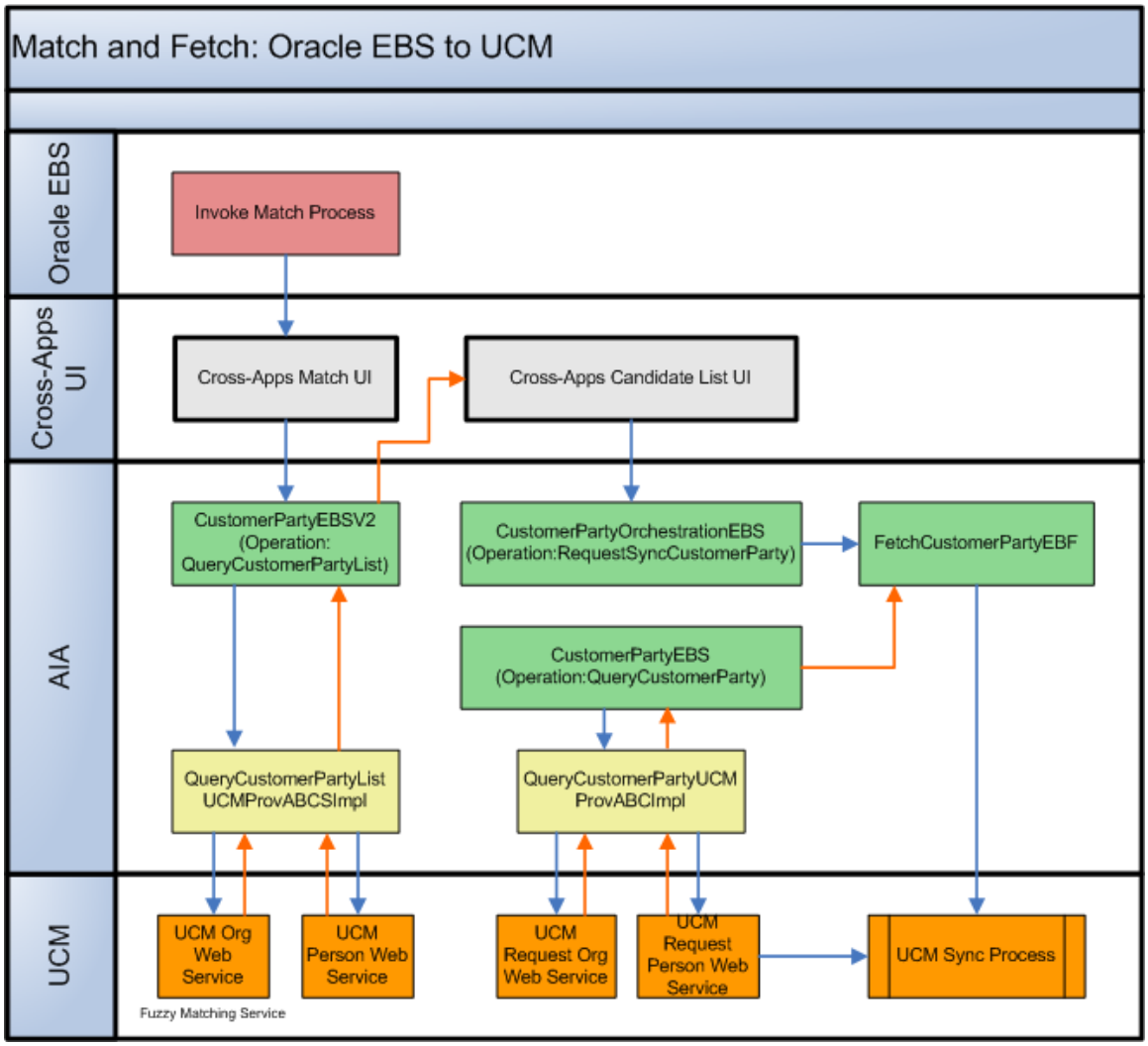
Match and Fetch between Oracle E-Business Suite and Oracle Customer Hub

The match or fetch integration flow is used to match organization and person parties between Oracle E-Business Suite and Oracle Customer Hub. This flow is initiated by Oracle E-Business Suite to generate a list of potential candidates from Oracle Customer Hub. From the list presented, a specific organization or person party entity is requested from Oracle Customer Hub, which returns the full profile of the selected record. The full profile from Oracle Customer Hub is then sent to Oracle E-Business Suite, at which point the parties are synchronized.

This integration flow leverages the Composite Application Framework to create a user interface (UI) that enables users to access Oracle Customer Hub to search for and to fetch the customer record from within Oracle E-Business Suite.

For more information about the match and fetch flow between Oracle E-Business Suite and Oracle Customer Hub, including the delivered Composite Application Framework UI, see the *My Oracle Support note 746109.1: Deploying the Universal Customer Master Match/Fetch Composite User Interface*.

This graphic illustrates the match and fetch functionality between Oracle E-Business Suite and Oracle Customer Hub.

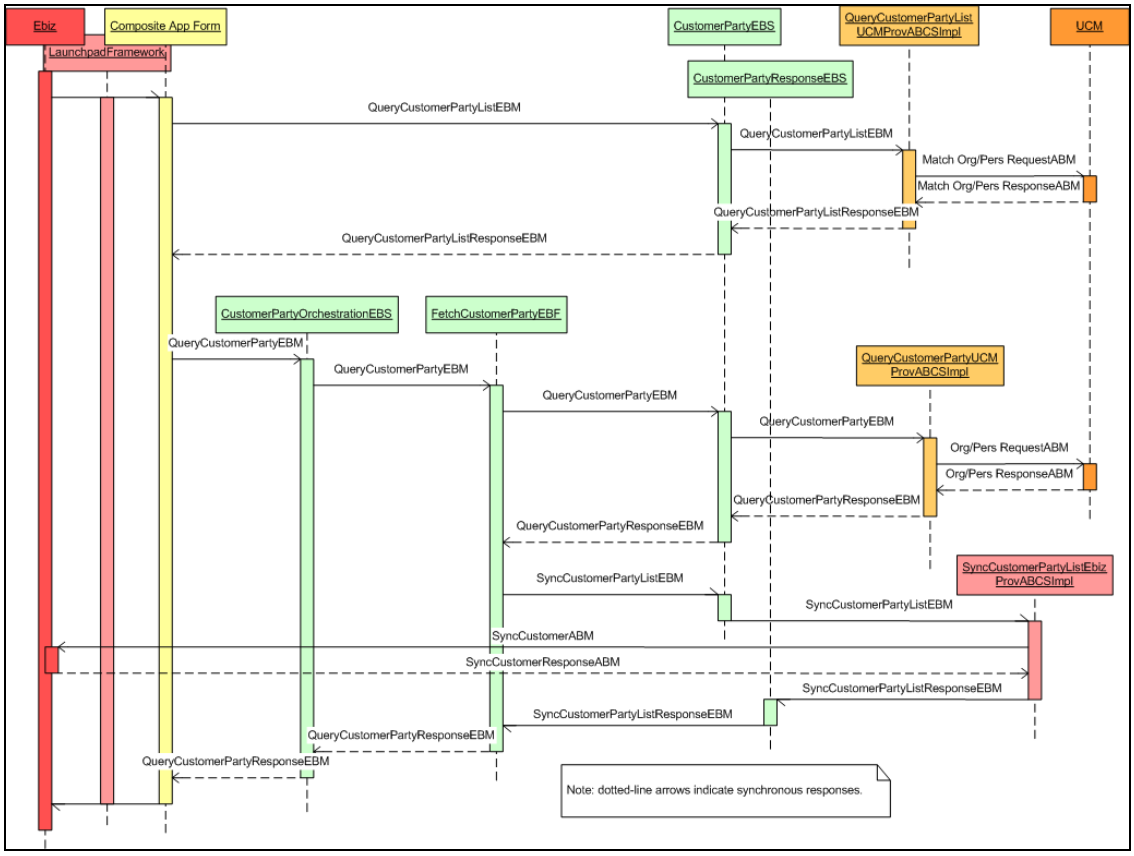


Match and fetch Oracle E-Business Suite to Oracle Customer Hub

This integration flow uses these interfaces:

- CustomerPartyEBSV2
- CustomerPartyResponseEBSV2
- QueryCustomerPartyListUCMProvABCSImpl
- CustomerPartyOrchestrationEBSV2
- FetchCustomerPartyEBF
- QueryCustomerPartyUCMProvABCSImpl
- SyncCustomerPartyListEbizProvABCSImpl

This sequence diagram illustrates the integration flow:



Match and fetch between Oracle E-Business Suite and Oracle Customer Hub

When you initiate this process, these events occur:

1. When a user accesses the Oracle E-Business Suite Customer Module to search for a party in the Customer Module, the composite UI application form appears.

   The user enters optional and required information (such as Organization Name and Zip Code) and clicks Search. This triggers the match flow, where the composite UI application synchronously invokes the QueryCustomerPartyList operation of the CustomerPartyEBSV2, and passes the QueryCustomerPartyListEBM.

2. The CustomerPartyEBSV2 routes the QueryCustomerPartyListEBM to the QueryCustomerPartyListUCMProvABCSImpl provider service is synchronously invoked.

3. The QueryCustomerPartyListUCMProvABCSImpl determines whether the request is for an organization or a person. It transforms the QueryCustomerPartyListEBM, which contains the match criteria, into the Oracle Customer Hub specific organization or person match requestor ABM.

   It invokes either the Oracle Customer Hub OrganizationMatchService or PersonMatchService.

4. The Oracle Customer Hub OrganizationMatchService or PersonMatchService queries the Oracle Customer Hub database based on the match criteria and returns the organization or person match response EBM, which contains a list of organization or person candidates back to the QueryCustomerPartyListUCMProvABCSImpl, which synchronously invoked it.

5. The QueryCustomerPartyListUCMProvABCSImpl receives the organization or person match response ABM and transforms it into the QueryCustomerPartyListResponseEBM.

   During the transformation, the common IDs are generated and the cross-reference database is populated where necessary. The response message is sent back to the CustomerPartyEBSV2.

6. The CustomerPartyEBSV2 receives the QueryCustomerPartyListResponseEBM response from the QueryCustomerPartyListUCMProvABCSImpl and passes the message back to the composite UI Candidates List form, which displays the records that matched the criteria.
7. The user clicks the Request button to select the most appropriate record from the composite UI Candidates List form.

   This triggers the fetch flow where the composite UI form synchronously invokes the FetchCustomerParty operation of the CustomerPartyOrchestrationEBSV2 business service. The QueryCustomerPartyEBM is passed with the Organization or Person common ID as the query criteria.

8. The CustomerPartyOrchestrationEBSV2 is a routing service that synchronously invokes the FetchCustomerParty operation of the FetchCustomerPartyEBF, and passes the QueryCustomerPartyEBM.
9. The FetchCustomerPartyEBF receives the QueryCustomerPartyEBM and synchronously invokes the QueryCustomerParty operation of the CustomerPartyEBSV2.
10. The CustomerPartyEBSV2 routes the QueryCustomerPartyEBM to the QueryCustomerPartyUCMProvABCSImpl provider service, which is synchronously invoked.
11. The QueryCustomerPartyUCMProvABCSImpl determines whether the request is for an organization or a person. It transforms the QueryCustomerPartyEBM, which contains the organization or person ID to the Oracle Customer Hub specific organization or person fetch service-requestor ABM.

    It invokes either the Oracle Customer Hub OrganizationFetchService or PersonFetchService.

12. The Oracle Customer Hub OrganizationFetchService or PersonFetchService queries the Oracle Customer Hub database based on the organization or person ID.

    It returns the organization or person fetch response ABM that contains the full organization or person data record from Oracle Customer Hub back to the QueryCustomerPartyUCMProvABCSImpl, which synchronously invoked it.

13. The QueryCustomerPartyUCMProvABCSImpl transforms the fetch response ABM into QueryCustomerPartyResponseEBM.

    The response message is sent back to the CustomerPartyEBSV2.

14. The CustomerPartyEBSV2 receives the QueryCustomerPartyResponseEBM response from QueryCustomerPartyUCMProvABCSImpl and passes the message back to the FetchCustomerPartyEBF, which invoked it.
15. The FetchCustomerPartyEBF receives the QueryCustomerPartyResponseEBM and transforms it into the SyncCustomerPartyListEBM.

    It asynchronously invokes the SyncCustomerPartyList operation of CustomerPartyEBSV2.

16. The SyncCustomerPartyList operation of the CustomerPartyEBSV2 asynchronously routes the SyncCustomerPartyListEBM message to the SyncCustomerPartyListEbizProvABCSImpl.

17. The SyncCustomerPartyListEbizProvABCSImpl receives the SyncCustomerPartyListEBM, and determines whether an account or contact synchronization is necessary.

    It transforms the SyncCustomerPartyListEBM into the Oracle E-Business Suite organization or person SyncCustomerABM and invokes the appropriate Oracle E-Business Suite synchronization service.

18. The Oracle E-Business Suite synchronization service creates the party and account records and returns the SyncCustomerResponseABM, which contains the details of the record added.
19. The SyncCustomerPartyListEbizProvABCSImpl receives the SyncCustomerResponseABM and transforms it into the SyncCustomerPartyListResponseEBM.

    As part of the transformation, the Oracle E-Business Suite ID is added to the cross-reference table. It then invokes the SyncCustomerPartyListResponse operation of the CustomerPartyResponseEBSV2, and passes the SyncCustomerPartyListResponseEBM.

20. The FetchCustomerPartyEBF receives the SyncCustomerPartyListResponseEBM callback from CustomerPartyResponseEBSV2, and transforms the SyncCustomerPartyListResponseEBM into the QueryCustomerPartyResponseEBM and returns the response to the CustomerPartyOrchestrationEBSV2.
21. The CustomerPartyOrchestrationEBSV2 returns the QueryCustomerPartyResponseEBM back to the composite UI with the full information of the party that was created.

For more information about the match and fetch flow between Oracle E-Business Suite and Oracle Customer Hub, including the delivered Composite Application Framework UI, see the *My Oracle Support note 746109.1: Deploying the Universal Customer Master Match/Fetch Composite User Interface*.

Synchronization of Organizations and Persons from Oracle E-Business Suite to Oracle Customer Hub

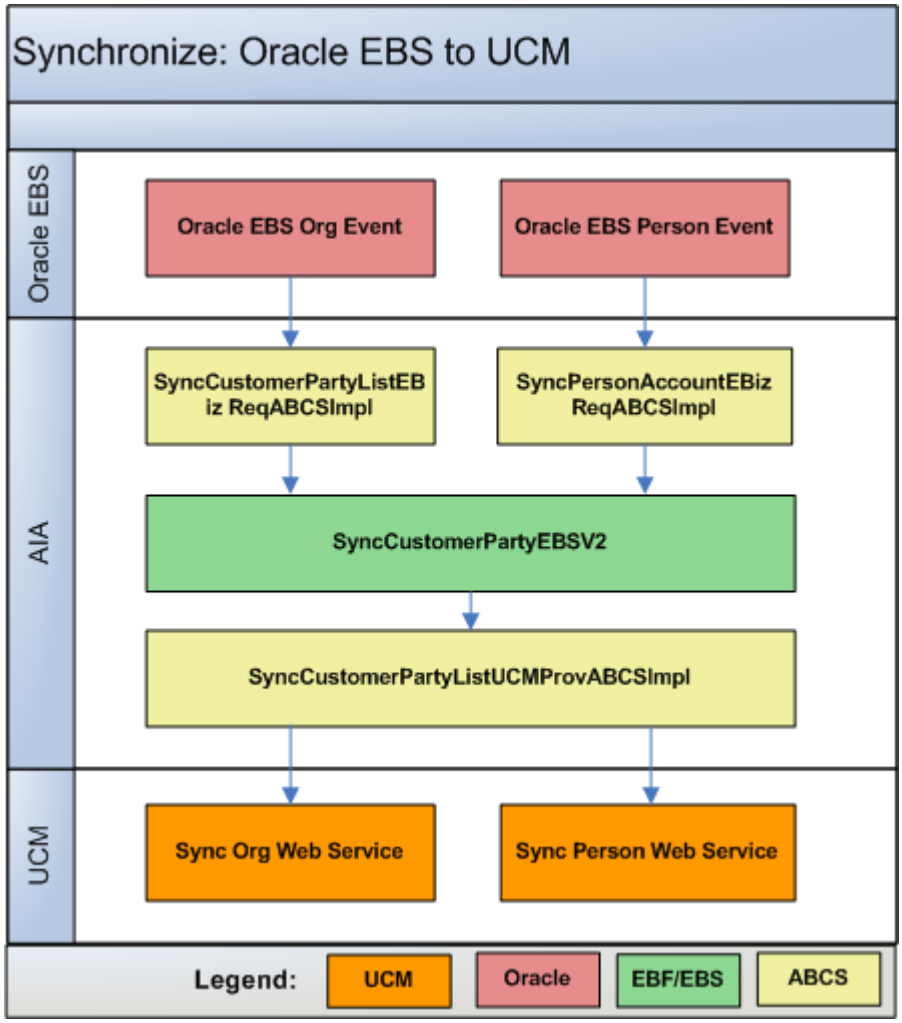
This integration flow synchronizes organization and person party and customer account information from Oracle E-Business Suite to Oracle Customer Hub. Oracle Customer Hub consumes the message using AIA and after applying configurable data quality and data cleansing processes to the record, determines whether to update an existing record or to create a new customer record based on survivorship and update precedence rules in Oracle Customer Hub.

For this integration flow:

- Only Oracle E-Business Suite parties with customer accounts are synchronized to Oracle Customer Hub. Customers are party entities (org party or person party) that have an associated account relationship.
- Creating a new customer account record in Oracle E-Business Suite for a party raises a business event to kick off a real time synchronization flow to synchronize this information to Oracle Customer Hub:
    - Party and customer account records to an account (org customer) or contact (person customer).

    - Synchronize account sites information to UCM address information.
    - Synchronize account contact information to UCM contact information.
- Updating any of these records for a customer account raises a similar synchronization event to synchronize information to Oracle Customer Hub.

This activity diagram illustrates the synchronize organizations and persons from Oracle E-Business Suite to Oracle Customer Hub:



Synchronizing organizations and persons from Oracle E-Business Suite to Oracle Customer Hub

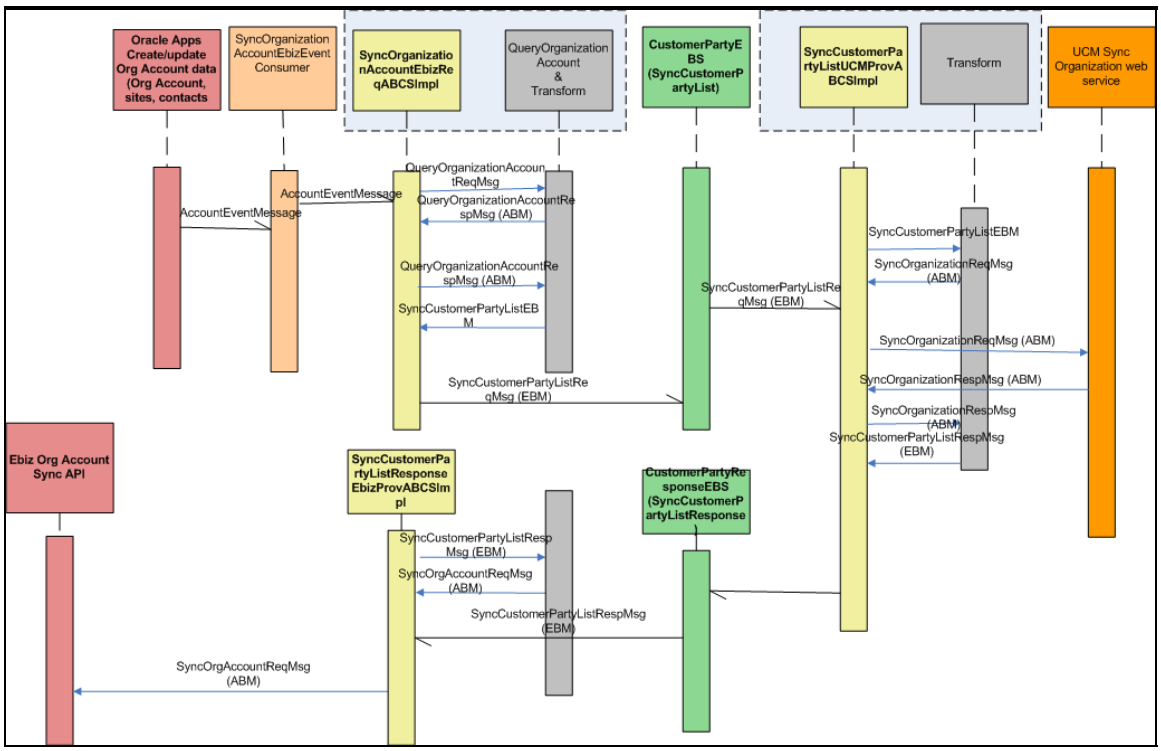
Synchronizing Organizations from Oracle E-Business Suite to Oracle Customer Hub

This integration flow is used to synchronize new organization accounts or updates from Oracle E-Business Suite to Oracle Customer Hub. This integration flow uses these interfaces:

- SynOrganizationAccountEbizEventConsumer
- SyncOrganizationAccountEbizReqABCSImpl

- QueryOrganizationAccountEbizAdapter
- CustomerPartyEBSV2
- SyncCustomerPartyListUCMProvABCSImpl
- CustomerPartyResponseEBSV2
- SyncCustomerPartyListResponseEbizProvABCSImpl

This sequence diagram illustrates the integration flow:



Synchronizing organizations from Oracle E-Business Suite to Oracle Customer Hub

When you initiate this process, these events occur:

1. Oracle E-Business Suite account events are raised when a new account is created or updated in Oracle E-Business Suite.

   This includes creates or updates to related addresses and contacts.

2. The SyncOrganizationAccountEbizJMSConsumer service dequeues the messages and invokes the SyncAccountEbizReqABCSImpl.

   The consumer has two adapters defined to dequeue both create and update events.

3. The Oracle E-Business Suite requester ABC implementation, SyncOrganizationAccountEbizReqABCSImpl first queries the entire payload using the keys in the event message using the QueryOrganizationAccountEbizAdapter and then transforms the Oracle E-Business Suite ABM into the SyncCustomerPartyListEBM and invokes the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

This implementation service has separate branches for create and update, as the APIs to query the entire payload from EBS are different; therefore, the maps to the EBM are also different.

4. Invoking the CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Oracle Customer Hub provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl.
5. The Oracle Customer Hub provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl, transforms the SyncCustomerPartyListEBM to the appropriate Oracle Customer Hub account ABM and invokes the Oracle Customer Hub API.
6. The response from the Oracle Customer Hub API is an integration object that has the account data if it was successfully created or updated, or both in Oracle Customer Hub, which is then transformed to the SyncCustomerPartyListResponseEBM in the process of which, the cross references tables are populated with the Oracle Customer Hub identifier values.

   The response is also sent to the CustomerPartyResponseEBSV2 if the response comes back from Oracle Customer Hub. If the Oracle Customer Hub ABM message was put in a queue, then the response IO status has to be checked and the cross-reference entry for the Oracle Customer Hub ID must be created at a later time when the publish from Oracle Customer Hub happens for this account using integration IDs.

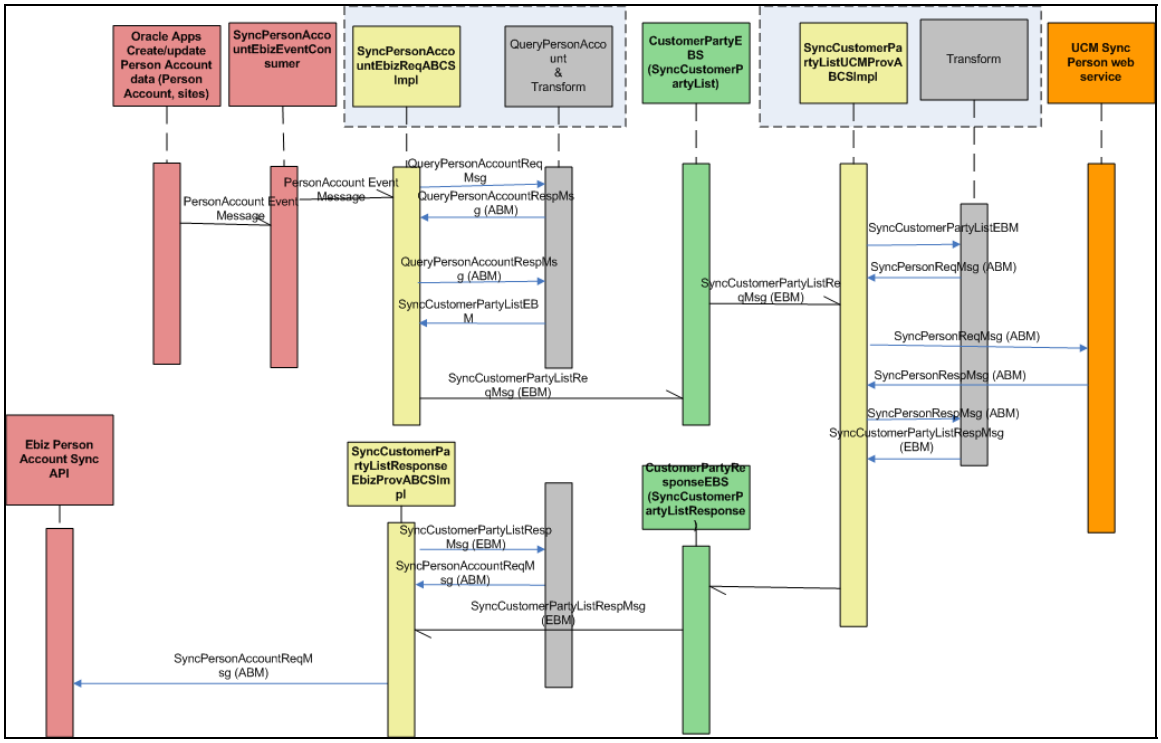
7. The SyncCustomerPartyListResponseEBM is transformed to Oracle E-Business Suite organization account ABM and the Oracle E-Business Suite organization account API is invoked.

Synchronizing Persons from Oracle E-Business Suite to Oracle Customer Hub

This integration flow synchronizes new persons or updates from Oracle E-Business Suite to Oracle Customer Hub. This integration flow uses these interfaces:

- SynPersonAccountEbizEventConsumer
- SyncPersonAccountEbizReqABCSImpl
- QueryPersonAccountEbizAdapter
- CustomerPartyEBSV2
- SyncCustomerPartyListUCMProvABCSImpl
- CustomerPartyResponseEBSV2
- SyncCustomerPartyListResponseEbizProvABCSImpl

This sequence diagram illustrates the integration flow:



Synchronizing persons from Oracle E-Business Suite to Oracle Customer Hub

When you initiate this process, these events occur:

1. The Oracle E-Business Suite person account events are raised when a new person account is created or updated in Oracle E-Business Suite.

   This includes creates/updates to related addresses.

2. The SyncPersonAccountEbizJMSConsumer service dequeues the messages and invokes the SyncPersonAccountEbizReqABCSImpl.

   This consumer has two adapters defined to listen on both create and update events.

3. The Oracle E-Business Suite requester ABC implementation, SyncPersonAccountEbizReqABCSImpl, first queries the entire payload using the keys in the event message using the QueryPersonAccountEbizAdapter, then transforms the Oracle E-Business Suite ABM into the SyncCustomerPartyListEBM, and invokes the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

4. This implementation service has separate branches for create and update, as the APIs to query the entire payload from EBS are different; therefore, the maps to EBM are also different.

5. Invoking the CustomerPartyEBSV2 with the SyncCustomerPartyList operation routes the SyncCustomerPartyListEBM to the Oracle Customer Hub provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl.

6. The Oracle Customer Hub provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl, transforms the SyncCustomerPartyListEBM into the appropriate Oracle Customer Hub person ABM and invokes the Oracle Customer Hub API.

7. The response from the Oracle Customer Hub API is an integration object that has the person data if it was successfully created or updated in Oracle Customer Hub.

   This object is then transformed into the SyncCustomerPartyListResponseEBM during which, the cross-reference tables are populated with the Oracle Customer Hub identifier values. The response is sent to the CustomerPartyResponseEBSV2 if the response comes back from Oracle Customer Hub. If the Oracle Customer Hub ABM message was put in a queue, then the response IO status has to be checked and the cross-reference entry for the Oracle Customer Hub ID must be created at a later time when the publish from Oracle Customer Hub happens for this person using integration IDs.

8. The SyncCustomerPartyListResponseEBM is transformed into the Oracle E-Business Suite person account ABM and the Oracle E-Business Suite person account API is invoked.

Oracle E-Business Suite Integration Services

These are the integration services for Oracle E-Business Suite:

- SyncCustomerPartyListEbizProvABCSImpl
- QueryPersonAccountEbizUpdateAdapter
- QueryCustomerPartyListEbizCreateAdapter
- QueryCustomerPartyListEbizUpdateAdapter
- SyncPersonAccountEbizReqABCSImpl
- SyncPersonAccountEbizEventConsumer
- SyncCustomerPartyListEbizEventConsumer
- SyncCustomerPartyListResponseEbizProvABCSImpl
- SyncCustomerPartyListEbizAdapter
- SyncCustomerPartyListPersonEbizAdapter
- SyncCustomerPartyListEbizReqABCSImpl
- QueryPersonAccountEbizCreateAdapter

SyncCustomerPartyListEbizProvABCSImpl

The match and fetch flows use some of the same services as the synchronization flows. The SyncCustomerPartyListEbizProvABCSImpl service is implemented in BPEL and is invoked by the SyncCustomerPartyList operation of the CustomerPartyEBSV2, passing in the synchronize CustomerPartyListEBM payload. This service transforms the EBM into an Oracle E-Business Suite ABM and invokes the Oracle E-Business Suite APIs.

QueryPersonAccountEbizUpdateAdapter

The QueryPersonAccountEbizUpdateAdapter is implemented as an ESB routing service in which the Oracle E-Business Suite adapter for querying the Person Account is registered as an ESB service so that the apps adapter call from the ABCS Requester Implementation becomes a SOAP web service call.

QueryCustomerPartyListEbizCreateAdapter

The QueryCustomerPartyListEbizCreateAdapter is implemented as an ESB routing service in which the Oracle E-Business Suite adapter for querying the Person Account is registered as an ESB service so that the Apps adapter call from the ABCS becomes a SOAP web service call.

QueryCustomerPartyListEbizUpdateAdapter

The QueryCustomerPartyListEbizUpdateAdapter is implemented as an ESB routing service in which the Oracle E-Business Suite adapter for querying the Person Account is registered as an ESB service so that the Apps adapter call from the ABCS becomes a SOAP web service call.

SyncPersonAccountEbizReqABCSImpl

The SyncPersonAccountEbizReqABCSImpl service is implemented in BPEL and is invoked by the SyncPersonAccountEbizEventConsumer service, passing in the Person Account event payload. This service queries the entire Person Account Oracle E-Business Suite ABM message and then transforms it into an EBM and invokes the SyncCustomerPartyList operation of the CustomerPartyEBSV2.

SyncPersonAccountEbizEventConsumer

The SyncPersonAccountEbizEventConsumer is implemented as an ESB routing service that listens on the Person Account event (create/update of person account data including child entities) in Oracle E-Business Suite and dequeues the event message and routes it to the SyncPersonAccountEbizReqABCSImpl service.

SyncCustomerPartyListEbizEventConsumer

The SyncCustomerPartyListEbizEventConsumer service is responsible for dequeueing the Oracle create or update account business event payloads from AQ and invoking the appropriate operation of the requestor ABCS, SyncCustomerPartyEbizReqABCSImpl. The dequeue operation is done depending on the CorrelationID in AQ. For customer create, correlation ID is oracle.apps.ar.hz.OrgCustBO.create. For customer update, correlation ID is oracle.apps.ar.hz.OrgCustBO.update.

There are two business events with one operation each for reading each of the three types of Oracle Apps messages from the AQ:

- oracle.apps.ar.hz.OrgCustBO.create

- oracle.apps.ar.hz.OrgCustBO.update

SyncCustomerPartyListResponseEbizProvABCSImpl

The SyncCustomerPartyListResponseEbizProvABCSImpl service is implemented in BPEL and is invoked by the SyncCustomerPartyListResponse operation of the CustomerPartyResponseEBSV2, passing in the SyncCustomerPartyListResponseEBM payload. This service transforms the EBM into an Oracle E-Business Suite ABM and invokes the Oracle E-Business Suite API to synchronize the response.

SyncCustomerPartyListEbizAdapter

The SyncCustomerPartyListEbizAdapter service is an Oracle Applications Adapter service registered in ESB. This adapter service exposes the HZ_AIA_CUSTOM_PUB.sync_acct_order PL/SQL API. This wrapper API is delivered as part of EBS HZ.N.

This service is the interface through which an order is created in Oracle E-Business Suite, and is invoked by the SyncCustomerPartyListEbizProvABCSImpl as part of the Update Customer and Create Order integration flow.

The SyncCustomerPartyListEbizAdapter service exposes the synchronize customer party operation of the PL/SQL wrapper API. This operation is a synchronous request + reply operation. By registering this adapter service in ESB, ESB exposes a SOAP binding, which is what is used in this integration to invoke the service from the SyncCustomerPartyListEbizProvABCSImpl.

SyncCustomerPartyListPersonEbizAdapter

The SyncCustomerPartyListPersonEbizAdapter service is an Oracle Applications Adapter service registered in ESB. It invokes the TCA API, HZ_PERSON_CUST_BO_PUB.SAVE_PERSON_CUST_BO.

SyncCustomerPartyListEbizReqABCSImpl

The SyncCustomerPartyListEbizReqABCSImpl service is responsible for transforming the Oracle Apps, account, contact, or address created or updated event message into the appropriate SyncCustomerPartyListEBM format and invoking the SyncCustomerPartyList operation of the CustomerPartyEBSV2. There are two asynchronous operations for this service: create and update.

QueryPersonAccountEbizCreateAdapter

The QueryPersonAccountEbizCreateAdapter is implemented as an ESB routing service in which the Oracle E-Business Suite adapter for querying the Person Account is registered as an ESB service so that the apps adapter call from the ABCSReqImpl becomes a SOAP web service call.

Oracle E-Business Suite Interfaces

These are the Oracle E-Business Suite interfaces.

Inbound Web Services: Fetch

- HZ_AIA_CUSTOM_PKG. SYNC_ACCT_ORDER
- HZ_PERSON_CUST_BO_PUB.SAVE_PERSON_CUST_BO

Outbound from EBS/Composite App Event Interfaces: Match and Fetch

- Account Match
- Contact Match
- Account Fetch
- Contact Fetch

Inbound Web Services: Person or Organization Synchronization

API Name	Schema
HZ_ORG_CUST_BO_PUB.GET_ORG_CUSTS_CREATED	HZ_ORG_CUST_BO
HZ_ORG_CUST_BO_PUB.GET_ORG_CUSTS_UPDATED	HZ_ORG_CUST_BO
HZ_PERSON_CUST_BO_PUB.GET_PERSON_CUSTS_CREATED	HZ_PERSON_CUST_BO
HZ_PERSON_CUST_BO_PUB.GET_PERSON_CUSTS_UPDATED	HZ_PERSON_CUST_BO
HZ_AIA_CUSTOM_PKG. SYNC_ACCT_ORDER	HZ_ORG_CUST_BO
HZ_PERSON_CUST_BO_PUB.SAVE_PERSON_CUST_BO	HZ_PERSON_CUST_BO

Outbound from EBS Event Interfaces: Organization or Person Synchronization

Event Name	Schema	Description
oracle.apps.ar.hz.OrgCustBO.create	Event Name CDH_EVENT_ID	This event is raised when an organization account is created
oracle.apps.ar.hz.OrgCustBO.update	Event Name CDH_EVENT_ID	This event is raised when an organization account is updated
oracle.apps.ar.hz.PersonCustBO.create	Event Name	This event is raised when a person

Event Name	Schema	Description
	CDH_EVENT_ID	account is created
oracle.apps.ar.hz.PersonCustBO.update	Event Name CDH_EVENT_ID	This event is raised when a person account is updated

For more information about EBS web services, see the *E-Business Suite references: E-Business Suite Electronic Technical Reference Manual (eTRM)* located on Oracle 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)

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

This chapter provides an overview of Oracle Billing and Revenue Management (Oracle BRM) integration option and covers:

- Oracle BRM Interfaces
- Integration Services

Overview

There is no process flow from Oracle BRM integration option to OCH as part of OCH PIP, for OCH to Oracle BRM flows, see Chapter 2: Oracle Customer Master Data Management Integration Base Pack.

Oracle BRM Interfaces

The SyncCustomerPartyListBRMCommsProvABCSImpl is the only service that directly calls Oracle BRM. A new configuration property is added for this ABCS called *StatusUpdateRequired.* This property is used to indicate the status whether the Account in Siebel needs to be communicated to Oracle BRM or not. In addition, there are appropriate configuration properties added supporting extensibility.

Oracle BRM Integration Services

These are the integration services for Oracle BRM spoke.

- CommunicationsCustomerPartyEBSV2
- SyncCustomerPartyListBRMCommsProvABCSImpl
- CommunicationsCustomerPartyResponseEBSV2
- SyncCustomerPartyListBRMCommsJMSProducer
- SyncCustomerPartyListBRM_01CommsJMSConsumer

CommunicationsCustomerPartyEBSV2

CommunicationsCustomerPartyEBSV2 is used to route the CustomerPartyEnterpriseObjects for creating a new customer or updating the customer in Oracle BRM. This enterprise business service routes the messages from Siebel and OCH to Oracle BRM.

If the incoming message is from Siebel, the message is routed through the SyncCustomerPartyListBRMCommsJMSProducer to a topic; then to the SyncCustomerPartyListBRM_01CommsJMSConsumer; and finally to the SyncCustomerPartyListBRMProvABCSImpl, which synchronizes the data to Oracle BRM.

If the incoming message is from OCH, the message is stamped with the *actionCode* value as *Update* because it is an updated message and then sent to the SyncCustomerPartyListBRMProvABCSImpl, which in turn synchronizes the data to Oracle BRM.

If the incoming message is from Siebel Order then this service routes the message to SyncCustomerPartyListBRMProvABCSImpl, which in turn synchronizes the data to Oracle BRM.

SyncCustomerPartyListBRMCommsProvABCSImpl

The SyncCustomerPartyListBRMCommsProvABCSImpl service is invoked by CommsProcessBillingAccountListEBF or SyncAccountSiebelReqABCSImpl or SyncOrganizationUCMReqABCSImpl service when the routing rules determine that Oracle BRM is to be the service provider of the SyncCustomerPartyList operation of the CommunicationsCustomerPartyEBSV2.

This either occurs during the integration flow that is initiated by the account-synchronization-processing portion of the order flow or during the account update flow.

CommunicationsCustomerPartyResponseEBSV2

CommunicationsCustomerPartyResponseEBSV2 is used for routing the CustomerPartyResponse EnterpriseObjects to synchronize the status back to the caller.

SyncCustomerPartyListBRMCommsJMSProducer

This process enqueues the customer messages from the communication customer party EBS to a topic, and routes the message to the SyncCustomerPartyListBRMProvABCSImpl.

SyncCustomerPartyListBRM_01CommsJMSConsumer

This process consumes the topic messages. For every billing instance, one such consumer must be deployed. It checks with the cross-reference data to ensure that the customer belongs to the corresponding instance. If the data belongs to the corresponding Oracle BRM instance, this process stamps the appropriate target system identifiers in EBM and routes the message to the provider.

Chapter 6: Oracle Customer Master Data Management Integration Option for SAP

This chapter provides an overview of SAP flows and covers:

- Process Flows
- SAP Interfaces
- SAP Integration Services

Overview

These flows enable you to synchronize new or updated records from SAP to the target system-OCH. The SAP application invokes the synchronization process to create or update records in OCH. The synchronization represents a single service to perform either a create call or an update call depending on the existence of the customer in the source and target applications.

Prerequisites

- SAP system must be registered in OCH system registration to provide create and update privileges to SAP.
- OCH configuration determines the method of publishing as realtime or batch.
- SAP and OCH must be set up for the customer synchronization to work properly. SAP must be configured in such a way that if the customer is created or changed, an event is be triggered to send the customer details.

Process Flows

These are the process flows.

Synchronizing Organizations and Persons from SAP to OCH

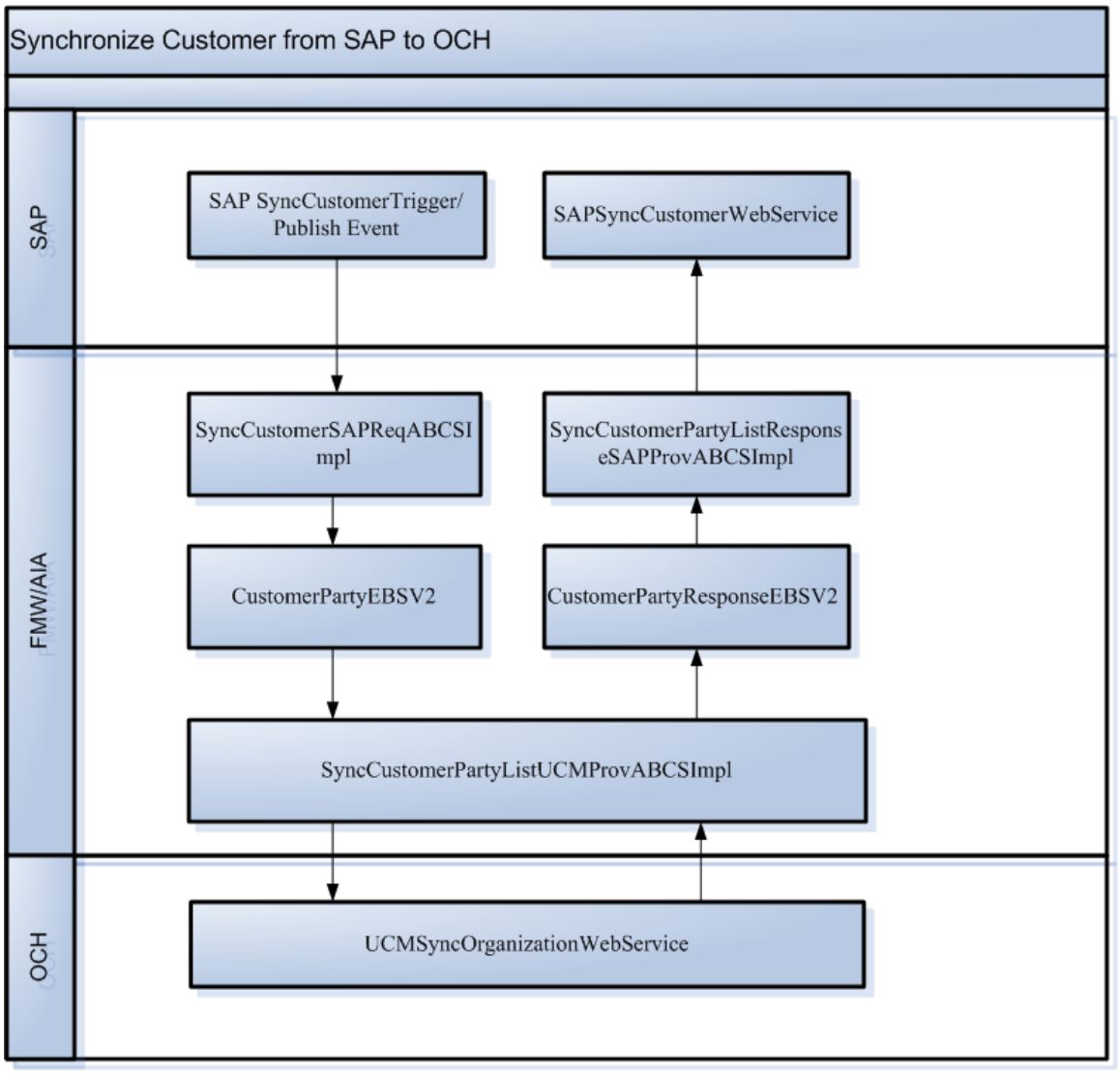
This integration flow synchronizes organization and person customer accounts from SAP to OCH.

SAP to OCH Create/Update:

OCH	SAP
Organization create/update	Mapped to 'Accounts' entity in OCH

OCH	SAP
(Mapped to Customer in SAP)	
Contact details for the Organization (Mapped to Customer -> Contact details in SAP)	Mapped to Account -> Contact in OCH
Person create /update(Mapped to Customer in SAP)	Mapped to 'Accounts' entity in OCH
Contact details for the Person (Mapped to Customer -> Contact details in SAP)	Mapped to Account -> Contact in OCH

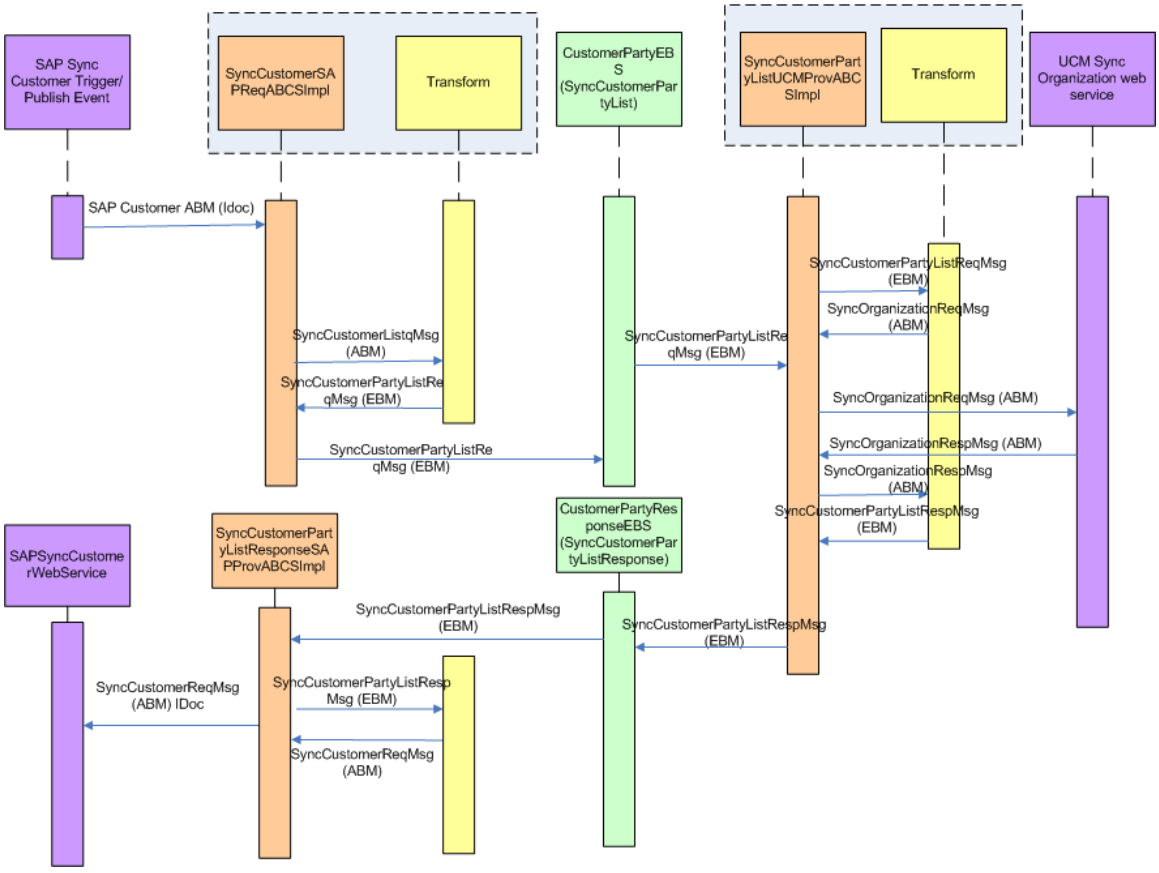
This graphic illustrates the synchronization of organizations and persons from SAP to UCM:



Synchronize customers from SAP to OCH

This integration flow synchronizes new accounts or updates from SAP to UCM. This integration flow uses these interfaces:

- SyncCustomerSAPReqABCSImpl
- CustomerPartyEBSV2
- SyncCustomerPartyListUCMProvABCSImpl
- CustomerPartyResponseEBSV2
- SyncCustomerPartyListResponseSAPProvABCSImpl



Synchronize customers from SAP to UCM

The SAP events are raised when a new customer is created or updated in SAP application. These events occur when the process is initiated:

1. Creation of a new customer or updating an existing customer initiates the SyncCustomerSAPReqABCSImpl with the SAPCustomerABM (IDoc) message.
2. The SAP Requester ABCS implementation, SyncCustomerSAPReqABCSImpl, transforms the SAPCustomerABM to SyncCustomerPartyListEBM and the ‘SyncCustomerPartyList’ operation of the CustomerPartyEBSV2 is invoked.
3. Invoking CustomerPartyEBSV2 with the ‘SyncCustomerPartyList’ operation routes the SyncCustomerPartyListEBM to the OCH Provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl.
4. The OCH provider ABC implementation service, SyncCustomerPartyListUCMProvABCSImpl, transforms the SyncCustomerPartyListEBM to the appropriate OCH Account ABM and invokes the OCH API.

5. The response from the OCH API is an integration object, which might have the account data if it was successfully created or updated in OCH, which is then transformed to the SyncCustomerPartyListResponseEBM in the process of which the cross references tables are populated with the OCH identifier values.

   The response is also sent to CustomerPartyResponseEBSV2 if the response comes back from OCH. If the OCH ABM message was put in a queue then the response IO status has to be checked and the cross-reference entry for the UCM ID needs to be created when the publish from UCM happens for this account using integration IDs.

6. The SyncCustomerPartyListResponseEBM is transformed to SAP Customer ABM and the SAP Customer web service is invoked.

SAP Interfaces

Inbound/outbound web services as applicable are provided here.

Inbound Web Services: Organization and Person Synchronization

Name	Schema/Element
DEBMAS04	DEBMAS04_invoke.wsdl

Outbound Web Services: Organization and Person Synchronization

Name	Schema
DEBMAS06	DEBMAS06_receive.wsdl

Inbound Web Services: Account Contact Synchronization

Name	Schema/Element
DEBMAS04	DEBMAS04_invoke.wsdl
BAPI_CUSTOMER_GETLIST	BAPI_CUSTOMER_GETLIST_invoke.wsdl
CUSTOMER_CONTACTPS_GET	CUSTOMER_CONTACTPS_GET_invoke.wsdl
HRCA_CUSTOMER_CONTACT_READ	HRCA_CUSTOMER_CONTACT_READ_invoke.wsdl

Outbound Web Services: Account Contact Synchronization

Name	Schema
DEBMAS06	DEBMAS06_receive.wsdl

SAP Integration Services

These are the services delivered with this integration:

- SyncCustomerSAPReqABCSImpl
- SyncCustomerPartyListSAPProvABCSImpl
- SyncCustomerPartyListResponseSAPProvABCSImpl
- SyncCustomerSAPReqABCSImpl

SyncCustomerSAPReqABCSImpl

The SyncCustomerSAPReqABCSImpl transforms the SAPCustomerABM to SyncCustomerPartyList EBM and the 'SyncCustomerPartyList' operation of the CustomerPartyEBSV2 is invoked.

SyncCustomerPartyListSAPProvABCSImpl

The SyncCustomerPartyListSAPProvABCSImpl service is responsible for transforming the SyncCustomerPartyListEBM to the appropriate SAP Customer ABM. It also helps in data enrichment of the SAP Customer ABM and invokes the SAP IDOC. After creating a customer, SAP application triggers back a response IDOC to SAP Requestor service that invokes CustomerPartyResponseEBSV2.

SyncCustomerPartyListResponseSAPProvABCSImpl

In the SyncCustomerPartyListResponseSAPProvABCSImpl, the SyncCustomerPartyListResponseEBM is transformed to SAP Customer ABM & the SAP SyncCustomer web service (DEBMAS04) is invoked.

SyncCustomerSAPReqABCSImpl

SyncCustomerSAPReqABCSImpl is triggered by SAP change pointer when a customer is created and customer data is sent back to populate cross references and then invokes CustomerPartyResponseEBSV2. This service is responsible for transforming response from SAP to SyncCustomerPartyListResponseEBM, populates cross references and routes the EBM message to CustomerPartyResponseEBS.

Chapter 7: Implementing the Process Integration Pack for Oracle Customer Hub

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

- Configuring Siebel CRM
- Setting Up Organizations
- Setting Up Cross References
- Identifying Cross-References
- Describing Domain Value Maps
- Creating Oracle E-Business Suite System Profiles
- Creating SAP System Profiles
- Scheduling Concurrent Processes
- Oracle AS Adapter
- Configuring Sequencing for the Synchronize Flows from Siebel CRM to Oracle Customer Hub
- Handling Errors
- Using the Composite Application Framework
- Viewing EBO Implementation Maps (EIMs)
- Setting Configuration Properties

Participating applications must be set up for the Oracle Customer Hub PIP to work properly. This includes setting up organizations, creating system profiles, setting up cross references, managing Domain Value Maps (DVMs) and setting configuration properties. These sections describe these setups in detail.

Configuring Siebel CRM

After you install and configure Siebel CRM and any required patches as described in the Oracle Application Integration Architecture Installation and Upgrade Guide for Prebuilt Integrations "Software Requirements", perform the following configuration steps within your Siebel CRM system.

To configure the Oracle Fusion Middleware (FMW) URLs for the outbound web services in the Siebel CRM application, follow these steps:

1. Go to Site Map.
2. Click **Administration - Web Services**.

3. Click **Outbound Web Service**. In the Name field of the web services applet, query for the following services -
    - SyncCustomerSiebelEventAggregator
    - MatchAccountSiebelReqABCSImpl
    - MatchContactSiebelReqABCSImpl
    - FetchAccountSiebelReqABCSImpl
    - FetchContactSiebelReqABCSImpl
4. Get the FMW URLs for these services from the EM console. Update the Address column of the Service Port applet to point to the correct FMW server and port number. For example, for SyncCustomerSiebelEventAggregator, the URL to be updated is - http://<SOA_HOST>:<SOA_PORT>/orabpel/default/SyncCustomerSiebelEventAggregator/1.0.
5. Enable the workflows that are listed in the following table:

Workflow Name	Status	Project
SWI Account Update Workflow	New	Web Service Integration
SWI Address Update Workflow	New	Web Service Integration
SWI Contact Update Workflow	New	Web Service Integration
SWI External Account Integration Process	New	Web Service Integration
SWI External Account Request Sync Process	New	Web Service Integration
SWI External Contact Integration Process	New	Web Service Integration
SWI External Contact Req Integration Process	New	Web Service Integration
SWI External Contact Request Sync Process	New	Web Service Integration
SWI External Customer Req Integration Process	New	Web Service Integration
SIA External Integration Process	New	Web Service Integration

For more information about how to enable workflows, see Siebel Business Process Framework: Workflow Guide, available on Oracle Technology Network.

6. Enable the component groups that are listed in the following table:

Component Group Name	Alias
Enterprise Application Integration	EAI

Siebel ISS	ISS
Workflow Management	Workflow

For more information about how to enable component groups, see Siebel System Administration Guide, available on Oracle Technology Network.

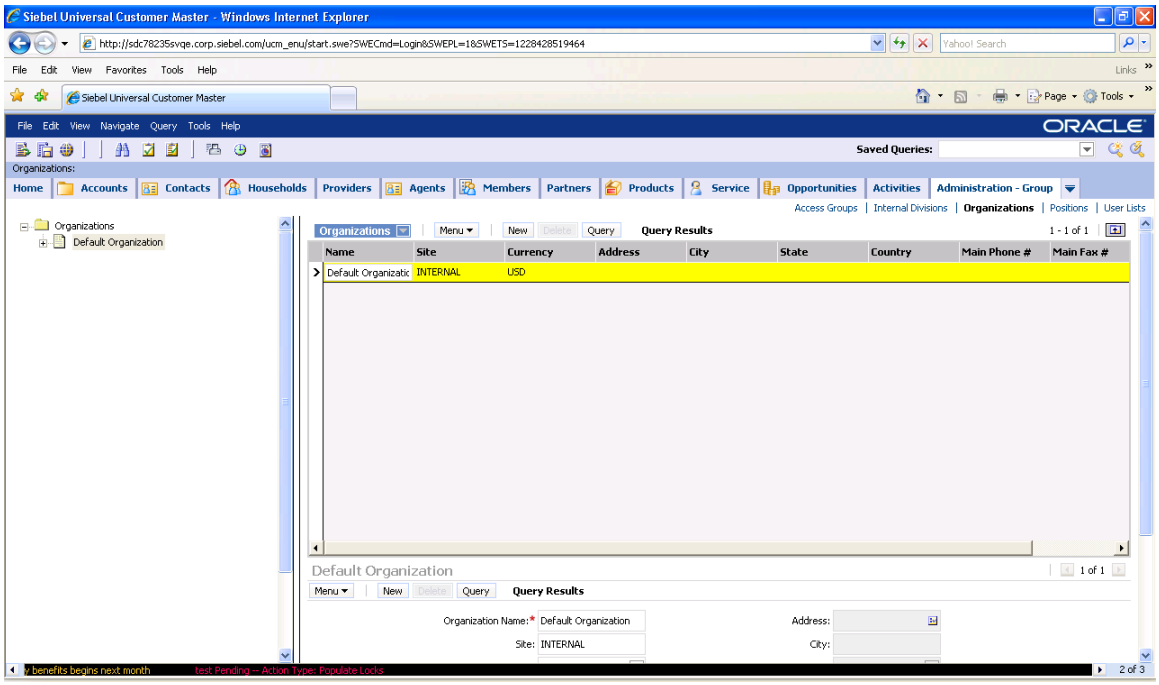
Setting Up Organizations

Set up Oracle Customer Hub.

Oracle Customer Hub

Users need to determine what organizations they want to support and then get the IDs for those organizations.

7. Log on to Oracle Customer Hub Applications.
8. Click Site Map.
9. Select Administration–Groups, Organizations.
10. Identify the default organization in Oracle Customer Hub or for the Oracle E-Business Suite Operating Unit that needs to be used as the default, create the same in Oracle Customer Hub. If multiple organizations are used, those also have to be set up. In addition, identify the organizations corresponding to every unique combination of the three-part SAP Sales Area key, and set it up



Administration – Group tab

Obtaining Oracle E-Business Suite Operating Unit IDs

Users need to determine what organizations they want to support and then get the IDs for those organizations.

To get the operating unit details:

1. Log on to Oracle E-Business Suite database.
2. Identify the operating units that need to be synchronized or maintained in Oracle E-Business Suite.

   If you want to pick other Operating Units, use this query:

   ```
   select organization_id, name from hr_operating_units
   ```

Obtaining SAP Sales Areas

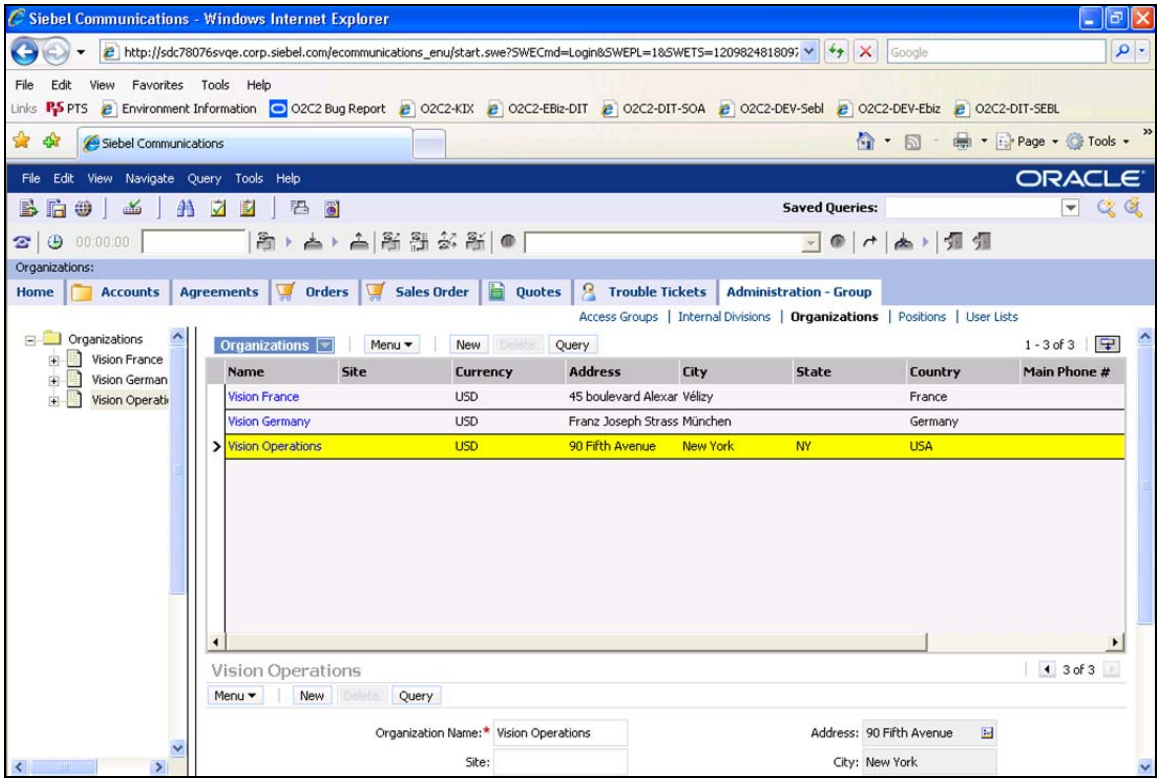
To get the sales area details:

1. Log on to SAP system.
2. Go to transaction SPRO and click SAP reference IMG.
3. Navigate to Enterprise Structure, Assignment, Sales and Distribution, Setup Sales Area.
4. Identify the Sales Areas that are created in the system.

Mapping Siebel CRM Organizations

To map Siebel organizations to EBS/Oracle Customer Hub operating units:

1. Log on to Siebel Applications.
2. Click Site Map.
3. Select Administration–Groups, Organizations.
4. For the Oracle E-Business Suite Operating Units that were identified previously, create the same in Siebel CRM. Here is a screen shot:



Administration – Group tab in Siebel

Setting Up Cross References

Cross-references can be created after organizations have been created in Siebel CRM, Oracle E-Business Suite, SAP, and Oracle Customer Hub.

Identifying Siebel Row IDs

To identify Siebel Row IDs:

1. Log on to the Siebel database as the table owner.
2. Run this query to get the IDs for the organizations created in the previous step:

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

Identifying Oracle E-Business Suite Entities

To identify Oracle E-Business Suite entities:

1. Login to Oracle E-Business Suite database (Apps/Apps).
2. Identify the Operating Units that need to be synchronized or maintained in Oracle E-Business Suite.

   If you want to pick other Operating Units, use this query:

```
select organization_id, name from hr_operating_units
```

Identifying SAP Entities

1. Log on to SAP.
2. Go to transaction xd03: Customer display initial screen.
3. Press F4 in the customer field to search for a customer account number based on a sales area (sales organization, distribution channel, and division)

Identifying Oracle Customer Hub Row IDs

To identify Oracle Customer Hub Row IDs:

1. Login to the Oracle Customer Hub database as the table owner.
2. Run this query to get the IDs for the Organizations you created:

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

Here is an alternative way to obtain the row ID:

1. Log on to Oracle Customer Hub.
2. Go to the site map.
3. Navigate to organization administration.
4. In the list box, select the appropriate organization.
5. From the Help menu, select About Record.

Populating Cross-References

To populate cross-references:

1. Create Organization.xml using the sample provided here. Update the XML file with the IDs from Siebel OCH, E-Business Suite, SAP, and Oracle BRM.

   The SAP entry is concatenated value of the 3-part SAP Sales Area key. For example: If the sales organization is 3110, distribution channel is 12 and division is 00, then the SAP entry for the ORGANIZATION_ID cross-reference table is like '3110|12|00'.

   Oracle IDs for this remain the same until a different organization is selected, created, or both.

   Sample **Organization.xml**:

```
<xref xmlns="http://xmlns.oracle.com/xref">
   <table name="ORGANIZATION_ID">
         <columns>
               <column name="EBIZ_01"/>
               <column name="SAP_01"/>
               <column name="SEBL_01"/>
               <column name="UCM_01"/>
               <column name="COMMON"/>

         </columns>
         <rows>
               <row>
                     <cell colName="EBIZ_01">204</cell>
                     <cell colName="SAP_01">3110|12|00</cell>
                     <cell colName="SEBL_01">88-25CHZ</cell>
                     <cell colName="COMMON_01">88-25CHZ</cell>
                     <cell colName="UCM_01">0-R9NH</cell>
               </row>
         </rows>
   </table>
</xref>
```

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

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

5. Set the environment variables as shown in this example:

```
-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"
```

Note: The values for OC4J_USERNAME, OC4J_PASSWORD, DB_USER, and DB_PASSWORD may change depending on the environment.

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

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

Note: The ORGANIZATION_ID cross-reference table must be pre-populated with at least one entry of the organization details for successful customer synchronization.

Validating Cross-References

To validate cross-references:

1. Log on to the AIA XREF database.
2. Query the Table XREF_DATA to confirm that every organization used in the XML files has three records. Use this query:

```
select value||'::'||Xref_column_name from xref_Data where row_number
in (select row_number from xref_data where xref_table_name =
'ORGANIZATION_ID' and value in ('204')).
```

3. Replace the value for the organizations. (The number of operating units depends on your setup.)

Identifying Cross-References

Cross-references map and connect the records within the application network, and enables these applications to communicate in the same language. The integration server stores the relationship in a persistent way so that others can refer to it.

For more information about cross-references, see Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite, "Working with Cross References."

These are the cross references for the Oracle Customer Hub PIP:

Cross-Reference Table Name	COLUMN NAME	DESCRIPTION
CUSTOMERPARTY_PARTYID	SEBL_01,COMMON, EBIZ_01,UCM_01, SAP_01	Customer Party IDs. For lookup and populate.
CUSTOMERPARTY_ACCOUNTID	SEBL_01,COMMON, EBIZ_01,UCM_01, BRM_01, SAP_01	Customer Party Account IDs. For lookup and populate.
CUSTOMERPARTY_BILLPROFILEID	SEBL_01,COMMON, BRM_01	Siebel bill profile ID is mapped one-to-one to the Oracle BRM bill info ID.
CUSTOMERPARTY_PAYPROFILEID	SEBL_01,COMMON, BRM_01	Siebel bill profile ID is mapped one-to-one to the Oracle BRM pay info ID.

Cross-Reference Table Name	COLUMN NAME	DESCRIPTION
CUSTOMERPARTY_CONTACTID	SEBL_01,COMMON, EBIZ_01,UCM_01, SAP_01	Customer Party Contact IDs. For lookup and populate.
CUSTOMERPARTY_PARTYLOCATIONID	SEBL_01,COMMON, EBIZ_01,UCM_01, SAP_01	Customer Party Location IDs. For lookup and populate.
CUSTOMERPARTY_LOCATIONREFID	SEBL_01,COMMON, EBIZ_01,UCM_01	Customer Party Location Reference Ids
CUSTOMERPARTY_ADDRESSID	SEBL_01,COMMON, EBIZ_01,UCM_01, BRM_01, SAP_01	Customer Party Address IDs. For lookup and populate. Oracle BRM account ID is cross-referenced here if the address is used as the billing address (nameinfo[1]) on that account. Oracle BRM pay info ID is cross-referenced here if the address is used as the pay info address on an account. The ACCOUNT and PAYINFO codes are prefixed to each ID to indicate what type of ID it is.
CUSTOMERPARTY_LOCATIONREFID	SAP_01,COMMON,UCM_01	Lookup and Populate
CUSTOMERPARTY_PARTYCONTACTID	SEBL_01,COMMON, EBIZ_01,UCM_01, BRM_01, SAP_01	Customer Party contact IDs. For lookup and populate. Oracle BRM account ID is cross-referenced here if the contact is used as the name (nameinfo[1]) on that account. Oracle BRM pay info ID is cross-referenced here if the contact is used as the name on the pay info on an account. The ACCOUNT and PAYINFO codes are prefixed to each ID to indicate what type of ID it is.
CUSTOMERPARTY_ACCOUNT_DEFAULTBALGRP	COMMON, BRM_01	Maps the default balance group to the common Account ID. This is populated after account creation in the CreateCustomerParty provider ABCS implementation service, and is referenced by the order flow during service creation.
ORGANIZATION_ID	SEBL_01,COMMON, EBIZ_01,UCM_01, SAP_01	Organization IDs, For lookup and populate.
CUSTOMERPARTY_ACCOUNT_PHONECOMMID	SEBL_01,COMMON, EBIZ_01,UCM_01,	Customer Party Account's Phone contact points. For lookup and populate.

Cross-Reference Table Name	COLUMN NAME	DESCRIPTION
	SAP_01	
CUSTOMERPARTY_ACCOUNT_FAXCOMMID	SEBL_01,COMMON, EBIZ_01,UCM_01, SAP_01	Customer Party Account's Fax contact points. For lookup and populate.
CUSTOMERPARTY_CONTACT_PHONECOMMID	SEBL_01,COMMON, EBIZ_01,UCM_01	Customer Party Contact's Phone contact points
CUSTOMERPARTY_CONTACT_FAXCOMMID	SEBL_01,COMMON, EBIZ_01,UCM_01	Customer Party Contact's Fax contact points
CUSTOMERPARTY_CONTACT_EMAILCOMMID	SEBL_01,COMMON, EBIZ_01,UCM_01	Customer Party Contact's Email/Web contact points. For lookup and populate.
CUSTOMERPARTY_ACCOUNT_WEBCOMMID	SEBL_01,COMMON, EBIZ_01,UCM_01	Customer Party Account's Web URL contact points. For lookup and populate.

Describing Domain Value Maps

Domain value maps (DVMs) are a standard feature of the Oracle SOA Suite and enable you to equate lookup codes and other static values across applications, for example, FOOT and FT; or US and USA.

DVMs are static in nature, though administrators can add additional maps as needed. Transactional business processes never update DVMs—they only read from them. They are stored in XML files and cached in memory at runtime.

DVM types are seeded for the Oracle Customer Hub PIP flows, and administrators can extend the list of mapped values by adding more maps. The DVM data is synchronized with what the participating applications use. This synchronization should be done prior to running any initial loads, or initiating any incremental transactional flows.

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

DVM Type	DVM Column Name	Comments
CUSTOMERPARTY_TYPECODE	Common, SEBL_01	Consumer vs. Business
CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE	Common, Siebel, BRM_01	Billing frequency (monthly, yearly, quarterly, and so on)
CUSTOMERPARTY_BILLPROFILE_BILLTYPECODE	Common, Siebel, BRM_01	Bill type (summary and detailed).
CUSTOMERPARTY_PAYPROFILE_PAYMETHODCODE	Common, Siebel, BRM_01	Payment-profile payment method types: credit card, direct debit, and invoice/bill me.
CUSTOMERPARTY_PAYPROFILE_PAYTERMCODE	Common, BRM_01	Payment term codes
CUSTOMERPARTY_PAYPROFILE_DELIVERYPREF	Common, BRM_01	Bill media/delivery preference (Email and Mail).
CUSTOMERPARTY_PAYPROFILE_CREDIT_CARDTYPE	Common, Siebel	Credit Card type (Visa, MasterCard, and so on.)

DVM Type	DVM Column Name	Comments
CUSTOMERPARTY_PAYPROFILE _BANKACCOUNTTYPE	Common, Siebel, BRM_01	Bank account type (checking, savings, and so on)
CUSTOMERPARTY_ACCOUNTTY PECODE	SEBL_01, COMMON, UCM_01	Account Type Codes
ADDRESS_COUNTRYID	SEBL_01, COMMON, EBIZ_01, UCM_01, SAP_01	Country Codes
STATE	SEBL_01, COMMON, EBIZ_01, UCM_01, SAP_01	State Codes
PROVINCE	SEBL_01, COMMON, EBIZ_01, UCM_01	Province
CURRENCY_CODE	SEBL_01, COMMON, EBIZ_01, UCM_01, SAP_01	Currency Codes
CUSTOMERPARTY_STATUSCODE	SEBL_01,COMMON,EBIZ_01,U CM_01, SAP_01	CustomerParty Status Code as Active, Inactive, and so on.
JOB_TITLE	Common, SAP_01, UCM_01	Manager, Executive Board, and so forth
CONTACT_SALUTATION	SEBL_01, COMMON, EBIZ_01, UCM_01, SAP_01	Salutation as head of sales and so on.
TAX_CATEGORY	Common, SAP_01, UCM_01	MWST, CTX2, and so forth
CONTACT_GENDERCODE	SEBL_01, COMMON, EBIZ_01, UCM_01	Gender Code
MARITAL_STATUS	UCM_01, COMMON, EBIZ_01	Marital Status
SOURCE_ID	UCM_01, COMMON, SAP_01	Contains System IDs. To map source Application ID (mandatory).
PHONENUMBER_PURPOSE	SEBL_01, COMMON, EBIZ_01,	Phone Number Purpose
PHONENUMBER_TYPE	SEBL_01,COMMON,EBIZ_01,	Phone Number Type
SITEUSAGE_CODE	SEBL_01, BRM_01, COMMON, EBIZ_01	Site Usage Code
TARGET_ID	UCM_01,COMMON, SAP_01	Contains System IDs. To map source Application ID (mandatory).
ADDRESS_DELIVERYTYPE	UCM_01, COMMON, SEBL_01	Residential Curb, Business, and so on. Used for third-party enrichment.
ADDRESS_SEASONALINDICATOR TYPE	UCM_01, COMMON, SEBL_01	Educational Facility, Seasonal Address, and so on. Used for third-party enrichment
ADDRESS_MOVETYPE	UCM_01, COMMON, SEBL_01	Individual Move, Business or Firm Move, and so on. Used for third-party enrichment
ADDRESS_POSTALPROCESSING	UCM_01, COMMON, SEBL_01	AA match at ZIP Code level, LOT default

DVM Type	DVM Column Name	Comments
CODE		match, and so on. Used for third-party enrichment

Creating Oracle E-Business Suite System Profiles

For the Oracle Customer Hub PIP, set these profile options:

1. Log on to Oracle E-Business Suite using the System Administrator responsibility.
2. Open the System Profile Values form.
3. Query these profile options E-Business Suite 11.5.10.2 and set the indicated values at the site level:
   - HZ: Execute API Callouts to ‘All Events Enabled’ or ‘Only Business Object Events Enabled’
   - HZ: Format Business Object Business Events as Bulk to ‘N’
   - HZ: Generate Party Number to 'Yes'
   - HZ: Generate Party Site Number to 'Yes'
4. Query these profile options E-Business Suite 12.1.1. and set the indicated values at the site level:
   - HZ: Raise API Events to All Events Enabled or Only Business Object Events Enabled
   - HZ: Format Business Object Business Events as Bulk to ‘N’
   - HZ: Generate Party Number to 'Yes'
   - HZ: Generate Party Site Number to 'Yes'

Creating SAP System Profiles

Set these profile options:

1. Log on to SAP
2. Enter Transaction code -XD01 in 'Command field' of SAP, Press 'ENTER'.
3. In Customer create: initial screen, Select the ‘Account group as -0001-Sold-to-party ‘Enter an existing ’ Sales area’ (Sales org, Distribution channel & Division), Press 'ENTER"
4. In 'General data' screen→ GOTO 'Address' Tab-Enter values (mapped fields). In 'Contact person" Tab →Enter" NAME"→Double click on "Name"-> Input values (mapped fields)→Choose 'Sales area data" menu (TOP)

5. In 'Sales area data" screen → GoTo → Sales, shipping, billing documents & partner functions' TABS-Enter the mandatory values & Press "SAVE" button (TOP).
6. 'Customer account number' is created for a sales area.

Scheduling Concurrent Processes

For the Oracle Customer Hub PIP, schedule these concurrent processes:

1. For TCA, a concurrent program must be executed to raise business events after data creation. Schedule the concurrent request 'TCA Business Object Events: Raise Events' to run periodically. This request can also be run manually.
2. Schedule the concurrent request 'TCA Business Object Events: Cleanse Infrastructure Program - clean up' to run once a day to purge the tracking table.

Oracle AS Adapter

The Oracle AS Adapter for SAP provides a means to exchange real time business data between SAP systems and other application, database, or external business partner systems. The adapter enables external applications for inbound and outbound processing with SAP. The adapter uses XML messages to enable non-SAP applications to communicate and exchange transactions with SAP.

The Oracle AS Adapter for SAP provides:

- Support for bidirectional message interactions.
- The Application Explorer, a GUI tool, which uses SAP object repository metadata to build XML schemas and Web services to handle adapter requests or event data.
- Support for Remote Function Calls (RFC), Business Application Programming Interfaces (BAPI), and Intermediate Documents (IDoc) interfaces to SAP.

Enabling SAP Events

These ports are to be created in Oracle AS Adapter

Port	Type	URL
DEBMAS04PortType	HTTP	http://SOAHOST:SOAPORT/orainfra/wsil/adapters/applications/DEBMAS04_invoke.wsdl
BAPI_CUSTOMER_GETLISTPortType	HTTP	http://SOAHOST:SOAPORT/orainfra/wsil/adapters/applications/BAPI_CUSTOMER_GETLIST _invoke.wsdl
CUSTOMER_CONTACTPS_GETPortType	HTTP	http://SOAHOST:SOAPORT/orainfra/wsil/adapters/applications/CUSTOMER_CONTACTPS_GET_invoke.wsdl

HRCA_CUSTOMER_CONTACT_READPortType	HTTP	http://SOAHOST:SOAPORT/orainfra/wsil/adapters/applications/HRCA_CUSTOMER_CONTACT_READ_invoke.wsdl

Configuring Sequencing for the Synchronize Flows from Siebel CRM to Oracle Customer Hub

These steps must be done manually to enable the re-sequencer functionality in ESB and for better performance. The FMW 10.1.3.4 sequencing feature ensures better scalability and performance. Not defining these does not break anything in the code, but the synchronize integration flows from Siebel CRM to Oracle Customer Hub only works as single threaded flow.

To configure sequencing:

1. Set the ESB configuration parameter *EnableResequencer* to true in the <SOA_HOME>/integration/esb/config/esb_config.ini file. This configuration makes it possible for the routing services to act as a re-sequencer.

```
# Cluster name
cluster_name=ESB

# Central OC4J or not
primary_oc4j=true

# JTA
xa_timeout=60

# JMS receive timeout (in seconds)
jms_receive_timeout=30

#BPEL system property
esbSystemForBPEL=BPELSystem

#ESBSequencer
EnableResequencer = true

#Retry
InboundRetryCount = 3
InboundRetryInterval = 5
InboundRetryEnabled = true

OutboundRetryCount = 3
OutboundRetryInterval = 5
OutboundRetryEnabled = true
```

Snapshot of the esb_config.ini file

2. Configure these properties in the same esb_config.ini file for better performance of the sequencing feature:

   ResequencerWorkerThreadPoolSize = 5

   ResequencerLockerThreadSleep = 1

   ResequencerMaxRowsRetrieved = 10

3. Restart the SOA server for the changes to take effect.

Note: The value of these properties can vary based on the environment configuration and the same can be set appropriately.

For more information about using the Oracle ESB Resequencer as a part of an Oracle AIA implementation, see *Oracle Application Integration Architecture – Foundation Pack: Integration Developer's Guide*, "Using the Oracle Enterprise Service Bus Resequencer."

Handling Errors

Based on the roles defined for the services, email notifications are sent if a service errors out. There are no AIA specific errors thrown by the Oracle Customer Hub PIP services.

For more information about the errors thrown by Siebel CRM, Oracle E-Business Suite, or Oracle Customer Hub, see that product's 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."

Using the Composite Application Framework

The Match and Fetch integration flow between Oracle E-Business Suite and Oracle Customer Hub leverages the Composite Application Framework to create a user interface (UI) that enables users to access Oracle Customer Hub to search for and to fetch the customer record from within Oracle E-Business Suite.

The Composite Application Framework supports the development of application independent user-interfaces based on enterprise business services. This loosely coupled user interface is reusable and enables changes to the underlying application architecture with little to no impact on the end user.

For more information about the delivered Composite Application Framework UI, see the *My Oracle Support note 746109.1: Deploying the Universal Customer Master Match/Fetch Composite User Interface*.

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

Settings for System properties:

System Property	Value/Default Value	Description
Routing.ActiveRuleset	DEFAULT	Property that governs how routing rules are handled when PIPs are interoperable. Leave the value as DEFAULT if only one PIP exists.

Settings for Module level properties:

Module Property	Value/Default Value	Description
ModuleName – 'EBIZ' EBIZ_01.SERVER_TIMEZONE	GMT-08:00	Property that governs the timezone setting.

Settings for the FetchAccountSiebelReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence
Routing.CustomerPartyOrchestrationEBSV2.FetchCustomerPary.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyOrchestrationEBSV2.FetchCustomerPary.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformABMtoEBMABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA

Property Name	Value/Default Value	Description
		ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeEBSEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreXformEBMtoABMEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreReplyABMCustomerPartyIOABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreReplyABMMatchAccountABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
FetchCustomerParty.InvokeAsync	True/false. Default = false.	If this it set to false the flow gets invoked in Synchronous process otherwise in Asynchronous process
FetchCustomerParty.AsyncTimeoutDuration	PT1M	Used to set the time out duration

Settings for the FetchContactSiebelReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence
Routing.CustomerPartyOrchestrationEBSV2.FetchCustomerParty.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyOrchestrationEBSV2.FetchCustomerParty.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformABMtoEBMABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Property Name	Value/Default Value	Description
ABCSExtension.PreInvokeEBSEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreXformEBMtoABMEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreReplyABMCustomerPartyIOABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreReplyABMMatchContactABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
FetchCustomerParty.InvokeAsync	True/false. Default = false.	If this it set to false the flow gets invoked in Synchronous process otherwise in Asynchronous process
FetchCustomerParty.AsyncTimeoutDuration	PT1M	Used to set the time out duration

Settings for the MatchAccountSiebelReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.CustomerPartyEBSV2.QueryCustomerPartyList.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.QueryCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformABMtoEBMABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeEBSE	True/false. Default = false.	Property that governs whether ABCS

Property Name	Value/Default Value	Description
BM		Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreXformEBMtoABMEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreReplyABMABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the MatchContactSiebelReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.CustomerPartyEBSV2.QueryCustomerPartyList.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.QueryCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformABMtoEBMABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeEBSEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreXformEBMtoABMEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreReplyABMABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the

Property Name	Value/Default Value	Description
		Extension process (defined along AIA ABCS Extension guidelines) is invoked.
MDM.MaximumMatchCandidates	0	Maximum match candidates to be returned in the response should be zero or greater than zero

Settings for the FetchCustomerPartyEBFservice property:

Property Name	Value/Default Value	Description
Routing.CustomerPartyEBSV2.QueryCustomerParty.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.QueryCustomerParty.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}:7832/AIAValidationSystemServlet/asyncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.CustomerPartyOrchestrationEBSV2.SyncCustomerPartyList.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyOrchestrationEBSV2.SyncCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.CustomerPartyOrchestrationResponseEBSV2.FetchCustomerPartyResponse.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyOrchestrationResponseEBSV2.FetchCustomerPartyResponse.CAVS.EndpointURI	${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponserecipient	CAVS Endpoint URI, when CAVS is enabled.
SyncCustomerPartyList.AsyncTimeoutDuration	PT1M	Used to set the time out duration

Settings for the QueryCustomerPartyListUCMProvABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	UCM_01	OCH system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request

Property Name	Value/Default Value	Description
		message, then that takes precedence.
Routing.OrganizationMatchService.UCM_01.EndpointURI	http://${ucm.http.host}:${ucm.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&UserName=${ucm.eai.user}&Password=${ucm.eai.password}	OCH Organization Match web service endpoint location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence.
Routing.OrganizationMatchService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.OrganizationMatchService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.PersonMatchService.UCM_01.EndpointURI	http://${ucm.http.host}:${ucm.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&UserName=${ucm.eai.user}&Password=${ucm.eai.password}	OCH Person Match web service endpoint location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence.
Routing.PersonMatchService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.PersonMatchService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformEBMtoABMEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeOrganizationMatchABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokePersoMatchABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostInvokeOrganizationMatchABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostInvokePers	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined

Property Name	Value/Default Value	Description
onMatchABM		plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostXformABMtoEBMEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the QueryCustomerPartyUCMProvABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	UCM_01	OCH system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.OrganizationService.UCM_01.EndpointURI	http://${ucm.http.host}:${ucm.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&UserName=${ucm.eai.user}&Password=${ucm.eai.password}	OCH getOrganization web service endpoint location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.OrganizationService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.OrganizationService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.PersonService.UCM_01.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.PersonService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.PersonService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformEBMtoABMEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeGetOrganizationABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Property Name	Value/Default Value	Description
ABCSExtension.PostInvokeGetOrganizationABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeGetPersonABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostInvokeGetPersonABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostXformABMtoEBMEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the SyncCustomerPartyListResponseEbizProvABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	EBIZ_01	EBIZ system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.SyncCustomerPartyListEbizAdapter.EBIZ_01.EndpointURI	http://${http.hostname}:${http.port}/event/AIASystem/Ebiz/ABCS/SyncCustomerPartyListEbizAdapter	Endpoint URL of the Oracle E-Business Suite adapter
Routing.SyncCustomerPartyListEbizAdapter.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.SyncCustomerPartyListEbizAdapter.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.SyncCustomerPartyListPersonEbizAdapter.EBIZ_01.EndpointURI	http://${http.hostname}:${http.port}/event/AIASystem/Ebiz/ABCS/SyncCustomerPartyListPersonEbizAdapter	Endpoint URL of the Oracle E-Business Suite adapter
Routing.SyncCustomerPartyListPersonEbizAdapter.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.SyncCustomerPartyListPersonEbizAdapter.RouteToCAVS	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the

Property Name	Value/Default Value	Description
		Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreProcessOrgEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostProcessOrgEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreProcessOrgABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostProcessOrgABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreProcessPersonEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostProcessPersonEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreProcessPersonABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostProcessPersonABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the SyncCustomerPartyListResponseSiebelProvABCSImpl service property:

Property Name	Value/Default Value	Description

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.SWI_spcCustomer_spcParty_spcService.SEBL_01.EndpointURI	http://${siebel.http.host}:${siebel.http.port}/eai_enu/start.swe?SWEExtSource=WebService&SWEExtCmd=Execute&UserName=${siebel.eai.user}&Password=${siebel.eai.password}	Siebel web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.SWI_spcCustomer_spcParty_spcService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.SWI_spcCustomer_spcParty_spcService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.SWI_spcContact_spcService.SEBL_01.EndpointURI	http://${siebel.http.host}:${siebel.http.port}/eai_enu/start.swe?SWEExtSource=WebService&SWEExtCmd=Execute&UserName=${siebel.eai.user}&Password=${siebel.eai.password}	Siebel web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.SWI_spcContact_spcService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.SWI_spcContact_spcService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformEBMtoABMEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeSWICustServiceABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeSWIContactServiceABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the SyncOrganizationUCMReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	UCM_01	OCH system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
ABCSExtension.PreProcessABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreProcessEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
Routing.TransformAppContextUCMService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.TransformAppContextUCMService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.

Settings for the SyncPersonUCMReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	UCM_01	OCH system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
ABCSExtension.PreProcessABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Property Name	Value/Default Value	Description
ABCSExtension.PreProcessEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.TransformAppContextUCMService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.TransformAppContextUCMService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.

Settings for the SyncCustomerPartyListResponseUCMProvABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	UCM_01	OCH system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.UCMOrganizationCrossReferenceService.UCM_01.EndpointURI	http://${ucm.http.host}:${ucm.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&UserName=${ucm.eai.user}&Password=${ucm.eai.password}	OCH web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.UCMPersonCrossReferenceService.UCM_01.EndpointURI	http://${ucm.http.host}:${ucm.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&UserName=${ucm.eai.user}&Password=${ucm.eai.password}	OCH web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.UCMOrganizationCrossReferenceService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.UCMOrganizationCross	http://${http.hostname}:${http.port}/A	CAVS Endpoint URI, when CAVS is

Property Name	Value/Default Value	Description
ReferenceService.CAVS.EndpointURI	IAValidationSystemServlet/syncresponsesimulator	enabled.
Routing.UCMPersonCrossReferenceService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.UCMPersonCrossReferenceService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreProcessEBM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreProcessOrgABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostProcessPersonABM	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the SyncCustomerPartyListUCMProvABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	UCM_01	OCH system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.SyncCustomerPartyListUCMOrganizationService.UCM_01.EndpointURI	http://${ucm.http.host}:${ucm.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&UserName=${ucm.eai.user}&Password=${ucm.eai.password}	OCH web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.SyncCustomerPartyListUCMOrganizationService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.SyncCustomerPartyListUCMOrganizationService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.UCMPersonService.UCM_01.EndpointURI	http://${ucm.http.host}:${ucm.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExt	OCH web service end point location. This is a SOAP endpoint URL. If the request message contains the target

Property Name	Value/Default Value	Description
	Cmd=Execute&UserName=${ucm.eai.user}&Password=${ucm.eai.password}	URL, then that takes precedence
Routing.UCMPersonService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.UCMPersonService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.CustomerPartyEBSV2.SyncCustomerPartyListResponse.MessageProcessingInstruction.EnvironmentCode	Production	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.CustomerPartyEBSV2.SyncCustomerPartyListResponse.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.SyncCustomerPartyListResponse.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyListResponse.MessageProcessingInstruction.EnvironmentCode	Production	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
ABCSExtension.PreProcessEBMForOrg	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreProcessABMForOrg	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostProcessABMForOrg	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostProcessEBMForOrg	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Property Name	Value/Default Value	Description
ABCSExtension.PreProcessPersonABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostProcessPersonABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostProcessPersonEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the SyncPersonAccountEbizReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	EBIZ_01	Oracle E-Business Suite instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.QueryPersonAccountEbizCreateAdapter.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.QueryPersonAccountEbizUpdateAdapter.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
Routing.QueryPersonAccountEbizUpdateAdapter.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.QueryPersonAccountEbizCreateAdapter.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.MessageProcessingInstruction.EnvironmentCode	Production	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.

Property Name	Value/Default Value	Description
Routing.TransformAppContextEbizService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.TransformAppContextEbizService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformCreateABMtoEBMABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreXformUpdateABMtoEBMABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeCustEBSEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
Routing.QueryPersonAccountEbizCreateAdapter.EBIZ_01.EndpointURI	http://${http.hostname}:${http.port}/event/AIASystem/Ebiz/ABCS/QueryPersonAccountEbizCreateAdapter	Endpoint URL of the Oracle E-Business Suite adapter
Routing.QueryPersonAccountEbizUpdateAdapter.EBIZ_01.EndpointURI	http://${http.hostname}:${http.port}/event/AIASystem/Ebiz/ABCS/QueryPersonAccountEbizUpdateAdapter	Endpoint URL of the Oracle E-Business Suite adapter

Settings for the SyncAccountSiebelReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.SWICustomerParty.SEBL_01.EndpointURI	http://${siebel.http.host}:${siebel.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&UserName=${siebel.eai.user}&Password=${siebel.eai.password}	Siebel web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.SWICustomerParty.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.SWICustomerParty.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.

Property Name	Value/Default Value	Description
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.MessageProcessingInstruction.EnvironmentCode	Production	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
Routing.TransformAppContextSiebelService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.TransformAppContextSiebelService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformABMtoEBMABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeEBSEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
Account.ProcessUpdateEventsOnly	True/false. Default = false.	Set to true to process the update events

Settings for the SyncContactSiebelReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.SWIContactIO.SEBL_01.EndpointURI	http://${siebel.http.host}:${siebel.http.port}/eai_enu/start.swe?SWEExtSource=SecureWebService&SWEExtCmd=Execute&UserName=${siebel.eai.user}&Password=${siebel.eai.password}	Siebel web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.SWIContactIO.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.SWIContactIO.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.

Property Name	Value/Default Value	Description
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.MessageProcessingInstruction.EnvironmentCode	Production	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
Routing.TransformAppContextSiebelService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.TransformAppContextSiebelService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformABMtoEBMABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeEBSEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
Contact.ProcessUpdateEventsOnly	True/false. Default = false.	Property that governs whether ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the SyncCustomerPartyListEbizProvABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	EBIZ_01	Oracle E-Business Suite system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.SyncCustomerPartyListEbizAdapter.EBIZ_01.EndpointURI	http://${http.hostname}:${http.port}/event/AIASystem/Ebiz/ABCS/SyncCustomerPartyListEbizAdapter	Endpoint URL of the Oracle E-Business Suite adapter
Routing.SyncCustomerPartyListEbizAdapter.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.SyncCustomerPartyListEbizAdapter.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system

Property Name	Value/Default Value	Description
Routing.SyncCustomerPartyListPersonEbizAdapter.EBIZ_01.EndpointURI	http://${http.hostname}:${http.port}/event/AIASystem/Ebiz/ABCS/SyncCustomerPartyListPersonEbizAdapter	Endpoint URL of the Oracle E-Business Suite adapter
Routing.SyncCustomerPartyListPersonEbizAdapter.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.SyncCustomerPartyListPersonEbizAdapter.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyListResponse.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyListResponse.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncresponserecipient	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformEBMtoABMEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeEbizAdapterABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostInvokeEbizAdapterABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostXformABMtoEBMEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreXformEBMtoPersonABMEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Property Name	Value/Default Value	Description
ABCSExtension.PreInvokePersonEbizAdapterABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostInvokePersonEbizAdapterABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostXformPersonABMtoEBMEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the SyncCustomerPartyListSiebelProvABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	Siebel system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.SyncCustomerPartyListSiebelService.SEBL_01.EndpointURI	http://${siebel.http.host}:${siebel.http.port}/eai_enu/start.swe?SWEExtSource=WebService&SWEExtCmd=Execute&UserName=${siebel.eai.user}&Password=${siebel.eai.password}	Siebel web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.SyncCustomerPartyListSiebelService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.SyncCustomerPartyListSiebelService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.MergeCustomerPartyListSiebelService.SEBL_01.EndpointURI	http://${siebel.http.host}:${siebel.http.port}/eai_enu/start.swe?SWEExtSource=WebService&SWEExtCmd=Execute&UserName=${siebel.eai.user}&Password=${siebel.eai.password}	Siebel web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.MergeCustomerPartyListSiebelService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.MergeCustomerPartyListSiebelService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.

Property Name	Value/Default Value	Description
Routing.SWI_spcContact_spcService.SEBL_01.EndpointURI	http://${siebel.http.host}:${siebel.http.port}/eai_enu/start.swe?SWEExtSource=WebService&SWEExtCmd=Execute&UserName=${siebel.eai.user}&Password=${siebel.eai.password}	Siebel web service end point location. This is a SOAP endpoint URL. If the request message contains the target URL, then that takes precedence
Routing.SWI_spcContact_spcService.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.SWI_spcContact_spcService.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.
Routing.CustomerPartyEBSV2.SyncCustomerPartyListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.CustomerPartyEBSV2.SyncCustomerPartyListResponse.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.SyncCustomerPartyListResponse.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
ABCSExtension.PreInvokeSWICustSyncServiceABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeSWICustUpsertServiceABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreXformEBMtoABMEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostInvokeSWICustSyncServiceABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the

Property Name	Value/Default Value	Description
		Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostInvokeSWICustUpsertServiceABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeSWICustMergeServiceABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostInvokeSWICustMergeServiceABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostXformABMtoEBMEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeSWIContactUpsertServiceABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PostInvokeSWIContactUpsertServiceABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for the SyncCustomerPartyListEbizReqABCSImpl service property:

Property Name	Value/Default Value	Description
Default.SystemID	EBIZ_01	Oracle E-Business Suite system instance code (defined in BSR) from which messages originate. If the instance ID is present in the request message, then that takes precedence.
Routing.QueryCustomerPartyListEbizCreate.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
Routing.QueryCustomerPartyLis	True/false. Default = false.	Controls whether to route messages to

Property Name	Value/Default Value	Description
tEbizCreate.RouteToCAVS		the verification system
Routing.QueryCustomerPartyListEbizUpdate.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
Routing.QueryCustomerPartyListEbizUpdate.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.RouteToCAVS	True/false. Default = false.	Controls whether to route messages to the verification system
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.CustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/asyncrequestrecipient	CAVS Endpoint URI, when CAVS is enabled.
ABCSExtension.PreXformCreateABMtoEBMABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreXformUpdateABMtoEBMABM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.
ABCSExtension.PreInvokeCustEBSEBM	True/false. Default = false.	Property that governs whether an ABCS Extension is enabled at the pre-defined plug-into point. If set to true, then the Extension process (defined along AIA ABCS Extension guidelines) is invoked.

Settings for SyncCustomerSAPReqABCSImpl Service property

Property Name	Value/Default Value	Description	Example
Default.SystemID	SAP_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 picks the default SystemID from this config property.	<Property name="Default.SystemID">SAP_01</Property>
Routing. CustomerPartyEBSV2.SyncCustomerPartyList.RouteToCAVS	True/false. Default = False	Controls whether SyncCustomerSAPReqABCSImpl should route messages to the CAVS or to the OCH system.	<Property name="Routing. CustomerPartyEBSV2. SyncCustomerPartyList. RouteToCAVS">false</Property>
Routing. CustomerPartyEBSV2. SyncCustomerPartyList. CAVS.EndpointURI	No Default value	CAVS Endpoint URL	<Property name=" Routing. CustomerPartyEBSV2.SyncCustomerPartyList.CAVS.EndpointURI">[CAVSEndpointURL]</Property>

Settings for SyncCustomerPartyListSAPProvABCSImpl Service property

Property Name	Property Value	Description	Example
Default.SystemID	SAP_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 picks the default SystemID from this config property.	<Property name="Default.SystemID">SAP_01</Property>
Routing. DEBMAS04PortType.SAP_01.EndpointURL	No default value	URL for SAP Instance Web service for “SAPSyncCustomerWS” web service	<Property name=" Routing. DEBMAS04PortType.SAP_01.EndpointURL ">[AppEndpointURL]</Property>
Routing.DEBMAS04PortType.RouteToCAVS	True/false. Default = False	Controls whether SyncCustomerSAPListProvABCSImpl should route messages to the CAVS or to the SAP system.	<Property name="Routing. DEBMAS04PortType.RouteToCAVS">false</Property>
Routing. DEBMAS04PortType.CAVS.EndpointURI	No Default Value	CAVS Endpoint URL	<Property name=" Routing. DEBMAS04PortType.CAVS.EndpointURI">[CAVSEndpointURL]</Property>

Property Name	Property Value	Description	Example
Routing. HRCA_CUSTOMER_CONTACT_READPortType.SAP_01.EndpointURL	No default value	URL for SAP Instance Webservice for “SAPSyncCustomerWS” web service	<Property name=" Routing. HRCA_CUSTOMER_CONTACT_READPortType.SAP_01.EndpointURL ">[AppEndpointURL]</Property>
Routing. HRCA_CUSTOMER_CONTACT_READPortType.RouteToCAVS	True/false. Default = False	Controls whether SyncCustomerSAPListProvABCSImpl should route messages to the CAVS or to the SAP system.	<Property name="Routing. HRCA_CUSTOMER_CONTACT_READPortType.RouteToCAVS" >false</Property>
Routing. HRCA_CUSTOMER_CONTACT_READPortType.CAVS.EndpointURI	No Default Value	CAVS Endpoint URL	<Property name=" Routing. HRCA_CUSTOMER_CONTACT_READPortType.CAVS.EndpointURI">[CAVSEndpointURL]</Property>
Routing. BAPI_CUSTOMER_GETLISTPortType.SAP_01.EndpointURL	No default value	URL for SAP Instance Webservice for “SAPSyncCustomerWS” web service	<Property name=" Routing. BAPI_CUSTOMER_GETLISTPortType.SAP_01.EndpointURL ">[AppEndpointURL]</Property>
Routing. BAPI_CUSTOMER_GETLISTPortType.Route ToCAVS	True/false. Default = False	Controls whether SyncCustomerSAPListProvABCSImpl should route messages to the CAVS or to the SAP system.	<Property name="Routing. BAPI_CUSTOMER_GETLISTPortType.RouteToCAVS">false</Property>
Routing. BAPI_CUSTOMER_GETLISTPortType.CAVS.EndpointURI	No Default Value	CAVS Endpoint URL	<Property name=" Routing. BAPI_CUSTOMER_GETLISTPortType.CAVS.EndpointURI">[CAVSEndpointURL]</Property>
Routing. CUSTOMER_CONTACTPS_GETPortType.SAP_01.EndpointURL	No default value	URL for SAP Instance Webservice for “SAPSyncCustomerWS” web service	<Property name=" Routing. CUSTOMER_CONTACTPS_GETPortType.SAP_01.EndpointURL ">[AppEndpointURL]</Property>
Routing. CUSTOMER_CONTACTPS_GETPortType.Route ToCAVS	True/false. Default = False	Controls whether SyncCustomerSAPListProvABCSImpl should route messages to the CAVS or to the SAP system.	<Property name="Routing. CUSTOMER_CONTACTPS_GETPortType.RouteToCAVS">false</Property>
Routing. CUSTOMER_CONTACTPS_GETPortType.CAVS.Endpo	No Default Value	CAVS Endpoint URL	<Property name=" Routing. CUSTOMER_CONTACTPS_GETPortType.CAVS.EndpointURI">[CAVSEndpointURL]</Property>

Property Name	Property Value	Description	Example
intURI			

Settings for SyncCustomerPartyListResponseSAPProvABCSImpl Service property

Property Name	Property Value	Description	Example
Default.SystemID	SAP_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 picks the default SystemID from this config property.	<Property name="Default.SystemID">SAP_01</Property>
Routing. CustomerPartyResponseEBSV2.SAP_01.EndpointURL	No default value	URL for SAP Instance web service for “SAPSyncCustomerWS” web service	<Property name=" Routing. CustomerPartyResponseEBSV2.SAP_01.EndpointURL ">[AppEndpointURL]</Property>
Routing. CustomerPartyResponseEBSV2.Route ToCAVS	True/false. Default = False	Controls whether SyncCustomerSAPListProvABCSImpl should route messages to the CAVS or to the SAP system.	<Property name="Routing. CustomerPartyResponseEBSV2.RouteToCAVS">false</Property>
Routing. CustomerPartyResponseEBSV2.CAVS.EndpointURI	No Default Value	CAVS Endpoint URL	<Property name=" Routing. CustomerPartyEBSV2.CAVS.EndpointURI">[CAVSEndpointURL]</Property>

Settings for InterfaceContactToAccountEBF Service property

Property Name	Value/Default Value	Description	Example
ContactUpdate.BRM_01	True/False Default = True	This property checks if only an account needs to be synchronized. When set to True, both the account and the contact updates are synchronized to BRM. When set to False, only the account updates are synchronized to Oracle BRM.	<Property name="ContactUpdate.BRM_01 ">true</Property>

Chapter 8: Customer Data Enrichment

This chapter provides an overview of data enrichment and covers:

- Process Flow
- Data Enrichment Integration Services
- Cross-References
- Domain Value Maps
- Handling Errors
- Viewing EBO Implementation Maps (EIMs)
- Setting Configuration Properties

Overview

Data enrichment represents a spoke service to the CustomerPartyEBO to make a request for clean or enriched information from third-party data provider services. This third-party provider cleanses, recognizes, enriches, and protects Oracle Customer Hub contact records.

- Clean – build a foundation of accurate customer contact information.
- Recognize – enables customers to consistently recognize consumers and remove duplications, even with name changes, unreported moves, or incorrectly keyed information.
- Enrich – enrich, enhance the data by adding other contact, demographic, socio-economic, or lifestyle enhancements.
- Protect –compliance and regulatory expertise combined with suppression, risk, and compliance products.

Data enrichment enables the Oracle Customer Hub to enrich its Contact (B2C) information by sending a request message to pre-built third-party AIA connector. Depending on configuration, manual or automatic, a process is invoked to send basic name/address information to the third-party system for cleansing.

Two types of processing can occur here for the request message: realtime or batch; however, the scope of this integration is realtime only, where the Oracle Customer Hub is configured to synchronize updated/enriched data automatically to participating applications as and when an EBM is received from the third-party enrichment application. There is no support for batch flows in this integration.

Prerequisites

Prerequisites for implementing customer data enrichment are:

- Oracle Customer Hub setup as provided in Siebel UCM Data Enrichment Web Services

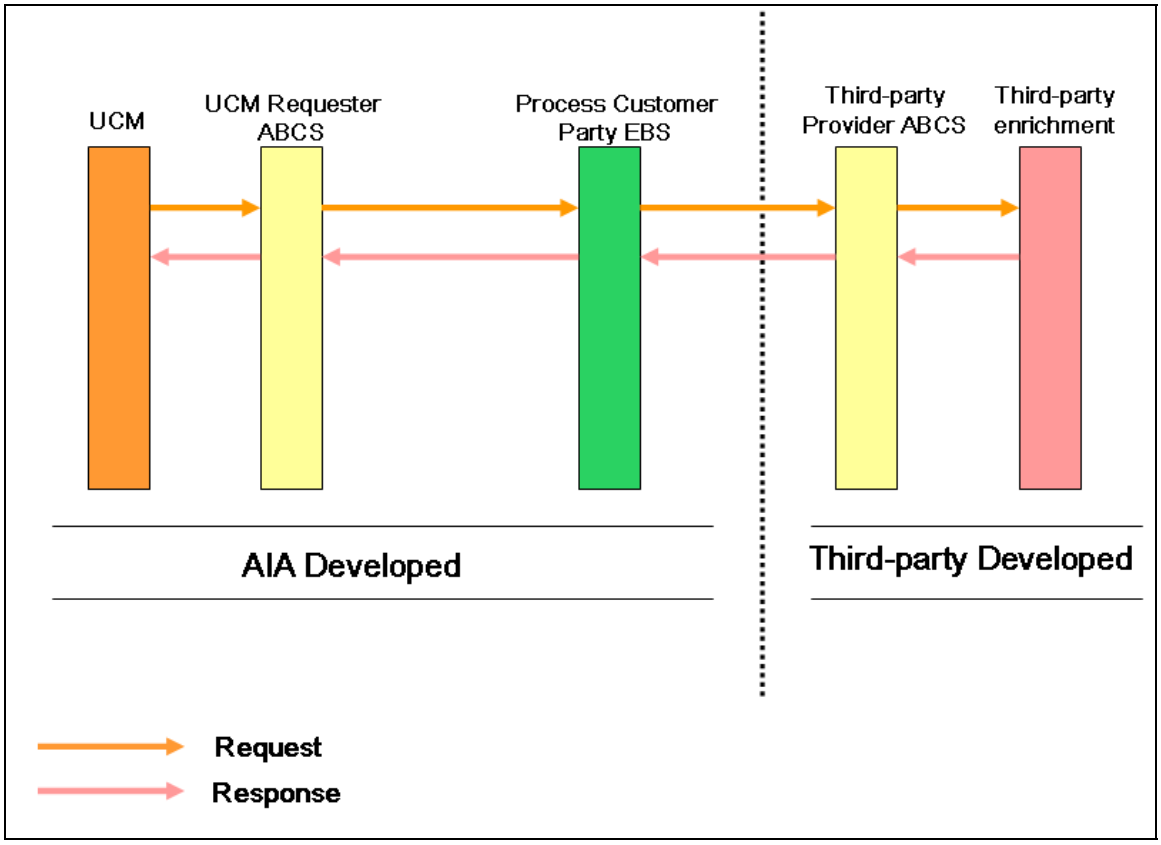
chapter at My Oracle Support note 763962.1.

- Oracle AIA configuration as mentioned in Handling Errors.
- Oracle AIA setup for the third-party provider, which would depend on the provider chosen by the customer.

Note: Set up of the third-party ABCS is not part of the Customer PIP. The third party provides the ABCS code and instructions.

Process Flow

This diagram depicts a high-level enrichment process.



High-level process

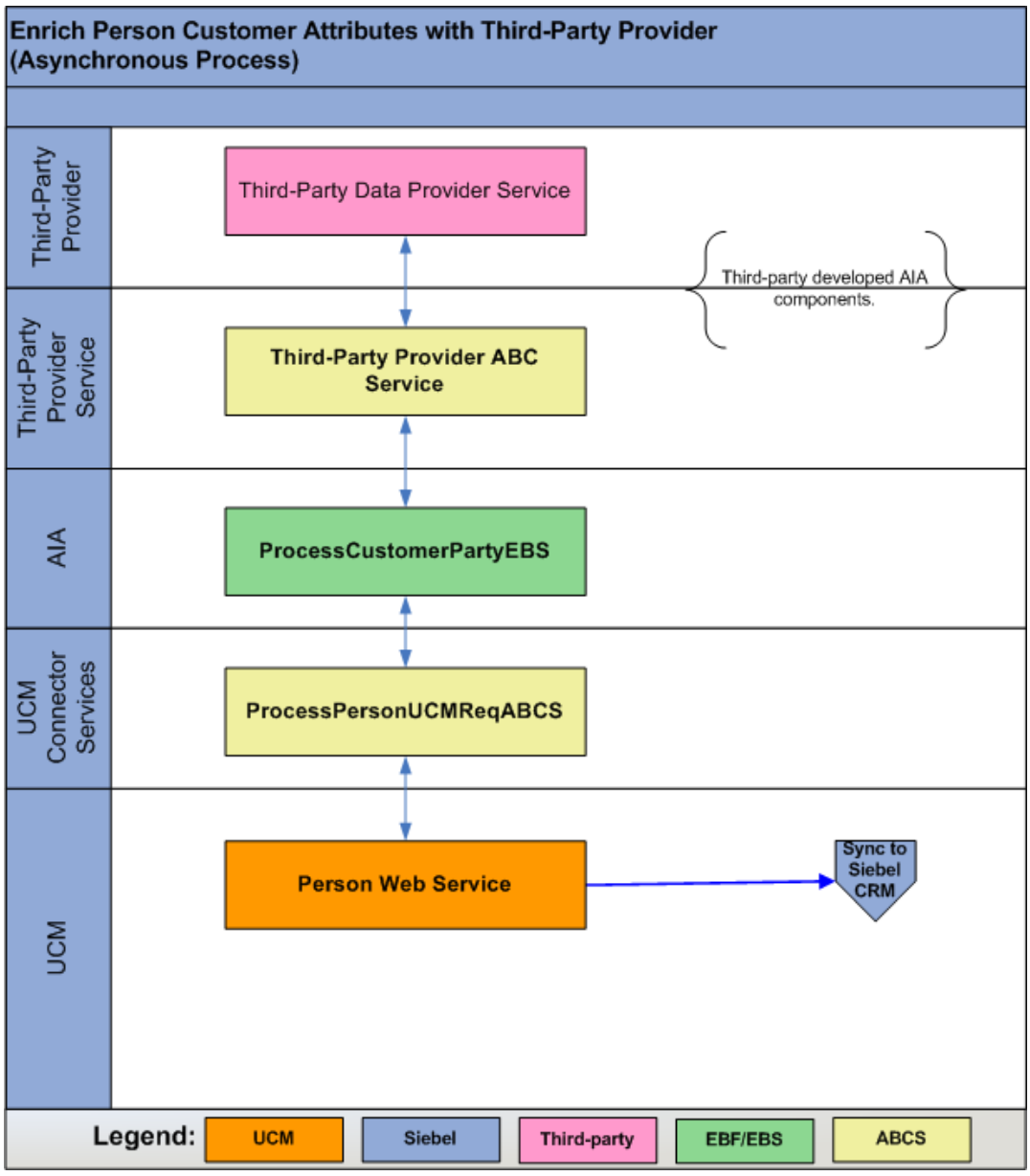
The high-level process is defined here:

- A set of parameters are sent using Oracle Customer Hub to the third-party enrichment application in the form of an EBM. Depending on the third party services to which an organization has subscribed, the third party consumes the message parameters and returns EBM of additional attributes.
- The Oracle Customer Hub consumes the inbound message and performs the required updates received as enriched data from the third-party enrichment application.

- Oracle Customer Hub can optionally synchronize these updated contact records to Siebel CRM or Oracle E-Business Suite if these are set up as participating applications.

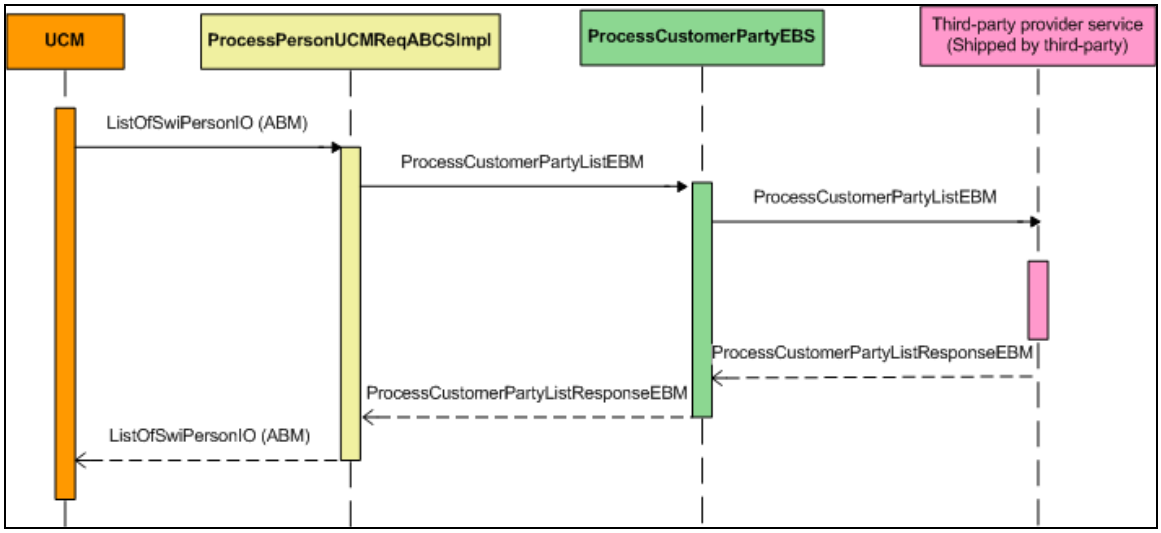
Note: A connector to AIA must be established by the third-party enrichment service to enable the applications to use the services provided.

This graphic illustrates the process for data enrichment from Oracle Customer Hub to third party and back.



Enriching person and customer attributes

Integration sequence is depicted here:



Customer data enrichment

The flow depicted in the sequence diagram is described here.

1. Oracle Customer Hub invokes the ABCS service for data enrichment by passing person records. A person record is sent from either Siebel or Oracle E-Business Suite, which invokes the Oracle Customer Hub web service to synchronize the record.

   Before synchronizing the record, Oracle Customer Hub invokes the ProcessPersonUCMreqABCSImpl to get the enriched data from the third-party provider.

2. ProcessPersonUCMReqABCSImpl: The Oracle Customer Hub requester ABC implementation, ProcessPersonUCMReqABCSImpl, transforms the Oracle Customer Hub ABM to the ProcessCustomerPartyListEBM and invokes the ProcessCustomerPartyListRequestResponse operation of the ProcessCustomerPartyEBS.

   This message contains one or more addresses to be enriched.

3. ProcessCustomerPartyEBS: Invoking ProcessCustomerPartyEBS with the ProcessCustomerPartyListRequestResponse operation routes the ProcessCustomerPartyListEBM to the third-party provider ABC implementation service.

4. ProcessPersonUCMReqABCSImpl: ProcessCustomerPartyListResponseEBM with enriched data from third-party provider is then transformed back to Oracle Customer Hub ABM and sent back as a synchronous response.

Data Enrichment Integration Services

The services that support the third-party integration are:

Service	Type	Owner	Description
ProcessPersonUCMReqABCSImpl	BPEL	AIA	Oracle Customer Hub ABCS Connector service initiated from Oracle Customer Hub for data enrichment. This synchronous service maps the Oracle Customer Hub ABM to ProcessCustomerPartyListEBM and invokes the ProcessCustomerPartyEBS. After getting a response, maps

Service	Type	Owner	Description
			the enriched response EBM back to Oracle Customer Hub ABM.
ProcessCustomerPartyEBS	ESB	AIA	ESB Routing service, which routes the ProcessCustomerPartyListEBM to the third-party provider and passes the response EBM back to the Oracle Customer Hub requester.
Third-party provider service	BPEL	Third-party	Third-party provider ABCS connector service. This service is not packaged with the AIA release, but delivered by third-party provider. This service maps the ProcessCustomerPartyListEBM to third-party provider input ABM, invokes the third-party provider service to get enriched data and maps the enriched data back as ProcessCustomerPartyListResponseListEBM

Cross-References

No cross-reference populate or lookups done in the data enrichment flow. Third-party IDs are directly passed into Oracle Customer Hub columns.

For a list of cross-references applicable for Oracle Customer Hub PIP, see Identifying Cross-References.

Domain Value Maps

The DVMs here pertain to the data enrichment flow only. For a complete list of DVMs in Oracle Customer Hub PIP, see Describing Domain Value Maps.

The DVMs used for the data enrichment flow are:

Name	DVM Column Name	Description	New/Existing
CONTACT_GENDERCODE	SEBL_01, COMMON, EBIZ_01, UCM_01	To map gender codes like Male, Female, M, and so on.	Existing
MARITAL_STATUS	UCM_01, COMMON, EBIZ_01	To map marital status codes like Married, Single, and so on.	Existing
ADDRESS_COUNTRYSUBDIVID	SEBL_01, COMMON, EBIZ_01, UCM_01	To map country sub division codes.	Existing
ADDRESS_COUNTRYID	SEBL_01, COMMON, EBIZ_01, UCM_01	To map country codes like USA, US, and so on.	Existing
STATE	SEBL_01, COMMON, EBIZ_01, UCM_01	To map state codes like CA, FL, TX, and so forth	Existing
ADDRESS_DELIVERYTYPE	UCM_01, COMMON, SEBL_01	To map delivery types like Residential Curb, Business, and so on.	New
ADDRESS_SEASONALINDICATORTYPE	SEBL_01, COMMON, UCM_01	To map seasonal address indicator types like Educational Facility, Seasonal Address, and so on.	New
ADDRESS_MOVETY	SEBL_01, COMMON,	To indicate the type of move like	New

Name	DVM Column Name	Description	New/Existing
PE	UCM_01	Individual, Business or Firm, and so on.	
ADDRESS_POSTAL PROCESSINGCODE	SEBL_01, COMMON, UCM_01	To map postal processing codes like AA match at ZIP Code level, LOT default match, and so on.	New

Handling Errors

Standard AIA error handling is implemented with catch blocks for remote, binding, and AIA faults. Fault policy is provided. Any error from third-party provider is propagated to Oracle Customer Hub using the fault schema as in any synchronous service.

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

The standard AIA configuration properties defined for data enrichment flow are provided here. For a complete list of properties in the **AIAConfigurationProperties.xml** file, see Setting Configuration Properties.

- System ID: The default system ID is set.
- Endpoint URLs: Properties are provided for the application service endpoint URLs.
- CAVS: Properties for enabling CAVS.
- ABCS Extension: Properties to govern ABCS Extension points.

Property Name	Value/Default Value	Description
Default.SystemID	UCM_01	OCH system code (like UCM_01, defined in BSR) from which requests originate for this process
ABCSExtension.PreProcessABM	True/False Default = False	Property that governs whether ABCS Extension is enabled to pre-process ABM at the pre-defined plug-in point. If set to true, then the Extension process is invoked.
ABCSExtension.PreProcessEBM	True/False Default = False	Property that governs whether ABCS Extension is enabled to pre-process EBM at the pre-defined plug-in point. If set to true, then the Extension process is invoked.
ABCSExtension.PostProcessEBM	True/False	Property that governs whether ABCS

Property Name	Value/Default Value	Description
	Default = False	Extension is enabled to post-process EBM at the pre-defined plug-in point. If set to true, then the Extension process is invoked.
ABCSExtension.PostProcessABM	True/False Default = False	Property that governs whether ABCS Extension is enabled to post-process ABM at the pre-defined plug-in point. If set to true, then the Extension process is invoked.
Routing.ProcessCustomerPartyEBS.ProcessCustomerPartyListRequestResponse.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTION Default = PRODUCTION	Property to govern whether the message is routed to CAVS or to the specified target service. Default value is PRODUCTION, which routes to the target service.
Routing.ProcessCustomerPartyEBS.ProcessCustomerPartyListRequestResponse.RouteToCAVS	True/False Default = False	Property that governs whether the service should route the message to the CAVS endpoint or not. Default value is false, which does not route to CAVS. If set to true, it routes to CAVS using the endpoint specified in the CAVS.EndpointURI property.
Routing.ProcessCustomerPartyEBS.ProcessCustomerPartyListRequestResponse.CAVS.EndpointURI	http://${http.hostname}:${http.port}/AIAValidationSystemServlet/syncresponsesimulator	CAVS Endpoint URI, when CAVS is enabled.

Index

ADF runtime, 75

AIA Integration Services, 11

Assumptions and Constraints, 51

CommsProcessBillingAccountListEBF, 14

CommsProcessFulfillmentOrderBillingAccountListEBF, 13

CommunicationsCustomerPartyEBSV2, 12, 92

CommunicationsCustomerPartyResponseEBSV2, 13, 92

Core AIA Components, 10

Customer Data Enrichment, 139

- Configuration Properties, 144
- Data Enrichment Integration Services, 142
- Domain Value Maps, 143
- UCM configuration, 140

Customer Master Data Management Integration Base Pack, 17

CustomerPartyEBSV2, 11

CustomerPartyOrchestrationEBSV2, 12

CustomerPartyOrchestrationResponseEBSV2, 12

CustomerPartyResponseEBSV2, 11

EBO Implementation Maps, 111

FetchCustomerPartyEBF, 13

Implementing OCH PIP

- Composite Application Framework, 111
- Configuring Sequencing for the Synchronize Flows, 110
- Domain Value Maps, 106
- Error Handling, 111
- Identifying Cross-references, 104
- Identifying OCH Row IDs, 102
- Identifying Oracle E-Business Suite Entities, 102
- Identifying SAP Entities, 102
- Identifying Siebel Row IDs, 102
- Oracle E-Business Suite System Profiles, 108
- Populating Cross-References, 103
- SAP System Profiles, 108
- Scheduling Concurrent Processes, 109
- Setting Configuration Properties, 112
- Setting Up Cross-References, 101
- Setting Up Organizations, 99
- Validating Cross-References, 104

InterfaceContactToAccountEBF, 15

Match and Fetch between Oracle E-Business Suite and Oracle Customer Hub, 76

Match and Fetch between Siebel CRM and Oracle Customer Hub, 60

Merge Flow from Oracle Customer Hub to Oracle BRM, 46

Merge Flow from Oracle Customer Hub to Oracle E-Business Suite, 46

Merge Flow from Oracle Customer Hub to Siebel CRM, 42

OCH Integration Services, 48

OCH Publishes Updates to BRM

- Existing Customer Data, 33
- New Customer Data, 30

Oracle AS Adapter, 109

Oracle BRM Integration Option, 91

Oracle BRM Integration Services, 91

Oracle BRM Interfaces, 91

Oracle Customer Hub Interfaces, 50

Oracle E-Business Suite Integration Option, 75

Oracle E-Business Suite Integration Services, 85

Oracle E-Business Suite Interfaces, 88

Overview

- Data Enrichment Flow, 10
- Match and Fetch Organizations and Persons, 8
- Merge Flows, 9
- Process Integration Pack, 5
- Synchronize Organizations and Persons, 8

Periodic Cron-Job, 36

ProcessCustomerPartyEBS, 12

ProcessPersonUCMReqABCSImpl, 50

QueryCustomerPartyListEbizCreateAdapter, 86

QueryCustomerPartyListEbizUpdateAdapter, 86

QueryCustomerPartyListUCMProvABCSImpl, 49

QueryCustomerPartyUCMProvABCSImpl, 49

QueryPersonAccountEbizCreateAdapter, 87

QueryPersonAccountEbizUpdateAdapter, 86

SAP Integration Option, 93

SAP Integration Services, 97

SAP Interfaces, 96

SAP to OCH Create Update:, 94

Siebel CRM Integration Option, 59

Siebel CRM Integration Services, 69

Siebel CRM Interfaces, 72

SyncAccountSiebelAggregatorAdapter, 70

SyncAccountSiebelReqABCSImpl, 70

SyncAcctSiebelAggrEventConsumer, 71

SyncAddressSiebelAggregatorAdapter, 70

SyncBPSiebelAggregatorAdapter, 71

SyncContactSiebelAggregatorAdapter, 70

SyncContactSiebelReqABCSImpl, 71

SyncContSiebelAggrEventConsumer, 71

SyncCustomerPartyListBRM_01CommsJMSConsumer, 92

SyncCustomerPartyListBRMCommsJMSProducer, 92

SyncCustomerPartyListBRMCommsProvABCSImpl, 92

SyncCustomerPartyListEbizAdapter, 87

SyncCustomerPartyListEbizEventConsumer, 86

SyncCustomerPartyListEbizProvABCSImpl, 85

SyncCustomerPartyListEbizReqABCSImpl, 87

SyncCustomerPartyListPersonEbizAdapter, 87

SyncCustomerPartyListResponseEbizProvABCSImpl, 87

SyncCustomerPartyListResponseSAPProvABCSImpl, 97

SyncCustomerPartyListResponseSiebelProvABCSImpl, 71

SyncCustomerPartyListResponseUCMProvABCSImpl, 50

SyncCustomerPartyListSAPProvABCSImpl, 97

SyncCustomerPartyListSiebelProvABCSImpl, 71

SyncCustomerPartyListUCMProvABCSImpl, 50

SyncCustomerSAPReqABCSImpl, 97

SyncCustomerSiebelEventAggregator, 70

Synchronization of Account Contacts from Oracle Customer Hub to SAP, 39

Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle BRM, 28

Synchronization of Organizations and Persons from Oracle Customer Hub to Oracle E-Business Suite, 18

Synchronization of organizations and Persons from Oracle Customer Hub to SAP, 37

Synchronization of Organizations and Persons from Oracle Customer Hub to Siebel CRM, 23

Synchronization of Organizations and Persons from Oracle E-Business Suite to Oracle Customer Hub, 80

Synchronizing Organizations and Persons from SAP to OCH, 93

Synchronizing Organizations and Persons from Siebel CRM to Oracle Customer Hub, 64

SyncOrganizationUCMJMSConsumer, 49

SyncOrganizationUCMJMSProducer, 48

SyncOrganizationUCMReqABCSImpl, 49

SyncPersonAccountEbizEventConsumer, 86

SyncPersonAccountEbizReqABCSImpl, 86

SyncPersonUCMJMSConsumer, 49

SyncPersonUCMJMSProducer, 49

SyncPersonUCMReqABCSImpl, 50