Oracle® Order Management Integration Pack for Oracle® Transportation Management, Oracle® E-Business Suite and Siebel CRM 2.5 - Implementation Guide

Release 2.5

Part No. E15790-01

October 2009

ORACLE

Oracle Order Management Integration Pack for Oracle Transportation Management, Oracle E-Business Suite and Siebel CRM 2.5 - Implementation Guide

Part No. E15790-01

Oracle Order Management Integration Pack for Oracle Transportation Management, Oracle E-Business Suite and Siebel CRM 2.5 - Implementation Guide

Contents

Oracle Order Management Integration Pack for Oracle Transportation Management, Oracle E-Business Suite and Siebel CRM 2.5 - Implementation Guide Preface

This preface discusses:

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

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

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

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

Oracle Application Integration Architecture Foundation Pack Concepts and Technologies Guide

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

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

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

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

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

- Create an integration scenario.
- Define business service patterns.
- Design and develop enterprise business services.
- Design and develop enterprise business flows.
- Design and construct application business connector services.

- Work with message transformation, enrichment, and configuration.
- Develop custom xpath functions.
- Design and construct JMS Adapter services.
- Work with enterprise message headers.
- Work with message routing.
- Work with transactions.
- Develop Oracle AIA services to work with the Composite Application Validation System (CAVS).
- Configure Oracle AIA processes to be eligible for error handling and logging.
- Extend enterprise business objects.

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

Oracle Application Integration Architecture Process Integration Packs

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

Additional Resources

The following resources are available:

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

Chapter 1: Understanding the Oracle Order Management Integration Pack for Oracle Transportation Management, Oracle E-Business Suite and Siebel CRM

This chapter provides an overview of the Transportation Order Management Integration Pack and discusses the:

- Participating Applications.
- Business process flows.
- Solution assumptions and constraints.

Process Integration Pack for Transportation Order Management Overview

Transportation Order Management Process Integration Pack (PIP) provides a best-of-breed solution that gives streamlined and end-to-end order management business process flows that enable faster time to process the Transportation orders, and market new products, and services.

This PIP interacts with the applications, such as Siebel CRM, Oracle Transportation Management (OTM), and Oracle E-Business Suite that are required to enable the integration process.

The Transportation Order Management PIP enables implementers to synchronize Siebel customer information to Oracle E-Business Suite and OTM. It also enables you to synchronize location, and product information from Siebel to OTM. An order is rated in Siebel from OTM and subsequently synchronized in OTM upon submission. Finally any change in order status in OTM is synchronized back to Siebel.

Key Benefits

- Customer is kept up to date in Siebel CRM, OTM, and Oracle E-Business Suite. By having a single view of the customer, they can be served better by providing the correct products/services, up-sell/cross-sell etc.
- Order capturing (Siebel), and Order fulfillment system (OTM) should have up-to-date correct information about Customers, Locations, and Products for order rating, capture, and execution.
- No manual revision of Order Release or Sell Shipment is necessary in the Fulfillment (OTM) system.
- Improved CSR productivity: The CSR/Sales Reps will be able to execute a rating query from the Order capture system (Siebel CRM) into the rating engine to provide the customer with various options about Itineraries (Rates and Routing).
- Increased operational efficiencies. There is no need to manually replicate any data.
- Pre-integrated solution leads to lower cost and risk for implementation.
- Faster time from Order capture to Shipment delivery, and Invoicing.
- Improved customer service: Real-time synchronization leads to better service to customers.
- Enhance customer visibility and accuracy.
- Faster time-to-market with new products/services.

Security

The Order Management Integration Pack also allows the organization to add one more level of security in the Integration. Integration Pack has been enhanced with Password Encryption feature for the Oracle Transportation Management Inbound flows, which allows the users to store the encrypted OTM password in the AIAConfigurationProperties file.

Participating Applications Overview

This section provides an overview of the following applications participating in the process integration:

- Siebel CRM and Order Management

  Siebel applications maximize sales effectiveness in real time by accelerating the quote-to-cash process, aligning sales channels, increasing pipeline and win rates, and raising average transaction values.

Siebel Order Management allows employees such as salespeople and call center agents to create and manage orders through their entire life cycle. Siebel Order Management can be tightly integrated with back-office systems, allowing users to perform tasks such as confirming availability and monitoring the fulfillment process.

Siebel Order Management allows employees to:

- Create orders for new products and services
- Modify in-process orders that have been submitted for fulfillment

For more information, see the *Oracle Siebel CRM User Guide*.

- Oracle Transportation Management

  Oracle Transportation Management (OTM) allows organizations to gain control of transportation and logistics operations while minimizing costs and eliminating inefficient and redundant procedures. A single, Web-architected application for all the transportation needs, Oracle Transportation Management combines broad logistics capabilities with deep transportation-management functionality.

For more information, see the *Oracle Transportation Management User Guide*.

- Oracle E-Business Suite

  Oracle Trading Community Architecture (TCA) allows you to manage complex information about the parties, or customers, who belong to your commercial community, including organizations, and locations. The information is maintained in TCA is the single source of trading community information for Oracle E-Business Suite applications. These applications, as well as TCA itself, provide user interfaces, and other features for you to view, create, and update customer information.

For more information, see the *Oracle E-Business Suite TCA User Guide*.

Transportation Order Management Business Process Flow

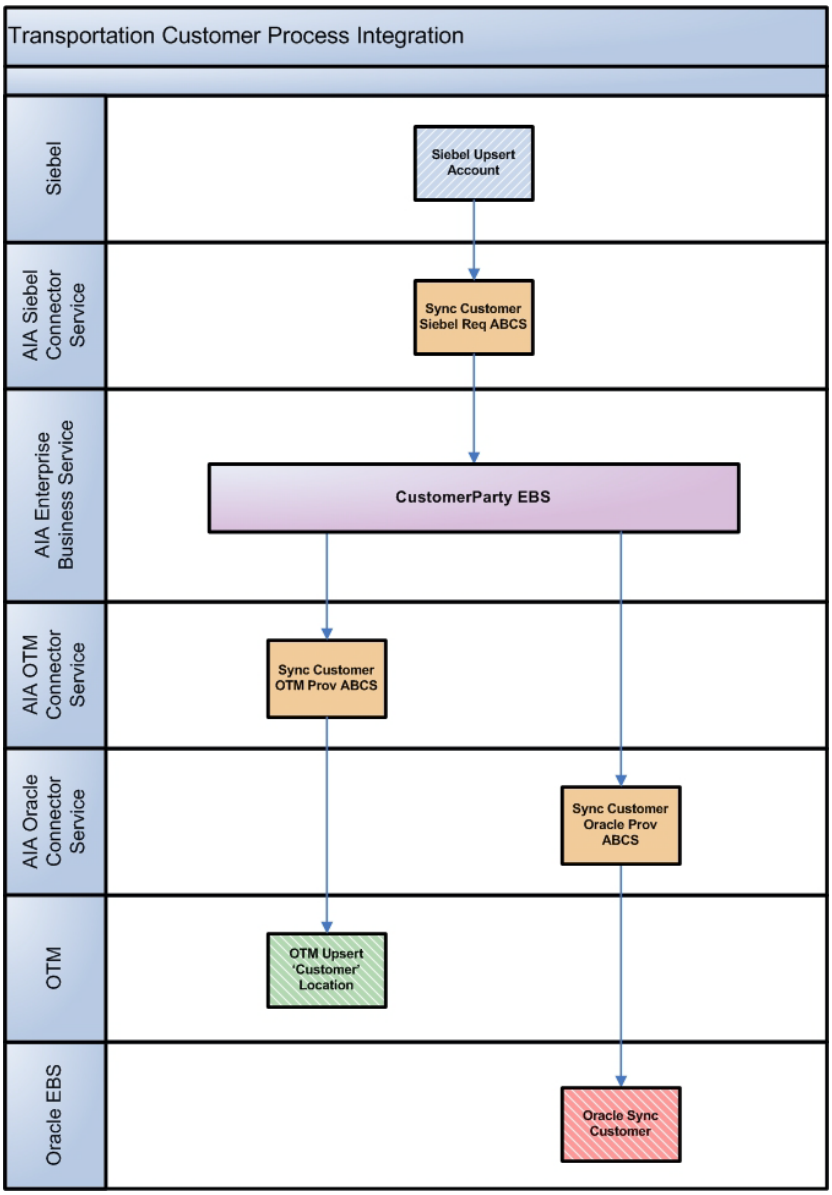
The Transportation Order Management PIP consists of these integration flows:

- Synchronizing Customer
- Synchronizing Location
- Synchronizing Order
- Querying Order Itinerary (Rates and Routes)
- Synchronizing Order Status
- Synchronizing Product

The main focus of this integration is to support the Transportation Order Management business flow that spans across the three applications.

Synchronizing Customer

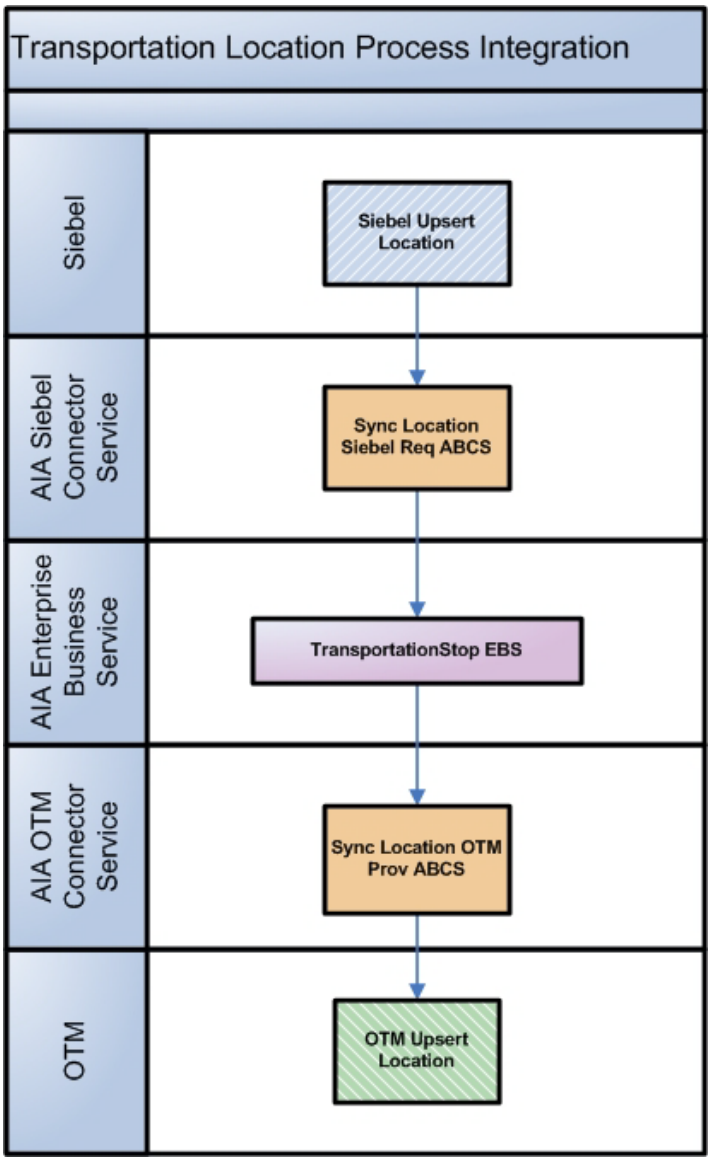
The following diagram shows the Customer Process Integration in the Transportation Order Management PIP:



Transportation Order Management – Customer Process Integration flow

Synchronizing Location

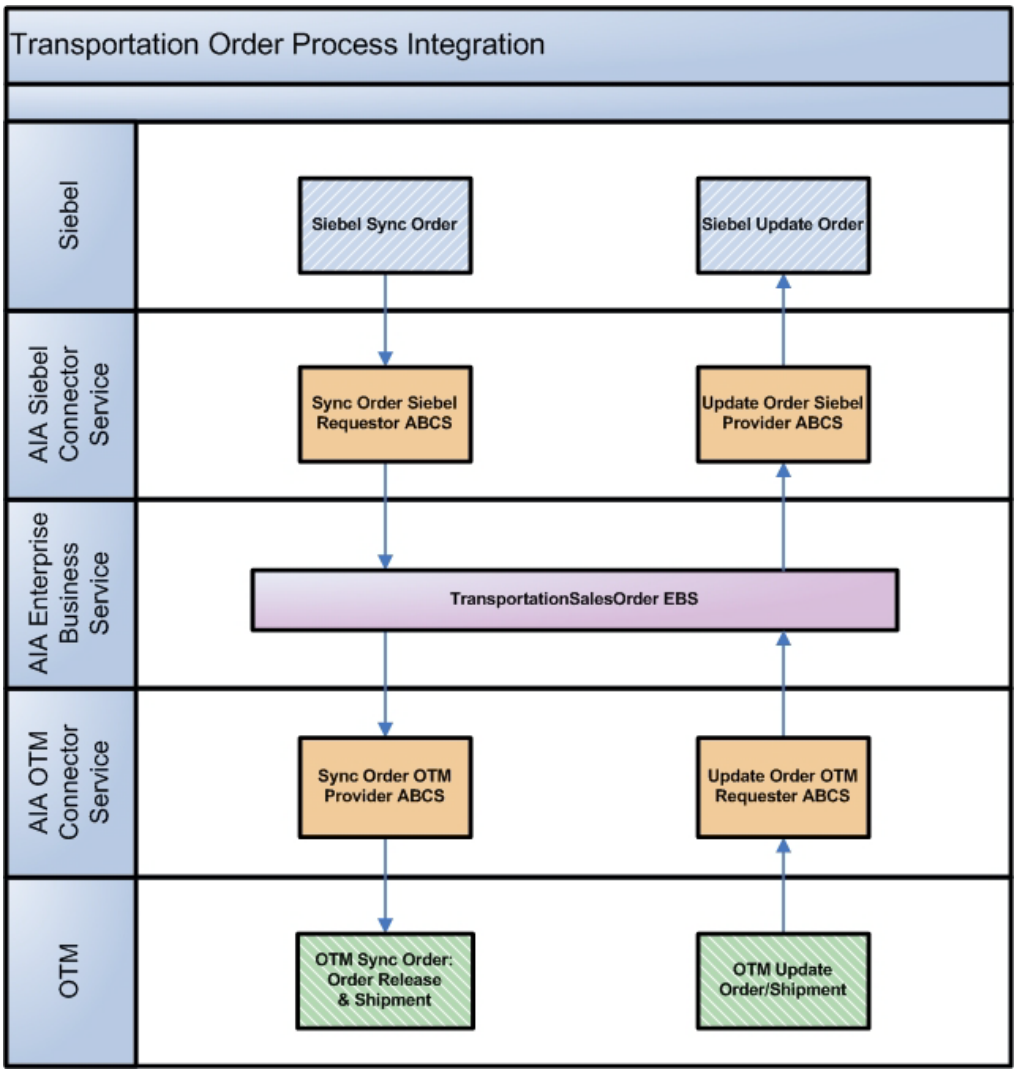
The following diagram shows the Location Process Integration in the Transportation Order Management PIP:



Transportation Order Management – Location Process Integration flow

Synchronizing Order and Order Status

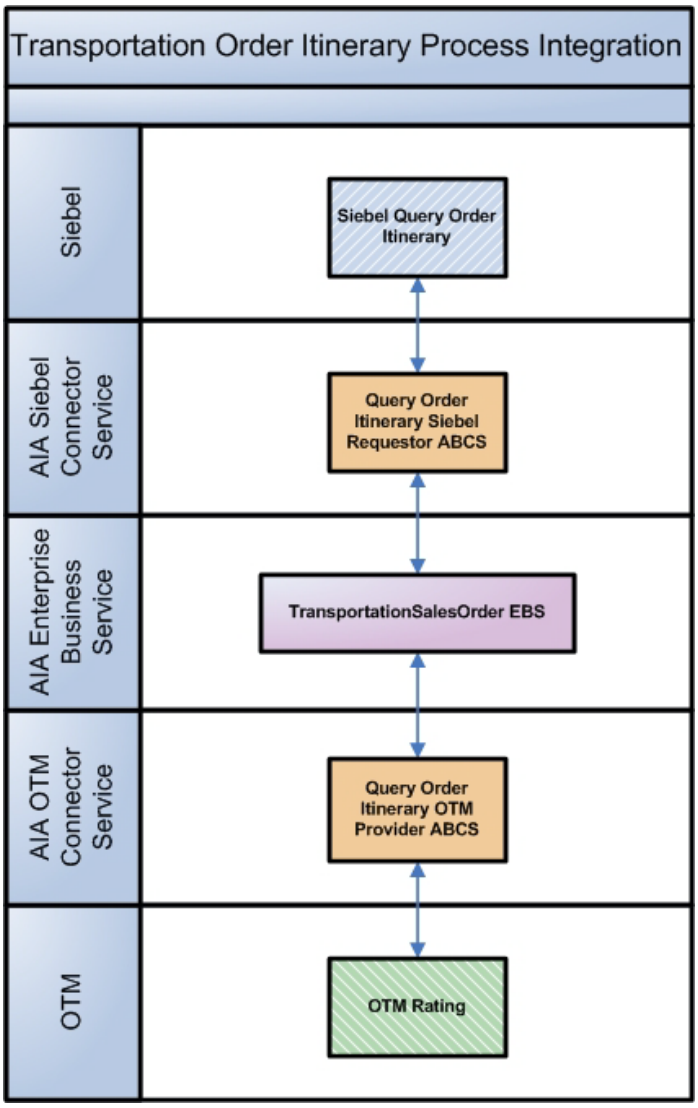
The following diagram shows the Transportation Order Process Integration in the Transportation Order Management PIP:



Transportation Order Management –Order Process Integration flow

Querying Order Itinerary and Rate

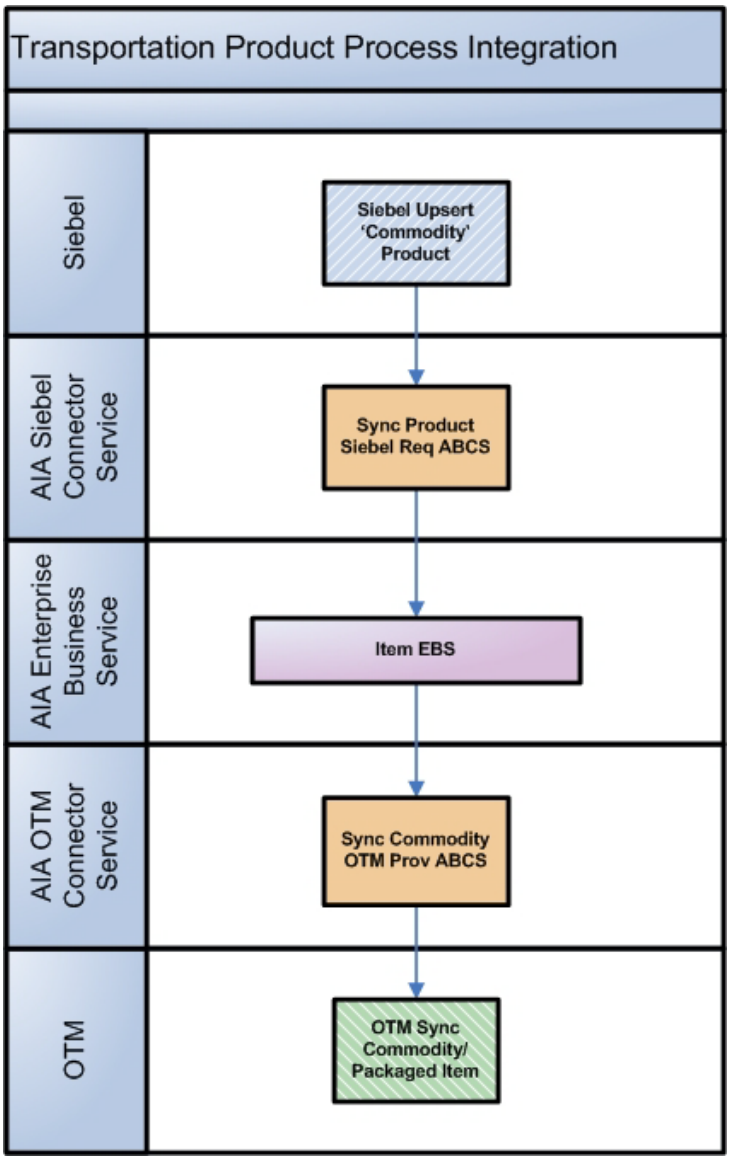
The following diagram shows the Query Order Itinerary Process Integration in Transportation Order Management PIP:



Transportation Order Management – Order Itinerary Process Integration flow

Synchronizing Product

The following diagram shows the Product Process Integration in Transportation Order Management PIP:



Transportation Order Management – Product Process Integration flow

Order Management Solution Assumptions and Constraints

The assumptions and constraints for the Transportation Order Management PIP are:

- The main focus of this integration is to support the Transportation Order Management business flow that spans across the three applications.

- This integration supports the “Business to Business” Transportation Order Management flows, and not the “Business to Customer” scenarios.
- The Customer Accounts in Siebel (Tendering Party, Liable Party, Invoice To Party) should be synchronized as part of Customer synchronization. Customers must be synchronized before they can be used in Order flow.
- Siebel “Commodity” Products should be synced to OTM’s Commodity and Packaged Items prior to the Order flow.
- The Locations in the Source, Destination, or any other Stop should also be synced as part of the Location flow. Locations must be set up and synchronized to OTM before they are used in Order flow.
- The Accessorial and the Special Services (Actions) set-up are done manually.
- There is no initial bulk load of any data between any applications.
- Deleting an order in Siebel is not supported currently.
- Delivered support is for Siebel SIA version only

Chapter 2: Describing the Process Integration for Customers

This chapter provides an overview of the process integration for customers and discusses:

- Integration flows.
- Data requirements.
- Siebel interfaces.
- Oracle E-Business Suite interfaces.
- OTM interfaces.
- Core AIA components.
- Integration services.

Process Integration for Customers Overview

This section provides an overview of the process integration for creating and synchronization of customer accounts between Siebel, E-Business Suite, and Oracle Transportation Management (OTM) and discusses the following integrations:

- Synchronize Create Account from Siebel CRM to OTM, and Oracle EBS
- Synchronize Update Account from Siebel CRM to OTM, and Oracle EBS

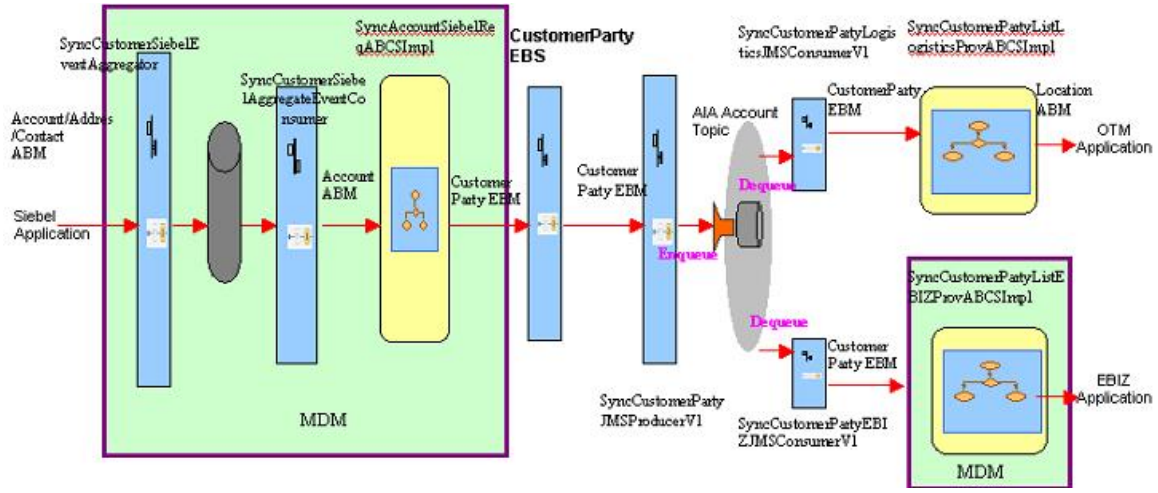
Customers are important for any transportation order. In Siebel the customers are created as accounts. These accounts are needed in OTM for execution of the order and the same accounts are needed in Oracle EBS Suite for billing purpose.

In this integration Siebel sends accounts to OTM and Oracle EBS. Whenever a new account is created in Siebel, a real time synchronization flow is initiated to synchronize it to a customer account in Oracle E-Business Suite. A new Location and the corresponding Corporation are created in OTM simultaneously for that Siebel account.

The addresses of the Siebel account are synced to Oracle E-Business Suite to create locations, party sites and account sites, as well as synched as separate locations in OTM. The “Bill To,” and “Ship To” address relationships from Siebel, are synced to ‘’Party Site Use’’ and “Account Site use’ in Oracle E-Business Suite, and to Location Roles in OTM.

If an update to a customer record or any of the details e.g. addresses, contacts occurs in Siebel, the updated account record will be synchronized to both OTM and Oracle e-Business Suite.

This diagram shows the overall flow for the Customer Process Integration:



Customer Process Integration Flow

Prerequisites

There are no prerequisites for this integration.

The Customer Management process integration does not have a dependency on other processes being run. However, the Organization cross-reference must be set up first.

For more information about setting up cross-references, see Chapter 8: Implementing the Order Management Process Integration Pack.

Solution Assumptions and Constraints

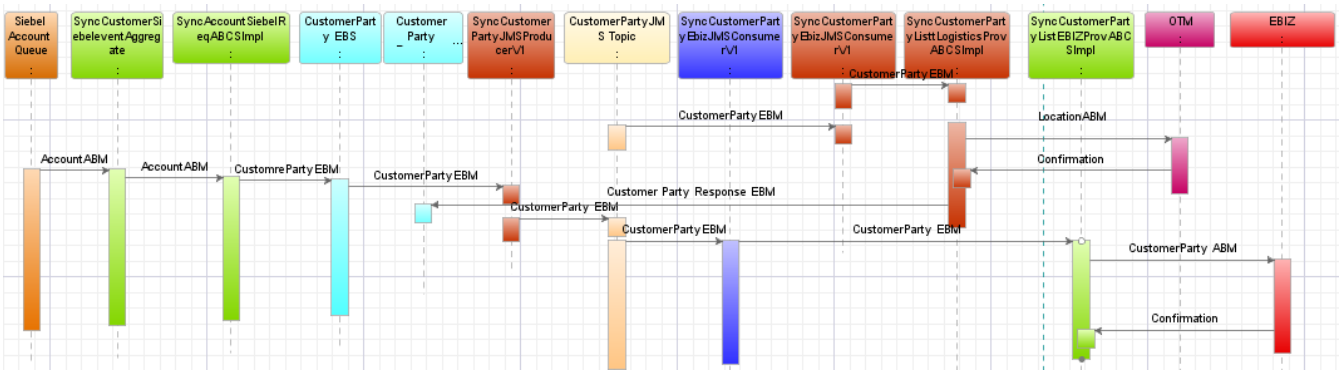
The integration assumes that the following statements are true:

- Only B2B scenarios are supported. B2C scenarios are not supported.
- Siebel Contacts are synchronized as Account Contacts to Oracle EBS, and not as Person Parties in Oracle EBS. However, contacts related to synced locations are synced to OTM
- Deletion of accounts, contacts, or address associated to a customer is not supported in the synchronization.
- If any error occurs and the customer message does not reach the target application, then AIA Error Handling Framework will notify the user. The user should then manually re-submit that transmission that failed for re-processing.
- This integration will not perform any business validation and thus will not raise errors for business data issues.
- The customer integration only supports the synchronization of accounts (and its addresses and contacts).

- Contacts can be shared across to multiple Accounts in Siebel. In OTM the same contact cannot be associated to multiple locations. Thus, if a contact is associated to two different locations in Siebel, then the same contact will be created twice in OTM for each location.
- Prospect Contacts are not synchronized from Siebel.
- Account Hierarchy synchronization is not in the scope of this release.
- The Customer Hub MDM PIP will be used for transformation from Siebel to EBO and E-Business Suite provider (EBO to Oracle e-Business Suite) transformation. For more details on Customer Hub PIP, please refer to its implementation guide.
- OTM domain value will be derived from business unit mapping in XREF. If implementers want to use their own logic for deriving the domain, they must use the extensible transformation template to do so.
- The OTM specific columns in cross-references will be populated based on the identifiers derived in the OTM connector. If implementers want to pass a different value in the OTM identifier then transformation's extensibility can be used to update the cross reference as well.
- The direction of data flow is only from Siebel to OTM and Oracle EBS. The reverse data flow is not supported in this integration. If any customer data is modified or created in OTM or Oracle EBS, those changes will not be reflected in Siebel. Any subsequent synchronization of the same data from Siebel to OTM or EBS may overwrite the changes already made in OTM or Oracle EBS.

Synchronizing Account Information

This sequence diagram shows the Incremental changes in the account:



Incremental changes in the Account

Siebel application can send one of the following three messages:

- **Account Message** – This message contains the details about an account and its associated addresses and contacts. This message is sent to the "SyncAccountSiebelReqABCSImpl" requestor process. The requestor process converts the message to "SyncCustomerPartyListEBM" and sends it to CustomerPartyEBS. From EBS the message then flows through the OTM provider and E-Business Suite provider to create the entries in the respective systems.

- **Address Message** – This message comes to AIA layer when an address is modified in Siebel. Siebel sends an Address message that contains the changed address data and all the associated accounts and locations for that address. MDM code then deciphers that message, picks up the associated accounts, and for each account it creates an account message and sends it to the MDM account requestor. Then the requestor processes the message in the same manner as described in the above paragraph for the “Account Message.”
- **Contact Message** – This message comes to AIA layer when a Contact is modified in Siebel. Siebel sends a Contact message that contains the changed contact data and all the associated accounts and locations for that contact. The MDM code then deciphers that message, picks up the associated accounts, and for each account it creates an account message and sends it to the MDM account requestor. Then the requestor processes the message in the same manner as described in the above paragraph for “Account Message.”

Data Requirements

The data requirements for Customer Process integration are:

- The business units being used must be seeded in all applications and in cross-references.
- Address is required in order to create an account in Oracle EBS.
- The address must have Address Line 1, City, State, Country, and Zip Code.

Siebel Interfaces

These are three schemas available for this outbound integration from Siebel.

- Account schema
- Change Address schema
- Change Contact schema.

For more information on Siebel customer WSDS and schema definition, see Customer Hub MDM PIP documentation.

Oracle E-Business Suite Interfaces

For more information on E-Business Suite WSDL and schema definition, see Customer Hub MDM PIP documentation.

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

Oracle OTM Interfaces

The OTM provides an interface through a web service to connect to its application. This connectivity will be established as a partner link in the Provider Service. The Logistics web service on being called will immediately return an Acknowledgement with a transmission number. Once the processing is complete, it will then send a Transmission Report back indicating the success or the failure.

For more information about the Logistics Service, see *Oracle Transportation Management Integration Guide.*

Core AIA Components

The integration flow uses the following components:

- CustomerPartyEBO
- CustomerPartyEBM

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 will 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."

Integration Services

The services delivered with the Customer process integration are:

- SyncCustomerPartyListLogisticsProvABCSImpl
- JMS Topic
- SyncCustomerPartyJMSProducerV1
- SyncCustomerPartyEbizJMSConsumerV1
- SyncCustomerPartyLogisticsJMSConsumer

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

For more information on other Customer integration services, refer to Customer Hub *MDM Implementation Guide*.

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.

SyncCustomerPartyListLogisticsProvABCSImpl

The SyncCustomerPartyListLogisticsProvABCSImpl service is a Business Process Execution Language (BPEL) process that receives SyncCustomerPartyListEBM as input from CustomerPartyListLogisticsJMSConsumerV1, transforms that message to Logistics ABM, and finally calls the OTM web service to process that message in OTM. Cross-reference values are populated upon successful message processing.

JMS Topic

A JMS topic is used to receive the message from customer party EBS and route that to both OTM connector and EBIZ connector. Topic is created in the AIA database under the user JMSUSER. The queue table for this topic is AIA_CUSTOMERPARTYJMSTTAB and the queue name for the topic is AIA_CUSTOMERPARTYJMSTV1.

SyncCustomerPartyJMSProducerV1

This service will publish the SyncCustomerPartyListEBM JMS message from EBS into the Topic.

This is an ESB Service with JMS Adapter and a Routing Rule. This will accept SyncCustomerPartyListEBM as input from CustomerPartyEBSV2. The received Customer Party List EBM payload message is published into the JMS Topic AIA_CUSTOMERPARTYJMSTV1.

SyncCustomerPartyEbizJMSConsumerV1

SyncCustomerPartyEbizJMSConsumerV1service is one of the subscribers to the topic that dequeues the SyncCustomerPartyListEBM JMS message from the JMS Topic for E-Business suite connector.

This is an ESB Service with JMS Adapter. This adapter dequeues the SyncCustomerPartyListEBM message for the subscribed consumer from the JMS Topic AIA_CUSTOMERPARTYJMSTV1 and routes the message to the SyncCustomerPartyListEbizProvABCSImpl provider service.

SyncCustomerPartyLogisticsJMSConsumer

SyncCustomerPartyLogisticsJMSConsumerV1 service is the other subscriber of the topic that dequeues the SyncCustomerPartyListEBM JMS message from the Topic for Logistics connector.

This is an ESB Service with JMS Adapter. This adapter dequeues the SyncCustomerPartyListEBM message for the subscribed consumer from the JMS Topic AIA_CUSTOMERPARTYJMSTV1 and routes to the SyncCustomerPartyListLogisticsProvABCSImpl.

Chapter 3: Describing the Process Integration for Location

This chapter provides an overview of the Location Process Integration and discusses:

- Integration flows.
- Data requirements.
- Siebel interfaces.
- OTM interfaces.
- Core AIA components.
- Integration services.

Process Integration for Location Overview

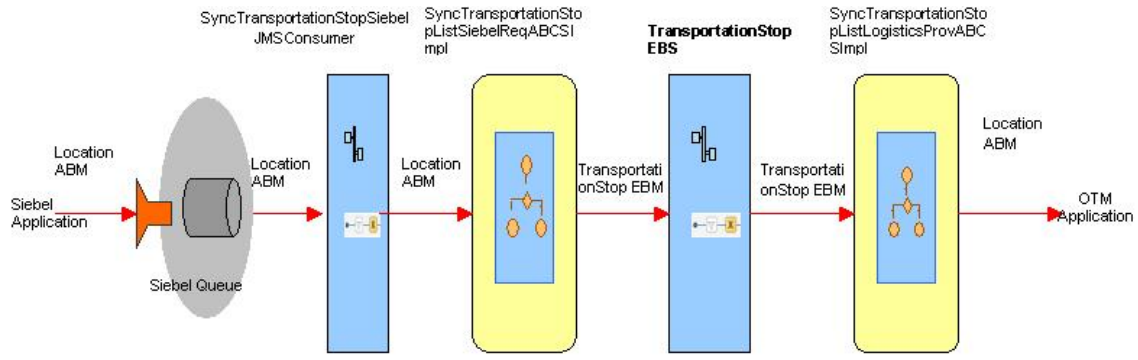
In Siebel CRM, the locations are created separately from an address or an account. In OTM these locations are needed for execution of the order.

In this integration process Siebel sends locations to OTM. Whenever a location is created or updated in Siebel, a corresponding location record will be created or updated in OTM. When an address, contact, or “Bill To”/”Ship To” details change in Siebel, all impacted locations and roles are synced properly to reflect the changes in OTM.

The process integration for location supports the following integration flows:

- Creating Location
- Updating Location

This diagram shows the overall flow for the process integration of location:



Location Process Integration flow

The Location Process Integration flow diagram displays the design where Siebel will send their location message to a queue. A JMS consumer will read the message, and send it to the requestor. The Siebel message is transformed to SyncTransportationStopListEBM in requestor connector, then the requestor connector will send the EBM message to EBS, and the Enterprise Business Service (EBS) will route the message to OTM provider. Then provider connector transforms the EBM to logistics schema and sends that logistics message to OTM web service to create Location in OTM.

Prerequisites

Prior to synchronizing a location, the corresponding parent account must be synchronized to OTM.

Solution Assumptions and Constraints

This integration assumes that the following statements are true:

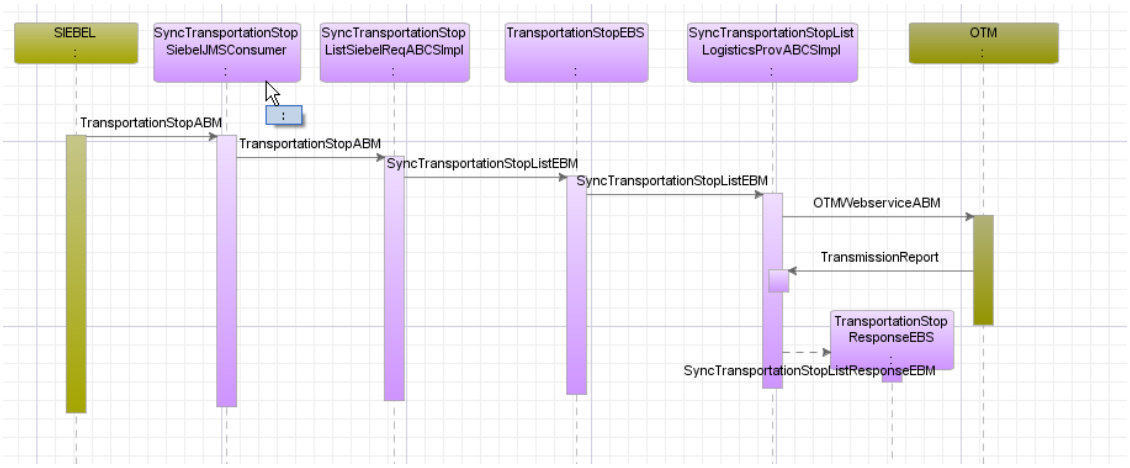
- Initial loading of existing data of locations is not supported in this process integration.
- No delete operations for the location, contacts, or roles on accounts are supported in this integration.
- Contacts are created once in Siebel, and associated to multiple locations. In OTM the same contact cannot be associated to multiple locations. Thus, if a contact is associated to two different locations in Siebel, then the same contact will be created in OTM for each location.
- In case if any error is occurred in the service layer and the location message does not reach the target application then AIA Error Handling Framework will be invoked and that will notify the user. The user should manually re-submit that transmission that failed for re-processing.
- This integration will not perform any business validation, and thus will not raise errors for those business validation issues.
- The location integration only supports the synchronization of locations.
- OTM domain value will be derived from business unit mapping. If implementers want to use their own logic for deriving the domain, they must use the extensible transformation template to do so.
- The OTM specific columns in cross-references will be populated based on the identifiers derived from main transformation to OTM. If implementers want to pass a different value in the OTM identifier then transformation's extensibility can be used to update the cross reference as well.
- The list of accounts for a location from Siebel will be stored till EBM. OTM provider will not be able to use the data stored in EBM since OTM doesn't allow multiple accounts to be associated with a location.
- To sync a location, parent account information is mandatory. It is also assumed that the selected parent account data is already synched prior to synching the location information.
- To use the aggregator services with location flow, couple of properties should be set to true in

the AIA configuration file (by default they will be set to false).

- Siebel Locations are not synced to Oracle E-business Suite.
- This integration flows in only one direction, Siebel to OTM. Any changes in OTM location or any new location created in OTM will not be synchronized back into Siebel.

Synchronizing Location Information

This sequence diagram shows the synchronization in the location:



Location Synchronization Sequence diagram

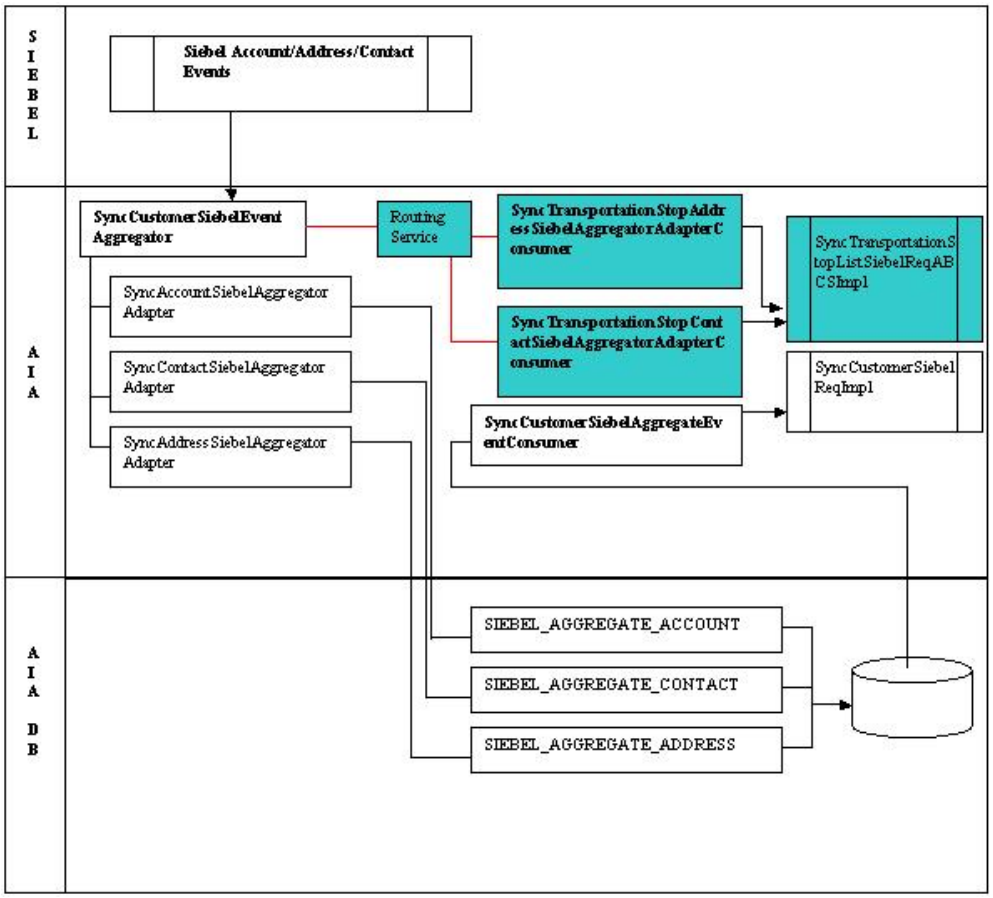
Overall Integration Flow

Addresses and Contacts are created independent of a location. When a location is created addresses and contacts are associated with that location. When the location is synced to OTM it creates a new location, and the address information is attached with that location in OTM. For the contacts, OTM creates new contacts for each associated contacts.

If any of the address information or contact information is updated in Siebel, all corresponding locations must be changed to reflect the changes. For example, when an address changes in Siebel, it sends out an address message with a list of impacted locations, and accounts. Similarly for contact changes it sends out a contact message with the list of impacted locations and accounts. These messages are sent to MDM aggregator process.

Existing MDM process receives these messages, goes through the list of accounts, and prepares the account message. With these account messages it calls the account integration flow. Since location is a new object MDM cannot update the data for impacted locations. To address that gap new services are added to MDM aggregator process.

The following diagram shows the new services added in the MDM Aggregator scheme. The new services are indicated in blue background.



Enhanced MDM Aggregator Process diagram

Data Requirements

Location name should be provided.

Siebel Interfaces

These are three schemas available for this integration from Siebel:

- Location schema
- Change Address schema
- Change Contact Schema.

For more information about Siebel customer WSDL and schema definition, see Customer Hub MDM PIP.

OTM Interfaces

Oracle Transportation Management provides an interface through a web service to connect to its application. This connectivity will be established as a partner link in the Provider Service. The Logistics web service on being called will immediately return an acknowledgement with a transmission number. Once the processing is complete, it will then send a Transmission Report back indicating the success or the failure.

For more information about the Logistics Service, see *Oracle Transportation Management Integration Guide.*

Core AIA Components

The integration flow uses the following components:

- TransportationStopEBO
- SyncTransportationStopListEBM

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 will 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."

Integration Services

These are the services delivered with this integration:

- AI A Transportation Stop Queue
- SyncTransportationStopSiebelJMSConsumer
- SyncTransportationStopListSiebelReqABCSImpl

- TransportationStopEBS
- TransportationStopResponseEBS
- SyncTransportationStopListLogisticsProvABCSImpl
- SyncTransportationStopAddressSiebelAggregatorAdapterConsumer
- SyncTransportationStopContactSiebelAggregatorAdapterConsumer
- SyncTransportationStopAggregatorRoutingService

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

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

AIA Transportation Stop Queue

The Transportation Stop Queue is used to queue Siebel location messages. This is an AQ. This queue will reside on any oracle database. For this integration, AIA database will be used as the place for this queue. The table used for this queue is AIA_TRANSPORTATIONSTOPJMSQTAB and the queue name is AIA_TRANSPORTSTOPJMSQUE.

SyncTransportationStopSiebelJMSConsumer

SyncTransportationStopSiebelJMSConsumer is the consumer for the JMS queue that will be invoked when the Siebel sends a location message into AIA_TRANSPORTSTOPJMSQUE queue. This service will route and invoke SyncTransportationStopListSiebelReqABCSImpl service.

SyncTransportationStopListSiebelReqABCSImpl

SyncTransportationStopListSiebelReqABCSImpl is a Business Process Execution Language (BPEL) process. This process receives Siebel Location ABM as input from Siebel system and transforms that message to SyncTransportationStopListEBM. Upon successful transformation, in addition to the mapping, EBM Header and Xref tables are populated. BPEL process invokes TransportationStopEBS with SyncTransportationStopListEBM as input.

TransportationStopEBS

TransportationStopEBS is an Enterprise Business Service. This service is used to route all location related actions like CreateTransportationStopList, QueryTransportationStop, SyncTransportationStopList, and UpdateTransportationStopList etc.

TransportationStopResponseEBS

TransportationStopResponseEBS is the Enterprise Business Service. This service is used to route all location related actions like CreateTransportationStopListResponse, QueryTransportationStopResponse, SyncTransportationStopListResponse and UpdateTransportationStopListResponse etc.

SyncTransportationStopListLogisticsProvABCSImpl

SyncTransportationStopListLogisticsProvABCSImpl is a Business Process Execution Language (BPEL) process. This process receives the SyncTransportationStopListEBM as a request from the TransportationStopEBS and invokes the LogisticsWebService. The web service will immediately send an acknowledgement and sends a transmission report about the success or failure of the data sent to OTM. Upon successful processing of location in OTM this process updates XREF data with OTM information.

SyncTransportationStopAddressSiebelAggregatorAdapterConsumer

SyncTransportationStopAddressSiebelAggregatorAdapterConsumer is a BPEL process. This process receives Siebel Address ABM as an input from the SyncCustomerSiebelEventAggregator, returns a Siebel Location ABM message, and invokes SyncTransportationStopListSiebelReqABCSImpl service.

SyncTransportationStopContactSiebelAggregatorAdapterConsumer

SyncTransportationStopContactSiebelAggregatorAdapterConsumer is a BPEL process. This process receives Siebel Contact ABM as input from the account aggregator service SyncCustomerSiebelEventAggregator and will be transformed to Siebel Location ABM and SyncTransportationStopListSiebelReqABCSImpl is invoked.

SyncTransportationStopAggregatorRoutingService

This is an ESB service that is used to route the ABM message received from SyncCustomerSiebelEventAggregator to SyncTransportationStopAddressSiebelAggregatorAdapterConsumer or SyncTransportationStopContactSiebelAggregatorAdapterConsumer.

Chapter 4: Describing Process Integration for Order

This chapter provides an overview of the Process Integration for Transportation Order, and discusses:

- Integration flows.
- Data requirements.
- Siebel interfaces.
- OTM interfaces.
- Core AIA components.
- Integration services.

Process Integration for Order Overview

In the Transportation Order Management Solution, Transportation Orders are created in Siebel CRM application and executed in Oracle Transportation Management (OTM) application. New or revised Orders will be synchronized in one direction from Siebel to OTM. Whenever a new Transportation Order is submitted in Siebel, a real time synchronization flow is initiated to publish it to AIA. AIA creates a corresponding Order Release, and Sell Shipment in OTM.

The process integration for order supports the following integration flows:

- Creating an Order
- Revising an Order (including cancellation)

Each Siebel Transportation Order includes following components:

- **Customers** involved: Liable, Tendering, and Invoice To Party.
- **Locations:** The Source and Destination locations.
- **Date-Time:** The Starting/Ending date-time for Pickup and Delivery.
- **Stops:** The intermediate Stops (with their locations) besides the Source and Destination.
- **Order Line Items:** The following four types of Products can be associated to Order lines:
- **Commodities:** These are the goods being shipped. No price is associated to Commodity line items.
- **Transportation**: The price for the Transportation Service.
- **Accessorial**: Any additional equipment that would be required to fulfill the order viz. forklift.
- **Special Services**: Any additional services to fulfill the order viz. Hand Unload.

- **Actions:** These are Special Services to be performed at any Stop. These could be associated to a commodity as well.

The prices of the Transportation, Accessorial, and Special Service lines are retrieved from OTM through the Order Itinerary/Rating integration as described in Chapter 7.

Corresponding to a Transportation Order in Siebel, the AIA synchronization process creates an Order Release, and a “Sell” Shipment in OTM.

The major components of Order Release synced to OTM are:

- **Involved Parties:** The Tendering/Liable/Invoice-To parties from Siebel Order are referenced as Involved Parties on the OTM Release Header.
- **Locations:** The Source and Destination Locations are on Order Release Header. The intermediate Stops are mapped to "OrStop" component of Release.
- **Date-Time:** The Pickup and Delivery time specified are mapped to Timeline in Release on the header and the stops.
- **Release Lines and ShipUnits:** For each Siebel Order line with a Commodity, a corresponding Order Release Line, and Release ShipUnit are created. For the “Commodity” product in Siebel Order line, a “Packaged Item” is specified in Release line, and a Commodity on the Release ShipUnit.

  If any Order line Commodity is of type of ‘Hazardous’ or ‘High Value’, then a Special Service is also created for it in the Release Header.

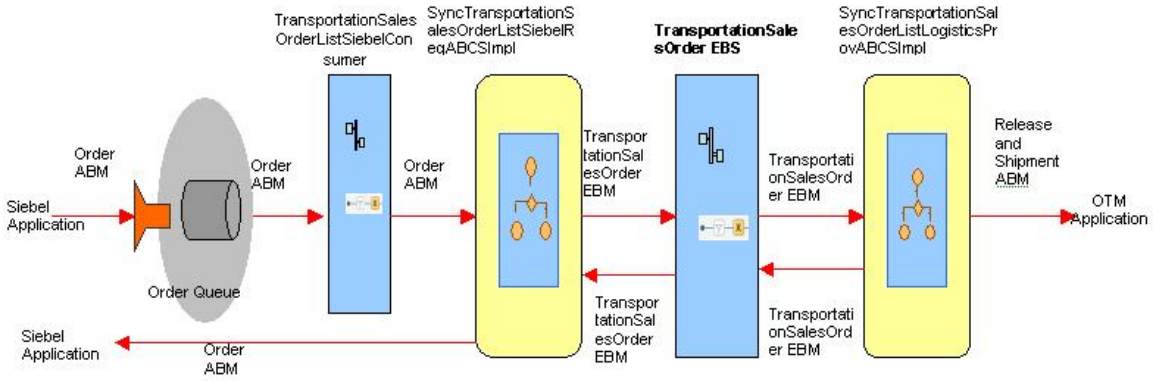
- **Special Services**: Corresponding to the Actions captured in Siebel Transportation Order, the Special Services for the Origin and Destination Stops will be created in the Release Header, whereas for the Intermediate Stops, the Special Services are created in the Release OrStop element.

  In addition, if the Commodity Order line in Siebel has Actions associated to it, then Special Service are created for the corresponding Release ShipUnit in OTM.

The major components of Sell Shipment synced to OTM are:

- **Involved Parties**: The Tendering/Liable/Invoice-To parties in Siebel Order are referenced as Involved Parties on the OTM Sell Shipment.
- **Shipment Costs**: The “Accepted Prices” for the Transportation, Accessorial, and Special Service Order lines are synced to Shipment Costs.
- **Shipment ShipUnits**: For each Siebel Order line specifying a Commodity, a corresponding Shipment ShipUnit is created in OTM. For the “Commodity” Product in Siebel Order line, a “Packaged Item” is specified in the Shipment ShipUnit.
- **Shipment Stops:** A Shipment Stop is created for every Stop specified in Siebel order.

This diagram shows the overall flow for the Order Process Integration:



Order Process Integration flow

This diagram shows that the Siebel application sends all its order messages in a queue. An AIA service reads that queue, gets the messages, and further sends the message to requestor. Inside the requestor service Siebel ABM message is transformed to the SyncTransportationSalesOrderListEBM, and then this service sends that EBM to Transportation Sales Order EBS.

The TransportationSalesOrderEBS will route that SyncTransportationSalesOrderListEBM to OTM provider. Inside the provider the EBM message is transformed to OTM ABM and that OTM ABM is send to OTM web service to create or modify the order.

Prerequisites

- The prerequisites for the Transportation Order Process Integration are:
- Customer Synchronization
- Location Synchronization
- Product Synchronization

For more information about these process integrations**,** see Chapter 2: Describing the Process Integration for Customers, Chapter 3: Describing the Process Integration for Locations, and Chapter 6: Describing the Process Integration for Products.

Solution Assumptions and Constraints

This integration assumes that the following statements are true:

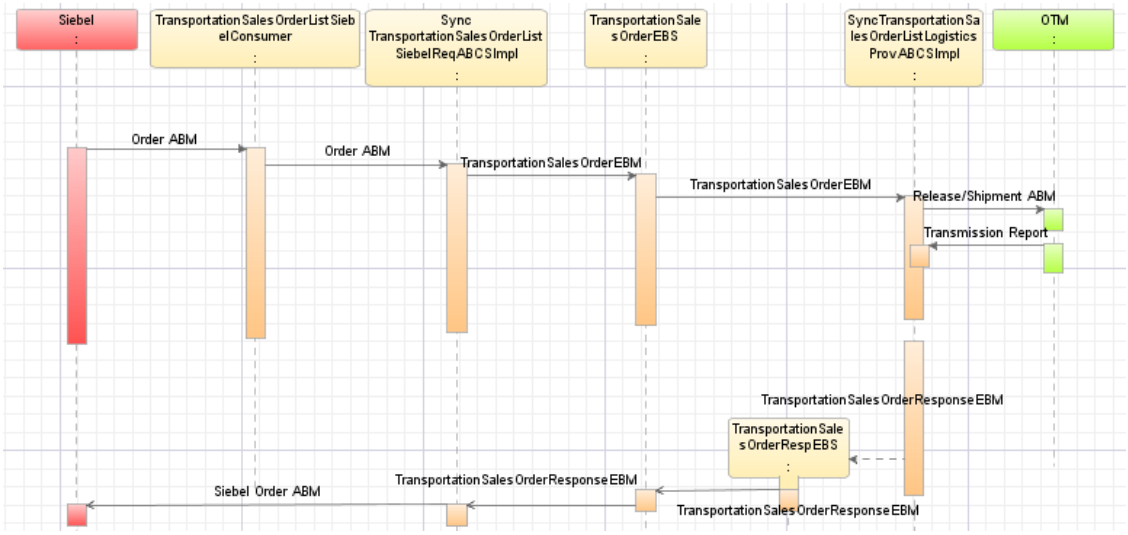
- Initial loading of existing orders is not supported in this solution.
- Orders have a status of ‘Accepted’ (or ‘Cancelled’) when they are submitted. Siebel CRM validates this status before any integration services are invoked.
- Outbound orders Statuses from Siebel are ignored except when canceling the order.
- No delete operations of an order are supported in this synchronization. Sync Operation is

used to create and update the orders.

- Order Process integration creates ‘Order Release’ and ‘Sell Shipment’ in OTM, but not the ‘Buy Shipment’.
- Stop action cross-references can’t be established from the OTM since OTM doesn’t have any equivalent identifier.
- The Customers, Locations, and ‘Commodity’ Products used in the Transportation Order must be synched prior to this order sync so that this process can inquire in the cross reference for their existence.
- This integration does not support manual updates to orders in Oracle Transportation Management.
- Liable party and the tendering party identifiers derived from Siebel will be validated against the cross references established in the account sync.
- For the liable party address and tendering party address this integration expects Siebel address identifiers (not location identifiers), and will look up the cross references established during account sync.
- In case if any error is occurred in the service layer and the order message does not reach target application, then AIA Error Handling Framework will be invoked, and it will notify the user. The user should manually re-submit that transmission that failed for re-processing.
- This process integration will not perform any business validation, and thus will not raise errors for business validation failures.
- OTM will receive Order name concatenated with GUID as their ID. OTM domain value will be derived from business unit mapping in XREF. If implementers want to use their own logic for deriving the domain, they must use the extensible transformation template to do so.
- The OTM specific columns in cross-references will be populated based on the identifiers derived from main transformation to OTM. If implementers want to pass a different value in the OTM identifier then transformation’s extensibility can be used to update the cross reference as well.
- The configuration property parameter for response message should be set to true while preparing an EBM indicating that the EBM wants a response.
- Siebel can send the messages in different languages; OTM can handle only single language. The OTM installation language will be captured in the configuration file. If the incoming Siebel message is of any other language that that is recorded in the configuration file then that message won’t be sent to OTM.
- The direction of the process is one way from Siebel to OTM.
- TRANSPORTATIONSALESORDER_ORDERLINEITEMPROPERTY DVM is a seeded DVM for name value pair attributes for Transportation product and Commodity. Any alteration to the seeded content will cause the transformation to fail, however new records can be added

Synchronizing Order Information

This sequence diagram shows the Incremental changes in the account:



Order synchronization sequence diagram

Data Requirements

This design assumes that the following statements are true:

- The Customers, Locations, and 'Commodity' Products used in the Transportation Order must be synched prior to submitting the Order.
- The 'Verify' button can be used in the Siebel Transportation Sales Order to validate that all the data has been appropriately populated.
- Order validation processes are successfully executed before the start of the Transportation Order integration flow.
- The order rating must be done to determine a rating solution for the Transportation Order to be submitted.

Siebel Interfaces

- The schemas that is available for this integration from Siebel is Order schema.
- The interface that is required for updating the order in Siebel is the Siebel order WSDL.

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.*

Oracle OTM Interfaces

The OTM application provides an interface through a web service to connect to its application. This connectivity will be established as a partner link in the Provider Service. The Logistics web service will immediately return an Acknowledgement with a transmission number when called and will send a Transmission Report back indicating the success or the failure once the process completes.

For more information about the Logistics Service, see *Oracle Transportation Management Integration Guide.*

Core AIA Components

The integration flow uses the following components:

- TransportationSalesOrderEBO
- SyncTransportationSalesOrderListEBM

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 will 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."

Integration Services

These are the services delivered with this integration:

- AIA Transportation Sales Order Queue
- TransportationSalesOrderListSiebelConsumer
- SyncTransportationSalesOrderListSiebelReqABCSImpl

- TransportationSalesOrderEBS
- TransportationSalesOrderResponseEBS
- SyncTransportationSalesOrderListLogisticsProvABCSImpl

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

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

AIA Transportation Sales Order Queue

This Transportation Sales Order Queue is used to queue Siebel order messages. This is an AQ. This queue can reside any oracle database. For this integration AIA DB is going to be used as the place for this queue. The table used for this purpose is AIA_TRANSSLSORDJMSQTAB and the queue name is AIA_TRANSSALESORDJMSQ.

TransportationSalesOrderListSiebelConsumer

TransportationSalesOrderListSiebelConsumer is the consumer to the queue that will be invoked when OTM places a message into the AIA_TRANSSALESORDJMSQ queue. This service will route and invoke SyncTransportationSalesOrderListSiebelReqABCSImpl service.

SyncTransportationSalesOrderListSiebelReqABCSImpl

SyncTransportationSalesOrderListSiebelReqABCSImpl is a BPEL process. This process receives Siebel ABM as an input from Siebel system and transforms the message to SyncTransportationSalesOrderListEBM. During transformation, Siebel ABM is transformed to SyncTransportationSalesOrderListEBM and EBM Header and Xref tables are populated. This process invokes TransportationSalesOrderEBS with SyncTransportationSalesOrderListEBM as an input. This process then waits for the response back from the TransportationSalesOrderResponseEBS. Based on the response received this service updates the Siebel application indicating whether the order was synced or failed.

TransportationSalesOrderEBS

TransportationSalesOrderEBS is an Enterprise Business Service. This service is used to route all transportation sales order related actions like CreateTransportationSalesOrder, QueryTransportationSalesOrder, SyncTransportationSalesOrderList, and UpdateTransportationSalesOrder etc.

TransportationSalesOrderResponseEBS

TransportationSalesOrderResponseEBS is the Enterprise Business Service. This service is used to route all transportation sales order response actions like CreateTransportationSalesOrderResponse, QueryTransportationSalesOrderResponse, SyncTransportationSalesOrderListResponse, UpdateTransportationSalesOrderResponse, and so on.

SyncTransportationSalesOrderListLogisticsProvABCSImpl

SyncTransportationSalesOrderListLogisticsProvABCSImpl is a BPEL process. This process receives SyncTransportationSalesOrderListEBM as a request from TransportationSalesOrderEBS and returns a response to TransportationSalesOrderResponseEBS. This service transforms the EBM to OTM ABM and invokes the LogisticsWebService using the OTM ABM. At the end the XREF table is updated with OTM data. Then a response message is prepared and sent back to TransportationSalesOrderResponseEBS.

Chapter 5: Describing Process Integration for Order Status

This chapter provides an overview of the Process Integration for Order Status, and discusses:

- Integration flows.
- Data requirements.
- Siebel interfaces.
- OTM interfaces.
- Core AIA components.
- Integration services.

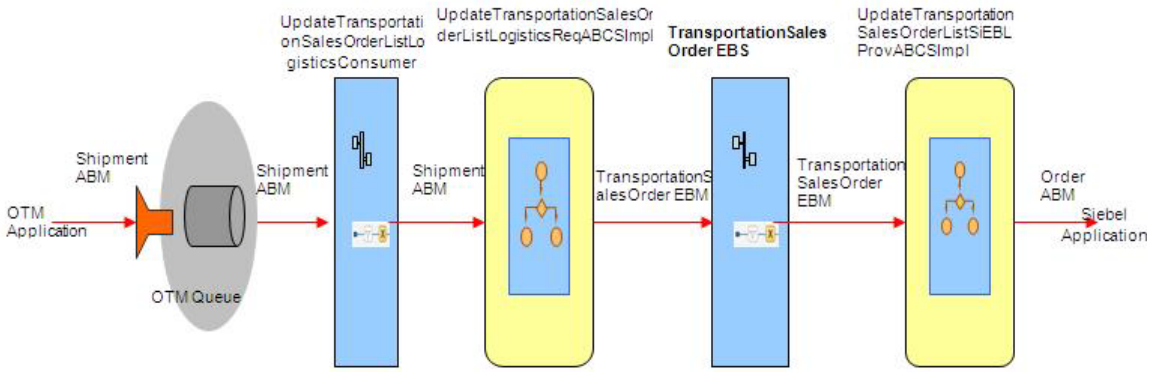
Process Integration for Order Status Overview

As the Order Release, and the Shipments (both buy and sell) get executed in OTM, the status of the Shipment in OTM will change. Also, as Shipments at various stops get Picked-up or Delivered, their actual time get updated in OTM.

OTM will publish these messages to AIA via Automation Agents to be configured by the implementer to synchronize the information to Siebel, so that the CSR/ Sales Agent can communicate the following to the customers:

- Order status
- Actual time of Pickup and Delivery at the Stops.

This diagram shows the overall flow for the Order Status Process Integration:



Order Status Integration flow

This integration assumes that OTM ABM message is sent in a queue inside OTM. A consumer service in AIA reads that message and calls the requestor service. Requestor service transforms the OTM ABM to the UpdateTransportationSalesOrderListEBM, and then the EBM is routed to the TransportationSaleOrderEBS. TransportationSaleOrderEBS service will route that UpdateTransportationSalesOrderListEBM to Siebel provider. Inside the Siebel provider this message will be transformed into Siebel ABM and a Siebel web service will be called to update the Siebel order.

Prerequisites

In OTM an external system must be defined for element “PlannedShipment” for the OTM outbound messages to be published.

Automation agents must be defined in OTM to listen for certain events and trigger the “Planned Shipment” out of OTM upon this event taking place, or when the dates or defined statuses are updated on the buy shipment, and to send the Sell Shipment out when its status is set to BILLED_APPROVED.

The cross-references for the order header and lines are already established through order synchronization prior to this flow.

Solution Assumptions and Constraints

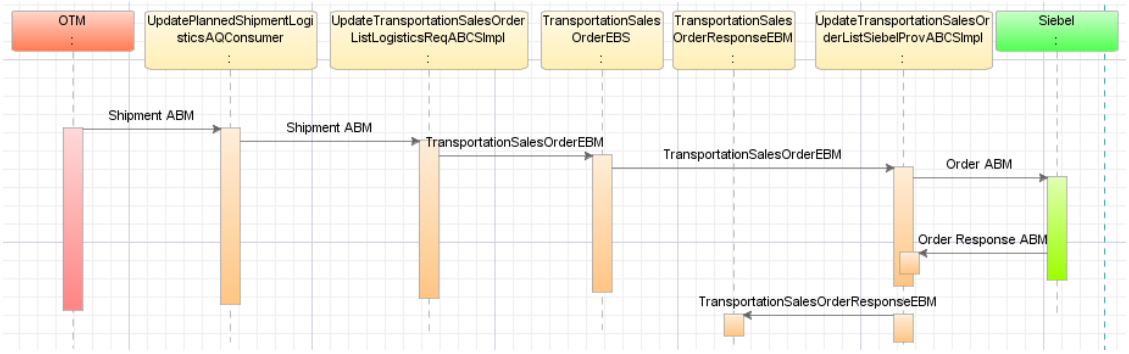
This integration assumes that the following statements are true:

- The order status integration only supports updating of order status in the header level, and date time elements at the stop level. No other data elements are updated in this flow.
- Header level statuses are sent from OTM to Siebel. For stop level, only the date time values are sent, Siebel will derive the statuses on stop level and or any other levels within the application.
- A buy shipment in OTM is necessary to have the status message. Order synchronization flow does not create a buy shipment automatically. The Order synchronization flows create the “Order Release”, and the “Sell Shipment”; the implementers must generate buy shipments in OTM either via agent configuration, or using manual actions.
- For the Siebel “Assigned” and “Moving” statuses message comes from OTM buy shipment, whereas the “Billed” status the message comes from sell shipment.
- The agents required in OTM to send out the messages from buy or sell shipment needs to be defined by the implementers
- This integration will not perform any business validation and thus will not raise errors for those business validation issues.
- In case of multi-leg itinerary in Siebel Order, there will be multiple buy shipments in OTM. Corresponding to each buy shipment, the same order status in Siebel will iterate through Assigned, and Moving values,
- When the buy shipment is created in OTM for the order status message, the user should copy all the reference numbers from the sell shipment.

- In case if any error occurs and the message does not reach the target application then AIA Error Handling Framework will notify the user. The user should manually re-submit that transmission that failed for re-processing.
- OTM will continue to send the message into an Oracle AQ. SOA suite AQ adapter will de-queue those messages in the AIA layer.
- Requests are made from OTM in “Fire and Forget” mode. Hence, OTM doesn’t wait for the response from Siebel.
- OTM will use AQ mechanism to en-queue the outbound shipment messages. The queue will be created within OTM application.
- Inbound Siebel web service requires a user id and password to supply as part of the end point URL.

Order Status Integration Flow

This sequence diagram shows the order status update flow:



Order Status Update Sequence diagram

Data Requirements

The solution design assumes that the following statements are true:

- A buy shipment in OTM is necessary to provide the Assigned and Moving status values to Siebel.
- A Sell Shipment has to be invoiced in OTM to send the Billed status to Siebel.

Siebel Interfaces

The Siebel ABM used for this integration is the Siebel Fleet Order schema.

The interface that is required for updating the order in Siebel is the Siebel Order WSDL.

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.*

Oracle OTM Interfaces

OTM publishes the shipment ABM message in AQ. The element in OTM schema that is used in this process is “PlannedShipment.” This is the only OTM interface used in this process.

Core AIA Components

The integration flow uses the following components:

- TransportationSalesOrderEBO
- UpdateTransportationSalesOrderListEBM

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 will 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.”

Integration Services

These are the services delivered with this integration:

- UpdateTransportationSalesOrderListLogisticsReqABCSImpl
- TransportationSalesOrderEBS
- TransportationSalesOrderResponseEBS

- UpdateTransportationSalesOrderListSiebelProvABCSImpl
- UpdatePlannedShipmentLogisticsAQConsumer

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

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

UpdateTransportationSalesOrderListLogisticsReqABCSImpl

UpdateTransportationSalesOrderListLogisticsReqABCSImpl is a BPEL process. This process receives logistics ABM as input from OTM. This message will be transformed to UpdateTransportationSalesOrderListEBM message and TransportationSalesOrderEBS service is invoked.

TransportationSalesOrderEBS

TransportationSalesOrderEBS is an Enterprise Business Service and is used to create, query, synchronize, and update all transportation sales orders.

TransportationSalesOrderResponseEBS

TransportationSalesOrderResponseEBS is the Enterprise Business Service and is used to create, query, synchronize, and update all the transportation sales order responses.

UpdateTransportationSalesOrderListSiebelProvABCSImpl

UpdateTransportationSalesOrderListSiebelProvABCSImpl is a BPEL process. This process receives the UpdateTransportationSalesOrderListEBM message from TransportationSalesOrderEBS and will be transformed to Siebel Order ABM. The Siebel Web Service updates the order status in Siebel system and returns a success or failure message to the calling service.

UpdatePlannedShipmentLogisticsAQConsumer

This service will be invoked the moment OTM enqueue a shipment message into AIA_TRANSPORTATIONSALESORDER_AQ queue. This service will take the xml element from the wrapper AQ schema, which contains the entire Transmission element as CLOB. This service is designed using ESB and hence it will simply route and invoke UpdateTransportationSalesOrderListLogisticsReqABCSImpl.

Chapter 6: Describing Process Integration for Product

This chapter provides an overview of the Process Integration for Product, and discusses:

- Integration flows.
- Data requirements.
- Siebel interfaces.
- OTM interfaces.
- Core AIA components.
- Integration services.

Process Integration for Product Overview

This chapter provides an overview of the Process Integration for Product and discusses:

In Transportation Order Management PIP, Siebel has four different types of products. They are:

- Commodity
- Transportation
- Accessorial
- Special Services
- Whenever a product is created/updated in Siebel, a synchronization flow is initiated to route these to AIA layer. However, AIA will route only the product type of “Commodity” to OTM.

For every “Commodity” type product from Siebel, the following objects will be created/updated in OTM:

- Item
- Commodity
- Packaged Item

There should be one to one mapping between the Item, Commodity, and Packaged Item in OTM.

- The Transportation Order in Siebel will have certain Order lines that will have Product of type “Commodity” associated to them.
- In OTM application, the Order Release will have “Release Lines,” and “Ship Units” associated to it.
- In the Release Line, the “Packaged Item” is associated to it, whereas in the Release Ship

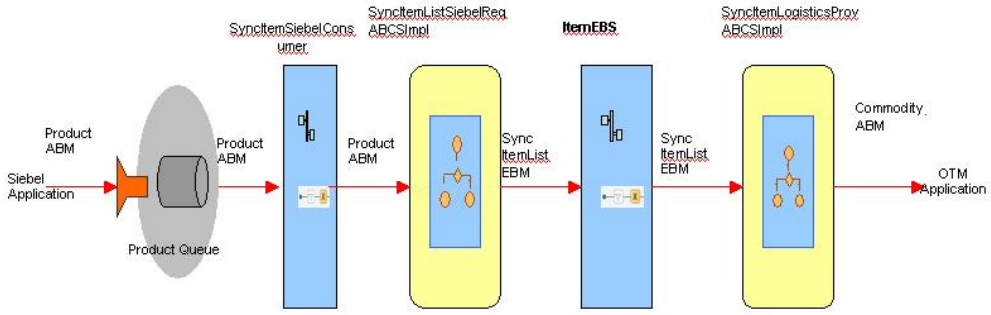
Unit, a “Commodity” is associated.

- Additionally in OTM, the “Packaged Item” will be referred to in the Ship Unit in the Sell Shipment.

The process integration for product supports the following integration flows:

- Creating Product
- Updating Product

This diagram shows the overall flow for Product Process Integration:



Product Process Integration flow

Whenever a product is created/updated in Siebel, Siebel sends a message in their schema in a queue; AIA receives that message and convert it to the EBM format, then convert the EBO into appropriate OTM format and finally send it to OTM

There should be one to one mapping between the Item, Commodity, and Packaged Item in OTM.

Prerequisites

There are no prerequisites for this integration.

Solution Assumptions and Constraints

- Initial loading of existing data of products is not supported in this synchronization.
- There are no delete transactions for Product records.
- This integration synchronizes only products of type ‘COMMODITY’ to the OTM application.
- If any error occurs in the service layer, AIA Error Handling Framework will be invoked to notify the user/s. The user should manually re-submit that transmission at various places that failed for re-processing.
- This integration will not validate and raise errors due to any business validation failure in OTM. It assumes such validations happen in OTM system.
- This integration supports only synchronization of products whenever a product is created or updated.
- OTM domain value will be derived from business unit mapping. If implementers want to use

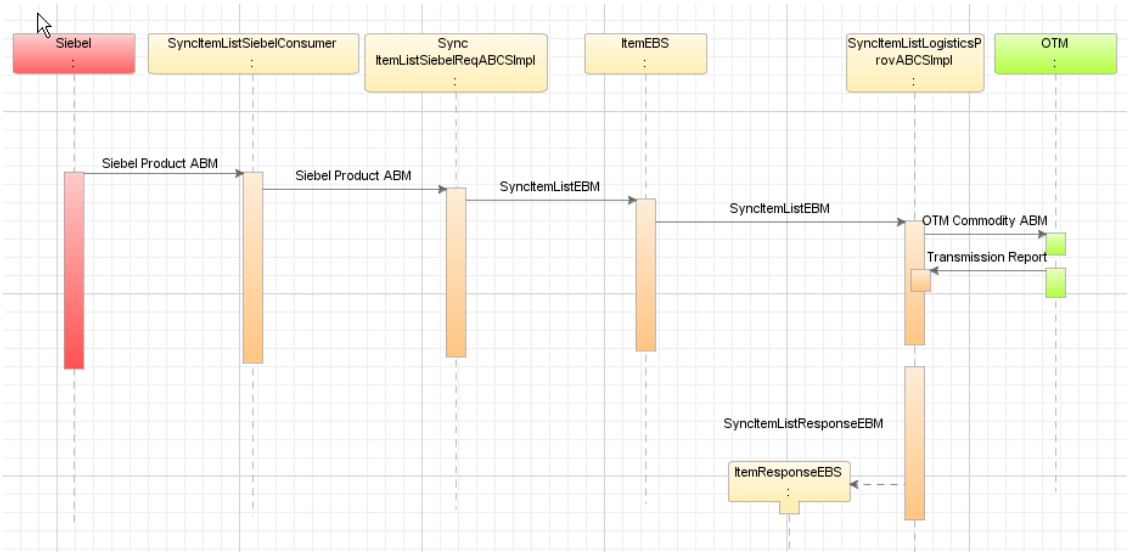
their own logic for domain they must use the extensible transformation template to do so.

- The provider side cross references will be populated based on the identifiers passed from main transformation to OTM. If implementers want to pass a different value in the OTM identifier then transformation's extensibility can be used to update the cross reference.

Synchronizing Product Information

Whenever a product of type "Commodity" is created or updated in Siebel, the updated record has to be synchronized to OTM.

This sequence diagram shows the synchronization of a product details from Siebel to OTM:



Product Synchronization flow

1. Whenever a product is created or updated, Siebel publishes a ProductABM.
2. The Siebel Requester ABCS receives this message, transforms the ABM to EBM, updates the Siebel, and invokes the ItemEBS service.
3. The ItemEBS service routs this message to OTM Provider ABCS.
4. OTM Provider ABCS receives this EBM, checks if the Product type is COMMODITY. If so, it transforms to OTM Commodity ABM and Invokes the OTM web service. It then waits for the Transmission report from OTM.
5. Once the transmission report is received, the OTM Provider ABCS checks the status. If status is OK, it updates the OTM column in the ITEM_ITEMID xref with the Commodity GID.

Data Requirements

There are no data requirements for this integration.

Siebel Interfaces

The schema that is available for this integration from Siebel is Siebel Product Schema.

Oracle OTM Interfaces

The OTM provides an interface through a web service to connect to its application. This connectivity is established as a partner link in the Provider Service. The Logistics WebService immediately returns an acknowledgement with a transmission number. Once the processing is complete, it will then send a Transmission Report back indicating the success or the failure.

Core AIA Components

The integration flow uses the following components:

- Item EBO
- Item EBM

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 will 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.”

Integration Services

These are the services delivered with this integration:

- Siebel Product Queue
- SyncItemSiebelConsumer

- SyncItemListSiebelReqABCSImpl
- ItemEBS
- SyncItemListLogisticsProvABCSImpl

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

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

Siebel Product Queue

Siebel Product Queue is be used to queue Siebel product messages. This queue can reside on any oracle database. For this integration AIA DB is going to be used as the place for this queue.

SyncItemSiebelConsumer

This service will be invoked the moment OTM enqueue a message into AIA_ITEMJMSQUEUE queue. This service routes and invokes SyncItemListSiebelReqABCSImpl process.

SyncItemListSiebelReqABCSImpl

The SyncItemListSiebelReqABCSImpl is a BPEL Process. This process receives the Siebel Product ABM as input from Siebel system, transforms it to the SyncItemListEBM message, and invokes the ItemEBS service. In this transformation in addition to the mapping, the EBM Header and the Xref tables are populated.

ItemEBS

ItemEBS is the Enterprise Business Service to route all Item related operations

SyncItemListLogisticsProvABCSImpl

SyncItemListLogisticsProvABCSImpl is a BPEL Process. This process receives the SyncItemListEBM as input from the ItemEBS, transforms the input into the LogisticsWebServiceABM and invokes the LogisticsWebService. When Logistics sends the Transmission Report and the status in the Transmission Report is success, it updates the XREF tables with OTM Ids.

Chapter 7: Describing Process Integration for Query Transportation Order Itinerary

This chapter provides an overview of the Process Integration for query transportation order Itinerary, and discusses:

- Integration flows.
- Data requirements.
- Oracle E-Business Suite interfaces.
- OTM interfaces.
- Core AIA components.
- Integration services.

Process Integration for Query Transportation Order Itinerary Overview

This integration flow allows the Siebel users to query OTM for various itineraries, schedules, and their rates.

In the Siebel transportation order, the user can query the ratings based on origin and destination location details, pickup and delivery time, and commodity (to be shipped) details. The corresponding OTM itineraries with details will be pulled into Siebel Solution view via AIA. The user could select one of the solutions and update the rate upon negotiation with the customer.

Prerequisites

The prerequisites for Query Transportation Order Itinerary Process Integration are:

- Location Synchronization

Note: If the location is used as an Origin and Destination, then the Role type is mandatory in OTM.

- Product Synchronization

For more information about these process integrations**,** see Chapter 3: Describing the Process Integration for Locations, and Chapter 6: Describing the Process Integration for Products.

Solution Assumptions and Constraints

This integration assumes that the following statements are true:

- The user authentication required for OTM Web Service will also be obtained as an input from the AIA Configuration File as a service level property.
- Exchange Rates required as part of the rate negotiation are manually setup in Siebel, and will not be synchronized as part of this integration.
- Itineraries and Rates have been defined within OTM.
- This query requests only Sell Rates from OTM, and not the buy rates.

Query Transportation Order Itinerary Integration Flow

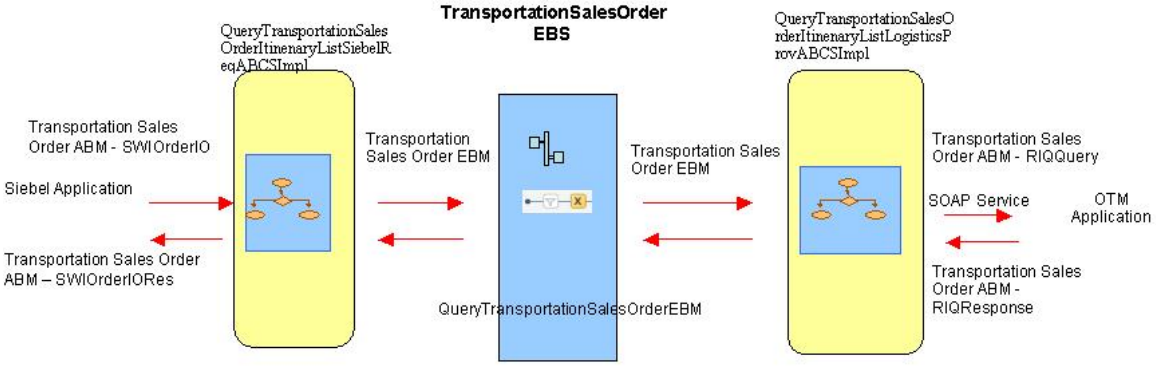
This integration flow supports the following services:

- QueryTransportationSalesOrderItineraryListSiebelReqABCSImpl
- TransportationSalesOrderEBS
- QueryTransportationSalesOrderItineraryListLogisticsProvABCSImpl

Query Transportation Order Itinerary Process

Query Transportation Sales Order process is a synchronous BPEL process. This process gets the itinerary information and rates from OTM and sends the data to Siebel for order confirmation.

This diagram shows the overall flow for Query Transportation Order Itinerary:



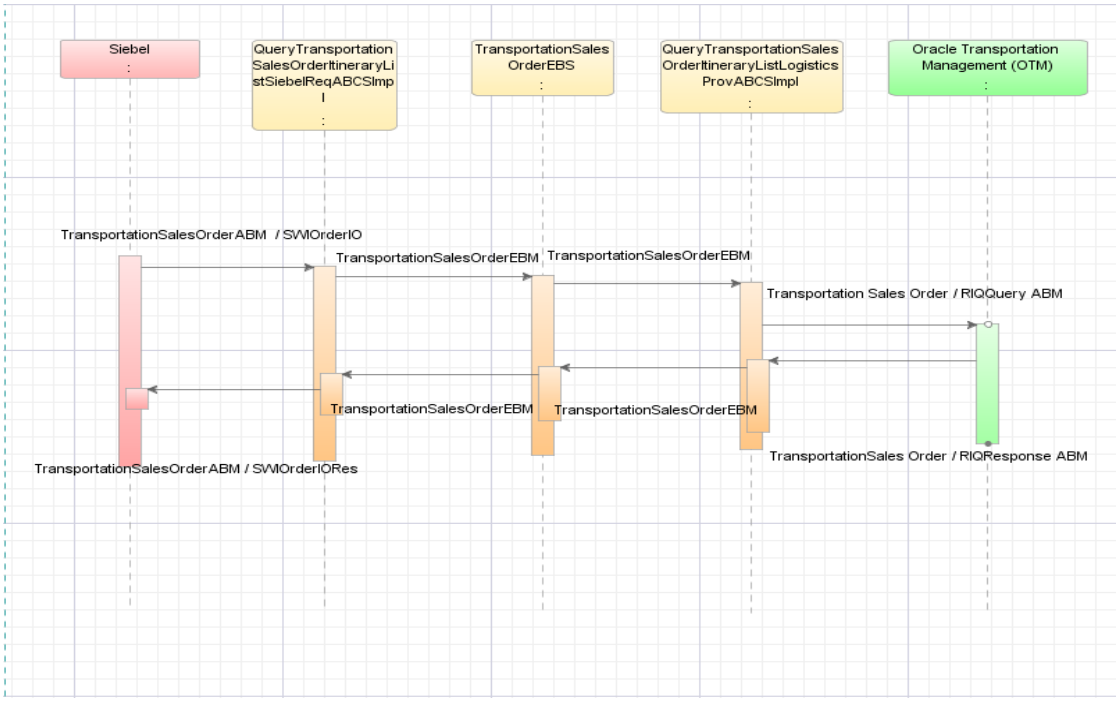
Query Transportation Order Itinerary process flow

1. Siebel Application invokes the QueryTransportationSalesOrderItineraryListSiebelReqABCSImpl with the SWIOrderIO ABM as an input.

6. If the preprocess ABM property is true in the AIA Configuration file, then the QueryTransportationSalesOrderItineraryListSiebelReqABCSImpl service invokes the QueryTransportationSalesOrderSiebelReqABCSImplExt through the PreProcessABM Operation, as a synchronous process, which accepts and replies with the TransportationSalesOrderSiebelABM message.
7. The TransportationSalesOrderSiebelABM message is transformed to the QueryTransportationSalesOrderItineraryListEBM and the EBMHeader is populated.
8. If the PreProcessEBM property is set to true in the AIAConfigFile, the QueryTransportationSalesOrderSiebelReqABCSImplExt will be invoked through the PreProcessEBM Operation, which is a synchronous process, which accepts and replies with the QueryTransportationSalesOrderItineraryListEBM.
9. The QueryTransportationSalesOrderSiebelReqABCSImpl then invokes TransportationSalesOrderEBS .
10. TransportationSalesOrderEBS then in-turn invokes the QueryTransportationSalesOrderItineraryListLogisticsProvABCSImpl.
11. The QueryTransportationSalesOrderItineraryListLogisticsProvABCSImpl in-turn then invokes the Logistics Web Service that returns the Itineraries and Rates.
12. This response is then sent to TransportationSalesOrderEBS
13. The TransportationSalesOrderEBS in-turn returns the response to QueryTransportationSalesOrderItineraryListSiebelReqABCSImpl
14. The QueryTransportationSalesOrderItineraryListSiebelReqABCSImpl returns the response to Siebel after transformation.

Query Transportation Sales Order Itinerary List

This is the sequence diagram for QueryTransportationSalesOrderItineraryList.



Sequence Diagram for QueryTransportationSalesOrderItineraryList

Data Requirements

This integration assumes that the following statements are true:

For the Rating query to get the appropriate data from OTM the Origin and Destination Location, Earliest Pickup time, and Latest Delivery time for the order should be included. Also the user would also need to provide Commodity to be shipped, Number of units to be shipped, Weight, and Volume.

Siebel Interfaces

The following outbound Web Services are configured in Siebel for this flow:

- QueryTransportationSalesOrderItineraryListSiebelReqABCSImpl
- This service takes the SWIOrder as input, and returns the SWIOrderEntrySalesIORes or a Fault.

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.*

Oracle OTM Interfaces

OTM provides an interface through a WebService to connect to its application. This connectivity will be established as a partner link in the Provider Service. The Logistics WebService on being called will immediately return a response.

The 'RIQQuery' element in OTM is used to request rate information for the order to be shipped.

For more information about the Logistics Service, see *Oracle Transportation Management Integration Guide.*

Core AIA Components

The integration flow uses the following components:

- TransportaionSalesOrderEBO
- TransportationSalesOrderEBM

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 will 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."

Integration Services

These are the services delivered with this integration:

- QueryTransportationSalesOrderItineraryListSiebelReqABCSImpl
- TransportationSalesOrderEBS
- QueryTransportationSalesOrderItineraryListLogisticsProvABCSImpl

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

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

QueryTransportationSalesOrderItineraryListSiebelReqABCSImpl

This BPEL process receives the TransportationSalesOrderABM as input from the Siebel Application, transforms the message to the QueryTransportationSalesOrderEBM, and invokes the TransportationSalesOrderEBS service. After transformation, based on the response, the message is again transformed into Siebel ABM and sent to Seibel application.

TransportationSalesOrderEBS

TransportationSalesOrderEBS is an Enterprise Business Service that exposes all the enterprise operations related to the TransportationSalesOrder like Create TransportationSalesOrder, Update TransportationSalesOrder, Delete TransportationSalesOrder etc. For this flow the "QueryTransportationSalesOrderItineraryList" operation is implemented. This will route the request to the appropriate provider like QueryTransportationSalesOrderItineraryListLogisticsProvABCSImpl or CAVS based on the filter condition and operation.

QueryTransportationSalesOrderItineraryListLogisticsProvABCSImpl

This is a Synchronous BPEL process. This process accepts a QueryTransportationSalesOrderItineraryListEBM as input from TransportationSalesOrderEBS transforms this message into TransportationSalesOrderItineraryListOTMABM, and invokes the LogisticsWebService. This service transforms the LogisticsResponseEBM message to QueryTransportationSalesOrderItineraryListResponseEBM, and sends the message back to TransportationSalesOrderEBS.

Chapter 8: Implementing the Order Management Process Integration Pack

This chapter discusses how to:

- Set up participating applications
- Extracting Organizations Data to Setup Cross-References
- Setting Up Organizations
- Setting Up Cross-References for Siebel IDs and Oracle EBS Entities
- Identifying cross-references.
- Describing domain value maps
- Creating Oracle EBS System Profiles
- Handling errors
- Setting Configuration Properties
- EBO Implementation Maps (EIMs)

Setting Up the Participating Applications

Siebel CRM, OTM, and Oracle E-Business Suite must be set up appropriately for the Order Process Integration Pack to work properly. The following sections describe these setups in detail.

Setting Up Organizations

This section describes how to set up organizations in Siebel CRM, Oracle EBS, and Oracle Transportation Management.

Obtaining Oracle EBS Operating Unit IDs

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

To get the Operating Unit details:

1. Login to Oracle EBS database.
2. Identify the Operating Units that need to be synchronized or maintained in Oracle EBS.

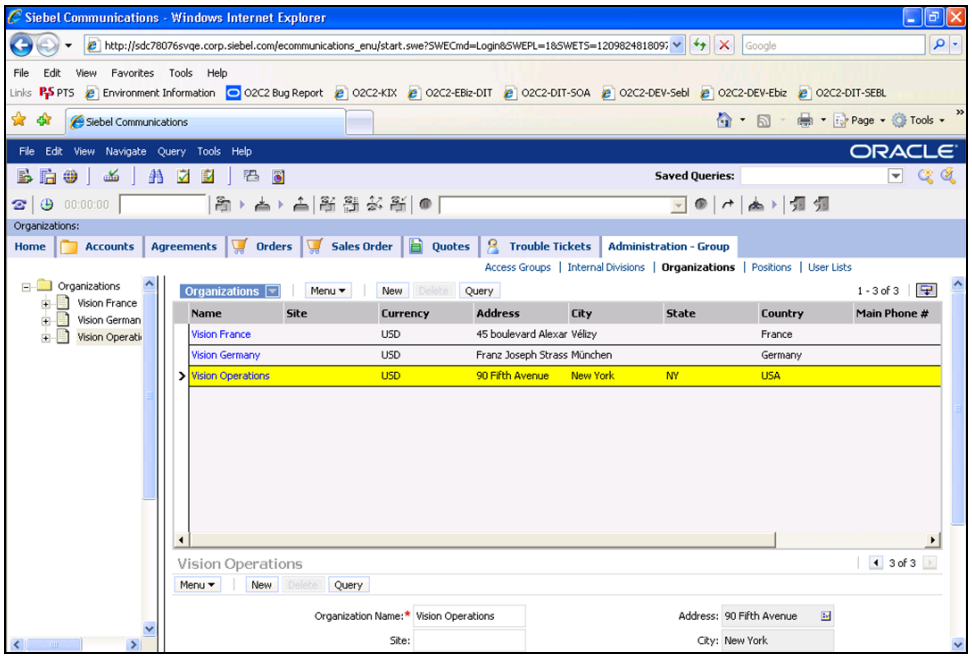
a. If you want to pick other Operating Units, use the following query:

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

Mapping Siebel CRM Organizations

To map Siebel Organizations to EBS Operating Units:

1. Login to Siebel Application.
2. Click Site Map.
3. Select Administration–Groups, Organizations.
4. For the Oracle EBS Operating Units that were identified previously, create the same in Siebel CRM. Here is a screen shot:



"Administration – Group" View tab in Siebel

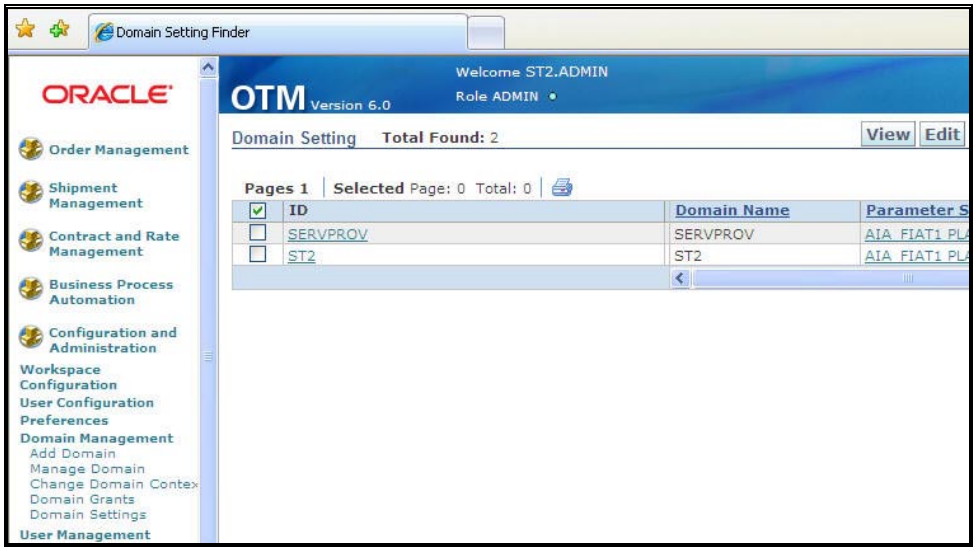
Obtaining Oracle Transportation Management Domains

Implementers need to determine what Domains in OTM have to support.

To get the Domain details:

1. Login to OTM application.
2. Navigate to "Configuration and Administration."
3. Click on "Domain Management."

4. Select “Domain Settings” → Click ‘Search’ button. The screen as in following screenshot appears with Domains setup.

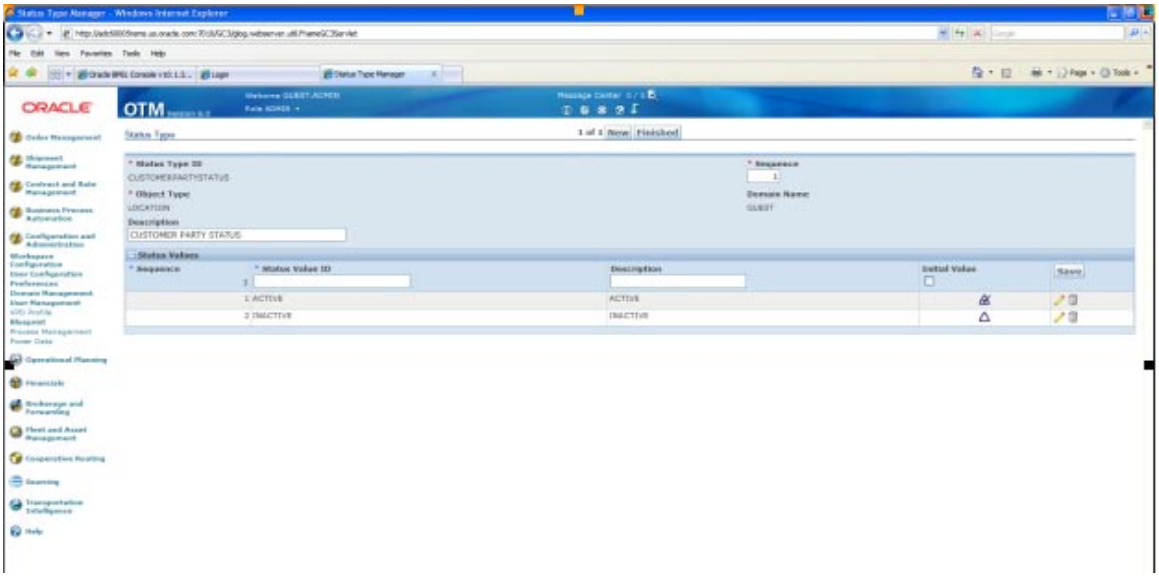


Creating StatusType for Location in Oracle Transportation Management

Implementers need to determine what StatusType in OTM is required. For example, CUSTOMERPARTYSTATUS.

To Create a StatusType:

1. Login to OTM application.
2. Navigate to '"Configuration and Administration”.
3. Click New button.
4. Enter Status Type ID. E.g: CUSTOMERPARTYSTATUS
5. Select 'LOCATION' as Object Type.
6. Enter Sequence 1.
7. Enter two rows ACTIVE and INACTIVE.
8. Check the initial Value for ACTIVE row.



Creating Oracle EBS System Profiles

Specific profile options for the Customer process integrations are set in Oracle EBS.

Creating System Profile Values for the Customer Integration

To set specific profile options for the Customer Management integration:

1. Log in to Oracle EBS using the System Administrator responsibility.
2. Open the System Profile Values form.
3. Query these profile options and set the indicated values at the site level:

For E-Business Suite 11.5.10.2 and 12.1.1:

- HZ: Generate Party Number to 'Yes'
- HZ: Generate Party Site Number to 'Yes'
- Setting Up Cross-References for Siebel IDs, Oracle EBS Entities, and OTM Domains

Cross-references can be created after organizations have been created in Siebel CRM, Operating Unit in Oracle EBS, and domain in OTM.

Identifying Siebel Row IDs

To set up a cross-reference:

1. Log in to the Siebel database as the table owner.
2. Run the following 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 EBS Entities

To get the operating unit details:

1. Log in to Oracle EBS database (Apps/Apps).
2. Identify the operating units that must be synchronized or maintained in Oracle EBS.
3. Log in to Oracle Applications and get the exact name for the operating units. For example:
   a. Vision Operations (204)
   a. Vision Germany (888)
4. To pick other operating units, use the following query:

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

Populating Cross-References

To populate cross-references:

1. Create Organization.xml using the following samples. Update the XML files with the IDs from Siebel CRM.

   Oracle IDs for this will remain the same until different organizations are 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="SEBL_01"/>
<column name="OTM_01"/>
         </columns>
         <rows>
               <row>
                     <cell colName="EBIZ_01">204</cell>
                     <cell colName="SEBL_01">88-25CHZ</cell>
                     <cell colName="OTM_01">ST2</cell>
               </row>

         </rows>
   </table>
</xref>
```

For more information about creating the cross references, see *Oracle Enterprise Service Bus Developer's Guide 10g (10.1.3.4.0),* Creating Cross References.

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

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

5. Set the following env variables:

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

6. Run the import for ORGANIZATION xref using the following command:

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

Setting Up Cross-References for Accessorial and Special Services Products

Cross-references need to be manually maintained for accessorial and special service products that are part of the Orders.

Identifying Siebel Row IDs

To set up a cross-reference:

1. Log in to the Siebel application.
2. Navigate to the Accessorial Product created in Siebel through Administration - Product.
3. Click on the Product and then Menu > About Record, a Row # label will be displayed. This is the Row ID of the Accessorial Product in Siebel.
4. In the same manner, the row id can be determined for the Special Service product as well.

Identifying XREF Row Number

To get the row number for the equivalent product id in Siebel:

1. Log in to AIA XREF database (aia/aia).

2. Identify the xref row number of the accessorial or special service product maintained in Siebel by executing the following query:

```
select row_number from xref_data where value = '<SIEBEL ROW ID>' and
xref_table_name = 'ITEM_ITEMID'
```

3. To pick other row_number, repeat the above query with appropriate value for the siebel row id.

Create the Accessorials/Special Services in OTM

Create the equivalent accessorials and special services in OTM

For more information about creating the accessorials and specials, see *Oracle Transportation Management Guide*.

Populating Cross-References

To populate cross-references:

1. Log in to the AIA XREF database (aia/aia)
2. Now create the corresponding entries for the OTM in XREF table. Siebel and COMMON entries would already be available in the XREF table.
3. Following is an example of how one can create the corresponding OTM entries in XREF manually.
4. Run the following query to insert the xref values for the accessorials created in OTM.

```
insert into XREF_DATA (XREF_TABLE_NAME,
XREF_COLUMN_NAME,ROW_NUMBER,VALUE,IS_DELETED,LAST_MODIFIED,LAST_ACCE
SSED) values ('ITEM_ITEMID','ACCESSORIAL_OTM_01','<row number found
in the section Identifying XREF section>','<otm domain:: otm
accessorial product id'>,'N','1236852128400','1236852128400');

For example to enter the accessorial item "Forklift" present in the
"GUEST" domain in OTM, the query will be
insert into XREF_DATA (XREF_TABLE_NAME,
XREF_COLUMN_NAME,ROW_NUMBER,VALUE,IS_DELETED,LAST_MODIFIED,LAST_ACCE
SSED) values ('ITEM_ITEMID','ACCESSORIAL_OTM_01','
F40CCE8024C911DE8F559994B8D73F6F','GUEST::
Forklift','N','1236852128400','1236852128400');
```

For more information about creating the cross references, see *Oracle Enterprise Service Bus Developer's Guide 10g (10.1.3.4.0),* Creating Cross References.

5. Run the following query to insert the xref values for the special services created in OTM.

```
insert into XREF_DATA
```

```
(XREF_TABLE_NAME,XREF_COLUMN_NAME,ROW_NUMBER,VALUE,IS_DELETED,LAST_M
ODIFIED,LAST_ACCESSED) values ('ITEM_ITEMID','
SPECIALSERVICE_OTM_01','<row number >','<otm domain:: otm special
service product id','N','1236852128400','1236852128400');

For example to enter the special service “Customer Unload” present
in the “GUEST” domain in OTM, the query will be
insert into XREF_DATA
(XREF_TABLE_NAME,XREF_COLUMN_NAME,ROW_NUMBER,VALUE,IS_DELETED,LAST_M
ODIFIED,LAST_ACCESSED) values ('ITEM_ITEMID',
SPECIALSERVICE_OTM_01,' F40CCE8024C911DE8F559994B8D73F6F','GUEST::
Customer Unload','N','1236852128400','1236852128400');
```

Identifying Cross-References

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

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

This table lists the Order Management cross-references:

Name	Description	Columns	Mapping Details
CUSTOMERPARTY_PARTYID	BillToPartyReference	SEBL_01, COMMON, OTM_01, EBIZ_01	Common column value is a GUID generated by BPEL. OTM column value is concatenation of Domain and OTM Identifier with "::" separating them.
CUSTOMERPARTY_ACCOUNTID	BillToPartyReference/AccountID	SEBL_01, COMMON, OTM_01, EBIZ_01	Common column value is a GUID generated by BPEL. OTM column value is concatenation of Domain and OTM Identifier with "::" separating them.
CUSTOMERPARTY_LOCATIONREFID	BillToPartyReference/LocationReference	SEBL_01, COMMON, OTM_01, EBIZ_01	Common column value is a GUID generated by BPEL. OTM column value is concatenation of Domain and OTM Identifier with "::" separating them.
CUSTOMERPARTY_ADDRESSID	BillToPartyReference/Address	SEBL_01, COMMON, OTM_01, EBIZ_01	Common column value is a GUID generated by BPEL. OTM column value is concatenation of Domain and OTM Identifier with "::" separating them.

Name	Description	Columns	Mapping Details
CUSTOMERPARTY_CONTACTID		SEBL_01, COMMON, OTM_01, EBIZ_01	Common column value is a GUID generated by BPEL. OTM column value is concatenation of Domain and OTM Identifier with "::" separating them.
CUSTOMERPARTY_CONTACT_PHONECOMMID		SEBL_01, COMMON, OTM_01, EBIZ_01	Common column value is a GUID generated by BPEL. OTM column value is concatenation of Domain and OTM Identifier with "::" separating them.
CUSTOMERPARTY_CONTACT_FAXCOMMID		SEBL_01, COMMON, OTM_01, EBIZ_01	Common column value is a GUID generated by BPEL. OTM column value is concatenation of Domain and OTM Identifier with "::" separating them.
CUSTOMERPARTY_CONTACT_EMAILCOMMID		SEBL_01, COMMON, OTM_01, EBIZ_01	Common column value is a GUID generated by BPEL. OTM column value is concatenation of Domain and OTM Identifier with "::" separating them.
ORGANIZATION_ID	This XREF will be used to map the Siebel ORG_UNIT to OTM Domain and EBIZ ORG_UNIT.	SEBL_01, COMMON, OTM_01, EBIZ_01	
TRANSPORTATIONSTOP_ACCOUNTID		SEBL_01, COMMON, OTM_01	Common is GUID generated by BPEL. Xref value for OTM is TransportationStopCustomerParty/Identification/ApplicationObjectID
TRANSPORTATIONSTOP_ID		SEBL_01, COMMON, OTM_01	Common is GUID generated by BPEL. Xref value for OTM is concatenation of /LocationGID/GID/DomainName & LocationGID/GID/Xid
TRANSPORTATIONSTOP_CONTACTID		SEBL_01, COMMON, OTM_01	Common is GUID generated by BPEL. Xref value for OTM is Location/ContactGid/Gid.
TRANSPORTATIONSTOP_ROLEID		SEBL_01, COMMON, OTM_01	TransportationStopUsage/Identification/ApplicationObjectID

Name	Description	Columns	Mapping Details
ORGANIZATION_ID			Location/LocationGID/GID/DomainName
TRANSPORTATIONSALESORDER_ID	Header cross reference	SEBL_01, COMMON, OTM_01_RELEASE, OTM_01_SHIPMENT	Siebel ID is populated in the SEBL_01, Common is GUID generated by BPEL. Xref value for OTM columns is concatenation of OTM Domain name, Order number coming from Siebel along with the corresponding COMMON column GUID value.
TRANSPORTATIONSALESORDER_LINEID	Order Line cross reference	SEBL_01, COMMON, OTM_01_RELLINE, OTM_01_RELSHIPUNIT, OTM_01_SELLSHIPUNIT	Siebel LineID is populated in the SEBL_01, Common is GUID generated by BPEL. Xref value for OTM columns is concatenation of OTM Domain name, hard coded string 'TSOL', and the corresponding COMMON column GUID value
TRANSPORTATIONSALESORDER_STOPID	Order Stop cross reference	SEBL_01, COMMON, OTM_01_RELEASE, OTM_01_SHIPMENT	Siebel StopID is populated in the SEBL_01, Common is GUID generated by BPEL. Xref value for OTM columns is concatenation of OTM Domain name, hard coded string 'TSOS', and the corresponding COMMON column GUID value
TRANSPORTATIONSALESORDER_STOPACTIONID	Stop Action Cross reference	SEBL_01, COMMON, OTM_01_RELEASE	Siebel StopActionID is populated in the SEBL_01, Common is GUID generated by BPEL. No value for OTM column is populated.
TRANSPORTATIONSALESORDER_STOPLINEID	Line and Stop association cross reference	SEBL_01, COMMON, OTM_01_RELEASE	Siebel Line Stop ID will be populated in the SEBL_01, Common will be a GUID generated by BPEL. Xref value for OTM columns is concatenation of OTM Domain name, hard coded string 'TSOS', and the corresponding COMMON column GUID value

Name	Description	Columns	Mapping Details
TRANSPORTATIONSALESORDER_ID	To determine the Siebel Order Id based on Common or OTM Release ID	SEBL_01, COMMON, OTM_01_RELEASE, OTM_01_SHIPMENT	This is used only for look up purpose.
TRANSPORTATIONSTOP_ID		SEBL_01, COMMON, OTM_01	This is used only for look up purpose.
TRANSPORTATIONSALESORDER_STOPID		SEBL_01, COMMON, OTM_01_RELEASE, OTM_01_SHIPMENT	This is used only for look up purpose
ITEM_ID		SEBL_01, COMMON, OTM_01, ACCESSORIAL_OTM_01, SPECIALSERVICE_OTM_01	Concatenation of DomainName and ComodityXID for OTM_01. Siebel ROW ID for SEBL_01

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.”

These are the domain value mappings (DVMs) for the customer process flow

Name	Description	Columns	Mapping Details
LOCATION_ROLE	Ship To, Bill To etc	COMMON, OTM_01	
STATE	State code	COMMON, Siebel, OTM_01	
ADDRESS_COUNTRYID		COMMON, Siebel, OTM_01	
CONTACT_SALUTATION	Mr., Mrs., etc	COMMON, Siebel, OTM_01	
COMMUNICATION_METHOD	Fax, Phone etc.	COMMON, OTM_01	

Note: Customer flow reuses several components from Customer Hub MDM PIP. Those components use some more DVMs. For a list of those DVMs please refer to the implementation guide of the Customer Hub MDM PIP

These are the domain value mappings (DVMs) for the location process flow:

Name	Description	Columns	Mapping Details
LOCATION_ROLE	Ship To, Bill To etc	COMMON, OTM_01	
STATE	State code	Siebel, COMMON, OTM_01	
ADDRESS_COUNTRYID	Country code	Seibel, COMMON, OTM_01	
TRANSPORTATIONSTOP_TYPE	Transportation StopType	Seibel, COMMON, OTM_01	
TIMEZONE_ID	Time zone description	Seibel, COMMON, OTM_01	
CONTACT_SALUTATION	Salutation of a contact	Siebel, COMMON, OTM_01	
PHONENUMBER_TYPE	Types of Phone number		
COMMUNICATION_METHOD	Phone, Fax etc	Siebel, COMMON, OTM_01	

These are the domain value mappings (DVMs) for the order process flow:

Name	Description	Columns	Mapping Details
CURRENCY_CODE	Used to convert the Siebel Currency Data to OTM Currency Data	SEBL_01,COMMON, OTM_01	
TRANSPORTATIONSALESORDER_ORDERLINEITEMPROPERTY	Relates the Siebel Items to the OTM Items	SEBL_01,COMMON, OTM_01	
TRANSPORTATIONSALESORDER_PRODUCTTYPECODE	Relates the type of Product at the Line Item level, whether a Accessorial product or Transportation product or Commodity	SEBL_01,COMMON, OTM_01	
ORDER_STATUSCODE	Relates the status of the order whether it is accepted or rejected	SEBL_01,COMMON, OTM_01	
TRANSPORTATIONSALESORDER_STOPACTIONTYPE	Relates the Stop action types	SEBL_01,COMMON, OTM_01	

Name	Description	Columns	Mapping Details
UNIT_OF_MEASURE	Relates the Volume, Weight and Distance units compatible to OTM	SEBL_01,COMMON, OTM_01	
TRANSPORTSALESORDER_PACKAGING	Relates the Packaging of the Items	SEBL_01,COMMON, OTM_01	
TRANSPORTSALESORDER_TYPE	Relates the CarrierRouteStopPickUp or DropOff	SEBL_01,COMMON, OTM_01	
ORDER_DIVISION	Siebel Division Code equivalent	SEBL_01,COMMON, OTM_01	
TRANSPORT_MODE	Transport Mode DVM	SEBL_01,COMMON, OTM_01	
EQUIPMENT_TYPE	Equipment Type DVM	SEBL_01,COMMON, OTM_01	
SHIP_DEVICE	Shipping Device DVM	SEBL_01,COMMON, OTM_01	
STOP_TYPE	Type of Stops	SEBL_01,COMMON, OTM_01	

These are the domain value mappings (DVMs) for the order status process flow:

Name	Description	Columns	Mapping Details
ORDER_STATUS	Derive the status type value for one system	COMMON, OTM_01, Siebel_01	
TRANSPORTATIONSALESORDER_STATUSCODE	Derive the status values for one system	COMMON, OTM_01, Siebel_01	

These are the domain value mappings (DVMs) for the product process flow:

Name	Description	Columns	Mapping Details
ITEM_INDICATOR	Item Indicator	Siebel, COMMON, OTM	
PRODUCT_STATUS	Product Status	Siebel, COMMON, OTM	
UNIT_OF_MEASURE	Unit of Measure	Siebel, COMMON, OTM	
ITEM_TYPE.xml			

These are the domain value mappings (DVMs) for the query transportation sales order process flow:

Name	Description	Columns	Mapping Details
Country Code in Address (ADDRESS_COUNTRYID)	Used to identify the Country codes in the Address	SEBL_01,COMMON, OTM_01	
State Code in Address (STATE)	Used to identify the State Code in Destination and Origin Location	SEBL_01,COMMON, OTM_01	
UNIT_OF_MEASURE	Relates the Weight Units, Volume units and Distance Units compatible to OTM	SEBL_01,COMMON, OTM_01	
TRANSPORTATIONSTOP_TYPE	TransportationStopType	EBL_01,COMMON, OTM_01	
CURRENCY_CODE	Currency codes	SEBL_01,COMMON, OTM_01	
FEASIBLE_FLAG	Relates the Feasible Solution for the Itineraries time wise	SEBL_01,COMMON, OTM_01	
TRANSPORTATIONSALESORDER_TRANSPORTMODE	Relates mode of Transport for the commodities	SEBL_01,COMMON, OTM_01	Example: Rail, Road, and Truck.
ITEM_TYPE	Relates the type of Costs associated with the itineraries	SEBL_01,COMMON, OTM_01	

Handling Errors

There are no business errors captured for Transportation Order Management PIP.

Setting Configuration Properties

The table lists the properties that need to be set in the configuration file.

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. You can perform this reload by clicking the Reload button on the Configuration page in the Oracle AIA Console. 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 SyncCustomerPartyListLogisticsProvABCSImpl service

Property Name	Value/Default Value	Description
Default.SystemID	OTM_01	Based on the SenderHostName obtained from ABM, sender SystemID is derived, but if that value is not available in ABM, AIA reads it from the config file using this property.
Routing.LogisticsWebService.RouteToCAVS	True/False, Default=False	Determines whether the EndpointURI should be routed either to the end application service or CAVS for simulating the service.
Routing.LogisticsWebService.OTM_01.EndpointURI		This property is used to derive the EndpointURI for the target application.
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyList.CAVS.EndpointURI		This property is used to determine the end point URI when the response message should be routed to CAVS.
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyList.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTION	Sets the Response EBM message header EnvironmentCode element to the value depending on what is mentioned here.
OTM_01.USERNAME		Property specifies the OTM instance Username.
OTM_01.PASSWORD		Property specifies the OTM instance password
LogisticsWebService.LanguageCode		This property is used for checking the LanguageCode coming from requestor. If that code matches with the acceptable language code of OTM, then the processing moves on forward If the language codes don't match, the process is terminated.
CallBackURL		Property specifies the URL used by OTM to return the response.
ABCSExtension.PreProcessABM	True/False Default=False	This property sets an extension point before EBM is transformed to ABM. It determines whether a service has to be invoked or not based on its value.
ABCSExtension.PreProcessEBM	True/False Default=False	This property sets an extension point before EBM is transformed to ABM. It determines whether a service has to be invoked or not based on its value.
ABCSExtension.PostProcessABM	True/False Default=False	This property is used as an extension point after EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
ABCSExtension.PostProcessEBM	True/False Default=False	This property sets an extension point before EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
Routing.LogisticsWebService.CAVS.EndpointURI		This property sets the EndpointURI for the CAVS simulator.

Property Name	Value/Default Value	Description
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyList.RouteToCAVS	True/False Default=False	Determines whether the response message from the provider application should be sent to the requestor application or to CAVS.
Transformation.EnableExtensions	True/False Default=False	This property determines enabling extensions in the transformations based on customer requirements.
Default.ComMethod	FAX	This property is used to read the default communication method of a contact for a location.
Default.CustomerPartyStatusType	CUSTOMERPARTYSTATUS	Determines the StatusType value to be used for the Location in OTM.
OTM_01.ISPASSWORDENCRYPTED	True/False Default=false	Property specifies whether the OTM password is encrypted in the Configuration Properties file. If true, DecryptionService is being called to decode the OTM password.

For more information about Customer Hub MDM service related configuration properties, please refer to the Customer Hub MDM PIP implementation guide.

Settings for SyncTransportationStopListLogisticsProvABCSImpl service:

Property Name	Value/Default Value	Description
ABCSExtension.PostProcessABM	True/False Default=False	This property is used as an extension point after EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
ABCSExtension.PostProcessEBM	True/False Default=False	This property sets an extension point before EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
Default.SystemID	OTM_01	Sender SystemId is obtained from ABM to be used in EBMHeader. If it is not available in ABM, AIA reads it from the config file using this property.
Routing.LogisticsWebService.RouteToCAVS	True/False Default=False	Determines whether the EndpointURI should be routed either to the end application service or CAVS for simulating the service.
Routing.LogisticsWebService.OTM_01.EndpointURI		This property sets the EndpointURI for the target application.
Routing.LogisticsWebService.CAVS.EndpointURI		This property defines the Endpoint URI for the CAVS simulator.

Property Name	Value/Default Value	Description
Routing.TransportationStopResponseEBS.SyncTransportationStopListResponse.RouteToCAVS	True/False Default=False	This property determines whether the response message should be routed to CAVS or not.
Routing.TransportationStopResponseEBS.SyncTransportationStopListResponse.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTION	Sets the response message EBM header Environment Code element to this value
Routing.TransportationStopResponseEBS.SyncTransportationStopListResponse.CAVS.EndpointURI		Defines the CAVS EndpointURI for response message.
OTM_01.USERNAME		Property specifies the OTM instance Username.
OTM_01.PASSWORD		Property specifies the OTM instance Password. These are the security credentials for OTM.
ABCSExtension.PreInvokeABS	True/False Default=False	This property is used as an extension point after EBM to ABM transformation and before invoking the target. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
ABCSExtension.PreXformEBMtoABM	True/False Default=False	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
Transformation.EnableExtensions	True/False Default=False	Property used to determine enabling extensions in the Transformations based on the customer requirements.
LogisticsWebService.LanguageCode	ENU	This property indicates the language code for the OTM system. It is used for matching the LanguageCode coming from requestor. If the language code doesn't match, the process is terminated.
OTM_01.ISPASSWORDENCRYPTED	True/False Default=false	Property specifies whether the OTM password is encrypted in the Configuration Properties file. If true, DecryptionService is being called to decode the OTM password.
CallBackURL		Property specifies the URL used by OTM to return the response.
Default.ComMethod	FAX	This property is used to read the default communication method for a location contact.
Default.SystemID	OTM_01	Sender SystemId is obtained from EBMHeader, but if it is empty AIA reads it from the config file using this property.

Property Name	Value/Default Value	Description
Routing.LogisticsWebService.RouteToCAVS	True/False	Determines whether the EndpointURI should be routed either to the end application service or CAVS for simulating the service.
Routing.LogisticsWebService.OTM_01.EndpointURI		This property sets the EndpointURI for the target application.
Routing.LogisticsWebService.CAVS.EndpointURI		This property sets the EndpointURI for the CAVS simulator.
Routing.TransportationStopResponseEBS.SyncTransportationStopListResponse.RouteToCAVS	True/False	This property determines whether the response message should be routed to CAVS or not.
Routing.TransportationStopResponseEBS.SyncTransportationStopListResponse.MessageProcessingInstruction.EnvironmentCode	CAVS	Sets the EnvironmentCode element to the value PRODUCTION.
Routing.TransportationStopResponseEBS.SyncTransportationStopListResponse.CAVS.EndpointURI		Defines the CAVS EndpointURI.
OTM_01.USERNAME		Property specifies the OTM instance Username.
OTM_01.PASSWORD		Property specifies the OTM instance Password. These are the security credentials for OTM.
ABCSExtension.PreInvokeABS	True/False	This property is used as an extension point after EBM to ABM transformation and before invoking the target. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
ABCSExtension.PreXformEBMtoABM	True/False	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
Transformation.EnableExtensions	False	Property used to determine enabling extensions in the Transformations based on the customer requirements.
LogisticsWebService.LanguageCode	ENG	This property is used for checking the LanguageCode coming from Seibel matches with the acceptable code of OTM. If the language code doesn't match, the process is terminated.
Default.ComMethod	FAX	This property is used to set the default communication method.

Settings for SyncTransportationStopContactSiebelAggregatorAdapterConsumer Service

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	It is the responsibility of the application to send the SystemID from which the request is being sent. If any requestor application fails to send this, AIA will pick the default SystemID from this property.
Routing.SiebelWebService.RouteToCAVS	True/False Default=False	If this property value is set to false, then invokes the Siebel web service endpoint WSDL. If the property value is set to true then it invokes simulator.
Routing.SiebelWebService.CAVS.EndpointURI		This holds the URI of CAVS simulator where the aggregator should send the request.
Routing.SiebelWebService.SEBL_01.EndpointURI		This holds the URI of Siebel webs service where the aggregator should send the request.

Settings for SyncTransportationStopAddressSiebelAggregatorAdapter Service

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	It is the responsibility of the application to send the SystemID from which the request is being sent. If any requestor application fails to send this, AIA will pick the default SystemID from this config property.
Routing.SiebelWebService.RouteToCAVS	True/false Default=False	If this property value is set to false, then the service invokes the Siebel web service endpoint WSDL. If the property value is set to true then it invokes simulator.
Routing.SiebelWebService.CAVS.EndpointURI		This holds the URI of CAVS simulator where the aggregator should send the request.
Routing.SiebelWebService.SEBL_01.EndpointURI		This holds the URI of Siebel web service where the aggregator should send the request.

Settings for SyncTransportationStopListSiebelReqABCSImpl service

Property Name	Value/Default Value	Description
Default.SystemID	SEBL_01	Based on the SenderHostName obtained from ABM, sender SystemID is derived. If ABM sends that as empty, AIA reads it from the config file using this property.
ABCSExtension.PreXformABMtoEBM	True/False Default=False	This property is used as an extension point before ABM is transformed to EBM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false
Routing.TransportationStopEBS.SyncTransportation	True/False Default=False	This property is used to determine whether to route the

Property Name	Value/Default Value	Description
StopList.RouteToCAVS		request to CAVS
Routing.TransportationStopEBS.SyncTransportationStopList.CAVS.EndpointURI		This property is used to get End Point URI when "Routing.TransportationStopEBS. SyncTransportationStopList.RouteToCAVS" is true
Routing.TransportationStopEBS.SyncTransportationStopList.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTION default=PRODUCTION	This property is used to set the environment code in the EBM header of the request message.
Transformation.EnableExtensions	True/False Default=False	This property should be set to True, when customers want to customize the attribute mapping done in XSL.
ABCSExtension.PreInvokeEBS	True/False Default=False	This property is used as an extension point after ABM to EBM transformation and before Invoking the EBS. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false

Settings for SyncTransportationSalesOrderListLogisticsProvABCSImpl service

Property Name	Value/Default Value	Description
ABCSExtension.PostProcessABM	True/False Default=False	This property is used as an extension point after EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
ABCSExtension.PostProcessEBM	True/False Default=False	This property sets an extension point before EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
Default.SystemID	OTM_01	Based on the SenderHostName obtained from ABM, sender SystemID is derived, but if it is empty, AIA reads it from the config file using this property.
Routing.LogisticsWebService.RouteToCAVS	True/False Default=False	Determines whether the EndpointURI should be routed either to the end application service or CAVS for simulating the service.
Routing.LogisticsWebService.OTM_01.EndpointURI		This property defines the EndpointURI for the target application.
Routing.LogisticsWebService.CAVS.EndpointURI		This property defines the EndpointURI for the CAVS simulator.
Routing.TransportationSalesOrderResponseEBS.SyncTransportationSalesOrderList.RouteToCAVS		This property determines whether the response message needs to be routed to CAVS or not.

Property Name	Value/Default Value	Description
Routing.TransportationSalesOrderResponseEBS.SyncTransportationSalesOrderList.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTION Default=PRODUCTION	This property value is used to set the EnvironmentCode element of the Response message.
Routing.TransportationSalesOrderResponseEBS.SyncTransportationSalesOrderList.CAVS.EndpointURI		Defines the CAVS EndpointURI for the response message.
OTM_01.USERNAME		Property specifies the OTM instance Username.
OTM_01.PASSWORD		Property specifies the OTM instance Password. These are the security credentials for OTM.
ABCSExtension.PreInvokeABS	True/False Default=False	This property is used as an extension point after EBM to ABM transformation and before invoking the target end point application for extensibility of ABM message. It determines invocation of extension point is to be made or not depending on whether it is true or false.
ABCSExtension.PreXformEBMtoABM	True/False Default=False	This property is used as an extension point before EBM is transformed to ABM for extensibility of EBM message. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
Transformation.EnableExtensions	True/False Default=False	Used for determining enabling extensions in the Transformations based on customer requirements.
LogisticsWebService.LanguageCode	ENU	This property indicates the language code for the OTM system. It is used for matching the LanguageCode coming from requestor. If the language code doesn't match, the process is terminated.
CallBackURL		Property specifies the URL used by OTM to return the response.
OTM_01.ISPASSWORDENCRYPTED	True/False Default=False	Property specifies whether the OTM password is encrypted in the Configuration Properties file. If true, DecryptionService is being called to decode the OTM password.

Settings for SyncTransportationSalesOrderListSiebelReqABCSImpl service

Property Name	Value/Default Value	Description
ABCSExtension.PostProcessABM	True/False Default=False	This property is used as an extension point after EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.

Property Name	Value/Default Value	Description
ABCSExtension.PostProcessEBM	True/False Default=False	This property sets an extension point before EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
Default.SystemID	SEBL_01	Based on the SenderHostName obtained from ABM, sender SystemID is derived, but if it is empty, AIA reads it from the config file using this property.
ABCSExtension.PreXformABMtoEBM	True/False Default=False	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
Routing.TransportationSalesOrderEBS.SyncTransportationSalesOrderList.RouteToCAVS	True/False Default=False	This property determines whether the message should be routed to CAVS or not.
Routing.TransportationSalesOrderEBS.SyncTransportationSalesOrderList.CAVS.EndpointURI		This property defines the CAVS Endpoint
Routing.TransportationSalesOrderEBS.SyncTransportationSalesOrderList.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTION Default=PRODUCTION	This property is used to set the EnvironmentCode element in the request message
ABCSExtension.PreInvokeEBS	True/false. Default=false.	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
Routing.SWIOrderUpsertRef.SEBL_01.EndpointURI		Property specifies the URL for routing to the Siebel application

Settings for UpdateTransportationSalesOrderListSiebelProvABCSImpl service

Property Name	Value/Default Value	Description
ABCSExtension.PostProcessABM	True/False Default=False	This property is used as an extension point after EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
ABCSExtension.PostProcessEBM	True/False Default=False	This property sets an extension point before EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
Default.SystemID	SEBL_01	Based on the SenderHostName obtained from ABM, sender SystemID is derived and set in EBMHeader. If it is

Property Name	Value/Default Value	Description
		empty, AIA reads it from the config file using this property.
Routing.SiebelWebService.RouteToCAVS	True/False Default=False	Determines whether the EndpointURI should be routed either to the end application service or CAVS for simulating the service.
Routing.SiebelWebService.SEBL_01.EndpointURI		This property defines the EndpointURI for the target application.
Routing.SiebelWebService.CAVS.EndpointURI		This property defines the EndpointURI for the CAVS simulator.
Routing.TransportationSalesOrderResponseEBS.UpdateTransportationSalesOrderListResponse.RouteToCAVS	True/False Default=False	This property determines whether the response message needs to be routed to CAVS or not.
Routing.TransportationSalesOrderResponseEBS.UpdateTransportationSalesOrderListResponse.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTION Default=PRODUCTION	This property is used to set the environment code element of the response message.
Routing.TransportationSalesOrderResponseEBS.UpdateTransportationSalesOrderListResponse.CAVS.EndpointURI		Defines the CAVS EndpointURI for response message.
ABCSExtension.PreInvokeABS	True/False Default=False	This property is used as an extension point after EBM to ABM transformation and before invoking the target. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
ABCSExtension.PreXformEBMtoABM	True/False Default=False	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
Transformation.EnableExtensions	True/False Default=False	Property used to determine enabling extensions in the Transformations based on customer requirements.

Settings for UpdateTransportationSalesOrderListLogisticsReqABCSImpl service

Property Name	Value/Default Value	Description
Default.SystemID	OTM_01	Sender SystemId is obtained from ABM and set into EBMHeader. If it is empty, AIA reads it from the config file using this property.
ABCSExtension.PreXformABMtoEBM	True/False Default=False	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on

Property Name	Value/Default Value	Description
		whether it is true or false
Routing.TransportationSalesOrderEBSV1.UpdateTransportationSalesOrderList.CAVS.EndpointURI	True/False Default=False	This property defines the end point URI of CAVS
Routing.TransportationSalesOrderEBSV1.UpdateTransportationSalesOrderList.RouteToCAVS	True/False Default=False	EnvironmentCode in the Header population will be derived based on this value. If this property value is set to true then the EnviromentCode value will be set to 'CAVS' and if the property value is not set, then the environment code is set to 'Production' by default.
Routing.TransportationSalesOrderEBSV1.UpdateTransportationSalesOrderList.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTTION Default=PRODUCTION	This property is used to set the Environment Code element of the request message.
ABCSExtension.PreInvokeEBS	True/false. Default = false.	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
Transformation.EnableExtensions	True/False Default=False	This property should be set to true when customers want to customize the attribute mapping done in xsl.
Default.Languages	ENG	This property is used to set the default language of OTM

Settings for SyncItemListSiebelReqABCSImpl service

Property Name	Property Value	Description
Default.SystemID	SEBL_01	Based on the SenderHostName obtained from ABM, sender SystemID is derived, but if it is empty, AIA reads it from the config file using this property.
ABCSExtension. PreXformABMtoEBM	True/False, Default = False	This property is used as an extension point before the ABM is transformed to EBM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false
ABCSExtension. PreInvokeEBS	True/False	This property is used as an extension point after the ABM is transformed to EBM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false
Transformation.EnableExtensions	True/False	Property used for determining enabling XSL 2.0 extensions in the Transformations.
Routing.ItemEBSV2.SyncItemList.RouteToCAVS	True/False	Determines whether the EndpointURI should be routed either to the end application service or CAVS for simulating the service.
Routing.ItemEBSV2.SyncIt	PRODUCTION/CAVS	Sets the EnvironmentCode element to the a value

Property Name	Property Value	Description
emList.MessageProcessingInstruction.EnvironmentCode		PRODUCTION
Routing.ItemEBSV2.SyncItemList.CAVS.EndpointURI		This property sets the EndpointURI for the CAVS simulator

Settings for SyncItemListLogisticsProvABCSImpl Service

Property Name	Property Value	Description
ABCSExtension.PostProcessABM	True/False Default=False	This property is used as an extension point after EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
ABCSExtension.PostProcessEBM	True/False Default=False	This property sets an extension point before EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
Default.SystemID	OTM_01	Sender SystemId is obtained from EBMHeader, but if it is empty, AIA reads it from the config file using this property.
ABCSExtension.PreProcessEBM	True/False, Default = False	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false
ABCSExtension.PreProcessABM	True/False	This property is used as an extension point After the EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false
ABCSExtension.PostProcessABM	True/False	This property is used as an extension point after the response ABM is received from the provider and before it is transformed to response EBM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false
ABCSExtension.PostProcessEBM	True/False	This property is used as an extension point after the response ABM is received from the provider and before it is transformed to response EBM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false
Transformation.EnableExtensions	True/False	Property used for determining enabling XSL 2.0

Property Name	Property Value	Description
		extensions in the Transformations.
Routing.LogisticsWebServiceV1.RouteToCAVS	True/False	Determines whether the EndpointURI should be routed either to the end application service or CAVS for simulating the service.
Routing.LogisticsWebServiceV1.CAVS.EndpointURI		This property sets the EndpointURI for the CAVS simulator.
Routing.LogisticsWebServiceV1.OTM_01.EndpointURI		This property sets the EndpointURI for the target application.
Routing.LogisticsWebServiceV1.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	Sets the EnvironmentCode element to the a value PRODUCTION.
ABCS.CallbackURL		Property specifies the URL used by OTM to return the response.
OTM_01.USERNAME		Property specifies the OTM instance user name.
OTM_01.PASSWORD		Property specifies the OTM instance password.
OTM_01.ISPASSWORDENCRYPTED	True / false Default=false	Property specifies whether the OTM password is encrypted in the Configuration Properties file. If true, DecryptionService is being called to decode the OTM password.

Settings for QueryTransportationSalesOrderItineraryListSiebelReqABCSImpl Service

Property Name	Property Value	Description
Default.SystemID	SEBL_01	Sender SystemId is obtained from ABM to be used in EBMHeader. If it is not available, then the AIA reads it from the config file using this property.
Routing.TransportationSalesOrderEBS.RouteToCAVS	True / false Default=false	This property, which is used for populating EBMHeader's EnvironmentCode, decides whether the TransportationSalesOrderEBS should invoke CAVS or the Provider application's business connector service. If the value is set to true, EBMHeader's Env Code is set to 'CAVS' and the EBS routes the request to CAVS. If the value is set to false, EBMHeader's Env Code is set to the EnvCode mentioned in AIAConfig property Routing.TransportationSalesOrderEBS.MessageProcessingInstruction.EnvironmentCode, or if this property is not set, then the default EnvCode will be 'PRODUCTION'. And in the EBS routing rules will decide based on the EnvCode where it should route.
Routing.TransportationSalesOrderEBS.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	This property defines the Environment Code to be populated in EBMHeader, which is used by the EBS to route it to the corresponding provider application business connector service or CAVS. This property is used while checking the RouteToCAVS property.

Property Name	Property Value	Description
Routing.TransportationSalesOrderEBS.CAVS.EndpointURI		This property defines the Definition Id to be populated in MessageProcessingInstruction of the EBMHeader, when the RouteToCAVS property is set to true. This holds the URI of CAVS simulator where the EBS should the request.
ABCSExtension.PreXformABMtoEBMTransportationSalesOrderItineraryListABM	True/false Default=false	This property is used as an extension point before ABM is transformed to EBM. It determines invocation of service at the extension point is to be made or not depending on its value.
ABCSExtension.PostXformABMtoEBMTransportationSalesOrderItineraryListABM	True/false Default=false	This property is used as an extension point before ABM is transformed to EBM. It determines invocation of service at the extension point is to be made or not depending on its value.
Transformation.EnableExtensions	True/false	This property determines enabling extensions in the transformations based on customer requirements.

Settings for QueryTransportationSalesOrderItineraryListLogisticsProvABCSImpl Service

Property Name	Property Value	Description
ABCSExtension.PostProcessABM	True/False Default=False	This property is used as an extension point after EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
ABCSExtension.PostProcessEBM	True/False Default=False	This property sets an extension point before EBM is transformed to ABM and after invoking Logistic web service. It determines whether a service has to be invoked or not based on its value.
Default.SystemID	OTM_01	Sender SystemId is obtained from ABM to be used in EBMHeader. If it is not available in ABM, AIA reads it from the config file using this property.
Routing.LogisticsWebService.RouteToCAVS	true/false Default=false	This property indicates whether the message should be sent to the target application or to CAVS. If this property is set to true, the message will be routed to CAVS, else it is routed to target application through adapter service if any. The URI of partnerlink is dynamically decided through a java activity based on this property.
Routing.LogisticsWebService.CAVS.EndpointURI		If the RouteToCAVS property is set to true, the URI of the simulator is dynamically derived by the java activity from this property

Property Name	Property Value	Description
Routing.LogisticsWebService.OTM_01.EndpointURI		If the RouteToCAVS property is set to false, the URI of the partnerlink is dynamically derived by the java activity from this property. This property should hold the endpoint URI of the provider application or that of the adapter service connected to provider application if any.
ABCSExtension.PreXformEBMtoABM	true/false Default=false	This property is used as an extension point before EBM is transformed to ABM. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
ABCSExtension.PostXformABMtoEBM	true/false Default=false	This property is used as an extension point after EBM is transformed to ABM and before invoking the target application. It determines invocation of service at the extension point is to be made or not depending on whether it is true or false.
Transformation.EnableExtensions	true/false	This property should be set to true when customers want to customize the attribute mapping done in xsl.
OTM_01.USERNAME		Property specifies the OTM instance user name.
OTM_01.PASSWORD		Property specifies the OTM instance password.
OTM_01.ISPASSWORDENCRYPTED	true/false Default=false	Property specifies whether the OTM password is encrypted in the Configuration Properties file. If true, DecryptionService is being called to decode the OTM password.

EBO Implementation Maps (EIMs)

For more information about how services are mapped, see the My Oracle Support document: EBO Implementation Maps (EIMs) 795541.1.

Chapter 9: Understanding Interoperability of AIA Process Integration Packs

This chapter provides an overview of interoperability of Application Integration Architecture (AIA) process integration packs (PIP) and discusses:

- Process Integration for Customer Objects Synchronization - Overview
- Interoperability of Fleet Financial Management PIP and Fleet Order Management PIP
- Interoperability of Fleet Financial and Fleet Order PIP
- Routing Rules

Process Integration for Customer Objects Synchronization - Overview

Starting with AIA 2.4 release vehicle, multiple PIPs are bundled in a single AIA release. This gives customers an opportunity to deploy more than one PIP if so desired.

PIPs that are bundled with AIA 2.4 are:

- Fleet Financial Management PIP
- Fleet Order Management PIP

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

In the following sections, scenarios for PIPs that can be potentially be deployed at the same site are described. Wherever applicable, recommendations are given for the best practices flow for these PIPs to work together. In addition, routing rule changes that customers need to make to suit the best practices flow are also given.

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

Interoperability of Fleet Financial Management Fleet Order Management PIPs

In the Fleet Financial PIP, Customer information is synchronized between Oracle EBS and Oracle Transportation Management. It is a one way feed from Oracle EBS to Oracle Transportation Management. The Fleet Order Management PIP supports exchanging of product information between Siebel CRM and Oracle Transportation Management.

Description of Customer Sync Related Flows in Fleet Financial Management PIP

- Sync Customer from Oracle EBS to OTM: When an item (product) is created/updated/Inactivated/Merged in Oracle EBS, it is synched to OTM.
- This customer is used in the Accounts Receivables flow for billing purposes.

Description of Customer Sync Related Flows in Fleet Order Management PIP

- Sync Customer from Oracle EBS to OTM: When an item (product) is created or updated in Siebel CRM, it is synched to OTM and Oracle EBS.
- This customer is used in the Order Sync for creating Orders in Siebel CRM.

Best Practice Flow(s) when Fleet Order and Fleet Financial PIPs Support Interoperability

The following are the best practices for Customer sync among the various participating applications when Fleet Financial and Fleet Order Management PIPs support interoperability:

- Customer Accounts are created only in Siebel CRM (Fleet Order PIP), which is the Customer master.
- The order is created in Siebel CRM and synched to Oracle Transportation Management for executing that Order.
- This customer is used in Fleet Financial PIPs Accounts Receivables flow for billing purposes.
- Customer Sync from EBS to OTM needs to be disabled. Disable the following ESB services in the ESB console:
  - SyncCustomerPartyListEbizEventCreateConsumer
  - SyncCustomerPartyListEbizEventUpdateConsumer
  - MergeAccountEbizEventConsumer
  - MergePartyEbizEventConsumer

Solution Assumptions and Constraints

- Customer should not be created/Updated in Oracle EBS.
- Siebel CRM is the Customer master.
- Orders should be created in Seibel CRM only.
- Billing happens through Accounts Receivables flow of Fleet Financial PIP.

Routing Rules

This section contains the routing rules to be used for the Interoperability.

For CustomerPartyEBSV2:

Operation	Filter Condition	Service Invoked	Description
SyncCustomerPartyList	{(/customerpartyebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode=' PRODUCTION' or not(/customerpartyebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode/text())) and (xp20:compare(/customerpartyebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:Sender/corecom:ID,'SEBL_01') = 0)};{ namespace xp20=http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.Xpath20 namespace corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2 namespace customerpartyebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CustomerParty/V2 }	SyncCustomerPartyJMSProducerV1::Produce_Message	This is Fleet Order PIP routing rule for synchronizing Customer to OTM and Oracle EBS.
SyncCustomerPartyList	{count(/customerpartyebo:SyncCustomerPartyListEBM/corecom:EBMHeader/corecom:MessageProcessingInstruction/corecom:EnvironmentCode[text() = 'CAVS'])!=0};{ namespace corecom=http://xmlns.oracle.com/EnterpriseObjects/Core/Common/V2 namespace customerpartyebo=http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CustomerParty/V2 }	AIASystem.ValidationSystem.AsyncRequestRecipient	CAVS

Index