Oracle® Application Integration Architecture

Oracle Financial Management Integration Pack Implementation Guide for Oracle Transportation Management and Oracle E-Business Suite

Release 3.1

E23563-07

June 2013

The Oracle Financial Management Integration Pack for Oracle Transportation Management (OTM) and Oracle E-Business Suite (Oracle EBS) implementation guide is a valuable resource for administrators and developers involved in the implementation, administration and deployment of robust financial business process provided by Oracle Application Integration Architecture (Oracle AIA) pre-built integrations release 11.4.

The first part of this guide focuses on understanding the pre-built integration between Oracle Financial Management and participating application such as OTM and Oracle EBS. It also lists the various assumptions and constrains, process flows, process integration, interfaces and integration services used by Oracle Financial Management pre-built integrations.

The second part of this guide discusses the customer data requirements, prerequisites, configuration steps and cross-references required for integrating Oracle Financial Management with OTM and Oracle EBS.

ORACLE®

Oracle Application Integration Architecture Oracle Financial Management Integration Pack Implementation Guide for Oracle Transportation Management and Oracle E-Business Suite, Release 3.1

E23563-07

Contents

3 Process Integration for Suppliers

4 Process Integration for Payable Invoices

5 Process Integration for Receivable Bills

6 Process Integration for Accruals and Reversals

7 Process Integration for Customer Synchronization

Part II   Implementing the Delivered Process Integrations

8 Setting Up Participating Applications

9   Data Requirements and Prerequisites

10   Performing an Initial Load

11   Working with Cross-References

12   Working with Domain Value Maps

13   Configuring Oracle Financial Management Integration Pack for OTM and Oracle EBS

List of Examples

List of Figures

List of Tables

Preface

Welcome to *Oracle Financial Management Integration Pack Implementation Guide for Oracle Transportation Management and Oracle E-Business Suite Implementation Guide*.

Common Oracle AIA Pre-Built Integration Guides

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

- Oracle Application Integration Architecture Installation and Upgrade Guide for Pre-Built Integrations Release 11.4

  This guide provides an overview of the installation process, including how to install, configure, and deploy your pre-built integrations. The steps required to upgrade your pre-built integrations to the latest release are also provided.

- Oracle Application Integration Architecture Pre-Built Integrations 11.1: Utilities Guide

  This guide describes:

  - How to work with and configure Session Pool Manager (SPM), which is a service in the Oracle SOA Suite web server whose primary function is to manage a pool of web server session tokens that can be reused by BPEL flows.

  - How to deploy and configure the AIACompositeScheduler. This is a utility component that is used by pre-built integrations to schedule a service-oriented architecture (SOA) composite to be invoked at the specified time interval.

- Oracle Application Integration Architecture Pre-Built Integrations 11.1: Product-to-Guide Index

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

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Master Notes

Master notes provide a one-stop reference point for the following:

- Latest manuals

- Alerts

- Troubleshooting details

- FAQs

- Patching information

- AIA community link and more.

For master notes for the Oracle Financial Management Integration Pack for Oracle Transportation Management and Oracle E-Business Suite, visit Master Note ID 1326049.2 on My Oracle Support (MOS): http://support.oracle.com.

Additional Resources

The following resources are also available:

- **Oracle Application Integration Architecture Foundation Pack:**

  Oracle AIA Pre-Built integrations require Foundation Pack 11.1.1.5.0 to be installed. Refer to the Foundation Pack documentation library on OTN to download the Foundation Pack guides at http://download.oracle.com/docs/cd/E21764_01/aia.htm.

- **Oracle Application Integration Architecture: Product-to-Guide Index:**

  Oracle Technology Network: http://www.oracle.com/technetwork/index.html

- **Known Issues and Workarounds:**

  My Oracle Support: https://support.oracle.com/

- **Release Notes:**

  Oracle Technology Network: http://www.oracle.com/technetwork/index.html

- **Documentation updates:**

  Oracle Technology Network: http://www.oracle.com/technetwork/index.html

Part I

Understanding the Delivered Process Integrations

This part includes the following chapters:

1

Oracle Financial Management Integration Pack for OTM and Oracle EBS

This chapter provides an overview of the Oracle Financial Management Integration Pack for Oracle transportation Management and Oracle E-Business Suite and discusses participating applications, business process flows, and assumptions and constraints.

This chapter includes the following sections:

- Section 1.1, "Overview"
- Section 1.2, "Participating Applications"
- Section 1.3, "Business Process Flow"
- Section 1.4, "Assumptions and Constraints"

1.1 Overview

This pre-built integration interacts with other applications, namely Oracle Transportation Management (OTM) and Oracle E-Business Suite (Oracle EBS) that are required to enable the integration process.

Oracle Financial Management for OTM and Oracle EBS pre-built integration provides best-of-breed solution that enables organizations to build a seamless and robust financial business process to manage invoice remittance between customers and service providers

1.1.1 Key Benefits

This pre-built integration streamlines the accounts payable and accounts receivable process between OTM and Oracle EBS Financials, including supplier synchronization, integrated freight payment, and customer billing, and accruals.

These are the key benefits for this integration pack

- Automate revenue recognition and cost accruals accurately.
- Single source of truth for supplier profiles and payments.
- Automated customer billing process to support complex business rules.
- Reduce integration time and costs.

1.1.2 Security

This pre-built integration has been enhanced with Oracle Web Services Manager (OWSM), which enables attaching OWSM policies to services and passing the OTM password information through *csf-keys*.

For more information about security validation and csf-key, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Working with Security" and *Oracle Fusion Middleware Installation Guide for Oracle Application Integration Architecture Foundation Pack.*

1.2 Participating Applications

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

- Section 1.2.1, "OTM"
- Section 1.2.2, "Oracle EBS Financials - General Ledger"
- Section 1.2.3, "Oracle EBS Financials - Payables"
- Section 1.2.4, "Oracle EBS Financials - Receivables"

1.2.1 OTM

OTM delivers robust transportation planning and execution capabilities to shippers and third-party logistics providers. It integrates and streamlines transportation planning, execution, freight payment, and business process automation on a single application across all modes of transportation, from full truckload to complex multi-leg air, ocean, and rail shipments.

Regardless of the size or the volume of the business, OTM delivers the capabilities needed in an open-standards-based architecture that allows to start with a single component or any mix of components and also gives the flexibility to grow easily, without installing or reinstalling added functionality.

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

1.2.2 Oracle EBS Financials - General Ledger

Oracle General Ledger (GL) is a comprehensive financial management solution that provides highly automated financial processing, effective management control, and real-time visibility to financial results. It provides everything needed to meet financial compliance and improve the organizations bottom line. Oracle GL is part of the Oracle E-Business Suite, an integrated suite of applications that drive enterprise profitability and reduce costs.

In this complex, global, and regulated environment, finance organizations face challenges in trying to follow local regulations and multiple reporting requirements. Oracle GL allows companies to meet these challenges in a very streamlined and automated fashion. Multiple ledgers can be assigned to a legal entity to meet statutory, corporate, regulatory, and management reporting. All accounting representations can be simultaneously maintained for a single transaction. For example, a single journal entered in the main, record-keeping ledger can be automatically represented in multiple ledgers even if each ledger uses a different chart of accounts, calendar, currency, and accounting principle.

Additionally, Oracle GL provides complete multi-currency and language functionality to satisfy the most demanding global financial requirements. The organization can

capture and report on any number of currencies from the balance level to the sub-ledger level. Currency conversion, revaluation, re-measurement, and translation are all performed in accordance with local and international accounting standards to improve internal controls and increase efficiency.

For more information about Oracle EBS Financials - General Ledger, see *Oracle E-Business Suite Financials - General Ledger Guide*.

1.2.3 Oracle EBS Financials - Payables

Oracle Payables improves margins, instills corporate and fiscal discipline, and optimizes business relationships. It is the cornerstone of Oracles Procure-to-Pay and Travel and Expense Management solutions, seamless and comprehensive business flows that help manage the total enterprise spend.

Oracle Payables provides the tools to control your cash flow, minimize errors and overpayments, and eliminate inefficiencies with:

- Real-time, accurate cash position, and forecast information, so the organization can plan payment runs and major disbursements appropriately.

- Minimum and maximum payment controls to better manage cash flow.

- Standard reports that allow you to eliminate duplicate and unauthorized payments.

- Disbursement requests that allow you to pay non-supplier payees and give detailed visibility of status.

For more information about Oracle EBS Financials - Payables, see *Oracle E-Business Suite Financials - Payables Guide*.

1.2.4 Oracle EBS Financials - Receivables

Oracle Receivables allows organizations to streamline invoicing, receipt, and customer deduction processing while improving cash flow, optimizing customer relationships, and providing strategic information. It provides the flexibility to meet the demands of a global market with strong financial controls to assist in installing corporate and fiscal discipline. Oracle Receivables is the cornerstone of Oracles Credit to Cash solution that helps you improve cash flow, increase efficiencies, and optimize customer relationships.

Oracle Receivables seamlessly manages invoicing requirements across the Oracle E-Business Suite and offers importing capabilities to extend this service to non-Oracle ordering systems. There is no need for complex coding and customization traditionally associated with creating customer or industry centric invoices. Invoicing types include traditional invoices, balance forward billing, installment billing, consolidated monthly billing, chargeback, and deposits.

For more information about Oracle EBS Financials - Receivables, see *Oracle E-Business Suite Financials - Receivables Guide*.

1.3 Business Process Flow

The Oracle Financial Management for OTM and Oracle EBS pre-built integration consists of these integration flows:

- Currency exchange rates
- Suppliers

- Payable invoices
- Receivable bills
- Accruals and reversals
- Customers

Figure 1–1 illustrates the Financial Management Business process flow:

Figure 1–1   Financial Management Business Process Flow
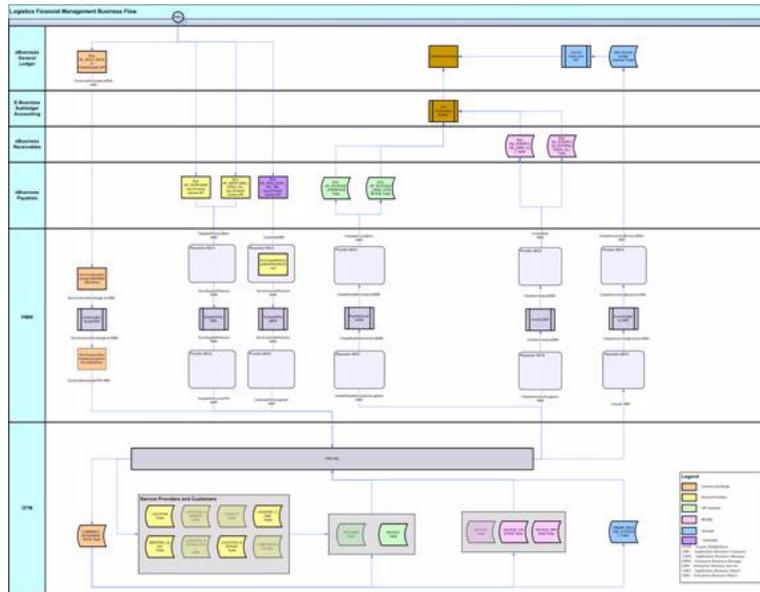
The integration pack for financial management allows an organization to synchronize their suppliers (service providers) and currency exchange rates between Oracle EBS and OTM. It also integrates the invoices created in OTM and sends them to the Oracle EBS Payables system so that they can be accounted and settled in the financial system. The business processes support multiple payment methods and ensure that the payment is not duplicated for the same invoice.

The integration pack for financial management also allows an organization to synchronize their customer information from Oracle EBS to OTM. This customer information can be used in the bills created in OTM to send as Accounts Receivables to get the payment from the customer.

The financial management process integration sends a bill created in OTM to Oracle EBS Receivables for a product or service performed for the customers, matches it with the customer invoice, and sends the bill to the customer to receive the payment.

In addition, any accruals and reversals created in OTM are sent to Oracle EBS General Ledger so they can be properly accounted.

1.4 Assumptions and Constraints

These are the assumptions and constraints for the Oracle Financials Management for OTM and Oracle EBS pre-built integration:

- An operating unit cannot be mapped to multiple domains.
- Several operating units can be mapped to one domain.

- OTM and Oracle EBS applications are implemented before the implementation of this PIP.

- The same values for Siebel CRM business units should be created in Oracle EBS for operating units.

- Customers switching from one financials application to another require re-implementation of the integration pack.

> **Note:**   Assumptions and constraints as applicable to process integrations are covered in the respective chapters.

2

Process Integration for Currency Exchange Rates

This chapter provides an overview of currency exchange rates process integration and discusses business process flows, assumptions and constraints, Oracle E-Business Suite (Oracle EBS) and Oracle Transportation Management (OTM) interfaces, core Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 2.1, "Overview"
- Section 2.2, "Business Process Flows"
- Section 2.3, "Assumptions and Constraints"
- Section 2.4, "Oracle EBS Interfaces"
- Section 2.5, "OTM Interfaces"
- Section 2.6, "Core Oracle AIA Components"
- Section 2.7, "Integration Services"

2.1 Overview

Currency exchange rate is the reference information used in the translation of monetary values from one currency to another. The exchange rate expresses the value of one currency in terms of another. The process integration for currency exchange rates enables you to use Oracle EBS Financials as an accounting engine and OTM for invoices and bills transactions.

2.2 Business Process Flows

The process integration for currency exchange rates between Oracle EBS and OTM supports these integration flows:

- Initial load of currency exchange rates: Extracts and loads the initial currency exchange rates from Oracle EBS to OTM.
- Incremental updates of currency exchange rates: Enables the synchronization of incremental creation and updates of the currency exchange rates from Oracle EBS to OTM.

This integration is not a point-to-point integration between Oracle EBS General Ledger and OTM. An Oracle AIA layer serves as an intermediate thin layer of application between Oracle EBS General Ledger (GL) and OTM. As a part of the currency

exchange rates integration, Oracle EBS GL sends the currency exchange rates to the Oracle AIA layer and the Oracle AIA layer delivers the information to OTM. This Oracle AIA layer performs message filtering, message transformation, and message routing.

Figure 2–1 illustrates the integration of currency exchange rates.

Figure 2–1   Currency Exchange Rates Process Integration Flow
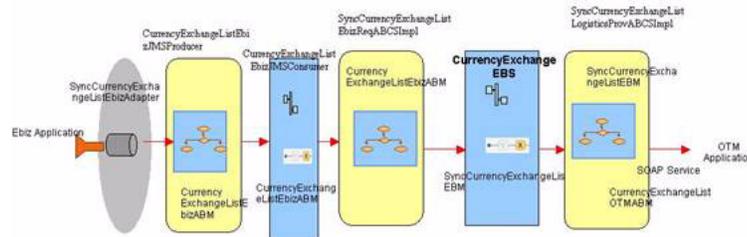
2.2.1 Initial Load of Currency Exchange Rates

The purpose of this flow is to load the existing currency exchange rates from Oracle EBS into OTM by triggering the CurrencyExchangeListEbizJMSProducer service.

Note: Contact your Oracle Service-Oriented Architecture (Oracle SOA) Administrator to obtain the end point of the CurrencyExchangeListEbizJMSProducer web service.

The service expects the inputs listed in Table 2–1.

Table 2–1   Parameters for Loading Currency Rates

Fields	Value
from_currency	xsd:string
to_currency	xsd:string
from_date	xsd:date
to_date	xsd:date
conversion_rate_type	xsd:string

The **from_date** field is mandatory. If the **to_date** is not specified it takes the current date as the **TO_DATE**. The valid date format is *YYYY-MM-DD (2008-02-09)*. The rest of the fields are optional. You can use those fields to limit the rates, which are required to be loaded from Oracle E-Business Suite to OTM. Use the appropriate date range based on the load of data and performance of the server.

Figure 2–2 illustrates the initial loading of currency exchange rates.

Figure 2–2   Initial Loading of Currency Exchange Rates
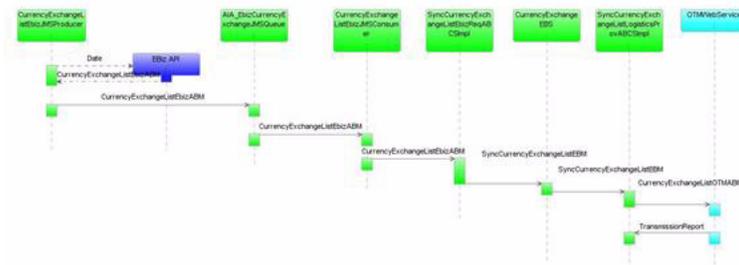
When you initiate the initial load of currency exchange rates process, these events occur:

1. Oracle E-Business Suite invokes the CurrencyExchangeListEbizJMSProducer whenever a currency exchange rate is created or loaded into Oracle E-Business Suite.

2. The CurrencyExchangeListEbizJMSProducer extracts all the currency exchange rates from the Oracle E-Business Suite database based on the dates provided in the Business Process Execution Language (BPEL) console and moves the currency exchange rates into the AIA_EbizCurrencyExchangeJMSQueue.

3. The CurrencyExchangeListEbizJMSConsumer service picks up the message from the AIA_EbizCurrencyExchangeJMSQueue and invokes the SyncCurrencyExchangeListEbizReqABCSImpl, which transforms the message into the SyncCurrencyExchangeListEBM. Then, the SyncCurrencyExchangeListEBM invokes the CurrencyExchangeEBS.

4. The CurrencyExchangeEBS receives the SyncCurrencyExchangeListEBM and invokes the SyncCurrencyExchangeListLogisticsProvABCSImpl.

5. The SyncCurrencyExchangeListLogisticsProvABCSImpl receives the SyncCurrencyExchangeListEBM and transforms it into the CurrencyExchangeOTMABM.

6. The LogisticsWebService is invoked with this transformed application business message (ABM) and the instance asynchronously waits for a transmission report from OTM. If the transaction is successful, then the cross-reference values are populated for OTM columns for each of the currency exchange IDs listed in the SyncCurrencyExchangeListEBM; otherwise it invokes the AIAAsyncErrorHandlingBPELProcess with an error message.

2.2.2 Incremental Updates of Currency Exchange Rates

The currency exchange rates that are created or updated in the Oracle EBS must be updated in the OTM.

Figure 2–3 shows the incremental updates for currency exchange rates:

Figure 2–3   Incremental Updates of Currency Exchange Rates
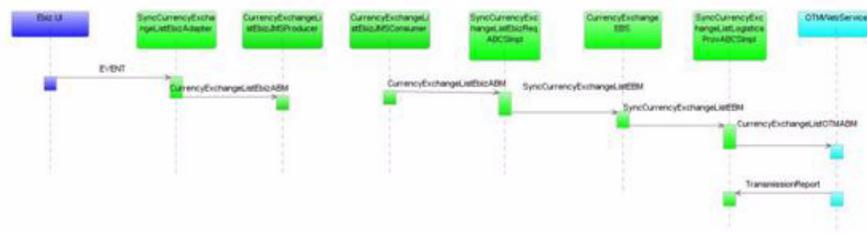
1. In the Oracle E-Business Suite user interface, the user updates or adds new currency exchange rates, which triggers and event that automatically invokes the SyncCurrencyExchangeListEbizAdapter. This adapter invokes the CurrencyExchangeListEbizJMSProducer, which queries the Oracle E-Business Suite database for currency exchanges rates and drops the message in AIA_ EbizCurrencyExchangeJMSQueue.

2. The CurrencyExchangeListEbizJMSConsumer service picks up the message from the AIA_EbizCurrencyExchangeJMSQueue and invokes the SyncCurrencyExchangeListEbizReqABCSImpl, which transforms the message into the SyncCurrencyExchangeListEBM, and the SyncCurrencyExchangeListEBM invokes the CurrencyExchangeEBS.

3. The CurrencyExchangeEBS receives the SyncCurrencyExchangeListEBM and invokes the SyncCurrencyExchangeListLogisticsProvABCSImpl.

4. The SyncCurrencyExchangeListLogisticsProvABCSImpl receives the SyncCurrencyExchangeListEBM and transforms it into the CurrencyExchangeListOTMABM.

5. The LogisticsWebService is invoked with this transformed application business message (ABM), and the instance asynchronously waits for a transmission report from OTM. If the transaction is successful then the cross-reference values are populated for OTM columns for each of the currency exchange IDs listed in the SyncCurrencyExchangeListEBM; otherwise it invokes the AIAAsyncErrorHandlingBPELProcess with an error message.

2.3 Assumptions and Constraints

These are the solution assumptions and constraints:

- The Oracle EBS is responsible for adding and maintaining the currency exchange rates and types. Oracle EBS is the currency exchange rate master and all data from the Oracle EBS is synchronized to OTM.

- The currency rate types and currency codes are different in Oracle EBS and OTM applications. The domain value maps (DVMs) are set up and maintained manually by both Oracle EBS and OTM in the Oracle AIA layer to have a common value.

- Oracle EBS currency precision limit is limited to four digits (Example: $120.0001). OTM precision length must be setup at four digits to avoid rounding errors.

- The currency exchange rates in OTM are synchronized to the public domain and this is changed by a property in the Oracle AIA configuration file.

2.4 Oracle EBS Interfaces

For the currency exchange rates integration flow, these are the Oracle EBS interfaces:

- Oracle EBS General Ledger (GL) exposes a business event called oracle.apps.gl.CurrencyConversionRates.dailyRate.specify as an interface to subscribe to the creation and updates of currency exchange rates.

- Oracle EBS also provides an API called the GET_CUR_CONV_RATES interface in the GL_EXCH_RATES_SYNC_PKG that helps to retrieve the complete information of exchange rates.

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

2.5 OTM Interfaces

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. Once invoked, the logistics web service immediately returns an acknowledgment with a transmission number. Once the processing is complete, it sends a transmission report back indicating the success or the failure. The application business message (ABM) details can be seen in the GLOG xsd with the ExchangeRate element.

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

2.6 Core Oracle AIA Components

The currency exchange rate integration flow uses these components:

- CurrencyExchangeEBO

- SyncCurrencyExchangeListEBM

- CurrencyExchangeEBS

The core enterprise business object (EBO) and enterprise business message (EBM) XSD files can be located by EBO within the $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/ parent folder.

The core enterprise business services (EBS) web services definition language (WSDL) files can be located by EBO within the $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/ parent folder.

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

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."

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

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Extensibility for AIA Artifacts".

2.7 Integration Services

These are the services involved in the currency exchange rate integration:

- SyncCurrencyExchangeListEbizAdapter
- CurrencyExchangeListEbizJMSProducer
- CurrencyExchangeListEbizJMSConsumer
- SyncCurrencyExchangeListEbizReqABCSImpl
- CurrencyExchangeEBS
- SyncCurrencyExchangeListLogisticsProvABCSImpl
- CurrencyExchangeResponseEBS

2.7.1 SyncCurrencyExchangeListEbizAdapter

This service is used for incremental changes. This service listens to the oracle.apps.gl.CurrencyConversionRates.dailyRate.specify business event and invokes the CurrencyExchangeListEbizJMSProducer, which then queries the Oracle E-Business Suite database for currency exchange rates with the Event ID.

2.7.2 CurrencyExchangeEbizListJMSProducer

The CurrencyExchangeEbizJMSProducer service is a Business Process Execution Language (BPEL) process. The Oracle E-Business application invokes this service when either is a new currency exchange rate is created, or an existing currency exchange rate is updated.

This service is also used when doing an initial load of the exchange rates from Oracle E-Business Suite. It takes the **From** and **To date** and the **From** and **To Currency** as inputs. These inputs can come from the event or are directly provided in an initial load. The database adapter reads all the currency exchange rates from the Oracle E-Business Suite database based on the dates entered. If the **To_Date** is not provided, it assumes today's date. Based on the **From_Date, To_Date** from the input and a property (NoOfDays) from the AIAConfigurationFile, the Oracle E-Business Suite API (get_cur_conv_rates) is called to obtain the currency exchange rates. Then, the Currency Exchange Rates are dropped in the AIA_EbizCurrencyExchangeJMSQueue. The NoOfDays property in the CURRENCYEXCHANGE module must be specified to determine the cursor size in which the exchange rates are retrieved. Based on the processing capabilities of the system, an appropriate value must be provided for this property.

2.7.3 CurrencyExchangeListEbizJMSConsumer

The CurrencyExchangeListEbizJMSConsumer is a mediator service. It has a JMS adapter called CurrencyExchangeListEbizJMSConsumer. This adapter listens to the AIA_EbizCurrencyExchangeJMSQueue and picks up the messages for which JMSCorrelationID is SyncEbizCurrencyExchange. This invokes the SyncCurrencyExchangeListEbizReqABCSImpl with the CurrencyExchangeListEbizABM.

2.7.4 SyncCurrencyExchangeListEbizReqABCSImpl

The SyncCurrencyExchangeListEbizReqABCSImpl is a Business Process Execution Language (BPEL) process, which receives the CurrencyExchangeListEbizABM from

the CurrencyExchangeListEbizJMSConsumer and transforms the message into the SyncCurrencyExchangeListEBM.

These domain value map (DVM) lookups are used by this service:

- CURRENCY_CODE: domain value mapping for currency codes.
- CURRENCYEXCHANGE_CONVERSIONTYPECODE: domain value mapping for currency type codes.
- CURRENCYEXCHANGE_STATUSCODE: domain value mapping for status code of currency exchange rates.

In addition, this service populates the EBM header variable and cross-reference table and invokes the CurrencyExchangeEBS.

2.7.5 CurrencyExchangeEBS

The CurrencyExchangeEBS is an enterprise business service that exposes all the enterprise operations related to the currency exchange like create currency exchange, update currency exchange, and so on. This integration uses the SyncCurrencyExchangeList operation.

This enterprise business service routes the request to the appropriate provider like the SyncCurrencyExchangeListLogisticsProvABCSImpl or the Composite Application Validation System (CAVS), based on the filter condition and operations. Updates and creates are done using the synchronize action. No transformations are done in this service. OTM determines whether this synchronize currency exchange rate message is for create or an update action.

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

2.7.6 SyncCurrencyExchangeListLogisticsProvABCSImpl

The SyncCurrencyExchangeListLogisticsProvABCSImpl is a Business Process Execution Language (BPEL) process, which receives the SyncCurrencyExchangeListEBM, transforms the message into the CurrencyExchangeListOTMABM, invokes the LogisticsWebService with the SyncCurrencyExchangeListEBM and waits for the transmission report from the LogisticsWebService. If the transmission report specifies the transaction is successful, it populates the cross-reference values otherwise it invokes the AIAAsyncErrorHandlingBPEL process to generate the error messages.

These DVM lookups are used by this service:

- CURRENCY_CODE: domain value mapping for currency codes.
- CURRENCYEXCHANGE_CONVERSIONTYPECODE: domain value mapping for currency type codes.

2.7.7 CurrencyExchangeResponseEBS

The CurrencyExchangeResponseEBS is the enterprise business service (EBS) used to route all currency exchange response related actions to the requesting application like create currency-exchange rates, update currency-exchange rates, delete currency-exchange rates, and synchronize currency-exchange-list rates.

3

Process Integration for Suppliers

This chapter provides an overview of the process integration for suppliers (service providers) and discusses business process flows, assumptions and constraints, Oracle E-Business Suite (Oracle EBS) and Oracle transportation Management (OTM) interfaces, core Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 3.1, "Overview"
- Section 3.2, "Business Process Flows"
- Section 3.3, "Assumptions and Constraints"
- Section 3.4, "Oracle EBS Interfaces"
- Section 3.5, "OTM Interfaces"
- Section 3.6, "Core Oracle AIA Components"
- Section 3.7, "Integration Services"

3.1 Overview

In the process integration for suppliers, Oracle EBS acts as a payable and accounting engine and OTM creates the invoices; therefore, Oracle EBS and OTM require the sharing of suppliers' information between the applications. OTM requires the supplier information for several key functions including the creation and management of order releases and invoices. Oracle EBS requires the supplier information for supplier payment. For end-to-end business integration, the same supplier instance and related information must be shared between these two applications.

3.2 Business Process Flows

Oracle EBS is the source of valid suppliers (service providers in OTM and their locations and payment terms. The supplier integration synchronizes supplier information from Oracle EBS to OTM through these integration flows:

- Initial loading of supplier information: Extracts and loads initial supplier information from Oracle EBS to OTM.
- Updating supplier information: Enables the synchronization of incremental creation and updates of the newly created, modified, or inactivated supplier information from Oracle EBS to OTM.

Figure 3–1 shows the process integration of suppliers (service providers):

Figure 3–1 Suppliers (Service Providers) Integration Flow
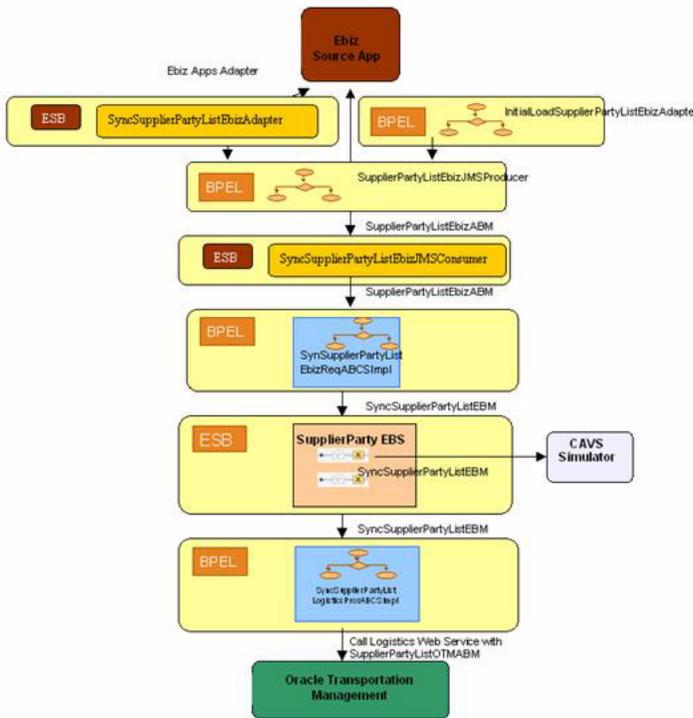
3.2.1 Initial Load of Suppliers

The purpose of this flow is to load into OTM the supplier information that was created in Oracle EBS by triggering the InitialLoadSupplierPartyListEbizAdapter service.

Figure 3–2 shows how supplier information from Oracle EBS is loaded into OTM:

Figure 3–2 Initial Load of Suppliers (Service Providers)
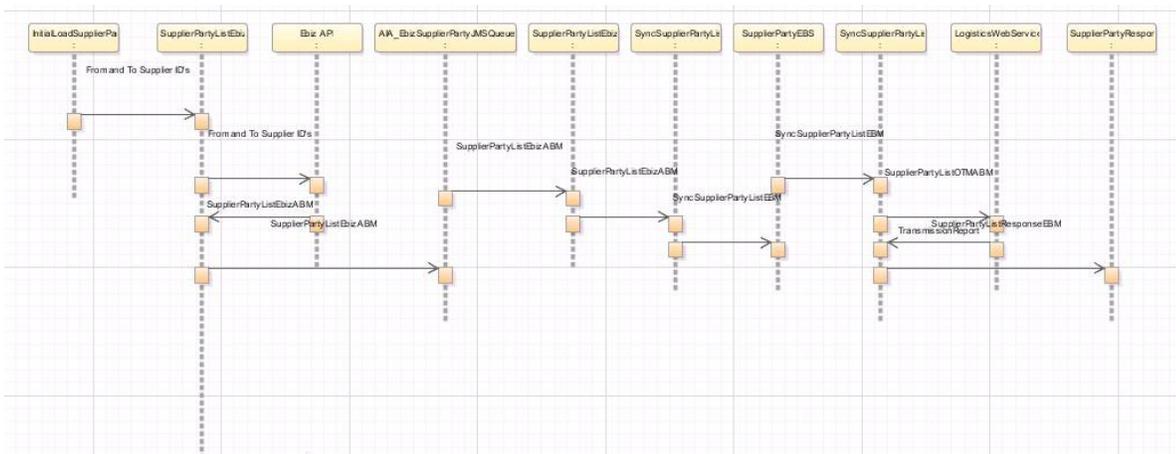
When you initiate the process, these events occur:

1. The InitialLoadSupplierPartyListEbizAdapter service queries the supplier information in Oracle EBS, counts the vendor IDs to obtain a minimum and maximum number of vendor IDs, and invokes the SupplierPartyListEbizJMSProducer.

2. The SupplierPartyListEbizJMSProducer invokes the Oracle EBS API to query for the SupplierPartyListEbizABM and drops the message into the AIA_ EbizSupplierPartyJMSQueue.

3. The SyncSupplierPartyListEbizJMSConsumer picks up the message from the AIA_ EbizSupplierPartyJMSQueue and invokes the SyncSupplierPartyListEbizReqABCSImpl.

4. The SyncSupplierPartyListEbizReqABCSImpl transforms the SupplierPartyListEbizABM message into the SyncSupplierPartyListEBM. The transformation does cross-referencing for systems specific values and calls the SupplierPartyEBS with operation SyncSupplierPartyList. The SupplierPartyEBS is a routing mediator service with several operations on the SupplierPartyEBO.

5. The SupplierPartyEBS routes the SyncSupplierPartyListEBM to the SyncSupplierPartyListLogisticsProvABCSImpl or Composite Application Validation System (CAVS).

6. The SyncSupplierPartyListLogisticsProvABCSImpl receives the message and transforms the SyncSuplierPartyListEBM into the SupplierPartyListOTMABM. The transformation invokes the LogisticsWebservice and waits for the transmission report from the LogisticsWebService. If the transaction is successful, it populates the cross-reference values. If the transaction is not successful, it invokes the AIAAsyncErrorHandlingBPELProcess.

3.2.2 Updating Supplier Information

The purpose of this flow is to load into OTM the supplier information that was updated in Oracle EBS by triggering the SupplierPartyListEbizJMSProducer service.

Figure 3–3 shows the incremental update of supplier information:

Figure 3–3 Incremental Load of Suppliers (Service Providers)
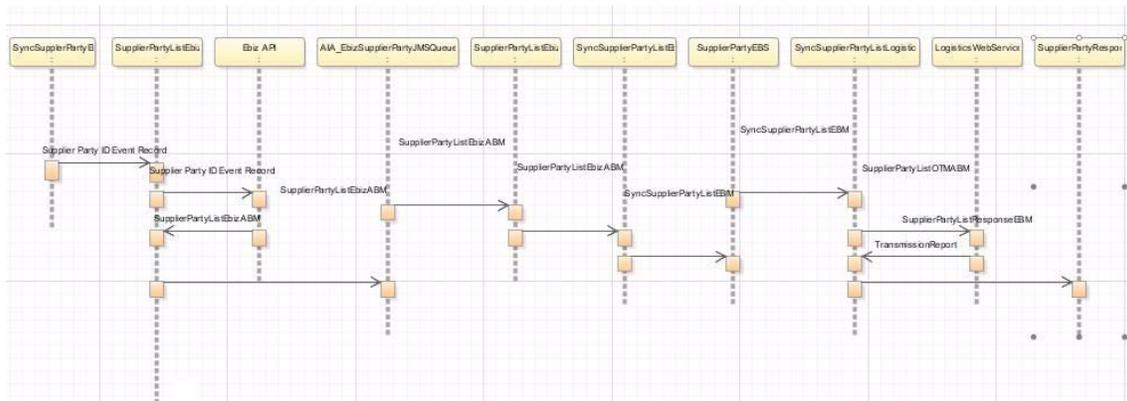
When you initiate the incremental load, these events occur:

1. In Oracle EBS, if you create or update the supplier information, the activity invokes the SyncSupplierPartyListEbizAdapter through a business event. This adapter service listens to the events and then invokes the SupplierPartyListEbizJMSProducer.

2. The SupplierPartyListEbizJMSProducer invokes the Oracle EBS API to query for the SupplierPartyListEbizABM and drops the message into the AIA_ EbizSupplierPartyJMSQueue.

3. The SyncSupplierPartyListEbizJMSConsumer picks up the message from the AIA_ EbizSupplierPartyJMSQueue and invokes the SyncSupplierPartyListEbizReqABCSImpl.

4. The SyncSupplierPartyListEbizReqABCSImpl transforms the SupplierPartyListEbizABM message into the SyncSupplierPartyListEBM. The transformation does cross-referencing for systems specific values and calls the SupplierPartyEBS with operation SyncSupplierPartyList. The SupplierPartyEBS is a routing mediator service with several operations on the SupplierPartyEBO.

5. The SupplierPartyEBS routes the SyncSupplierPartyListEBM to the SyncSupplierPartyListLogisticsProvABCSImpl or CAVS.

6. The SyncSupplierPartyListLogisticsProvABCSImpl receives the message and transforms the SyncSuplierPartyListEBM into the SupplierPartyListOTMABM. The transformation invokes the LogisticsWebservice and waits for the transmission report from the LogisticsWebService. If the transaction is successful, it populates the cross-reference values. If the transaction is not successful, it invokes the AIAAsyncErrorHandlingBPELProcess.

3.3 Assumptions and Constraints

These are the solution assumptions and constraints:

- Oracle EBS Financials AP is the source system for suppliers, contacts, locations, and addresses.

- This integration is a one-way synchronization from Oracle EBS into OTM. Any update to supplier information in OTM is not synchronized into Oracle EBS.

- The capability of creating service providers in the OTM application must be disabled.

- The supplier header information is synchronized to OTM. Any purchasing sites that exist for the supplier should be synchronized into OTM based on a configuration option.

  The supplier pay sites are not stored in OTM. When invoices are received from OTM into Oracle EBS, the default supplier pay site information is derived using the supplier number, organization unit details or purchasing site information.

- The service provider in OTM has the same value for location ID and corporation ID. In the case that purchasing sites are synchronized into OTM, the corporation ID is the supplier number for that site or location.

- The supplier information is created as locations and service providers in OTM in the OTM domain specified in the configuration file. This domain can be overridden through a Custom.xsl.

- If Oracle EBS is sending the purchasing sites, then the contact and address information is updated with the purchasing site address/contact information.

- Whenever an inactive date is entered for a supplier associated with multiple purchasing sites in Oracle EBS, the synchronization of the inactive date does not happen to all the corresponding purchasing sites for the supplier in OTM. Therefore, you should enter the inactive date for the purchasing sites in Oracle E-Business and this should be synchronized to OTM.

- The initial creation of supplier in Oracle EBS has no value set for the supplier type because the supplier type is not mandatory.

On the other hand, in OTM the supplier type/location role is not mandatory; therefore, the supplier records are created in OTM with the location role blank. In Oracle EBS, you can update the supplier type (through the organization form field) and the location role is created in OTM. If you want to filter by supplier type, a routing rule can be added during implementation.

- The Oracle EBS vendor name is 240 characters long and in OTM the corporation name is 120 characters; therefore, a vendor name longer than 120 are truncated in OTM.

- The Oracle EBS vendor alternate name is 320 characters long and in OTM the service provider alias is only 101 characters; therefore, an alternate name longer than 101 characters is truncated in OTM.

3.4 Oracle EBS Interfaces

These are the Oracle EBS interfaces for the supplier integration flow:

- Oracle EBS API exposes a business event called oracle.apps.ap.supplier.event as an interface to subscribe to the creation and updates of supplier data.

- Apart from the business event, Oracle EBS also provides an API called the AP_SUPPLIER_INFO_PKG that helps to retrieve the supplier information based on either a range of suppliers or a specific supplier.

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

3.5 OTM Interfaces

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. Once invoked, the logistics web service immediately returns an acknowledgment with a transmission number. Once the processing is complete, it then sends a transmission report back indicating the success or the failure. The application business message (ABM) details can be seen in the GLOG xsd with the location element.

For more information about the logistics service, see *Oracle Transportation Management Integration Guide*.

3.6 Core Oracle AIA Components

The supplier party integration flow uses these components:

- SupplierPartyEBO

- SyncSupplierPartyListEBM

- SupplierPartyEBS

The core enterprise business object (EBO) and enterprise business message (EBM) XSD files can be located by EBO within the $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/ parent folder.

The core enterprise business services (EBS) web services definition language (WSDL) files can be located by EBO within the $AIA_

HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/ parent folder.

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

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."

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

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Extensibility for AIA Artifacts".

3.7 Integration Services

These are the services delivered with this integration:

- InitialLoadSupplierPartyListEbizAdapter
- SyncSupplierPartyListEbizAdapter
- SupplierPartyListEbizJMSProducer
- SyncSupplierPartyListEbizJMSConsumer
- SyncSupplierPartyListEbizReqABCSImpl
- SupplierPartyEBS
- SyncSupplierPartyListLogisticsProvABCSImpl
- SupplierPartyResponseEBS

3.7.1 InitialLoadSupplierPartyListEbizAdapter

This service is an asynchronous Business Process Execution Language (BPEL) process. This process is used for the initial load of suppliers from Oracle EBS into the OTM web service. This service queries the supplier information in Oracle EBS, counts the vendor IDs to obtain a minimum and maximum number of vendor IDs, and invokes the SupplierPartyListEbizJMSProducer.

3.7.2 SyncSupplierPartyListEbizAdapter

The SyncSupplierPartyListEbizAdapter is used only for incremental changes. This is a mediator service with one Oracle applications adapter that subscribes to the oracle.apps.ap.supplier.event. Whenever a supplier, site, or contact is created or updated, this adapter service listens to the events and then invokes the SupplierPartyListEbizJMSProducer service.

3.7.3 SupplierPartyListEbizJMSProducer

The SupplierPartyListEbizJMSProducer service is a Business Process Execution Language (BPEL) process used for both initial and incremental loads. The Oracle EBS application invokes this service when either a new supplier is created or an existing supplier is updated

This service reads the supplier party ID information either from the InitalLoadSupplierPartyEbizAdapter, or the SyncSupplierPartyEbizAdapter for incremental changes. It invokes the Oracle EBS API (ap_supplier_info_pkg) to retrieve the SupplierPartyListEbizABM and drops the messages into the AIA_EbizSupplierPartyJMSQueue individually.

3.7.4 SyncSupplierPartyListEbizJMSConsumer

The SyncSupplierPartyListEbizJMSConsumer is a mediator service. It has a Java Message Service (JMS) adapter, which picks up the messages from AIA_EbizSupplierPartyJMSQueue and invokes the SyncSupplierPartyListEbizReqABCSImpl.

3.7.5 SyncSupplierPartyListEbizReqABCSImpl

The SyncSupplierPartyListEbizReqABCSImpl is a Business Process Execution Language (BPEL) process and a single operations service that has the SupplierPartyEBS as a partner service. This service receives the SupplierPartyListEbizABM message as a request and does not return a response to the calling service.

This service performs these actions:

- Accepts the SupplierPartyListEbizABM message from Oracle EBS. This message contains a cross-reference for suppliers, sites, and contacts.

- Transforms the SupplierPartyListEbizABM into the SyncSupplierPartyListEBM. While it is transforming from the application business message (ABM) to the enterprise business message (EBM), cross-references are looked up for:

  - SUPPLIERPARTY_ID

  - SUPPLIERPARTY_ADDRESS_ID

  - SUPPLIERPARTY_LOCATION_ID

  - SUPPLIERPARTY_CONTACT_ID

- Sends the SyncSupplierPartyListEBM message as an input to the SyncSupplierPartyList operation in the SupplierPartyEBS service.

These DVM lookups are used by this service:

- CURRENCY_CODE: domain value mapping for currency codes.

- ADDRESS_COUNTRYSUBDIVID: domain value mapping for the state code values in address of supplier.

- SUPPLIERPARTY_TYPECODE: domain value mapping for the type of supplier.

- ADDRESS_COUNTRYID: domain value mapping for the country code values.

- LANGUAGE_CODE: domain value mapping for language.

3.7.6 SupplierPartyEBS

The SupplierPartyEBS is an enterprise business service that exposes all the enterprise operations related to the supplier party like create supplier-party, update supplier-party, and so on. This integration only uses the SyncSupplierPartyList operation. This service routes the request to the appropriate provider like the SyncSupplierPartyListLogisticsProvABCSImpl or the Composite Application Validation System (CAVS) based on the filter condition and operations. Updates and

creates are done using the synchronize action. No transformations are done in this service. OTM determines whether this synchronize supplier party message is for create or an update action.

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

3.7.7 SupplierPartyResponseEBS

The SupplierPartyResponseEBS is the enterprise business service (EBS) used to route all supplier party response related actions to the requesting application like create supplier-party, update supplier-party, and synchronize supplier-party-list.

3.7.8 SyncSupplierPartyListLogisticsProvABCSImpl

The SyncSupplierPartyListLogisticsProvABCSImpl is a Business Process Execution Language (BPEL) process, which receives the SyncSupplierPartyListEBM, transforms the message into the SupplierPartyListOTMABM, invokes the logistics web service with the SyncSupplierPartyListEBM, and waits for the transmission report from the LogisticsWebService. If the transmission report specifies the transaction is successful, it populates the cross-reference values otherwise it invokes the AIAAsyncErrorHandlingBPEL process to generate the error messages

These DVM lookups are used by this service:

- CURRENCY_CODE: domain value mapping for currency codes.

- ADDRESS_COUNTRYSUBDIVID: domain value mapping for the state code values in address of supplier.

- SUPPLIERPARTY_TYPECODE: domain value mapping for the type of supplier.

- ADDRESS_COUNTRYID: domain value mapping for the country code values.

- LANGUAGE_CODE: domain value mapping.

The SupplierPartyResponseEBS is the enterprise business service (EBS) used to route all supplier party response related actions to the requesting application like create supplier-party, update supplier-party, and synchronize supplier-party-list.

3.7.9 SupplierPartyResponseEBS

The SupplierPartyResponseEBS is the enterprise business service (EBS) used to route all supplier party response related actions to the requesting application like create supplier-party, update supplier-party, and synchronize supplier-party-list.

4

Process Integration for Payable Invoices

This chapter provides an overview of the process integration for payable invoices and discusses business process flows, assumptions and constraints, Oracle E-Business Suite (Oracle EBS) and Oracle Transportation Management (OTM) interfaces, core Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 4.1, "Overview"
- Section 4.2, "Business Process Flows"
- Section 4.3, "Assumptions and Constraints"
- Section 4.4, "Oracle EBS Interfaces"
- Section 4.5, "OTM Interfaces"
- Section 4.6, "Core Application Integration Architecture Components"
- Section 4.7, "Integration Services"

4.1 Overview

The payables invoice integration flow allows OTM to pay its service providers, carriers, and other parties by sending its invoice (vouchers) transactions to the Oracle EBS Payables application. Oracle EBS Payables generates the payment and create the proper accounting for the payment transactions.

4.2 Business Process Flows

The process integration for payable invoices supports these integration flows:

- Voucher match and pay (no auto pay).

    OTM receives the invoice from the service provider:

    a. Matches it to the shipments based on the match rules and approves the invoice.

    b. Creates the voucher.

    c. Changes the voucher status to *issued* and sends the transaction using the VoucherXML to Oracle EBS Account Payables (AP).

- Voucher auto pay.

    OTM calculates the freight charges for the shipments:

     a. Approves the invoice.

     b. Creates the voucher.

     c. Changes the voucher status to *issued* and sends the transaction using the VoucherXML to Oracle EBS Account Payables (AP).

- Consolidated invoice batches.

  OTM receives a consolidated invoice from a service provider for multiple shipments:

     a. Matches the consolidated invoice with the shipments.

     b. Approves the invoice for payment.

     c. Creates a voucher.

     d. Changes the voucher status to *issued* and sends the transaction using the VoucherXML to Oracle EBS AP.

- One invoice with multiple vouchers.

  OTM receives an invoice for multiple payments from a service provider:

     a. Matches it with the shipments.

     b. Manually approves invoice for partial payment.

     c. Creates the voucher.

     d. Changes the voucher status to *issued* and sends the transaction using the VoucherXML to Oracle EBS AP.

- Prepaid invoice (comcheck).

  The user manually creates a prepayment invoice both in OTM and Oracle E-business suite. When the service provider invoice arrives:

     a. OTM creates a voucher containing the total invoice amount, prepayment amount, advance fee, and outstanding approved voucher amount.

     b. OTM changes the voucher status to *issued* and sends the transaction using the VoucherXML to Oracle EBS AP.

     c. The prepayment application is done automatically based on the reference fields entered in the voucher.

- Credit memo.

  OTM receives an invoice from the service provider:

     a. Matches the invoice to the shipments.

     b. Creates a credit memo for over payment, matches it with the original shipment, and creates the invoice with the status of *process hold*. The user approves the invoice and creates the approved voucher.

     c. Changes the approved voucher status to *issued*, and sends the transaction using the VoucherXML to Oracle EBS AP.

- Voucher with third-party payment

  OTM receives a supplier invoice, which specifies that a third-party supplier should be paid for the service. For example, OTM plans the shipment and sends a tender to supplier A. Supplier A wants supplier B to be paid for the service:

     a. Approves the invoice.

    b. Creates the voucher, changes the voucher status to *issued*, and sends the transaction using the VoucherXML to Oracle EBS AP to pay supplier B.

■ Foreign currency voucher.

  OTM receives an invoice in a foreign currency:

    a. Matches it to the shipments based on the match rules.

    b. Approves the invoice.

    c. Creates the voucher with the foreign currency.

    d. Changes the voucher status to *issued*, and sends the transaction using the VoucherXML to Oracle EBS AP.

■ Voucher with Disputed Amount and Reason.

  If the shipment is partially paid, OTM approves the invoice:

    a. Creates the voucher with the original invoice amount, approved payment amount, and dispute reason.

    b. Changes the voucher status to *issued*, and sends the transaction using the VoucherXML to Oracle EBS AP.

Figure 4–1 shows the process integration for payable invoices:

Figure 4–1   OTM (Logistics) to Oracle E-business Suite Payable Invoice Flow

4.2.1 Sending Payable Invoices to Oracle EBS Account Payables

The payable invoices integration flow creates the invoice in Oracle EBS AP based on the voucher sent from OTM.

Figure 4–2 shows the payable invoice flow from OTM to Oracle E-Business Suite:

Figure 4–2   OTM (Logistics) to Oracle E-business Suite Payable Invoice Flow
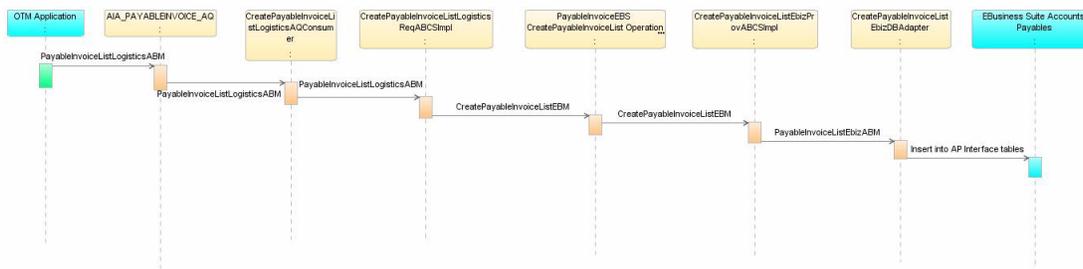
When you initiate the process, these events occur:

1. OTM enqueues the messages into the AIA_PAYABLEINVOICE_AQ queue. The CreatePayableInvoiceListLogisticsAQConsumer service dequeues the messages and invokes the CreatePayableInvoiceListLogisticsReqABCSImpl service.

2. The CreatePayableInvoiceListLogisticsReqABCSImpl service transforms the OTM PayableInvoiceListLogisticsABM into the CreatePayableInvoiceListEBM, populates the enterprise business message (EBM) Header, updates the cross-reference data, and invokes the PayableInvoiceEBS with the CreatePayableInvoiceList operation in fire and forget mode.

3. The PayableInvoiceEBS service with the CreatePayableInvoiceList operation routes the messages based on the Composite Application Validation Service (CAVS) flag to either the CreatePayableInvoiceListEbizProvABCSImpl service or the CAVS simulator in an synchronized fire and forget mode.

4. The CreatePayableInvoiceListEbizProvABCSImpl transforms the CreatePayableInvoiceListEBM into the Oracle E-Business Suite PayableInvoiceListEbizABM, and invokes the CreatePayableInvoiceListEbizDBAdapter.

5. The CreatePayableInvoiceListEbizDBAdapter service receives the PayableInvoiceListEbizABM and makes insert calls to the Oracle E-Business Suite accounts payable interface table.

4.3 Assumptions and Constraints

These are the solution assumptions and constraints:

- OTM sends only approved vouchers for payment.

- OTM creates the vouchers and enqueues them in the AIA_PAYABLEINVOICE_ AQ queue based on user defined agents.

- If any errors occur and the voucher does not reach Oracle EBS interface tables, the AIA error handling framework notifies you.

  You must manually re-submit the transactions that failed. In case a batch of vouchers is sent in a single transmission, if a single voucher fails, the whole batch fails and should be sent again.

- This integration does not perform any validations and raise errors due to any business validation failure in OTM or in Oracle EBS.

  For example, if the Oracle EBS invoice import process rejects the invoice due to a validation error, you should manually correct the data in the Oracle EBS interface tables and re-run the import process.

- This integration supports only the creation of a list of payable invoices. Update and query of payable invoices is not supported in this integration.

- OTM matches the invoices against shipments and sends the approved vouchers to Oracle EBS AP based on a user-defined agent. Duplicate payment checking is done in OTM.

- OTM only sends single currency invoices for payment. As the OTM requester ABCS does not validate if the currency is different at the line level and at the header level, OTM must reject the invoices that have different currency at the header and the line level.

- An invoice can only belong to one operating unit. Therefore, if there are consolidated invoices with multiple operating units, the payments are made from only one operating unit.

- Payment terms are set up and maintained in Oracle EBS AP. OTM does not send the payment terms with an invoice; therefore, the payment terms is defaulted by Oracle EBS AP system.

- OTM does not send the invoice due date information to Oracle E-Business AP. Oracle EBS AP derives the due date of an invoice using a setup of the default terms and date at the supplier site level.

  For auto pay, the invoice received date in OTM is blank; therefore, you must set up the suppliers site payment terms based on the system date.

- Payment method is maintained in Oracle EBS AP and can be overridden at the transaction level.

- Supplier pay site is derived by Oracle EBS AP based on the supplier number and the operating unit included in the invoice sent by OTM.

- If an invoice comes into Oracle E-Business AP for an inactive supplier or supplier site, Oracle EBS AP rejects the invoice.

- OTM populates the reference number to indicate the operating unit with a reference qualifier type as OP_UNIT based on the agents defined to copy this value from order release to shipment and invoice.

- All invoices are assigned the default invoice type of standard, unless there is a negative amount at the voucher level and then the invoice type is a credit memo.

- For consolidated invoices, the parent invoice in OTM is the invoice header in Oracle EBS AP and the child invoice lines in OTM are the invoice lines in Oracle EBS AP.

- For a prepayment application, OTM should send a negative line amount and the Oracle AP prepayment invoice number as a reference number in the reference qualifier name PREPAY_INV_NUMBER.

  This is considered a prepay line type and Oracle EBS AP applies the prepayment and reduce the total amount. If you do not put a reference number value in the invoice, the invoice is treated as a negative invoice and a prepayment is not applied.

- Only one prepayment application invoice line is allowed for each invoice. If there are multiple prepayment invoice lines in an invoice, it fails within the Oracle E-Business AP application.

- Users should use OTM for cancellations (either reverse, or issuing a credit note) instead of using standard Oracle E-Business Invoice cancellation functionality.

- For consolidated invoices, if a parent invoice has two or more invoices and each invoice has two or more lines, then there are four or more lines and one header in Oracle E-Business AP.

  For example, if an OTM consolidated invoice has two invoices and each invoice has two lines of $25 each, then there is an invoice of $100 in Oracle E-Business AP with four lines of $25 each.

- The OTM invoice number and the voucher XID should not be more than 50 characters. Oracle EBS invoice number should be unique and should not be more than 50 characters.

4.4 Oracle EBS Interfaces

These are the Oracle EBS interfaces for the payable invoices integration flow. The schema is based on these Oracle EBS tables:

- AP_INVOICES_INTERFACE

- AP_INVOICE_LINES_INTERFACE

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

4.5 OTM Interfaces

OTM uses PayableInvoiceListOTMABM to send approved vouchers for payment to accounts payable financials system. GLogXML.xsd, element - Transmission, subelement - Voucher

For more information about the logistics service, see *Oracle Transportation Management Integration Guide*.

4.6 Core Application Integration Architecture Components

The integration flow uses these components:

- PayableInvoiceEBO

- CreatePayableInvoiceListEBM

- PayableInvoiceEBS

The core enterprise business object (EBO) and enterprise business message (EBM) XSD files can be located by EBO within the $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/ parent folder.

The core enterprise business services (EBS) web services definition language (WSDL) files can be located by EBO within the $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/ parent folder.

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

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

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

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

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Extensibility for AIA Artifacts".

4.7 Integration Services

These are the services delivered with this integration:

- CreatePayableInvoiceListLogisticsAQConsumer

- CreatePayableInvoiceListLogisticsReqABCSImpl

- PayableInvoiceEBS

- CreatePayableInvoiceListEbizProvABCSImpl

- CreatePayableInvoiceListEbizDBAdapter

- UpdatePayableInvoiceListEbizXref

- PayableInvoiceResponseEBS

4.7.1 CreatePayableInvoiceListLogisticsAQConsumer

This mediator service dequeues the InvoiceListLogisticsABM message whenever OTM enqueues a message into AIA_INVOICE_AQ queue. This service takes the PayableInvoiceListLogisticsABM from the wrapper AQ xsd, which contains the entire transmission element as CLOB. This service invokes the CreateInvoiceListLogisticsReqABCSImpl service.

4.7.2 CreatePayableInvoiceListLogisticsReqABCSImpl

The CreatePayableInvoiceListLogisticsReqABCSImpl is a Business Process Execution Language (BPEL) process and a single operations service that has the PayableInvoiceEBS as a partner service. This service receives the PayableInvoiceListLogisticsABM message as a request and does not return a response to the calling service.

This service performs these actions:

- Accepts the PayableInvoiceListLogisticsABM message from OTM.

- Transforms the PayableInvoiceListLogisticsABM into the CreatePayableInvoiceListEBM. While it is transforming from the application business message (ABM) to the enterprise business message (EBM), cross-references are populated for the: PAYABLEINVOICE_PAYABLEINVOICEID.

- Sends the CreatePayableInvoiceListEBM message as an input to the CreatePayableInvoiceList operation in the PayableInvoiceEBS service.

These DVM lookups are used by this service:

- PAYMENTMETHOD_CODE: domain value mapping for the payment method codes.

- PAYABLEINVOICE_PAYABLEINVOICELINETYPE: domain value mapping for the payable invoice line types.

4.7.3 PayableInvoiceEBS

The PayableInvoiceEBS is an enterprise business service that exposes all the enterprise operations related to the payable invoice like create payable-invoice, update payable-invoice, and so on. This integration uses only the CreatePayableInvoiceList operation. This service routes the request to the appropriate provider like the CreatePayableInvoiceListEbizProvABCSImpl or the Composite Application Validation System (CAVS) based on the filter condition and operations. No transformations are done in this service.

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

4.7.4 CreatePayableInvoiceListEbizProvABCSImpl

The CreatePayableInvoiceListEbizProvABCSImpl is a BPEL process, which receives the CreatePayableInvoiceListEBM, transforms the message into the PayableInvoiceEbizABM, and invokes the CreatePayableInvoiceListEbizDBAdapter service.

These DVM lookups are used by this service:

- PAYABLEINVOICE_PAYABLEINVOICETYPE: domain value mapping for the invoice type (that is standard, credit memo).

- PAYMENTMETHOD_CODE: domain value mapping for the payment method codes.

- PAYABLEINVOICE_PAYABLEINVOICELINETYPE: domain value mapping for the payable invoice line types.

4.7.5 CreatePayableInvoiceListEbizDBAdapter

This service accepts the PayableInvoiceListEbizABM message from the CreatePaybleInvoiceListEbizProvABCSImpl and inserts the data into the Oracle EBS AP interface tables.

4.7.6 UpdatePayableInvoiceListEbizXref

The UpdatePayableInvoiceListEbizXref service is a Business Process Execution Language (BPEL) process. When the payables open interface program successfully imports an invoice, it raises the business event oracle.apps.ap.invoice.import. The subscription to this event calls the UpdatePayableInvoiceListEbizXref service. This service using the UpdatePayableInvoiceListEbizXrefAdapter, fetches the invoice IDs and corresponding AIA global unique identifier (GUID), and updates to the PAYABLEINVOICE_PAYABLEINVOICEID cross-reference table.

If you choose not to use this service, you can disable the event subscription in the Oracle EBS application.

4.7.7 PayableInvoiceResponseEBS

The PayableInvoiceResponseEBS is the enterprise business service (EBS) used to route all payable invoice response related actions to the requesting application like create payable-invoice response, update payable-invoice response, delete payable-invoice

response, and synchronize payable-invoice response.
CreatePayableInvoiceListResponse is implemented as a part of this integration.

5

Process Integration for Receivable Bills

This chapter provides an overview of the process integration for receivable bills and discusses business process flows, assumptions and constraints, Oracle E-Business Suite (Oracle EBS) and Oracle Transportation Management (OTM) interfaces, core Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 5.1, "Overview"
- Section 5.2, "Business Process Flows"
- Section 5.3, "Assumptions and Constraints"
- Section 5.4, "Oracle EBS Interfaces"
- Section 5.5, "OTM Interfaces"
- Section 5.6, "Core Application Integration Architecture Components"
- Section 5.7, "Integration Services"

5.1 Overview

The receivable bills integration flow allows OTM to manage and bill the sell-side settlement process by approving bills and sending the transaction to the financial application for collection and management of uncollectibles.

5.2 Business Process Flows

The process integration for receivable bills supports these integration flows:

- Manual Bill

  User creates the bill manually based on the sell shipment. OTM approves the bill amount to be received from the customer:

  1. Changes the bill status to *issued*.

  2. Sends the transaction using the BillingXML to Oracle EBS AR.

- Auto Bill

  OTM creates a sell shipment:

  1. Creates a bill based on the sell shipment.

  2. Approves the bill for the amount to be received from the customer.

3. Changes the bill status to *issued*.

4. Sends the transaction using the BillingXML to Oracle EBS AR.

- Consolidated Bill

    OTM creates a sell shipment:

    1. Creates a consolidated bill for multiple sell shipments.

    2. Approves the bill for the amount to be received from the customer.

    3. Changes the bill status to *issued*.

    4. Sends the transaction using the BillingXML to Oracle EBS AR.

- Credit Memo

    OTM creates a credit memo for over payment

    1. Matches it with the original shipment.

    2. Creates the bill with the status of *process hold*. The user approves the bill and creates the approved bill.

    3. Changes the approved bill status to *issued*.

    4. Sends the transaction using the BillingXML to Oracle EBS AR.

Figure 5–1 shows the process integration for receivable bills:

Figure 5–1   Receivable Bills Process Integration Flow
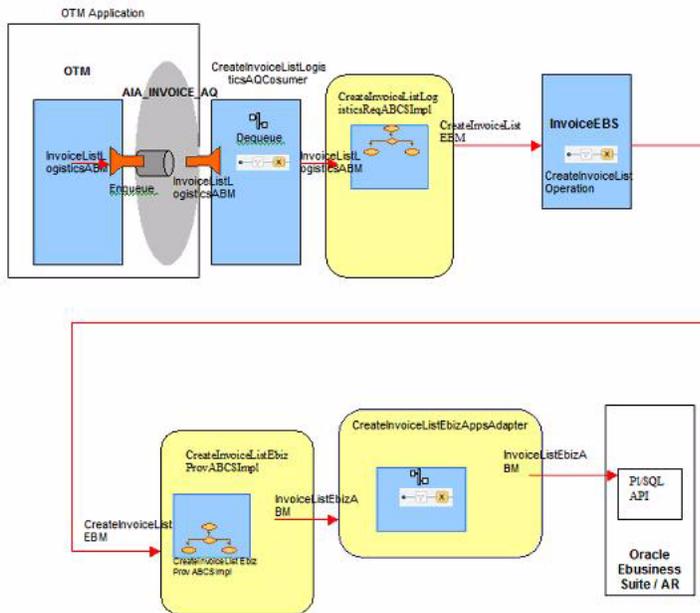
5.2.1 Sending Receivable Bills into Oracle EBS Account Receivables

The receivable bills process integration flow creates the bill in Oracle EBS Account Receivables (AR) based on the customer bill sent by OTM.

Figure 5–2 is a sequence diagram that shows the process integration for receivable bills:

Figure 5–2   OTM (Logistics) to Oracle EBS Receivable Bill Flow
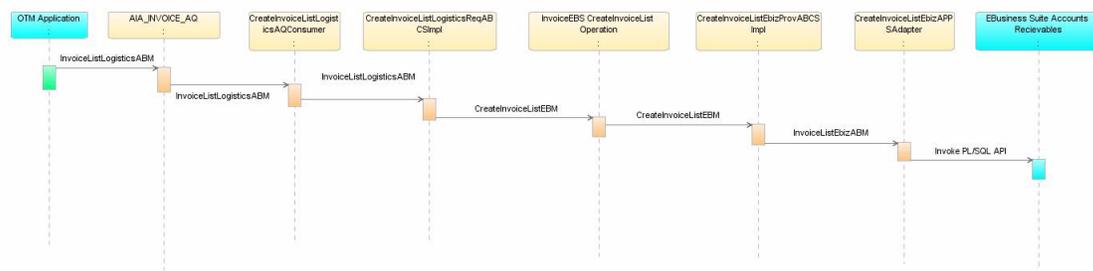
When you initiate the process these events occur:

1. OTM enqueues the messages into the AIA_INVOICE_AQ queue. This service dequeues the messages and invokes the CreateInvoiceListLogisticsReqABCSImpl service.

2. The CreateInvoiceListLogisticsReqABCSImpl service transforms the OTM CreateInvoiceListLogisticsABM into the CreateInvoiceListEBM, populates the enterprise business message (EBM) Header, updates the cross-reference data, and invokes the InvoiceEBS with the CreateInvoiceList operation in fire and forget mode.

3. The InvoiceEBS service with the CreateInvoiceList operation routes the messages based on the Composite Application Validation System (CAVS) flag to either the CreateInvoiceListEbizProvABCSImpl service or the CAVS simulator in an asynchronized fire and forget mode.

4. The CreateInvoiceListEbizProvABCSImpl transforms the CreateInvoiceListEBM into the Oracle EBS InvoiceListEbizABM, and invokes the CreateInvoiceListEbizAppsAdapter.

5. The CreateInvoiceListEbizAppsAdapter service receives the InvoiceListEbizABM, sets the apps context, and invokes the Oracle EBS AR PL/SQL API - AR_EBI_ INVOICE_PUB. ar_invoice_submission.

5.3 Assumptions and Constraints

These are the solution assumptions and constraints:

- OTM sends only approved bills for collection.

- OTM creates the bill and enqueues them in the named queue in the AIA_ INVOICE_AQ based on user defined agents.

- If any errors occur in the service layer and the bills do not reach Oracle EBS interface tables, the AIA error handling framework notifies you.

  You should manually re-submit the transactions that failed. In case that a batch of bills is sent in a single transmission, if a single bill fails, the whole batch fails and should be sent again.

- This integration does not perform any validations and raise errors due to any business validation failure in OTM or in Oracle EBS.

  For example, if the Oracle EBS invoice import process rejects the bill due to a validation error, you should manually correct the data in the Oracle EBS interface tables and re-run the import process.

- This integration supports only the creation of a list of receivable invoices. Update and query of receivable invoices is not supported in this integration.

- OTM matches the bills against shipments and sends the approved bills to Oracle EBS AR based on a user-defined agent. Duplicate bill checking is done in OTM.

- OTM only sends single currency bills for collection. Since the OTM requester ABCS does not validate if the currency is different at the line level and at the header level; OTM must reject the bills that have different currency at the header and line level.

- A bill can only belong to one operating unit. Therefore, if there are consolidated bills with multiple operating units, the collections are made for only one operating unit.

- OTM sends only bills and credit memos to Oracle EBS AR, other type of transactions - that is debit memos, must be created in Oracle EBS.

- In Oracle EBS AR, all imported bills are assigned the default transaction type (invoice). When there is negative amount, the bill has transaction type of credit memo.

- When there are adjustments to a bill, a new bill with a different invoice number must be created.

- Payment method and terms are maintained in Oracle EBS AR; therefore, imported bills use the default payment method and terms for the customer.

- If there is no value in the ServiceProviderGID in the OTM XML, the default legal entity for the operating unit is used as the service provider.

- OTM populates the reference number to indicate the operating unit with a reference qualifier type as OP_UNIT based on the agents defined to copy this value from order release to shipment and bill.

- For consolidated bills, the parent bill in OTM is the bill header in Oracle EBS AR and the child bill lines in OTM are the bill lines in Oracle EBS AR.

- All consolidated bills in AR must have the same due date.

- All bills within the consolidated bill must contain the same currency.

- An AR bill number should be no longer than 20 characters. Hence, OTM should ensure to send a bill with a bill number with less than 20 characters, else the process fails.

5.4 Oracle EBS Interfaces

The schema is based on AR_EBI_INVOICE_PUB.AR_INVOICE_SUBMISSION API.

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

5.5 OTM Interfaces

OTM uses InvoiceListLogisticsABM to send approved bills for collection to accounts receivable financials system. GLogXML.xsd, element - Transmission, sub-element - Billing

For more information about the logistics service, see *Oracle Transportation Management Integration Guide*.

5.6 Core Application Integration Architecture Components

The integration flow uses these components:

- InvoiceEBO
- CreateInvoiceListEBM
- InvoiceEBS

The core enterprise business object (EBO) and enterprise business message (EBM) XSD files can be located by EBO within the $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/ parent folder.

The core enterprise business services (EBS) web services definition language (WSDL) files can be located by EBO within the $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/ parent folder.

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

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."

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

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Extensibility for AIA Artifacts".

5.7 Integration Services

These are the services delivered with this integration:

- CreateInvoiceListLogisticsAQConsumer
- CreateInvoiceLogisticsReqABCSImpl
- InvoiceEBS
- CreateInvoiceListEbizProvABCSImpl
- CreateInvoiceListEbizAppsAdapter
- UpdateInvoiceListEbizXref
- InvoiceResponseEBS

5.7.1 CreateInvoiceListLogisticsAQConsumer

This mediator service is implemented to route the message whenever OTM enqueues a message into the AIA_INVOICE_AQ queue. This service takes the BillingXML element from the wrapper AQ xsd and invokes the CreateInvoiceListLogisticsReqABCSImpl service.

5.7.2 CreateInvoiceListLogisticsReqABCSImpl

The CreateInvoiceListLogisticsReqABCSImpl is a Business Process Execution Language (BPEL) process and a single operations service that has the InvoiceEBS as a partner service. This service receives the CreateInvoiceListLogisticsABM message as a request and does not return a response to the calling service.

This service performs these actions:

■ Accepts the CreateInvoiceListLogisticsABM message from OTM. The cross-reference table is populated for the unique identifier at the receivable invoice header level.

■ Transforms the CreateInvoiceListLogisticsABM into the CreateInvoiceListEBM. While it is transforming from the application business message (ABM) to the enterprise business message (EBM), cross-references are looked up for INVOICE_INVOICEID.

■ Sends the CreateInvoiceListEBM message as an input to the CreateInvoiceList operation in the InvoiceEBS service.

■ The INVOICE_INVOICELINETYPE: domain value mapping for the receivable invoice line type DVM is used by this service.

5.7.3 InvoiceEBS

The InvoiceEBS is an Enterprise Business Service that exposes all the enterprise operations related to the invoice like create invoice, update invoice, and so on. This integration uses only CreateInvoiceList operation. This Enterprise Business Service routes the request to the appropriate provider like the CreateInvoiceListEbizProvABCSImpl or the Composite Application Validation System (CAVS) based on the filter condition and operations. No transformations are done in this service.

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

5.7.4 CreateInvoiceListEbizProvABCSImpl

The CreateInvoiceListEbizProvABCSImpl is a Business Process Execution Language (BPEL) process, which receives the CreateInvoiceListEBM, transforms the message into the InvoiceListEbizABM, and invokes the CreateInvoiceListEbizAppsAdapter service.

These DVM lookups are used by this service:

■ INVOICE_INVOICETYPE: domain value mapping for the invoice type (standard, credit memo, and so on).

■ INVOICE_INVOICELINETYPE: domain value mapping for the receivable invoice line types.

5.7.5 CreateInvoiceListEbizAppsAdapter

This service accepts the InvoiceListEbizABM message from CreateInvoiceListEbizProvABCSImpl. It sets the application context and invokes the Oracle EBS PL/SQL API AR_EBI_INVOICE_PUB.ar_invoice_submission.

5.7.6 UpdateInvoiceListEbizXref

The UpdateInvoiceListEbizXref service is a BPEL process. When the receivables' AutoInvoice master program successfully imports an invoice, it raises a business event oracle.apps.ar.batch.AutoInvoice.run. The subscription to this event calls the UpdateInvoiceListEbizXref service. This service uses UpdateInvoiceListEbizXrefAdapter and fetches the invoice_IDs and corresponding AIA global unique identifier (GUID), and updates the cross-reference table.

If customers choose not to use this service, they can disable the event subscription in Oracle EBS application.

5.7.7 InvoiceResponseEBS

The InvoiceResponseEBS is the enterprise business service (EBS) used to route all invoice response related actions to the requesting application like create invoice-response, update invoice-response, delete invoice-response, and synchronize invoice-response. CreateInvoiceListResonse is implemented as a part of this integration.

6

Process Integration for Accruals and Reversals

This chapter provides an overview of the process integration for accruals and reversals and discusses business process flows, assumptions and constraints, Oracle E-Business Suite (Oracle EBS) and Oracle Transportation Management (OTM) interfaces, core Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 6.1, "Overview"

- Section 6.2, "Business Process Flows"

- Section 6.3, "Assumptions and Constraints"

- Section 6.4, "Oracle EBS Interfaces"

- Section 6.5, "OTM Interfaces"

- Section 6.6, "Core Oracle AIA Components"

- Section 6.7, "Integration Services"

6.1 Overview

The accruals and reversals integration flow allows OTM to send accruals and reversals transactions to the Oracle EBS General Ledger (GL) so that the accounting entries are generated in the general ledger.

The process integration for accruals and reversals supports these integration flows:

- Cost Accrual

OTM receives an order:

1. Plans a buy shipment.

2. Allocates the shipment.

3. Creates the accrual.

4. Sends the transaction using the AccrualXML to the Oracle EBS General Ledger.

   Once the shipment is delivered and completed, OTM creates a voucher for payment and sends it to Oracle EBS AP. The accounting for the invoice in Accounts Payable (AP) also triggers the accrual reversal transaction in the general ledger.

Here is an example with the steps needed to send an accrual when you create an Order Release in OTM and then do the Buy Shipment.

1. Once you create the Buy Shipment, click **Actions** to allocate cost for this Buy Shipment.

   This action creates the accrual in OTM, which is sent to Oracle EBS.

2. Click **Allocate Cost** to create the cost accrual in OTM.

   A confirmation is displayed that the cost was allocated in OTM implying that the cost accrual is ready to be sent to Oracle EBS.

3. Select **Business Process Automation**.

4. Click **Process Management**.

5. Click **Send Integration** to manually send the Cost Accrual to Oracle EBS.

6. Click the button to the right of the **Interface Type** field.

7. Click the scroll bar button to select the **Order Accrual interface** type.

8. Select **Order Accrual** entry in the list.

9. Click **Define**.

10. Select **Order Release** tab.

11. Enter the Order Release ID in the **Order Release ID** field.

12. Click **Save**.

13. Enter the COSTACCRUAL in the **Query Name** field.

14. Click **Save**.

15. Click **Select**.

16. Select **QUEUE** from the **Notify Method** drop down.

    OTM creates the accruals and enqueues them in the AIA_ACCOUNTINGENTRY_ AQ queue based on a scheduled Send Integration in the Process Management.

17. Enter the AIA_ACCOUNTINGENTRY_AQ in the **Receiving External System** field.

18. Click **Submit**.

19. A confirmation message that the Cost Accrual has been sent is displayed.

    The cost accrual was queued in the AIA_ACCOUNTINGENTRY_AQ queue. Now, the AIA layer consumes the inbound messages and transforms the data elements into the AccountingEntryEBO using the DVMs to translate values.

- Updated Cost Accrual

  OTM receives an order:

  1. Plans a buy shipment.

  2. Allocates the shipment.

  3. Creates the accrual.

  4. Sends the transaction using the AccrualXML to the Oracle EBS General Ledger.

The accessorial charge is added or changed for the shipment; therefore, OTM re-allocates the cost and creates a transaction that is sent using the VoucherXML to the Oracle EBS General Ledger.

Once the shipment is delivered and completed, OTM creates a voucher for payment and sends it to Oracle EBS AP. The accounting for the invoice in AP also triggers the accrual reversal transaction in the General Ledger.

- Recognize Revenue

  OTM receives an order:

  1. Plans a sell shipment.

  2. Allocates the shipment.

  3. Creates the accrual.

  4. Sends the transaction using the AccrualXML to the Oracle EBS General Ledger.

  Once the shipment is delivered and completed, OTM creates a bill for collection and sends it to Oracle EBS AR. The accounting for the bill in AR also triggers the accrual reversal transaction in the General Ledger.

- Updated Revenue Recognition

  OTM receives an order:

  1. Plans a sell shipment.

  2. Allocates the shipment.

  3. Creates the accrual.

  4. Sends the transaction using the AccrualXML to the Oracle EBS General Ledger. The accessorial charge is added or changed for the shipment; therefore, OTM re-allocates the cost and creates a transaction that is sent using the VoucherXML to the Oracle EBS General Ledger.

  Once the shipment is delivered and completed, OTM creates a bill for collection and sends it to Oracle EBS AR. The accounting for the bill in AR also triggers the accrual reversal transaction in the General Ledger.

6.2 Business Process Flows

Figure 6–1 shows the process integration flow for accruals and reversals:

Figure 6–1   Accruals and Reversals Process Integration Flow

6.2.1 Accruals and Reversals Integration Details

The accruals and reversals integration flow creates the accrual transactions in Oracle E-Business GL based on the accruals or reversals sent from OTM.

Figure 6–2 is a sequence diagram that shows the accruals and reversals flow from OTM to Oracle EBS:

Figure 6–2   Oracle Transportation Management (Logistics) to Oracle EBS General Ledger Accounting Entry Flow

When you initiate the process, these events occur:

1. OTM sends the messages into the AIA_ACCOUNTINGENTRY_AQ queue based on the scheduled send integration in process management.

2. The CreateAccountingEntryListLogisticsAQConsumer service dequeues the messages and invokes the CreateAccountingEntryListLogisticsReqABCSImpl service.

3. The CreateAccountingEntryListLogisticsReqABCSImpl service transforms the accrual application business message (ABM) into the

CreateAccountingEntryListEBM, populates the enterprise business message (EBM) Header, updates the cross-reference data, and invokes the AccountingEntryEBS with the CreateAccountingEntryList operation in fire and forget mode.

4. The AccountingEntryEBS service with the CreateAccountingEntryList operation routes the messages based on the Composite Application Validation System (CAVS) flag to either the CreateAccountingEntryListEbizProvABCSImpl service or the CAVS simulator in an asynchronized fire and forget mode.

5. The CreateAccountingEntryListEbizProvABCSImpl transforms the CreateAccountingEntryListEBM into the Oracle EBS CreateAccountingEntryListEbizABM and invokes the CreateAccountingEntryListEbizDBAdapter service.

6. The CreateAccountingEntryListEbizDBAdapter service receives the AccountingEntryListEbizABM and makes insert calls to the Oracle EBS General Ledger interface table.

6.3 Assumptions and Constraints

These are the solution assumptions and constraints:

- OTM creates the accruals and enqueues them in the AIA_ACCOUNTINGENTRY_ AQ queue based on a scheduled send integration in process management.

- If any errors occur in the service layer and the accruals do not reach Oracle EBS interface tables, then the AIA error handling framework notifies you.

  You should manually re-submit the transactions that failed. In case that a batch of bills is sent in a single transmission, if a single bill fails, the whole batch fails and should be sent again.

- This integration does not perform any validations and raise errors due to any business validation failure in OTM or in Oracle EBS.

  For example, if the Oracle EBS journal import process rejects the accrual due to a validation error, you should manually correct the data in the Oracle EBS interface tables and re-run the import process.

- This integration supports only the creation of a list of accounting entries (accruals). Update and query of accounting entries is not supported in this integration.

- For each accrual coming from OTM in Oracle EBS, two accrual lines are created in GL Interface table, one as a credit line, and one as a debit line. You must use the IDT tool to derive the account values based on the perspective (BUY versus SELL), the **is_reversal** flag, and if it is delta/base cost.

- The credit and debit accounts of the accruals are derived by IDT based on the information sent by OTM.

- IDT derives the period name and ledger ID.

- OTM should ensure to send the accrual only once. If OTM sends the same accrual ID more than once, the data is imported successfully to GL and separate journal batches are created. You must review the journals for correct accounting and reverse the incorrect entries.

6.4 Oracle EBS Interfaces

These are the Oracle EBS interfaces for the accruals and reversals integration flow. The schema is based on this Oracle EBS table: GL_INTERFACE.

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

6.5 OTM Interfaces

OTM uses the AccountingEntryListLogisticsABM to send the estimated cost and revenues to any financial systems general ledger for accounting. *GLogXML*.xsd, element - Transmission, sub-element - Accrual.

For more information about the logistics service, see *Oracle Transportation Management Integration Guide*.

6.6 Core Oracle AIA Components

The integration flow uses these components:

- AccountingEntryEBO
- AccountingEntryListEBM
- AccountingEntryEBS

The core enterprise business object (EBO) and enterprise business message (EBM) XSD files can be located by EBO within the $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/ parent folder.

The core enterprise business services (EBS) web services definition language (WSDL) files can be located by EBO within the $AIA_HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/ parent folder.

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

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."

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

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Extensibility for AIA Artifacts".

6.7 Integration Services

These are the services delivered with this integration:

- CreateAccountingEntryListLogisticsAQConsumer
- CreateAccountingEntryListLogisticsReqABCSImpl

- AccountingEntryEBS
- CreateAccountingEntryListEbizProvABCSImpl
- CreateAccountingEntryListEbizDBAdapter
- AccountingEntryResponseEBS

6.7.1 CreateAccountingEntryListLogisticsAQConsumer

This mediator service is implemented to route the message whenever OTM enqueues a message into the AIA_ACCOUNTINGENTRY_AQ queue. This service takes the AccrualXML from the wrapper AQ xsd and invokes the CreateAccountingEntryListLogisticsReqABCSImpl service.

6.7.2 CreateAccountingEntryListLogisticsReqABCSImpl

The CreateAccountingEntryListLogisticsReqABCSImpl is a Business Process Execution Language (BPEL) process and a single operations service that has the AccountingEntryEBS as a partner service. This service receives the CreateAccountingEntryListLogisticsABM message as a request and does not return a response to the calling service.

This service performs these actions:

- Accepts the AccrualABM message from OTM. The cross-reference table is populated for the unique identifier of the accounting entry element.

- Transforms the AccrualABM into the CreateAccountingEntryListEBM. While it is transforming from the application business message (ABM) to the enterprise business message (EBM), cross-references are looked up for ACCOUNTINGENTRY_ ACCOUNTINGENTRY ID.

- Sends the CreateAccountingEntryListEBM message as an input to the CreateAccountingEntry operation in the AccountingEntryEBS service.

6.7.3 AccountingEntryEBS

The AccountingEntryEBS is an Enterprise Business Service that exposes all the enterprise operations related to the accounting entry like create accounting-entry, update accounting-entry, and so on. This integration uses only CreateAccountingEntryList operation. This Enterprise Business Service routes the request to the appropriate provider like the CreateAccountingEntryListEbizProvABCSImpl or the CAVS based on the filter condition and operations. No transformations are done in this service.

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

6.7.4 CreateAccountingEntryListEbizProvABCSImpl

The CreateAccountingEntryListEbizProvABCSImpl is a Business Process Execution Language (BPEL) process, which receives the CreateAccountingEntryListEBM, transforms the message into the CreateAccountingEntryListABM, and invokes the CreateAccoutingEntryListEbizDBAdapter service.

These DVM lookups are used by this service:

- ACCOUNTINGENTRY_ACCOUNTINGENTRYTYPE: domain value mapping for the accrual type (that is actual, budget, and so on).

- ACCOUNTINGENTRY_JOURNALCATCODE: domain value mapping for the journal category code (that is accrual, adjustment, and so on).

6.7.5 CreateAccountingEntryListEbizDBAdapter

This service accepts the AccountingEntryListABM, message from the CreateAccountingEntryListEbizProvABCSImp and inserts the data into the GL_INTERFACE table in Oracle EBS.

6.7.6 AccountingEntryResponseEBS

The AccountingEntryResponseEBS is the enterprise business service (EBS) used to route all accounting entry response related actions to the requesting application like create accounting-entry response, update accounting-entry response, delete accounting-entry response, and synchronize accounting-entry response. CreateAccountingEntryResponse is implemented as a part of this integration.

7

Process Integration for Customer Synchronization

This chapter provides an overview of the process integration for customer synchronization and discusses business process flows, assumptions and constraints, Oracle E-Business Suite (Oracle EBS) and Oracle Transportation Management (OTM) interfaces, core Oracle Application Integration Architecture (Oracle AIA) components, and integration services.

This chapter includes the following sections:

- Section 7.1, "Overview"
- Section 7.2, "Business Process Flows"
- Section 7.3, "Assumptions and Constraints"
- Section 7.4, "Oracle EBS Interfaces"
- Section 7.5, "OTM Interfaces"
- Section 7.6, "Core Oracle AIA Components"
- Section 7.7, "Integration Services"

7.1 Overview

The customer synchronization integration supports these operations and corresponding synchronization of customer records from Oracle EBS to OTM:

- Synchronize Customer: Defines the ability to synchronize the new or updated records from Oracle EBS to OTM. Synchronization represents a single service to perform either a Create or an Update call depending on the existence of the customer in the source and target applications.

- Merge Customer: When customers are merged in Oracle EBS and are of the designated type, the data is synchronized to OTM. The types of mergers that can take place in Oracle EBS are:

  - Duplicate records are created for the same account or location and merged, making one of them inactive.

  - Two or more companies merge. As the result, there is a losing account and a winning account. All locations for the losing account (corporation in OTM) are modified to refer to the winning account.

  - Inactivate Customer: When an account is inactivated in Oracle EBS it is recorded in OTM. When customer locations are made inactive, the date of inactivation is recorded on the location within OTM. You must decide exactly

how to use this information in OTM (like setting a status, placing orders on hold, and so on).

7.2 Business Process Flows

Figure 7–1 is a flow diagram that shows the customer flow synchronization.

Figure 7–1   Customer Flow - Sync Operation
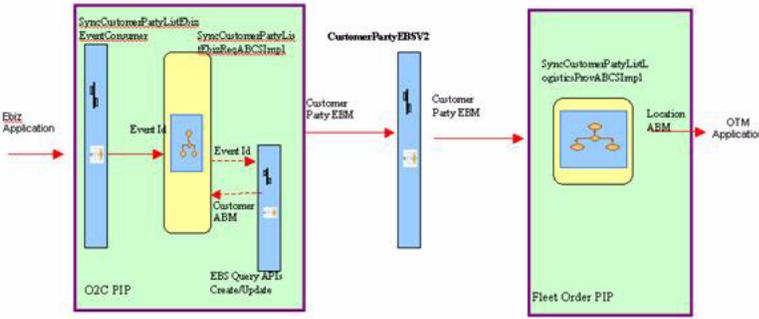
Figure 7–2 is a flow diagram that shows the customer flow for merge party:

Figure 7–2   Customer Flow - Merge Party Operation
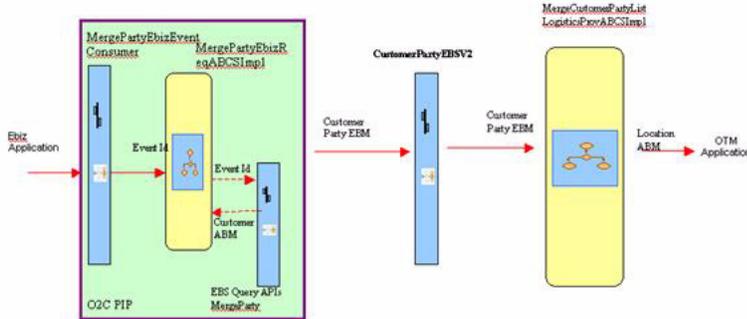
Figure 7–3 is a flow diagram that shows the customer flow for merge account:

Figure 7–3   Customer Flow - Merge Account Operation
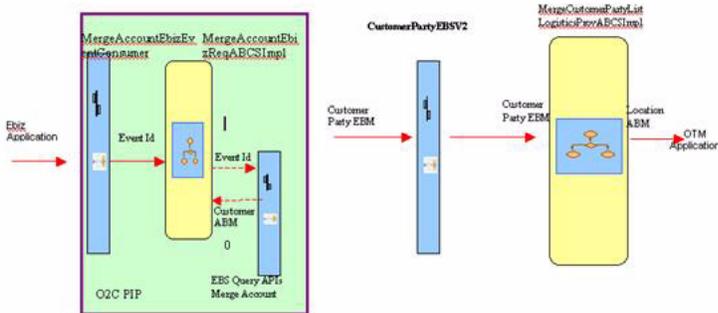
Figure 7–4 is a sequence diagram that shows the incremental changes in the customer party synchronization:

Figure 7–4   Incremental Changes - Customer Party Sync
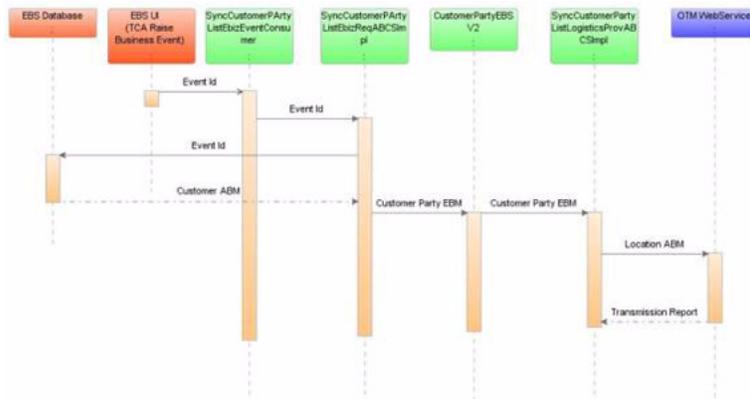
Figure 7–5 is a sequence diagram that shows the incremental changes in the customer party synchronization:

Figure 7–5   Incremental Changes - Customer Party Sync
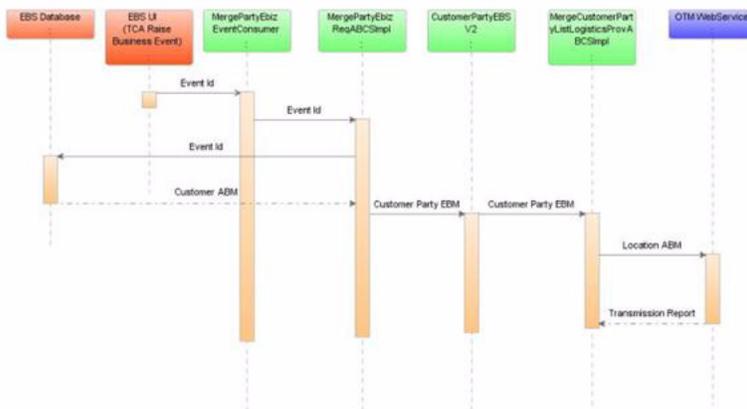
Figure 7–6 is a sequence diagram that shows the incremental changes in customer party merge:

Figure 7–6   Incremental Changes - Customer Party Merge
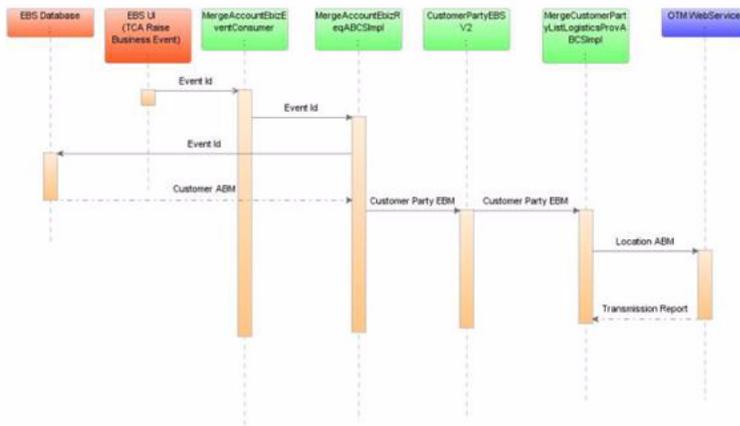
Figure 7–7 is a sequence diagram that shows the incremental changes in the customer account merge:

Figure 7–7   Incremental Changes - Account Merge
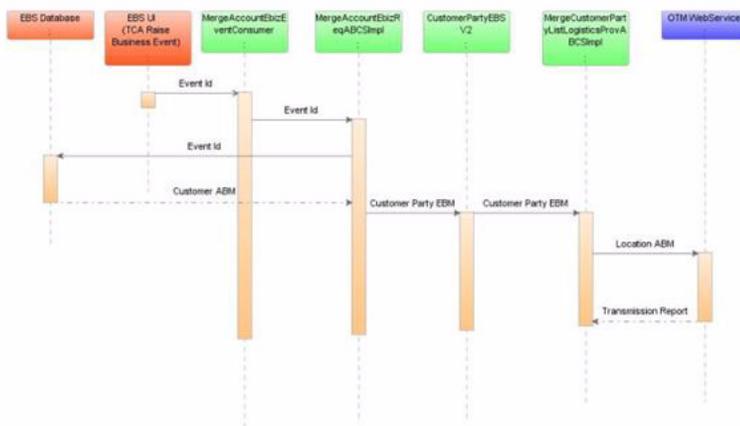
7.2.1 Order to Cash Integration Services

These services from Order to Cash (O2C) are available:

- SyncCustomerPartyListEbizEventConsumer

- MergeAccountEbizEventConsumer

- SyncCustomerPartyListEbizReqABCSImpl

- CustomerPartyEBSV2

- SyncCustomerPartyListLogisticsProvABCSImpl

- MergeAccountEbizReqABCSImpl

- MergePartyEbizReqABCSImpl

- MergeCustomerPartyListLogisticsProvABCSImpl

- CreatePayableInvoiceLogisticsReqABCSImpl

- QueryCustomerPartyListEbizCreateAdapter

- QueryCustomerPartyListEbizUpdateAdapter
- QueryCustomerPartyEbizAdapter
- MergeAccountEbizEventConsumer
- MergePartyEbizEventConsumer
- MergeAccountEbizReqABCSImpl
- MergePartyEbizReqABCSImpl
- TransformAppContextEbizService
- QueryMergeAccountEbizAdapter
- QueryPartyMergeEbizAdapter
- QueryMergeOrgCustEbizAdapter
- QueryRelatedOrgCustEbizAdapter

7.3 Assumptions and Constraints

These are the assumptions and constraints:

- Only parties of type organization are synchronized into OTM.
- Delete for the accounts (customers) is not covered in the synchronization.
- In case if any error occurs in the service layer and the customer message does not reach target application, then AIA error handling framework is invoked that notifies the administrator. The administrator should manually re-submit that transmission at various places that failed for re-processing.
- This integration does not validate and raise errors due to any business validation failure in OTM or Oracle EBS.
- O2C PIP code takes care of the requester transformation (from Oracle EBS to EBO) and Fleet Order PIP code takes care of OTM provider (EBO to OTM) transformation.
- The cross-references are populated based on the identifiers passed from main transformation to OTM. To pass a different value to the OTM identifier, you can use transformations extensibility to update the cross-reference.

7.4 Oracle EBS Interfaces

For more information about Oracle EBS web services definition language (WSDL) and schema definition, see Oracle Customer Master Data Management Integration documentation.

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

7.5 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. When the logistics web service is called it returns an acknowledgment with a transmission

number. Once the processing is complete, it then sends a transmission report back indicating the success or the failure.

For more information about the logistics service, see *Oracle Transportation Management Integration Guide.*

7.6 Core Oracle AIA Components

The integration flow uses these components:

- CustomerPartyEBO
- CustomerPartyEBM

The core enterprise business object (EBO) and enterprise business message (EBM) XSD files can be located by EBO within the $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseObjectLibrary/Core/EBO/ parent folder.

The core enterprise business services (EBS) web services definition language (WSDL) files can be located by EBO within the $AIA_ HOME/AIAMetaData/AIAComponents/EnterpriseBusinessServiceLibrary/Core/EBO/ parent folder.

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

For more information about using the Oracle Enterprise Repository and configuring it to provide the AIA Reference Doc link, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Configuring and Using Oracle Enterprise Repository as the Oracle AIA SOA Repository."

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

For more information, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*, "Extensibility for AIA Artifacts".

7.7 Integration Services

These are the services delivered with this integration:

- CustomerPartyEBSV2
- CustomerPartyResponseEBSV2
- MergeCustomerPartyListLogisticsProvABCSImpl

7.7.1 CustomerPartyEBSV2

CustomerPartyEBSV2 is the Enterprise Business Service to route all location related actions such as create customer, update customer, delete customer, and synchronize customer. The mediator service routes to SyncCustomerPartyListLogisticsProvABCSImpl in case of create and update operations of customer. The mediator service routes to MergeCustomerPartyListLogisticsProvABCSImpl in case of merge operations of customer. The mediator service routes to Composite Application Validation System (CAVS) based on the filter condition and operation.

7.7.2 CustomerPartyResponseEBSV2

CustomerPartyResponseEBSV2 is the Enterprise Business Service to route all customer related actions such as create customer, update customer, delete customer, and synchronize customer.

The mediator service routes the response message to MergeAccountEbizReqABCSImpl in case of merge of customer accounts. The mediator service routes to CAVS based on the filter condition and operation.

7.7.3 MergeCustomerPartyListLogisticsProvABCSImpl

MergeCustomerPartyListLogisticsProvABCSImpl is a Business Process Execution Language (BPEL) Process which receives SyncCustomerPartyListEBM as input from CustomerPartyEBSV2. This message is transformed to LogisticsABM. Cross-reference values are populated here. SyncCustomerPartyListEBM is transformed into CustomerPartyLogisticsABM and then the LogisticsWebService is invoked with this transformed application business message (ABM). This instance asynchronously waits for TransmissionReport from OTM. If the transaction is successful, the cross-reference values are populated; else, AIAAsyncErrorHandlingBPELProcess is invoked with an error message.

Part II

Implementing the Delivered Process Integrations

This part includes the following chapters:

8

Setting Up Participating Applications

This chapter provides information about how to set up Oracle E-Business Suite (Oracle EBS) and Oracle Transportation Management (OTM) in order for the Oracle Financial Management for OTM and Oracle EBS pre-built integration to function properly.

This chapter includes the following sections:

- Section 8.1, "Setting Up OTM"
- Section 8.2, "Setting Up Oracle EBS"

8.1 Setting Up OTM

Complete the following tasks to set up OTM:

- Enable security for fleet services
- Set up qualifiers
- Create contacts in OTM
- Set up external systems for queues
- Set up automation agents in OTM

8.1.1 Enabling Security for Fleet Services

- Inbound to OTM
- Outbound from OTM

To perform basic authentication (Inbound):
OTM 6.1 supports only basic http authentication.

1. Login to EM console. Go to **WebLogic Domain** --> **<soa_domain>**. Right click **<soa_domain>** --> **Security** --> **Credentials**.
2. Create **New Map**, enter a name and save it.
3. Click **Create Key**, enter **Key**, **Username** and **Password**.
4. Click the process in which you plan to make inbound call to OTM, and then click **<OTMWebService>** partnelink in **Services** and **References** section.
5. Click **Policies** section, and then click **Attach/Detach** to select **oracle/wss_http_token_client_policy** policy from the list. Click **Ok**.
6. Click the policy which you chose, and then enter the **Key** which you created in step **3**.

This takes the credentials from the key store and sends it to the http header while making an inbound call to OTM.

To perform basic authentication (Outbound):

1. Log in to OTM.

2. Go to **Business Process Automation**, **Communication Management**, **Web Services** and click **New**.

3. Click **New for WSDL Document**.

4. Click **Document Detail**.

    - Enter name for ID.

    - Upload the saved concrete WSDL.

    - Ensure **Storage** is set to *Text* and **Mime Type** as *text/xml*. Before uploading, you must add "style="document" to soap:binding element in your concrete WSDL.

5. Click **Finished**.

6. Click **Service Details** Fill **Service ID**, **Service Endpoint ID**, **Service Endpoint**, **User Name** and **Password** and save it. You must enter password after converting it using Base64 encoding.

7. Go to **Business Process Automation** --> **Communication Management** --> **External Systems**, and then click **New**.

8. Enter value for **External System ID** and select **Web Service**, **Operation** and **Service Endpoint** which are created in the step **7** and save it.

9. A **Contact** is automatically created in OTM with the same name that you have for **External System ID**. To verify, got to **Business Process Automation**, **Communication Management**, **Contacts**.

10. You must populate the **Contact ID** in **TransmissionHeader** while making an inbound call to OTM and expecting a **Transmission Report**.

11. Now you have to enable http authentication in FMW for your process.

12. Login to EM console and click the corresponding process.

13. Click **<TransmissionReport>** PartnerLink from **Services and References** section.

14. Click **Policies** section and then click **Attach/Detach** to select **oracle/wss_http_token_service_policy** policy from the list. Click **Ok**.

Example 8–1   Sample Transmission Header

```
<logisabm:TransmissionHeader>
<logisabm:SenderTransmissionNo>
<xsl:value-of select="orcl:generate-guid()"/>
</logisabm:SenderTransmissionNo>
<logisabm:AckSpec>
<logisabm:ComMethodGid>
<logisabm:Gid>
<logisabm:Xid>
<xsl:value-of select="string('SERVICE')"/>
</logisabm:Xid>
</logisabm:Gid>
</logisabm:ComMethodGid>
<logisabm:AckOption>
```

```
<xsl:value-of select="string('YES')"/>
</logisabm:AckOption>
<logisabm:ContactGid>
<logisabm:Gid>
<logisabm:DomainName>
<xsl:value-of select="'<DomainName where your Contact is created in OTM>'"/>
</logisabm:DomainName>
<logisabm:Xid>
<xsl:value-of select="'<Contact ID which is created in OTM>'"/>
</logisabm:Xid>
</logisabm:Gid>
</logisabm:ContactGid>
</logisabm:AckSpec>
<logisabm:IsProcessInSequence>
<xsl:value-of select="string('Y')"/>
</logisabm:IsProcessInSequence>
</logisabm:TransmissionHeader>
```

8.1.2 Setting Up Qualifiers

These qualifiers are used by OTM:

- Suppliers Integration Flow

  - ServiceProviderAliasQualifierXid = VENDOR_NAME_ALT: This qualifier value contains the alternate vendor name.

  - LocationRefnumXid = INACTIVE: This qualifier value contains the inactive date of supplier/site.

  - LocationRefnumXid = SUPPLIER_SITE_ID: This qualifier value contains the Site ID.

- Payable Invoices Integration Flow

  - ServiceProviderAliasQualifierXID = GLOG: This qualifier value contains the Supplier Number or Supplier Site Number (based on sync supplier site profile).

  - ServiceProviderAliasQualifierXID = SUPPLIER_ID: This qualifier value contains the Supplier Number (based on sync supplier site profile).

  - InvoiceRefNumQualifierXID = OP_UNIT: This qualifier value contains the AIA GUID That is ORGANIZATION_ID.COMMON value.

  - InvoiceRefNumQualifierXID = PREPAY_INV_NUMBER: This qualifier value contains the E-Business Prepayment Invoice Number

  - InvolvedPartyQualifierXID = REMIT-TO: This qualifier value contains the Remit to Supplier Number

- Receivable Invoices Integration Flow

  - InvoiceRefNumQualifierXID = OP_UNIT: This qualifier value contains the AIA GUID that is ORGANIZATION_ID.COMMON value.

  - InvolvedPartyQualifierXID = BILL-TO: This qualifier value contains the Bill to Customer Party ID.

  - InvolvedPartyQualifierXID = CUSTOMER: This qualifier value contains the Ship to Customer Party ID.

- Accruals and Reversals Integration Flow

■ ReleaseRefNumQualifierXID = OP_UNIT: This qualifier value contains the AIA GUID That is ORGANIZATION_ID.COMMON value.

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

8.1.3 Creating Contacts in OTM

Create external systems for all inbound flows to OTM. Table 8–1 lists the processes involved.

Table 8–1 Process Name - External System ID

Process Name	External System ID
SyncCurrencyExchangeListLogisticsProvABCSImpl	CURRENCY_ESID
MergeCustomerPartyListLogisticsProvABCSImpl	MERGECUSTOMERPARTY_ESID
SyncSupplierPartyListLogisticsProvABCSImpl	SUPPLIER_ESID
SyncCustomerPartyListLogisticsProvABCSImpl	CUSTOMERPARTY_ESID

> **Note:** External System IDs and their corresponding processes are in the *AIAConfigurationProperties*.xml file.

To create external systems:

1. Log in to Oracle Enterprise Manager Fusion Middleware Control [`http://<server host>:<server port>/em`].

2. Click the desired process.

3. Click **Show WSDL and endpoint URI**.

Figure 8–1 Creating Contacts in OTM
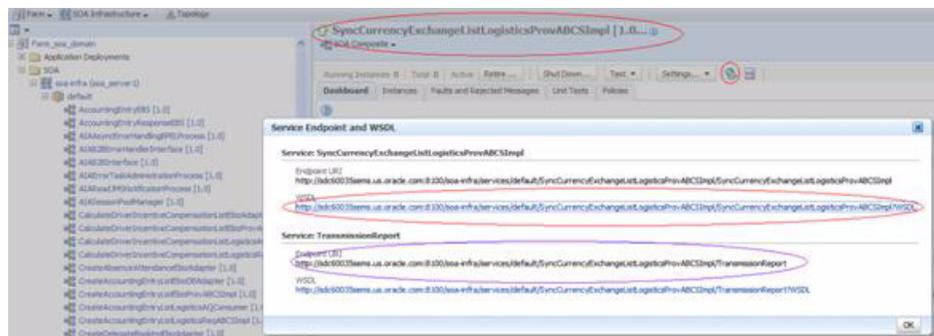
4. Click **concrete WSDL URI** and add *style="document"* attribute to **soap:binding** element.

5. Save the source on your local system.

6. Log in to OTM.

7. Go to **Business Process Automation**, **Communication Management**, **Web Services** and click **New**.

8. Click **New for WSDL Document**.

9. Click **Document Detail**.

   ■ Enter name for ID.

   ■ Upload the saved concrete WSDL.

   ■ Ensure **Storage** is set to *Text* and **Mime Type** as *text/xml*.

10. Click **Finished**.

11. Click **Service Details**.

    ■ Enter **Service ID** and **Service Endpoint ID**.

    ■ Enter **Service Endpoint** as *Endpoint URI* as depicted in the screen. See Section 8.1.3, "Creating Contacts in OTM".

    ■ Enter user name and password of your weblogic server.

    > **Note:** For OTM versions earlier than 6.1.2 encrypt the password before entering using Base64 encoding.

12. Click **Finished**.

13. Go to **Business Process Automation**, **Communication Management**, **External Systems**.

14. Click **New**.

    ■ Enter value for **External System ID**. See Section 8.1.3, "Creating Contacts in OTM" for these values.

    ■ Select webservice created previously.

    ■ Select **Operation** as **TransmissionReport**.

    ■ Select the **Service Endpoint** created earlier.

15. Click **Finished**.

    ■ These steps create a contact automatically in OTM with the same name as of **External System ID**.

    ■ To verify, go to **Business Process Automation**, **Communication Management**, **Contacts**.

    > **Note:** Update OTM_01.Contact_Domain Property in *AIAConfigurationProperties*.xml for each process with the domain in which External System is created for the corresponding process.

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

8.1.4 Setting Up External Systems for Queues

To setup external systems:

1. Login to OTM application.

2. Go to **Business Process Automation**, **Communication Management**, **External Systems**.

3. Click **New**.

4. Enter **External System ID** as AIA_PAYABLEINVOICE_AQ. Add a description

5. Enter the **Queue Name** as AIA_PAYABLEINVOICE_AQ under **For Queue** section.

Figure 8–2   External System Manager

6. In the **Out XML Profiles** section, click **n** for **New XML profile**.

7. Enter **Out XML Profile ID** as *AIA_VOUCHER_MAX*.

8. Choose **Default Mode** as *MAX*.

9. Toggle **Validate for Max**.

Figure 8–3   Out XML Profiles

10. Click **Finished**.

11. In the **Out XML Profiles** section, choose the **XML Element ID** as *Voucher*.

12. Click **Save**.

Figure 8–4   Out XML - Profile ID and Element ID

13. Click **Finished**.

External systems should be defined and set up in OTM for each of the flows with the queue name:

- AIA_PAYABLEINVOICE_AQ

- AIA_INVOICE_AQ

- AIA_ACCOUNTINGENTRY_AQ

To define and set up external system - AIA_INVOICE_AQ:

1. Login to OTM application.

2. Go to **Business Process Automation**-> **Communication Management**-> **External Systems**. Click **New**.

3. Enter **External System ID** as **AIA_INVOICE_AQ**. Add a description

4. Enter the queue name as **AIA_INVOICE_AQ** under **Queue** section

5. In the **Out XML Profiles** section, click **n** for **New XML** profile.

Figure 8–5   External System Manager
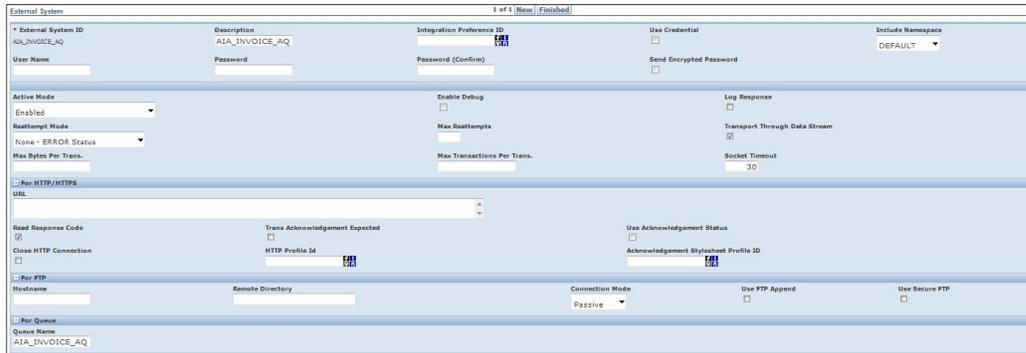
6. Enter **Out XML** profile id as **BILL_EXCL_SHIPMENT**.

7. Choose **Default Mode** as **MAX**.

8. Toggle Validate for Max. Click **Finished**.

Figure 8–6   Out XML Profiles
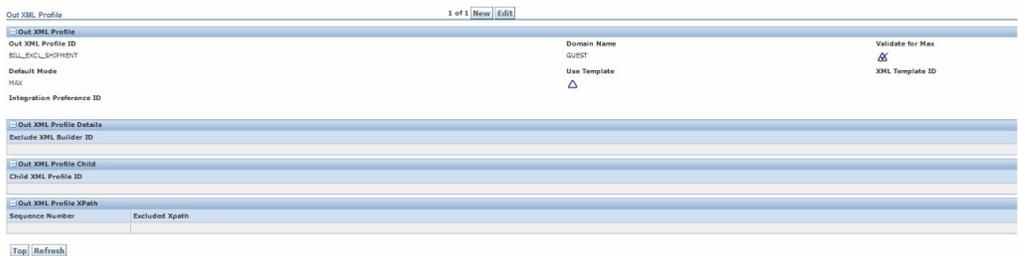
9. In the **Out XML Profiles** section, choose the **XML Element ID** as **Billing**. Click **Save**. Click **Finished**.

Figure 8–7   Out XML - Profile ID and Element ID

To define and set up external system - AIA_ACCOUNTINGENTRY_AQ:

1. Login to OTM application.

2. Go to **Business Process Automation**-> **Communication Management**-> **External Systems**. Click **New**.

3. Enter **External System ID** as **AIA_ACCOUNTINGENTRY_AQ**. Add a description

4. Enter the queue name as **AIA_ACCOUNTINGENTRY_AQ** under **Queue** section

Figure 8–8   External System Manager
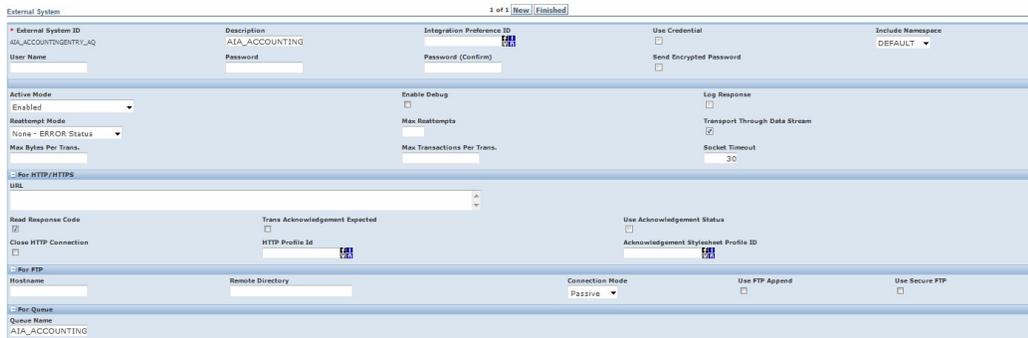
5. In the **Out XML Profiles** section, click **n** for **New XML** profile.

6. Enter **Out XML** profile id as **MAX**.

7. Choose **Default Mode** as **MAX**.

Figure 8–9   Out XML Profiles
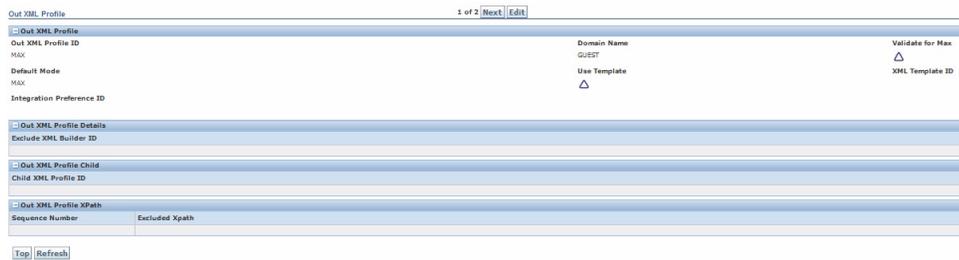
8. Click **Finished**.

9. In the **Out XML Profiles** section, choose the **XML Element ID** as **Accrual**. Click **Save**. Click **Finished**.

Figure 8–10   Out XML - Profile ID and Element ID

8.1.5 Set Up Automation Agents in OTM

This section describes how to setup automation agents:

- Create order release

- Create buy shipment

- Modify buy shipment cost

- Create sell shipment

- Sell shipment cost modified

- Bill approved

Create order release:

1. Navigate to **Business Process Automation**, **Agents and Milestones**, **Automation Agent**, **New**.

Figure 8–11   Agent Manager - Create Order Release (1 of 2)
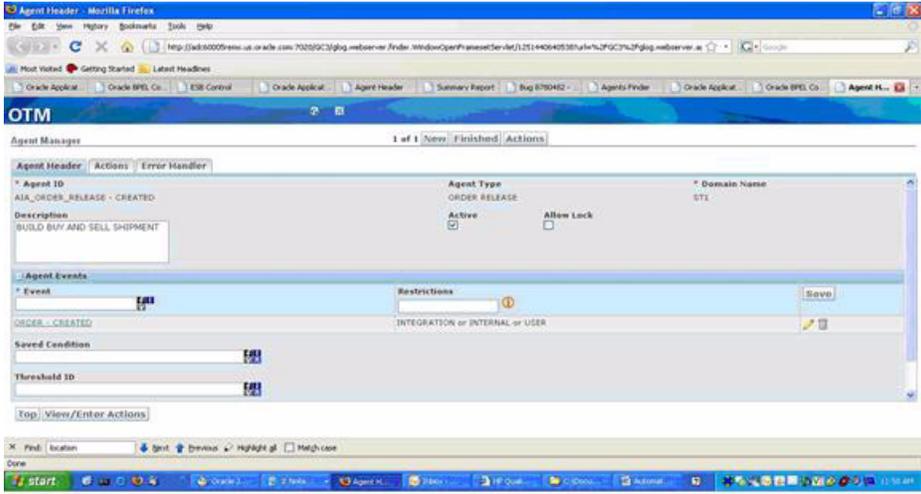
2. Select **Actions**.

Figure 8–12   Agent Manager - Create Order Release (1 of 2)
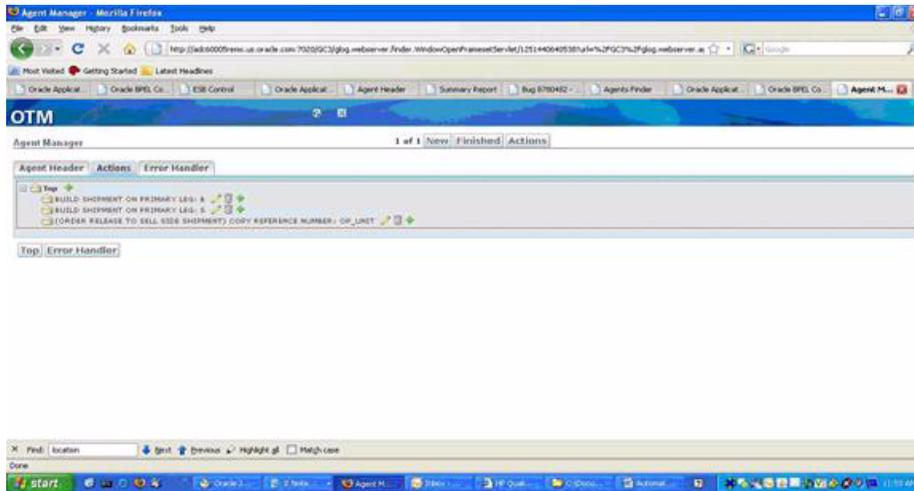
3.   Click **Finished**.

Create buy shipment:

1.   Navigate to **Business Process Automation**, **Agents and Milestones**, **Automation Agent**, **New.**

Figure 8–13   Agent Manager - Create Buy Shipment (1of 2)
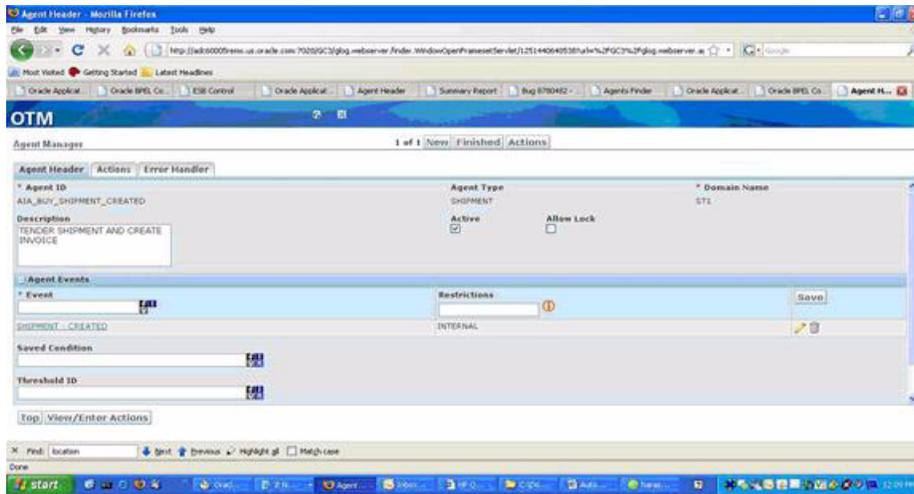
2.   Select **Actions**.

Figure 8–14   Agent Manager - Create Buy Shipment (2 of 2)
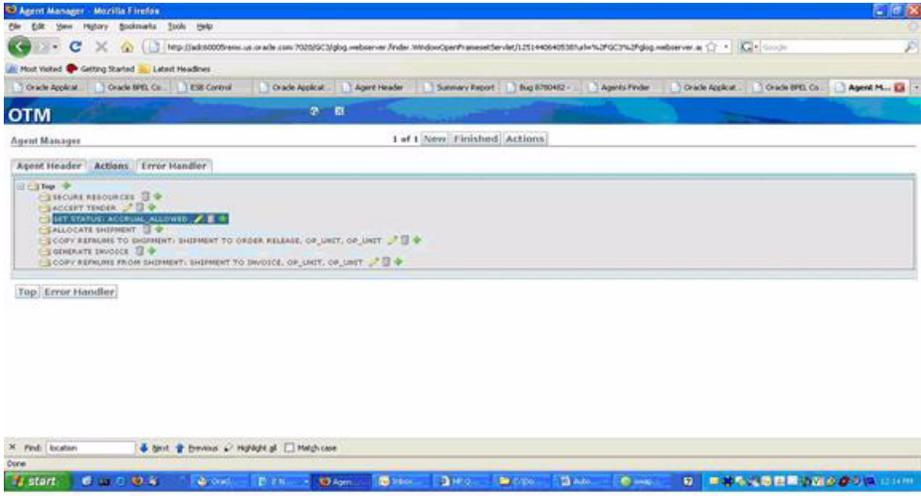
3. Click **Finished**.

Modify buy shipment cost:

1. Navigate to **Business Process Automation**, **Agents and Milestones**, **Automation Agent**, **New.**

Figure 8–15   Agent Manager - Modify Shipment Cost (1 of 2)

2. Select **Actions**.

Figure 8–16   Agent Manager - Modify Shipment Cost (2 of 2)

3. Click **Finished**.

Create sell shipment:

1. Navigate to **Business Process Automation**, **Agents and Milestones**, **Automation Agent**, **New.**

Figure 8–17   Agent Manager - Create Sell Shipment (1of 2)
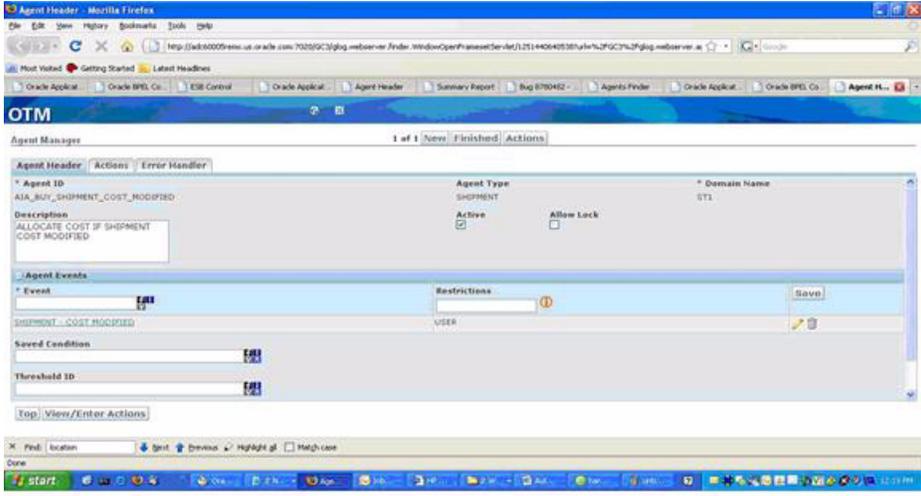
2. Select **Actions.**

Figure 8–18   Agent Manager - Create Sell Shipment (2 of 2)
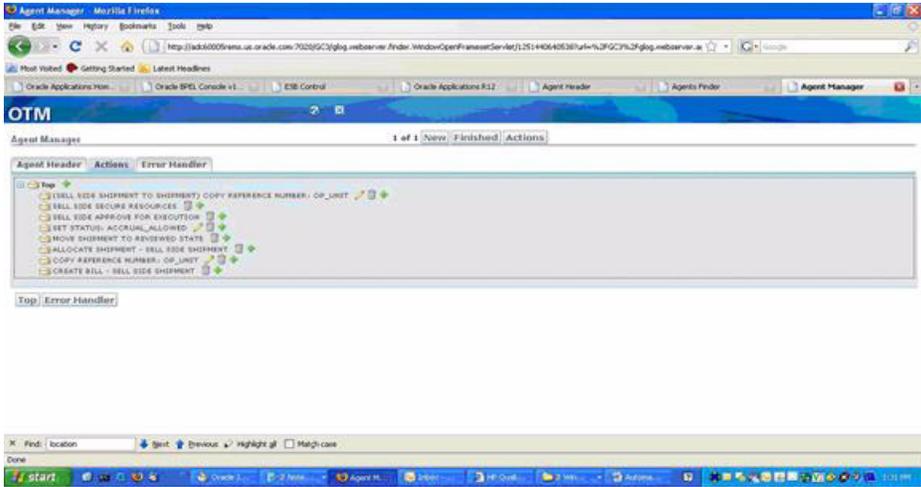
3. Click **Finished**.

Sell shipment cost modified

1. Navigate to **Business Process Automation**, **Agents and Milestones**, **Automation Agent**, **New**.

Figure 8–19   Agent Manager - Modify Sell Shipment (1 of 2)
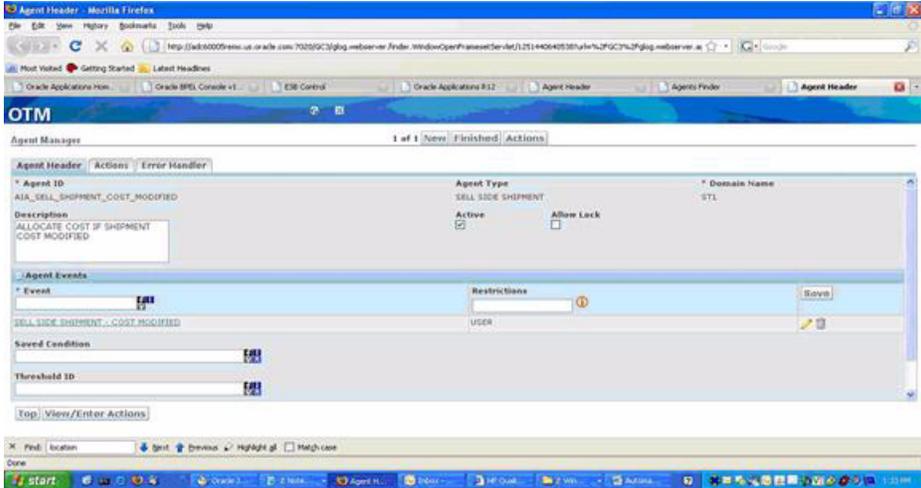
2. Select **Actions**.

Figure 8–20 Agent Manager - Modify Sell Shipment (2 of 2)
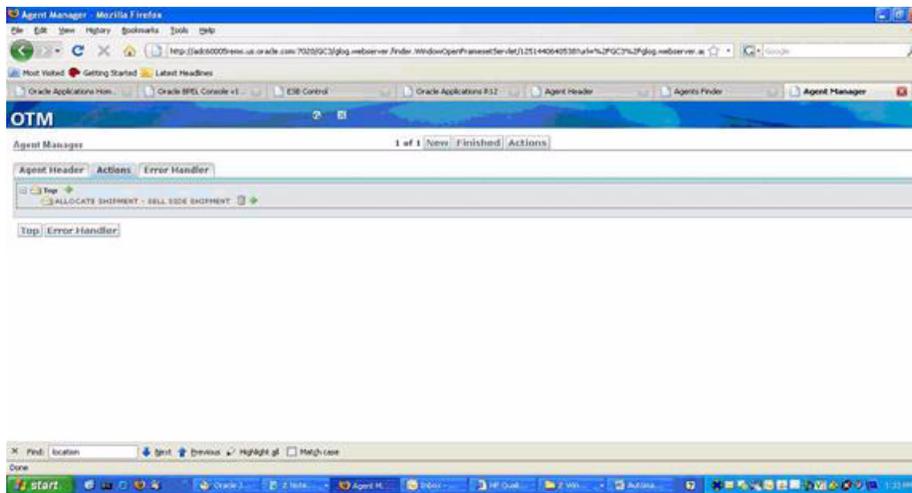
3. Click **Finished**.

Bill approved

1. Navigate to **Business Process Automation**, **Agents and Milestones**, **Automation Agent**, **New**.

Figure 8–21 Agent Manager - Bill Approved (1 of 2)
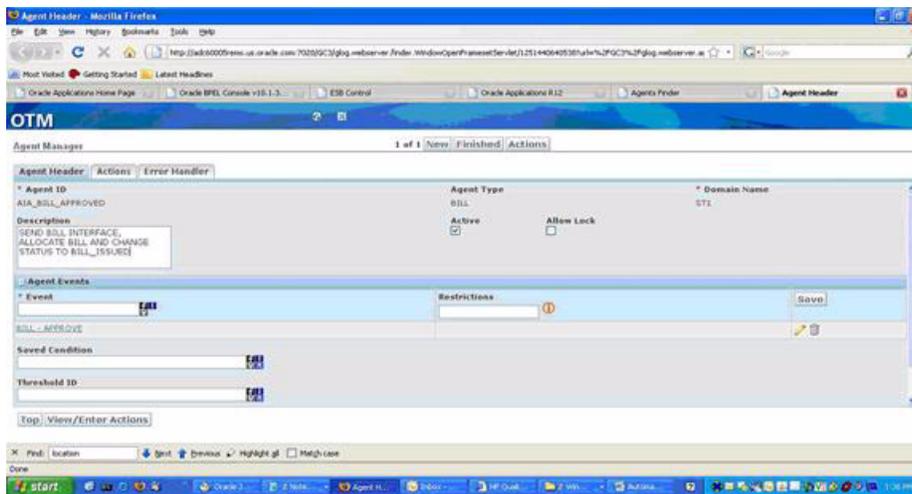
2. Select **Actions.**

Figure 8–22   Agent Manager - Bill Approved (2 of 2)
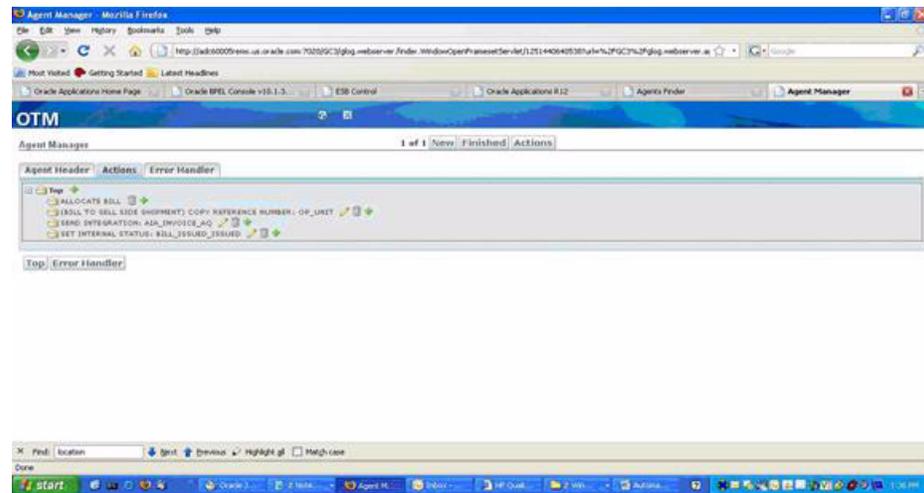
3. Click **Finished**.

Voucher created

1. Navigate to **Business Process Automation**, **Agents and Milestones**, **Automation Agent**, **New**.

Figure 8–23   Agent Manager - Voucher Created (1 of 2)
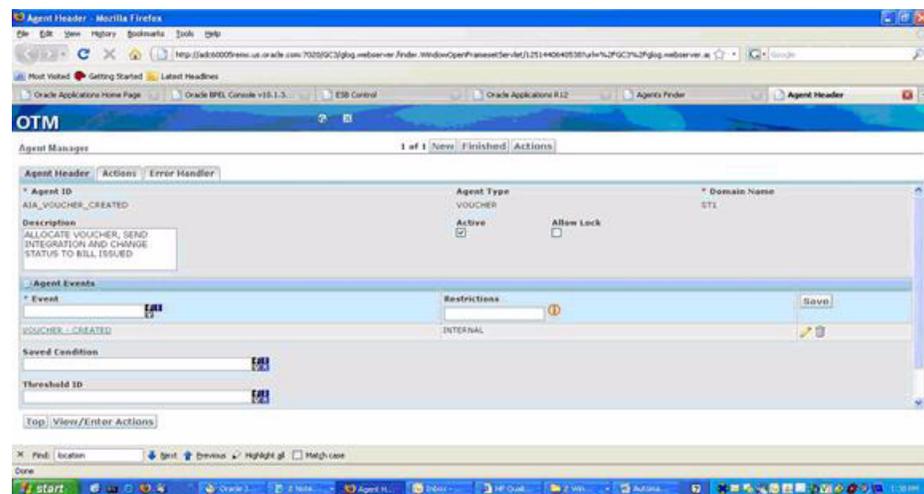
2. Select **Actions**.

Figure 8–24   Agent Manager - Voucher Created (2 of 2)
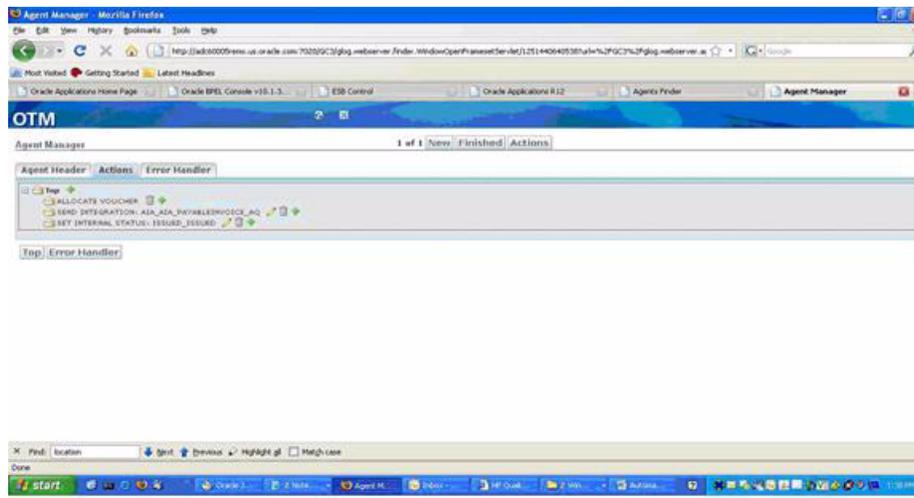
3. Click **Finished**.

8.2 Setting Up Oracle EBS

Complete the following tasks to set up Oracle EBS:

- Create Oracle EBS system profile options

- Synchronize currency exchange rates

- Synchronize supplier data

- Synchronize payable invoices

- Synchronize receivable invoices

- Synchronize accruals

- Synchronize customers

- Set up organizations

8.2.1 Creating Oracle EBS System Profile Options

To create Oracle EBS system profile options:

1. Log in to Oracle EBS as System Administrator.

2. Open **System Profile Values** form.

3. Query these profile options, and set the indicated values at the site level:

    For Oracle EBS:

    - HZ: Generate Party Number to *Yes*

    - HZ: Generate Party Site Number to *Yes*

    - HZ: Raise API Events to *All Events Enabled* or *Only Business Object Events Enabled*

    - HZ: Format Business Object Business Events as Bulk to *N*.

8.2.2 Synchronizing Currency Exchange Rates

To synchronize currency exchange rates:

1. Go to **Workflow Administrator**, **Business Event**.

2. Enable the Currency Exchange Business Event oracle.apps.gl.CurrencyConversionRates.dailyRate.specify in Oracle EBS GL: Daily Conversion Rates Specified.

> **Note:** The status of the event should be *Enable*.

If the status is not enabled:

1. Click **Update** and change the status to *Enabled*.

2. Click **Save**.

   The subscription is automatically created during the installation of the PIP.

   Verify the value for the **Phase** field for the subscription.

3. Click the pencil button to see the **Business Events Subscription** details. The value in the **Phase** field should be changed to *less than 100*.

4. Click **Save**.

8.2.3 Synchronizing Supplier Data

To synchronize supplier data:

■ Enable the payables supplier event oracle.apps.ap.supplier.event in Oracle EBS.

8.2.4 Synchronizing Payable Invoices

To synchronize payable invoices:

1. Define **Payment Terms**. To define payment terms, navigate to **Setup**, **Invoice**, **Payment Terms**.

2. Define **Distribution Sets** or **SLA Account Derivation Rules**. To define distribution sets, navigate to **Setup** -> **Invoice** -> **Distribution Sets**.

   You can either set up the distribution set with the correct accounting at the supplier site or if you are using Subledger Accounting (SLA) to derive the accounting, set up a dummy distribution set at the supplier site level (that is: 000-000-0000) and set up SLA with the account derivation rules to create the accounting. If you set up the distribution set at the supplier site level with the correct accounts and also set up the account derivation rules in SLA, the account derivation rules overrides the distribution set accounting and create the accounting in GL; therefore, when querying the invoices in AP, it has a different distribution versus the journal in GL.

3. Define **Source Lookups**, add *Code OTM_01*. To define Source Lookups, navigate to **Setup** -> **Lookups** -> **Payables**.

4. Define **Line Source Lookups**, add *Code OTM_01*. To define Line Source Lookups, navigate to **Setup** -> **Lookups** -> **Payables**.

5. Define **Supplier**, with primary pay site and attach payment term and distribution set to the pay site.

6. Open **Payables Accounting Period**.

8.2.5 Synchronizing Receivable Invoices

To synchronize receivable invoices:

1. Define **Transaction Source** - *OTM_01*. To define Transaction Source, navigate to **Setup** -> **Transactions** -> **Sources**.

   ■ Batch Source Options

   ■ Set Automatic Batch Numbering = *Unchecked*

   ■ Set Automatic Transaction Numbering = *Unchecked*

   ■ Set Copy Document Number to Transaction Number = *Unchecked*

   ■ Set Allow Duplicate Transaction Numbers = *Unchecked*

   ■ Set Reference Field Default Value= *interface_header_attribute2*

   ■ Customer Information Options

   ■ Set Bill To Customer = *ID*

   ■ Set Bill To Address= *ID*

   ■ Set Bill To Contact = *None*

   ■ Set Ship To Customer = *ID*

   ■ Set Ship To Address= *ID*

   ■ Set Ship To Contact = *None*

   ■ Set Payment Terms = *ID*

   ■ Accounting Information Options

2. Define **Payment Terms.**To define payment terms, navigate to **Setup** -> **Transactions** -> **Payment Terms**.

3. Define **Auto Accounting** or **SLA Account Derivation Rules**. To define auto accounting, navigate to **Setup**, **Transactions**, **AutoAccounting**

   a. If you are using Oracle EBS AR AutoAccounting functionality, you must define AutoAccounting to specify the general ledger accounts for transactions that you enter manually or import using AutoInvoice. AutoAccounting uses this information to create the default revenue, receivable, freight, tax, unearned revenue, unbilled receivable, bills receivable accounts, and AutoInvoice clearing (suspense) accounts. The default accounting that AutoAccounting creates is considered interim accounting only. Oracle EBS AR integrates with SLA that accepts the default accounts that AutoAccounting derives without change. However, you can modify the accounting rules in Subledger Accounting to create accounting that meets your business requirements.

   b. You must define **AutoAccounting** by **Operating Unit** before you can enter transactions in **Receivables**.

4. Define **Balance Forward Billing** for **Consolidated Billing**. To define Balance Forward Billing, navigate to **Setup**, **Print**, **Balance Forward Billing Cycles**.

5. Define **Line Transaction Flexfield**. To define line Transaction Flexfield, navigate to **Setup**, **Financials, Flexfields**, **Descriptive**, **Segments**.

   a. Context Code =*OTM*

   b. Segment1 = Domain, Column=*INTERFACE_LINE_ATTRIBUTE1*

   c. Segment2= InvoiceXid, Column=*INTERFACE_LINE_ATTRIBUTE2*

   d. Segment3=*InvoiceLineNumber*

6. Attach payment terms to customer account sites. To attach payment terms to customer account sites, perform these steps:

   1. Navigate to **Customer** -> **Profile Class**. Create a Profile Class and configure the Payment Terms described in step2.

   2. Navigate to **Customer Site details**, click **Profile** tab and associate the **Profile Class** to the Site.

8.2.6 Synchronizing Accruals

To synchronize accruals:

> **Note:**   The IDT is not supported from 12.2. version onwards. The IDT is supported only till 12.1.3 version.

1. Define **Interface Data Transformation Rules**.

   a. Derive **Period Name** from the Accounting Date

   b. Derive **Ledger ID** from the Operating Unit

   c. Derive **Code Combination ID**

   d. Create PL/SQL functions, if any to use in an IDT transformation rule set and register the functions in the metadata structure.

   e. Create value sets, if any, to use in an IDT Transformation Rule Set.

   f. Create lookup tables, if any, to use in an IDT Transformation Rule Set and register the tables in the metadata structure.

   g. Define **Transformation Rule Sets**.

      IDT rules should be set up to look at the **Perspective** and the **Is Reversal** flag to create the logic whether the entry is a debit or a credit. Based on the Perspective, IDT should be set up to create the accounting for the balancing entry. Same goes for the Delta costs.

   h. Run Transformation Rule Sets by launching the Program, GL Interface Data Transformer, either standalone or along with Journal Import.

2. Define **Journal** import.

   a. If you want Journal Import to assign sequential numbers to your accrual entries, enable sequential numbering, specifying Automatic as both your numbering and document generation method.

   b. Disable dynamic insertion. Journal Import runs much faster when it does not have to create new account combinations dynamically.

8.2.7 Synchronizing Customers

To synchronize customers:

1. Navigate to **Workflow Administrator**, **Business Event**.

2. Enable these customers business events in Oracle EBS:

    ■ oracle.apps.ar.hz.OrgCustBO.create

    ■ oracle.apps.ar.hz.OrgCustBO.update

    ■ oracle.apps.ar.hz.CustAccount.merge

    ■ oracle.apps.ar.hz.Party.merge

> **Note:** The status of the event should be *Enabled*.

If the status is not enabled, do these for each event listed:

1. Click **Update** and change the status to *Enabled* and save the event.

    The subscription is automatically created during the installation of the PIP. Verify the value for the Phase field for the subscription.

2. Click the pencil button to review the **Business Events Subscription** details. The value in the **Phase** field should be changed to *less than 100*.

3. Save the subscription.

8.2.8 Setting Up Organizations

Complete the following tasks to set up organizations.

■ Determine what organizations to support and then get the IDs for those organizations

■ Determine which domains in OTM require support

■ Determine which status type in OTM is required, for example, CUSTOMERPARTYSTATUS

■ Populate cross-reference values for ORGANIZATION_ID table

■ Validate cross-references

To get the Operating Unit details:

1. Login to Oracle EBS database.

2. Identify the Operating Units that must be synchronized or maintained in Oracle EBS.

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

select organization_ID, name from hr_operating_units

To get the OTM Domain details:

1. Login to OTM application.

2. Navigate to **Configuration and Administration**.

3. Click **Domain Management**.

4. Select **Domain Settings**.

5. Click **Search**.

   The following screen appears.

Figure 8–25   OTM Domains
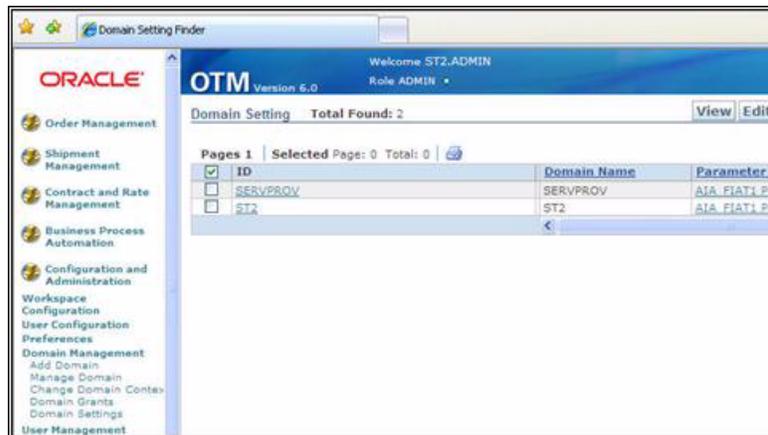
To create a Status Type for Location in OTM:

1. Login to OTM application.

2. Navigate to **Configuration and Administration**.

3. Click **New**.

4. Enter **Status Type ID**, for example, *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.

Figure 8–26   OTM Location Status Type
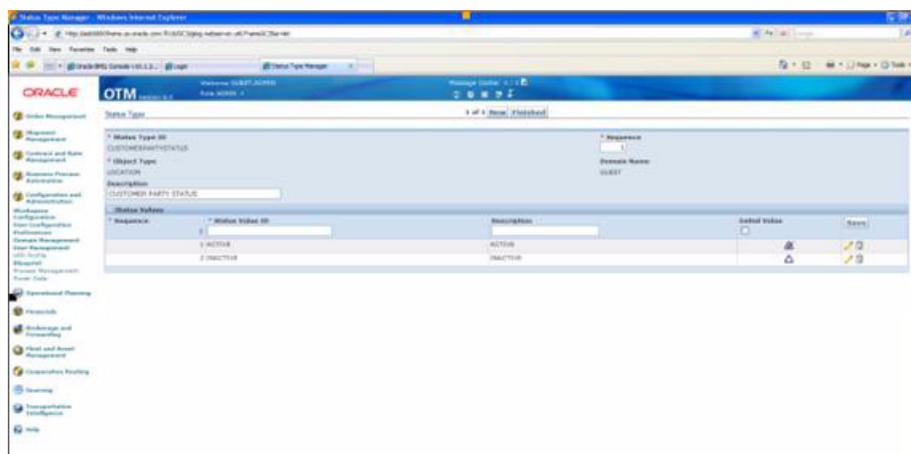
To populate cross-reference values for ORGANIZATION_ID table:

1. Login to database.

2. Connect to schema *<AIA_INSTANCE>_xref*.

3. Enter values into cross-reference table using the insert command.

See Example 8–2.

Example 8–2   Values for Cross-Reference Table

```
INSERT INTO XREF_DATA VALUES ('oramds:/apps/AIAMetaData/xref/ORGANIZATION_
ID.xref','EBIZ_01','27F4D6303B2511DFBFA11DB680CBD54F','204','N','25-AUG-10
02.26.11.000000000 AM')

INSERT INTO XREF_DATA VALUES ('oramds:/apps/AIAMetaData/xref/ORGANIZATION_
ID.xref','COMMON','27F4D6303B2511DFBFA11DB680CBD54F','COMMON_ORG_
ID','N','25-AUG-10 02.26.11.000000000 AM')

INSERT INTO XREF_DATA VALUES
('oramds:/apps/AIAMetaData/xref/ORGANIZATION_ID.xref','OTM_
01','27F4D6303B2511DFBFA11DB680CBD54F','OTM_ORG_VALUE','N','25-AUG-10
02.26.11.000000000 AM')
```

> **Note:**   Keep the row number (third column) same while inserting date for each pair of cross-reference values.

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

To validate cross-references:

1. Log in to the AIA XREF database.

2. Query the Table XREF_DATA to confirm that every organization used in the XML files has three records.

Use the query in Example 8–3.

Example 8–3   Table XREF_DATA Query

```
select value||::||Xref_column_name from xref_Data where row_number in (select row_
number from xref_data where xref_table_name =
oramds:/apps/AIAMetaData/xref/ORGANIZATION_ID.xref and value in (204))
```

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

9

Data Requirements and Prerequisites

This chapter discusses data requirements for each of the Oracle Financial Management for Oracle transportation Management (OTM) and Oracle E-Business Suite (Oracle EBS) process integrations.

This chapter includes the following sections:

- Section 9.1, "Process Integration for Currency Exchange Rates"
- Section 9.2, "Process Integration for Suppliers"
- Section 9.3, "Process Integration for Payables Invoices"
- Section 9.4, "Process Integration for Receivable Bills"
- Section 9.5, "Process Integration for Accruals and Reversals"
- Section 9.6, "Process Integration for Customer Synchronization"

9.1 Process Integration for Currency Exchange Rates

These are the prerequisites for the process integration for currency exchange rates:

- The list of valid currencies is maintained separately in each application and is mapped using a domain value map (CURRENCY_CODE) in the integration layer.
- Oracle E-Business applications maintain the exchange rate types manually and the rate types are mapped using a domain value map (CURRENCYEXCHANGE_ CONVERSIONTYPECODE) in the integration layer.
- To synchronize currency exchange rates from Oracle EBS to OTM, you must be an administrator because the currency exchange rates must be available to all the users and be synchronized to the public domain.
- To synchronize currency exchange rate from Oracle EBS to OTM, the oracle.apps.gl.CurrencyConversionRates.dailyRate.specify business event should be enabled in the Oracle E-Business application.

9.2 Process Integration for Suppliers

These are the prerequisites for the process integration for suppliers:

- The relevant Supplier Types/Location Roles must be set up in the SUPPLIERPARTY_TYPECODE domain value map (DVM).
- The required configuration properties must be specified. See Chapter 13, "Configuring Oracle Financial Management Integration Pack for OTM and Oracle EBS".

9.3 Process Integration for Payables Invoices

These are the prerequisites for the process integration for payable invoices:

- Suppliers, their locations and currency exchange rates must be synchronized between Oracle EBS and OTM.

- You can either set up the distribution set (distribution sets are an account derivation mechanism offered within Oracle EBS AP) with the correct accounting at the supplier site or, if you are using Subledger Accounting (SLA) to derive the accounting, set up a dummy distribution set at the supplier site level (that is: 000-000-0000) and setup SLA with the account derivation rules to create the accounting.

  If you setup the distribution set at the supplier site level with the correct accounts and also setup the account derivation rules in SLA, the account derivation rules overrides the distribution set accounting and create the accounting in GL; therefore, when querying the invoices in AP, it has a different distribution versus the journal in GL.

- Calendar must be setup for transactions.

- The required configuration properties should also be specified. See Chapter 13, "Configuring Oracle Financial Management Integration Pack for OTM and Oracle EBS".

9.4 Process Integration for Receivable Bills

These are the prerequisites for the process integration for receivable bills:

- Customers must be synchronized between Oracle EBS and OTM. The logistics order management process integration pack provides customer synchronization between Siebel and Oracle EBS and between Siebel and OTM.

- Currency exchange rates must be synchronized between Oracle EBS and OTM.

- If you are using Oracle EBS Accounts Receivables (AR) AutoAccounting functionality (an account derivation mechanism), you must define AutoAccounting to specify the general ledger accounts for transactions that you enter manually or import using AutoInvoice.

  AutoAccounting uses this information to create the default revenue, receivable, freight, tax, unearned revenue, unbilled receivable, bills receivable accounts, and AutoInvoice clearing (suspense) accounts.

  The default accounting that AutoAccounting creates is considered interim accounting only. Oracle EBS AR integrates with Subledger accounting (SLA) which accepts the default accounts that AutoAccounting derives without change. However, you can modify the accounting rules in SLA to create accounting that meets your business requirements.

- You must define AutoAccounting by operating unit before you can enter transactions in receivables.

- If you are using AutoInvoice, define the grouping rules to indicate how to group transaction lines imported by AutoInvoice.

  For example, to include specific transaction lines on a single transaction, certain attributes must be identical. Oracle EBS receivables provides many attributes that you can use to define your grouping rules.

- Define the invoice line ordering rules for transaction lines that you import into Oracle EBS AR using AutoInvoice. AutoInvoice uses these rules to order transaction lines when grouping the transactions it creates into invoices, debit memos, and credit memos.

- Calendar must be setup for transactions.

- All the required Configuration properties should also be specified. See Chapter 13, "Configuring Oracle Financial Management Integration Pack for OTM and Oracle EBS".

9.4.1 Account Receivable Functionality

To run the CreateInvoiceListEbizappsAdapter service properly, you must add the Oracle EBS **responsibility** and **username** parameter in the EM console. The values provided for these parameters are used by the AR flow in E-Business application.

> **Note:** Before you add the Oracle EBS **responsibility** and **username** parameter in the EM console, you must change the **Responsibility** name in the AIAConfigurationProperty file.

9.4.1.1 Changing Responsibility Name in AIAConfigurationProperty File

To change the Responsibility name in the AIAConfigurationProperty file, you must perform the following steps:

1. Update the AIAConfigurationProperty file.

2. Upload the AIAConfigurationProperty file into MDS.

Updating the AIAConfigurationProperty file

1. Open the AIAConfigurationProperties.xml file located at *<AIA_INSTANCE>/ AIAMetaData/config* directory.

2. Update the property, for example, *<Property name="RESPONSIBILITY">Receivables, Vision Operations (USA)</Property>* in AIAConfigurationProperties.xml file with the Responsibility name used by you for the CreateInvoiceListEbizProvABCSImpl service.

3. Save and close the AIAConfigurationProperties.xml file.

Uploading the AIAConfigurationProperty file into MDS

1. Source the aiaenv.sh file: *source <AIA_INSTANCE>/bin/aiaenv.sh*.

2. Navigate to the config directory: *cd <AIA_HOME>/Infrastructure/Install/config*.

3. Run the command: **ant -f UpdateMetaData.xml**.

9.4.1.2 Adding Responsibility and Username in WebLogic Console

To add the **Responsibility** and **Username** in WebLogic EM console, perform the following steps:

> **Note:** These steps must be performed only if you are using a different **responsibility** or **username** than the one which is shipped in the out of the box code.

1. Login to the EM Console.

2. Expand **SOA**, **soa-infra**, **default** as listed in Figure 9–1.

Figure 9–1   SOA-Infra (SOA _Server1) Details
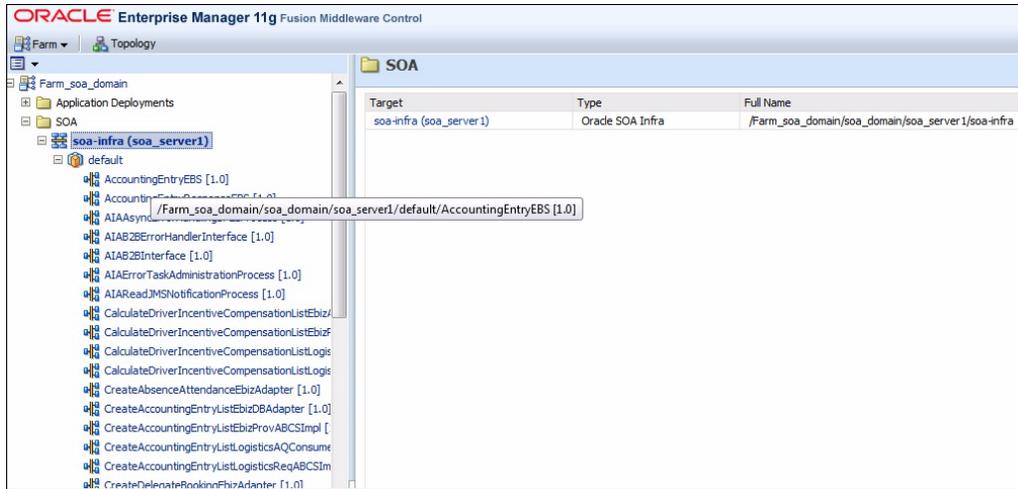
3. Right-click **CreateInvoiceListEbizAppsAdapter**, select **Service/Reference Properties**, click **CreateInvoiceListEbizAppsAdapter**.

Figure 9–2   CreateInvoiceListEbizAppsAdapter Details
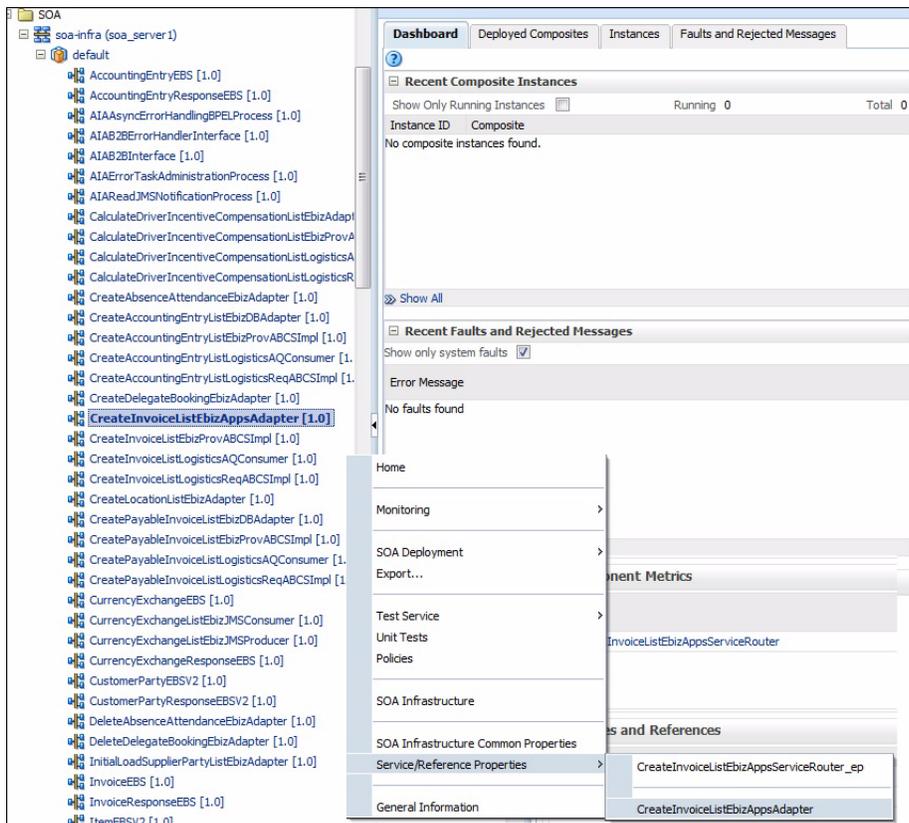
4. Select the **properties** tab and change the **Responsibility** and **Username** according to your requirement.

Figure 9–3   Parameters - Responsibility and Username
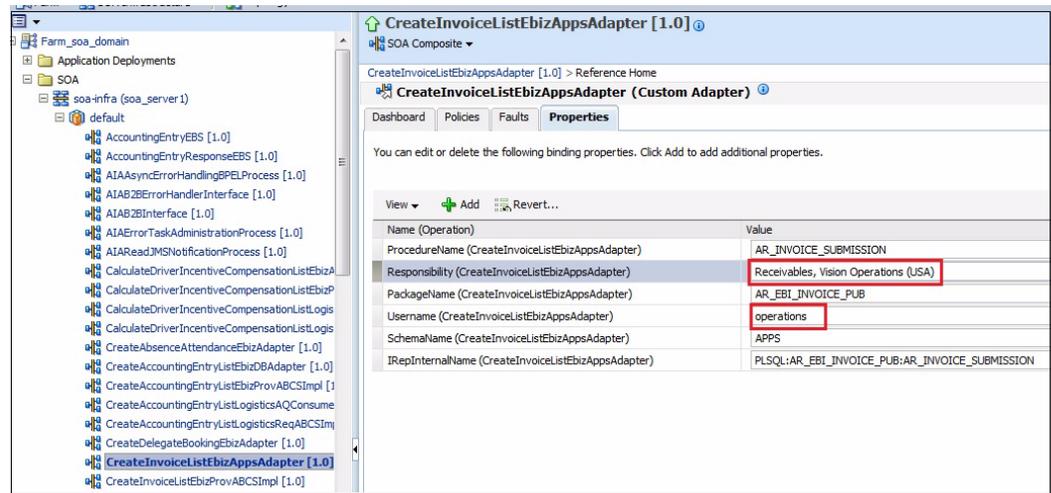
9.5 Process Integration for Accruals and Reversals

These are the prerequisites for the process integration for accruals and reversals:

- Currency exchange rates must be synchronized between Oracle EBS and OTM.

- The Oracle EBS journal import process uses the interface data transformer (IDT) functionality to derive the accounting entries for the accruals and reversals. IDT is a lightweight ETL tool that you can use to transform data in the GL interface table. As part of the IDT process, several steps must be completed to setup IDT:

  - Create PL/SQL functions, if any to use in an IDT transformation rule set and register the functions in the metadata structure.

  - Create value sets, if any, to use in an IDT transformation rule set.

  - Create lookup tables, if any, to use in an IDT transformation rule set and register the tables in the metadata structure.

  - Define transformation rule sets. IDT rules should be set up to look at the **perspective** and **is reversal** flag to create the logic whether the entry is a debit or a credit. Based on the perspective, IDT should be set up to create the accounting for the balancing entry. The same goes for the delta costs.

  - Run transformation rule sets by launching the program, GL interface data transformer, either standalone or along with journal import.

- To import and assign sequential numbers to your accrual entries, enable sequential numbering, specifying automatic as both your numbering and document generation method.

- Disable dynamic insertion. Journal import runs much faster when it does not have to create new account combinations dynamically.

- Calendar must be setup for transactions.

- All the required configuration properties should also be specified. See Chapter 13, "Configuring Oracle Financial Management Integration Pack for OTM and Oracle EBS".

9.6 Process Integration for Customer Synchronization

Data requirement is that you must populate the ORGANIZATION_ID transfer table with valid EBIZ_01, COMMON and OTM_01 columns.

10

Performing an Initial Load

This chapter discusses how to start and run an initial load of exchange rates and an initial load of suppliers.

This chapter includes the following sections:

- Section 10.1, "Initial Load of Currency Exchange Rates"
- Section 10.2, "Initial Load of Suppliers"

10.1 Initial Load of Currency Exchange Rates

The General Ledger application within Oracle E-Business Suite (Oracle EBS) is responsible for maintaining and storing currency exchange rates. Currency Exchange Rates data should be loaded from the Oracle EBS Financials - General Ledger - to the Oracle Transportation Management (OTM). Existing currency exchange rates present in Oracle EBS can be synchronized (initial load) to OTM using this process.

10.1.1 How to Start and Run Initial Load of Exchange Rates

To load the currency exchange rates initially from Oracle EBS to OTM:

1. Contact your Oracle Service-Oriented Architecture (Oracle SOA) Administrator and obtain the endpoint location of the process CurrencyExchangeListEbizJMSProducer.

   a. Login to Oracle Enterprise Manager Fusion Middleware Control (`http://<soa_server_host>:<soa_server_port>/em`).

   b. Click **CurrencyExchangeListEbizJMSProducer** process.

   c. Click **Test**.

   d. Click **Test WebService**.

   Note: If a web service is secured, enter credentials before invoking the web service.

   ---

2. Open the end point.

   The system displays the parameters listed in Table 10–1:

Table 10–1   Current Exchange Load Parameters

Parameter	Description
from_currency	xsd: string

Table 10–1   (Cont.) Current Exchange Load Parameters

Parameter	Description
to_currency	xsd: string
from_date	xsd: string
to_date	xsd: string
conversion_rate_type	xsd:string
SystemID	xsd: string
from_currency	xsd: string

The **from_date** field is a mandatory field. If the **to_date** is not specified, it takes the current date as the **TO DATE**. The valid date format is *YYYY-MM-DD (2008-02-09)*. The rest of the elements are optional. You can use those elements to limit the rates, which are required to be loaded from Oracle EBS to OTM. Use the appropriate date range based on the load of data and performance of the server.

If a user defined conversion rate type must be loaded, provide the internal ID/number of the rate type in the parameter list.

3. Click **Invoke** and the exchange rates within the input range specified are sent from Oracle EBS into OTM.

10.2 Initial Load of Suppliers

Oracle EBS Accounts Payable is responsible for adding and maintaining the Supplier and Supplier Site Profiles. For this integration, the supplier information should be initially loaded or synchronized from Oracle EBS to OTM using this process.

10.2.1 How to Start and Run Initial Load of Supplier Parties

To start the initial load of the supplier parties:

1. Contact your SOA Administrator and obtain the endpoint location of the process CurrencyExchangeListEbizJMSProducer.

    a. Login to Oracle Enterprise Manager Fusion Middleware Control (`http://<soa_server_host>:<soa_server_port>/em`).

    b. Click **InitialLoadSupplierPartyListEbizAdapter**.

    c. Click **Test**.

    d. Click **Test WebService**.

    > **Note:**   If a web service is secured, enter credentials before invoking the web service.

2. Click the link to display the parameters listed in :

Table 10–2   Current Exchange Load Parameters

Parameter	Description
from_currency	xsd: string
to_currency	xsd: string

Table 10–2   (Cont.) Current Exchange Load Parameters

Parameter	Description
from_date	xsd: string
to_date	xsd: string
conversion_rate_type	xsd:string
SystemID	xsd: string
from_currency	xsd: string

Enter an appropriate value for the chunk size field based on the memory of the system, the number of suppliers present in the Oracle EBS for initial load. It is appropriate to give this value between *50* and *100* or even smaller based on the kind of supplier profiles existing in the system.

3. Click **Invoke**.

This starts the initial load of the suppliers from Oracle EBS to OTM.

10.2.1.1 Performance Tuning Parameters

In case of any issues during the initial load, these properties can be fine-tuned by referring to the Oracle SOA guide.

In the Fusion Middleware machine, change the domain level configurations as described below.

To fine tune the properties:

1. Login into Oracle Enterprise Manager Fusion Middleware Control (`http://<soa_ server_host>:<soa_server_port>/em`).

2. Right-click **SOA-infra**, **SOA Administration**, **Common Properties**.

You can change the **Audit** level to either **Production** or **Off**.

3. Enable **Capture Composite Instance State** to see the instance state in the Oracle Enterprise Manager Fusion Middleware Control.

4. Click **More BPEL Configuration Properties**.

5. Right-click **SOA-infra**, **SOA Administration**, **BPEL Properties**.

Change SyncMaxWaitTime property based on your server performance. Provide a lesser value for the chunk size to run the initial load.

10.2.1.2 Alternative Option

An alternative approach to run the initial load of suppliers from Oracle EBS into OTM is described below.

The initial load of supplier information from Oracle EBS to OTM can also be done using these steps if the range of supplier IDs are known:

1. Contact your Oracle SOA Administrator and obtain the endpoint location of the process SupplierPartyListEbizJMSProducer.

    a. Login to Oracle Enterprise Manager Fusion Middleware Control (`http://:/em`).

    b. Click **SupplierPartyListEbizJMSProducer process**.

    c. Click **Test**.

    d. Click **Test WebService**.

> **Note:** If a web service is secured, enter credentials before invoking the web service.

2. Choose the operation as ProduceSupplierPartyABMInitLoad.

3. Enter the data in **From Vendor ID** and **To Vendor ID** fields for which the data should be synchronized from Oracle EBS to OTM.

4. Click **Invoke**.

   The suppliers present in Oracle EBS between the *from* and *to* values are synchronized to OTM.

11

Working with Cross-References

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

This chapter includes the following section:

- Section 11.1, "Oracle Financial Management Cross-References"

For more information about cross-references, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*.

11.1 Oracle Financial Management Cross-References

Table 11–1 lists the Oracle Financial Management cross-references for accounting.

Table 11–1   Oracle Financial Management Cross-References for Accounting

Name	Column	Description
ACCOUNTINGENTRY_ ACCOUNTINGENTRY ID	OTM_01, COMMON, EBIZ_01	To maintain the cross-reference between OTM and Oracle E-Business Suite AccountingEntryID.
		OTM_01 - <Domain>::<AccrualXid>
		COMMON AIA GUID
ACCOUNTINGENTRY_ ACCOUNTINGENTRY ID	OTM_01, COMMON, EBIZ_01	To maintain the cross-reference between OTM and Oracle E-Business Suite AccountingEntryID.
		OTM_01 - <Domain>::<AccrualXid>
		COMMON AIA GUID
		EBIZ_01 - ACCRUAL_ID
CURRENCYEXCHANGE_ID	EBIZ_01, COMMON, OTM_01	This is used to store the cross-reference for the exchange rates among the different applications.
		Cross-reference value for Oracle E-Business Suite and OTM is concatenation of FromCurrency, ToCurrency, CurrencyExchangeRateType, and ConversionDate with ::" separating them. GUID is a common value.
INVOICE_INVOICEID	OTM_01, COMMON, EBIZ_01	To maintain the cross-reference between OTM and Oracle E-Business Suite Invoice ID.
		OTM_01 - <Domain>::<BillingXid>
		COMMON AIA GUID
		EBIZ_01 - INVOICE_ID
INVOICE_INVOICEID	OTM_01, COMMON, EBIZ_01	To maintain the cross-reference between OTM and Oracle E-Business Suite Invoice ID.
		OTM_01 - <Domain>::<BillingXid>
		COMMON AIA GUID
		EBIZ_01 - INVOICE_ID

Table 11–1   (Cont.) Oracle Financial Management Cross-References for Accounting

Name	Column	Description
ORGANIZATION_ID	EBIZ_01, COMMON, OTM_01	This is used to look up the cross-reference values for the Operating Unit / Business Unit / Domain information of supplier.
		Common - GUID.
		OTM -Domain names
		EBIZ - ORG_ID
PAYABLEINVOICE_ PAYABLEINVOICEID	OTM_01, COMMON, EBIZ_01	To maintain the cross-reference between the PayableInvoice ID between OTM and E-Business Suite.
		Cross-reference value for Oracle E-Business Suite and OTM is concatenation of FromCurrency, ToCurrency, CurrencyExchangeRateType and ConversionDate with ::" separating them. GUID is a common value.
PAYABLENVOICE_ PAYABLEINVOICEID	OTM_01, COMMON, EBIZ_01	To maintain the cross-reference between OTM and Oracle E-Business Suite PayableInvoiceID.
		OTM_01 - <Domain>::<VoucherXid>
		COMMON AIA GUID
		EBIZ_01 - INVOICE_ID
PAYABLENVOICE_ PAYABLEINVOICEID	OTM_01, COMMON, EBIZ_01	To maintain the cross-reference between OTM and Oracle E-Business Suite PayableInvoiceID.
		OTM_01 - <Domain>::<VoucherXid>
		COMMON AIA GUID
		EBIZ_01 - INVOICE_ID
PAYMENTTERM_ID	EBIZ_01,Common,	This is used to look up the cross-reference values for the payment terms ID.
		This is used for Retek enhancement.
SUPPLIERPARTY_ADDRESS_ ID	EBIZ_01, COMMON	This is used to store the cross-reference for the Address information that exists for a supplier.
		Common - GUID
		Oracle E-Business Suite - Site ADDRESS ID.
		OTM does not have any specific ID for address
SUPPLIERPARTY_CONTACT_ ID	EBIZ_01, COMMON, OTM_01	This is used to store the cross-reference for contact information of a supplier.
		Common - GUID
		OTM is - Location/Contact/ContactGid/Gid/Xid
		Oracle E-Business Suite - VENDOR_Contact_ID
SUPPLIERPARTY_ID	EBIZ_01, COMMON, OTM_01	This is used to store the cross-reference for the Supplier ID among the different applications.
		Common - GUID
		OTM - LocationGid/DomainName and LocationGid/Xid.
		EBIZ_01 - VENDOR ID
SUPPLIERPARTY_ LOCATION_ID	EBIZ_01, COMMON, OTM_01	This is used to store the cross-reference for Sites that belong to a supplier.
		Common - GUID, Oracle E-Business Suite - VENDOR_ SITE_ID
		OTM - LocationGid/DomainName and LocationGid/Xid

Table 11–2 lists the Oracle Financial Management cross-references for Customer Party web service.

Table 11–2   Financial Management Cross-References for Customer Party

Name	Column	Description
CUSTOMERPARTY_ ACCOUNTID	SEBL_01, COMMON, OTM_01, EBIZ _01	Common is a GUID generated by Business Process Execution Language (BPEL). cross-reference value for OTM is concatenation of common and Domain",::" separating them.
CUSTOMERPARTY_ ADDRESSID	SEBL_01, COMMON, OTM_01, BIZ _01	Common is a GUID generated by BPEL. cross-reference value for OTM is concatenation of common and Domain,"::" separating them.
CUSTOMERPARTY_ CONTACT_COMMID	SEBL_01, COMMON, OTM_01, EBIZ _01	Common is a GUID generated by BPEL. cross-reference value for OTM is concatenation of common and UseCode,"::" separating them.
CUSTOMERPARTY_ CONTACTID	SEBL_01, COMMON, OTM_01, EBIZ _01	Common is a GUID generated by BPEL. cross-reference value for OTM is concatenation of common and Domain,"::" separating them.
CUSTOMERPARTY_ LOCATIONREFID	SEBL_01, COMMON, OTM_01, EBIZ _01	Common is a GUID generated by BPEL. cross-reference value for OTM is concatenation of common and Domain,"::" separating them.
CUSTOMERPARTY_ PARTYCONTACTID	SEBL_01, COMMON, OTM_01, EBIZ _01	Common is a GUID generated by BPEL. cross-reference value for OTM is concatenation of common and UseCode,"::" separating them.
CUSTOMERPARTY_PARTYID	SEBL_01, COMMON, OTM_01, EBIZ_01	Common is a GUID generated by BPEL. cross-reference value for OTM is concatenation of common and Domain,"::" separating them.
ORGANIZATION_ID	SEBL_01, COMMON, OTM_01, EBIZ_01	This table is used to determine the domain value in OTM from the ORG_ID that is being sent from Oracle E-Business Suite.
PAYMENTTERM_ID	COMMON, OTM_01, EBIZ 01, RETL _01	Common is a GUID generated by BPEL. cross-reference value. TBD. Payment Code is the value of EBIZ_01.

12

Working with Domain Value Maps

Domain value maps (DVMs) are a standard feature of the Oracle Service Oriented Architecture (Oracle SOA) suite. They are tables containing mapping between related information in the participating applications. They enable you to equate lookup codes and other static values across applications, for example, **FOOT** and **FT** or **US** and **USA**. These DVM tables are maintained in the AIA layer. The AIA layer uses these DVM tables in transforming the messages from one system in the expected format of the other system.

This chapter includes the following section:

- Section 12.1, "Oracle Financial Management Integration DVMs"

For more information about DVMs, see *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite,* "Working with Message Transformations" and "Working with Domain Value Maps."

12.1 Oracle Financial Management Integration DVMs

Table 12–1 lists the domain value mappings (DVMs) for the Financial Management Integration.

Table 12–1   DVMs for Financial Management Integration

Names	Columns	Description
ACCOUNTINGENTRY_ ACCOUNTINGENTRYTYPE	COMMON, EBIZ, OTM does not have corresponding value	Domain value mapping for the Accrual Type
ACCOUNTINGENTRY_ JOURNALCATCODE	COMMON, EBIZ	Domain value mapping for Journal Category Code
ADDRESS_COUNTRYID	EBIZ_01, COMMON, OTM_01	Domain value mapping for country codes
ADDRESS_ COUNTRYSUBDIVID	EBIZ_01, COMMON, OTM_01	Domain value mapping for state code in supplier address
APPS_USER	USER_NAME, LANG_CODE	Domain Value mapping for the Oracle E-Business Suite User and Language Code
CURRENCYEXCHANGE_ CONVERSIONTYPECODE	EBIZ_01, COMMON, OTM_01	Domain value mapping for currency type codes.
CURRENCYEXCHANGE_ STATUSCODE	EBIZ_01, COMMON,	Domain value mapping for status code of currency exchange rates.
INVOICE_INVOICELINETYPE	COMMON, OTM, EBIZ	Domain Value mapping for the Receivable Invoice Line Type
INVOICE_INVOICETYPE	COMMON, EBIZ, OTM does not have corresponding value	Domain Value mapping for the Invoice Type
LANGUAGE_CODE	EBIZ_01, COMMON, OTM_01	Domain value mapping for Languages

Table 12–1   (Cont.) DVMs for Financial Management Integration

Names	Columns	Description
PAYABLEINVOICE_ PAYABLEINVOICELINETYPE	COMMON, OTM, EBIZ	Domain Value mapping for the Payable Invoice Line Types.
PAYABLEINVOICE_ PAYABLEINVOICETYPE	COMMON, EBIZ, OTM does not have corresponding value.	Domain Value mapping for the Invoice Type.
PAYMENTMETHOD_CODE	COMMON, OTM, EBIZ	Domain Value mapping for the Payment Method Codes
SUPPLIERPARTY_ ADDRESSTYPE	EBIZ_01, COMMON	Domain value mapping to map type of site pay/purchase
SUPPLIERPARTY_ PAYSITEFLAG	EBIZ_01, COMMON, OTM_01	Domain value mapping to map Pay site flag Y/N to True/False
SUPPLIERPARTY_ PRIMARYSITEFLAG	EBIZ_01, COMMON, OTM_01	Domain value mapping to map Primary site flag Y/N to True/False
SUPPLIERPARTY_TYPECODE	EBIZ_01, COMMON, OTM_01	Domain value mapping for supplier type

Table 12–2 displays the domain value mappings (DVMs) seeded data.

Table 12–2   DVM Seed Data

Name	Column	Description
ADDRESS_COUNTRYID	Common, OTM_01, EBIZ_01	NA
COMMUNICATION_METHOD	Common, OTM_01, EBIZ_01	NA
CONTACT_SALUTATION	Common, OTM_01, EBIZ_01	Mr., Mrs., and so on.
CUSTOMERPARTY_ACCTSITESTATUS	Common, OTM_01, EBIZ_01	ACTIVE, INACTIVE
CUSTOMERPARTY_STATUSCODE	Common, OTM_01, EBIZ_01	ACTIVE, INACTIVE
LOCATION_ROLE	Common, OTM_01, EBIZ_01	Ship To, Bill To, and so on.
STATE	Common, OTM_01, EBIZ_01	NA

13

Configuring Oracle Financial Management Integration Pack for OTM and Oracle EBS

This chapter discusses how to configure the Oracle Financial Management Integration Pack for Oracle Transportation Management and oracle E-Business Suite including setting configuration properties, handling errors, and Enterprise Business Object (EBO) implementation maps (EIMs)

This chapter includes the following sections:

- Section 13.1, "Setting Configuration Properties"
- Section 13.2, "Handling Errors"
- Section 13.3, "Enterprise Business Object Implementation Maps"

13.1 Setting Configuration Properties

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

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

13.1.1 Currency Exchange Module

Table 13–1 Currency Exchange Module Configuration Properties

Property Name	Value/Default Value	Description
NoOfDays	Days in Number	This property is used to call Oracle E-Business Suite API by chunking the input date range for initial loads.

13.1.2 Logistics Module

Table 13–2 Logistic Module Configuration Properties

Property Name	Value/Default Value	Description
OTM_01.LANG	ENG	OTM works in a single language at a time. This property is set to OTM operating language. All AIA Inbound transactions read this property and set the OTM language attribute in the XML.

13.1.3 Oracle E-Business Suite Module

Table 13–3   Oracle E-Business Suite Module Configuration Properties

Property Name	Value/Default Value	Description
EBIZ_01.SERVER_TIMEZONE	/	Oracle E-Business Suite runs in a single time zone. This property is set to Oracle E-Business Suite time zone. All Oracle E-Business Suite requester service converts the date time data from this time zone to GMT and all Oracle E-Business Suite providers converts the enterprise business message (EBM) time zone to this server time zone.

13.1.4 SyncCurrencyExchangeListEbizReqABCSImpl

Table 13–4   SyncCurrencyExchangeListEbizReqABCSImp Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	Ebiz_01	It is the responsibility of the application to send the SystemID from which the request is being sent. If any requester application fails to send the SystemID, AIA picks the default SystemID from this configuration property.
Routing.CurrencyExchangeEBS.SyncCurrencyExchangeList.RouteToCAVS	True/False	This property populates EBMHeaders EnvironmentCode and decides whether the CurrencyExchangeEBS should invoke Composite Application Validation System (CAVS) or the provider applications business connector service. If the property value is set to true, EBMHeaders EnvironmentCode is set to CAVS and the enterprise business service (EBS) routes the request to CAVS. If the value is set to false, EBMHeaders Environment Code is set to the EnvironmentCode specified in the AIAConfiguration property Routing.CurrencyExchangeEBS.SyncCurrencyExchangeList.MessageProcessingInstruction.EnvironmentCode. If this property is not set, the default EnvironmentCode is PRODUCTION.
Routing.CurrencyExchangeEBS.SyncCurrencyExchangeList.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.
Routing.CurrencyExchangeEBS.SyncCurrencyExchangeList.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 property holds the URI of the CAVS simulator.
ABCSExtension.PreXformABMtoEBM	True/False	An Enterprise Business Flow can invoke custom code during its execution. These serve as extensibility points. Typical ABCS can have four extension points. This property is used as an extension point before application business message (ABM) is transformed to enterprise business message (EBM). It determines invocation of service at the extension point is to be made or not depending on the property value.
ABCSExtension.PreInvokeEBS	True/False	This property is used as an extension point before application business message (ABM) is transformed to EBM and before Invoking the EBS. It determines invocation of service at the extension point is to be made or not depending on the property value.
Transformation.EnableExtensions	True/False	This property should be set to true when customers want to customize the attribute mapping specified in xsl.

13.1.5 SyncCurrencyExchangeListLogisticsProvABCSImpl

Table 13–5   SyncCurrencyExchangeListLogisticsProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	OTM_01	It is the responsibility of the customers to set the SystemID in EBMHeader to which the request should be sent in the enterprise business service (EBS). If not set, the ProviderABCS routes the message to this DefaultSystemID picked from the configuration file.
Routing.LogisticsWebService.RouteToCAVS	True/False	This property indicates whether the message should be sent to the target application or to Composite Application Validation System (CAVS). If this property is set to true, the message is routed to CAVS, else it is routed to target application through adapter service. The URL of partner link 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.
Routing.LogisticsWebService.OTM_01.EndpointURI	/	If the RouteToCAVS property is set to false, the URI of the partner link is dynamically through the java activity based on this property. This property holds the endpoint URI of the provider application or the adapter service connected to the provider application.
Routing.CurrencyExchangeResponseEBS.SyncCurrencyExchangeListResponse.RouteToCAVS	True/False	RouteToCAVS property decides, whether the Response message from the provider application is sent to the requester application or to CAVS based on the value of the Environment Code.
		If RouteToCAVS is set to true, EnvironmentCode is set to Composite Application Validation System (CAVS) and then the simulator URI is picked up from Routing.CurrencyExchangeResponseEBS.SyncCurrencyExchangeList Response.CAVS.EndpointURI.
		If RouteToCAVS is set to false, EnvironmentCode is set to the value of Routing.CurrencyExchangeResponseEBS.SyncCurrencyExchangeList Response.MessageProcessingInstruction.EnvironmentCode. If this value is NULL, EnvironmentCode is set to PRODUCTION by default.
Routing.CurrencyExchangeResponseEBS.SyncCurrencyExchangeListResponse.CAVS.EndpointURI	/	/
Routing.CurrencyExchangeResponseEBS.SyncCurrencyExchangeList Response.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	/
ABCSExtension.PreXformEBMtoABM	True/False	This property is used as an extension point after enterprise business message (EBM) is transformed to application business message (ABM). It determines invocation of service at the extension point is to be made or not depending on the property value.
ABCSExtension.ABCSExtension.PreInvokeABS	True/False	This property is used as an extension point after EBM is transformed to application business message (ABM) and before invoking the target application. It determines invocation of service at the extension point is to be made or not depending on the property value.

Table 13–5   (Cont.) SyncCurrencyExchangeListLogisticsProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Transformation.EnableExtensions	True/False	This property should be set to true when customers want to customize the attribute mapping specified in xsl.
OTM_01.CONTACT_DOMAIN	PUBLIC	Domain in which Contact GID is created in OTM. Used by OTM to send TransmissionReport back to FMW.
OTM_01.CONTACT_GID	CURRENCY_ESID	This contact is created in OTM to send TransmissionReport to FMW

13.1.6 SyncCustomerPartyListLogisticsProvABCSImpl

Table 13–6   SyncCustomerPartyListLogisticsProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	OTM_01	Based on the SenderHostName obtained from application business message (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 Composite Application Validation System (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 enterprise business message (EBM) message header EnvironmentCode element to the value depending on what is mentioned here.
OTM_01.USERNAME	/	Property specifies the OTM instance User name.
OTM_01.PASSWORD	/	Property specifies the OTM instance password.
LogisticsWebService.LanguageCode	/	This property is used for checking the LanguageCode coming from requester. If that code matches with the acceptable language code of OTM, then the processing moves on forward. If the language codes do not match, the process is terminated.
CallBackURL	7	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 application business message (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.	/
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyList.RouteToCAVS	True/False Default=False	Determines whether the response message from the provider application should be sent to the requester application or to Composite Application Validation System (CAVS).
Transformation.EnableExtensions	True/False Default=False	This property determines enabling extensions in the transformations based on customer requirements.

Table 13–6 (Cont.) SyncCustomerPartyListLogisticsProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.ComMethod	FAX	This property is used to read the default communication method of a contact for a location.
OTM_01.ISPASSWORDENCRYPTED	True/False Default=False	Property specifies whether the OTM password is encrypted in the Configuration Properties file. If true, Decryption Service is being called to decode the OTM password.
Default.CustomerPartyStatusType	CUSTOMERPARTYSTATUS	Determines the StatusType value to be used for the Location in OTM.

For more information about customer master data management related configuration properties, see *Oracle Customer Master Data Management Integration 11.1 - Implementation Guide*.

13.1.7 MergeCustomerPartyListLogisticsProvABCSImpl

Table 13–7 MergeCustomerPartyListLogisticsProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	OTM_01	Based on the SenderHostName obtained from application business message (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.LogisticsWebService.MessageProcessingInstruction.EnvironmentCode	CAVS/PRODUCTION	Sets the Response enterprise business message (EBM) message header EnvironmentCode element to the value depending on what is mentioned here
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 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, Decryption Service is being called to decode the OTM password.
LogisticsWebService.LanguageCode	/	This property is used for checking the LanguageCode coming from requester. If that code matches with the acceptable language code of OTM, then the processing moves on forward If the language codes do not match, the process is terminated.
CallBackURL	/	Property specifies the URL used by OTM to return the response.
ABCSExtension.PreProcessEBM	True/False Default=False	This property sets an extension point before enterprise business message (EBM) is transformed to application business message (ABM). It determines whether a service has to be invoked or not based on its value.
ABCSExtension.PreProcessABM	True/False Default=False	This property sets an extension point after 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 sets an extension point before ABM is transformed to EBM after getting the response from the Target System. It determines whether a service has to be invoked or not based on its value.

Table 13–7   (Cont.) MergeCustomerPartyListLogisticsProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
ABCSExtension.PostProcessEBM	True/False Default=False	This property sets an extension point after ABM is transformed to EBM after getting the response from the Target System. 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.
Routing.CustomerPartyResponseEBSV2.SyncCustomerPartyList.RouteToCAVS	True/False Default=False	Determines whether the response message from the provider application should be sent to the requester 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.

13.1.8 CreatePayableInvoiceListLogisticsReqABCSImpl

Table 13–8   CreatePayableInvoiceListLogisticsReqABCSImp

Property Name	Value/Default Value	Description
Default.SystemID	OTM_01	It is the responsibility of the application to send the SystemID from which the request is being sent. If any requester application fails to send the SystemID, AIA picks the default SystemID from this configuration property.
Routing.PayableInvoiceEBS.CreatePayableInvoiceList.RouteToCAVS	True/false. Default = false	This property populates EBMHeaders EnvironmentCode and decides whether the PayableInvoiceEBS should invoke Composite Application Validation System (CAVS) or the Provider applications business connector service. If the value is set to true, EBMHeaders EnvironmentCode is set to CAVS and the enterprise business service (EBS) routes the request to CAVS. If the value is set to false, EBMHeaders EnvironmentCode is set to the EnvironmentCode specified in the AIAConfiguration property Routing.PayableInvoiceEBS.CreatePayableInvoiceList.MessageProcessingInstruction.EnvironmentCode. If this property is not set, then the default EnvironmentCode is PRODUCTION.
Routing.PayableInvoiceEBS.CreatePayableInvoiceList.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	This property defines the Environment Code to be populated in EBMHeader, which is used by the enterprise business service (EBS) to route it to the corresponding provider application business connector service or CAVS.
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 based on the property value.
Routing.PayableInvoiceEBS.CreatePayableInvoiceList.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.
ABCSExtension.PostXformABMtoEBM	True/false. Default = false.	/
ABCSExtension.PreInvokeEBS	True/false. Default = false	This property is used as an extension point after transforming application business message (ABM) to EBM and before invoking the EBS.
ABCSExtension.PostInvokeEBS	True/false. Default = false	/
Transformation.EnableExtensions	True/false. Default = false	This property should be set to true when customers want to customize the attribute mapping specified in xsl.

13.1.9 CreatePayableInvoiceListEbizProvABCSImpl

Table 13–9   CreatePayableInvoiceListEbizProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	EBIZ_01	It is the responsibility of the customers to set the SystemID in EBMHeader to which the request should be sent in the enterprise business service (EBS). If the SystemID is not set, the ProviderABCS routes the message to this DefaultSystemID picked from the config file.
Routing.CreatePayableInvoiceListEbizDBAdapter.RouteToCAVS	True/false. Default = false	This property indicates whether the message should be sent to the target application or to Composite Application Validation System (CAVS).
		If this property is set to true, the message is routed to CAVS, else it is routed to target application through adapter service. The URI of the partner link is dynamically decided through a java activity based on this property.
Routing.CreatePayableInvoiceListEbizDBAdapter.CAVS.EndpointURI	/	If the RouteToCAVS property is set to true, the URI of the simulator is dynamically derived through the java activity from this property.
Routing.CreatePayableInvoiceListEbizDBAdapter.EBIZ_01.EndpointURI	/	If the RouteToCAVS property is set to false, the URI of the partner link is dynamically derived by the java activity from this property.
		This property should hold the endpoint URI of the provider application or the adapter service connected to provider application.
Routing.PayableInvoiceResponseEBS.CreatePayableInvoiceListResponse.RouteToCAVS	True/false. Default = false	RouteToCAVS property decides, whether the Response message from the provider application is sent to the requester application or to CAVS based on the value of the Environment Code.
		If RouteToCAVS is set to true, EnvironmentCode is set to CAVS and then the simulator URI is picked up from Routing.PayableInvoiceResponseEBS.CreatePayableInvoiceListResponse.CAVS.EndpointURI.
		If RouteToCAVS is set to false, EnvironmentCode is set to the value of Routing.PayableInvoiceResponseEBS.CreatePayableInvoiceListResponse.MessageProcessingInstruction.EnvironmentCode.
		If this value is NULL, it is set to PRODUCTION by default.
Routing.PayableInvoiceResponseEBS.CreatePayableInvoiceListResponse.CAVS.EndpointURI	/	/
Routing.PayableInvoiceResponseEBS.CreatePayableInvoiceListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	/
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 the property value.
ABCSExtension.PostXformEBMtoABM	/	/
ABCSExtension.ABCSExtension.PreInvokeABS	True/false. Default = false	This property is used as an extension point after transforming enterprise business message (EBM) to application business message (ABM) and before invoking the target application. It determines invocation of service at the extension point is to be made or not depending on the property value.
ABCSExtension.ABCSExtension.PostInvokeABS	True/false. Default = false	This property is used as an extension point after transforming enterprise business message (EBM) to application business message (ABM) and before invoking the target application. It determines invocation of service at the extension point is to be made or not depending on the property value.
Transformation.EnableExtensions	True/false. Default = false	This property should be set to true when customers want to customize the attribute mapping specified in xsl.

13.1.10 CreateInvoiceListLogisticsReqABCSImpl

Table 13–10   CreateInvoiceListLogisticsReqABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	OTM_01	Based on the SenderHostName obtained from ABM, sender SystemID is derived but if it is not derived from the SenderHostName, AIA reads it from the config file using this property.
Routing.InvoiceEBS.CreateInvoiceList.RouteToCAVS	True/false. Default = false	EnvironmentCode in the Header population is derived based on this value. If this property value is set to true then the EnviromentCode value is set to Composite Application Validation System (CAVS). If this property value is set to false, read the Routing.InvoiceEBS.CreateInvoiceList.MessageProcessingInstruction.EnvironmentCode property from the config file and set that value for EnvironmentCode.
If Routing.InvoiceEBS.CreateInvoiceList.MessageProcessingInstruction.EnvironmentCode property is not set, the default EnvironmentCode is set to PRODUCTION.
Routing.InvoiceEBS.CreateInvoiceList. MessageProcessingInstruction.EnvironmentCode	PRODUCTION	This property is used at the time of checking the RouteToCAVS property.
Routing.InvoiceEBS.CreateInvoiceList.CAVS.EndpointURI	/	This property is used for setting the Definition ID at the time of EBMHeader population. This holds the URI of CAVS simulator.
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 based on the property value.
ABCSExtension.PreInvokeEBS	True/false. Default = false	This property is used as an extension point after transforming ABM to EBM and before Invoking the EBS. It determines invocation of service at the extension point is to be made or not depending on the property value.
Transformation.EnableExtensions	True/false. Default = false	Setting this property value to true enables the override of the existing mappings.

13.1.11 CreateInvoiceListEbizProvABCSImpl

Table 13–11   CreateInvoiceListEbizProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	EBIZ_01	Target System ID is obtained from EBMHeader but if it is not obtained from the EBMHeader read it from the config file using this property.
Routing.CreateInvoiceListEbizAppsAdapter.RouteToCAVS	True/false. Default = false	TargetEndPointLocation is derived using Java code in the Provider based on this value. If this property is true then the message routes to Composite Application Validation System (CAVS), otherwise it is routed to target application through the Adapter.
Routing.CreateInvoiceListEbizAppsAdapter.CAVS.EndpointURI	/	This property is used to get the EndPointURI when Routing.CreateInvoiceListEbizAppsAdapter.RouteToCAVS is set to true.
Routing.CreateInvoiceListEbizAppsAdapter.EBIZ_01.EndpointURI	/	This property is used to get the EndPointURI when Routing.CreateInvoiceListEbizAppsAdapter.RouteToCAVS is set to false.
Routing.InvoiceResponseEBS.CreateInvoiceListResponse.RouteToCAVS	True/false. Default = false	EnvironmentCode in the Header population for InvoiceEBSResponse is obtained based on this value.
If this property value is set to true, the EnviromentCode value is set to CAVS. If this property value is set to false, read the Routing.InvoiceResponseEBS.CreateInvoiceListResponse.MessageProcessingInstruction.EnvironmentCode property from the config file and set that value for EnvironmentCode.

Table 13–11　(Cont.)　CreateInvoiceListEbizProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Routing.InvoiceResponseEBS.CreateInvoiceListResponse.MessageProcessingInstruction.EnvironmentCode	PRODUCTION	This property is used at the time of checking the RouteToCAVS property.
ABCSExtension.PreXformEBMtoABM	True/false. Default = false	This property is used as an extension point before transforming enterprise business message (EBM) to application business message (ABM). It determines invocation of service at the extension point is to be made or not depending on the property value.
ABCSExtension.PreInvokeApps	True/false. Default = false	This property is used as an extension point after transforming 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 the property value.
Transformation.EnableExtensions	True/false. Default = false	Setting this property to true enables the override of the existing mappings.
RESPONSIBILITY	/	This property along with the User Name is used for setting the AppsContext in SetAppsContext.xsl. This xsl is used for setting the Oracle E-Business Suite User and Responsibility values before invoking PL/SQL API.
EBIZ_01.P_RUN_AUTOINVOICE_CP_Flag Property	T/F. Default = T	This property decides whether AutoInvoice Master Program should be called immediately after inserting Invoice in the Oracle E-Business Suite interface tables. If the value is set to true then the AutoInvoice Master Program is triggered immediately after inserting invoice in the Oracle E-Business Suite interface table, otherwise administrator must manually trigger at a later point of time.
EBIZ_01.P_COMMIT_Flag	T/F. Default = F	This property is used internally within the PL/SQL API along with EBIZ_01.P_RUN_AUTOINVOICE_CP_Flag property for calling the AutoInvoice Master Program.

13.1.12 CreateAccountingEntryListLogisticsReqABCSImpl

Table 13–12　CreateAccountingEntryListLogisticsReqABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	OTM_01	It is the responsibility of the application to send the SystemID from which the request is being sent. If any requester application fails to send this SystemID, AIA picks the default SystemID from this config property.
Routing.AccountingEntryEBSV1.CreateAccountingEntryList.RouteToCAVS	True/false	This property populates the EBMHeaders EnvironmentCode and decides whether the AccountingEntryEBS should invoke CAVS or the Provider applications business connector service.
		If the value is set to true, EBMHeaders EnvironmentCode is set to CAVS and the enterprise business service (EBS) routes the request to Composite Application Validation System (CAVS).
		If the value is set to false, EBMHeaders EnvironmentCode is set to the EnvironmentCode specified in AIAConfiguration property Routing.AccountingEntryEBSV1.CreateAccountingEntryList.MessageProcessingInstruction.EnvironmentCode.
		If this property is not set, the EnvironmentCode is set to PRODUCTION as a default value.
Routing.AccountingEntryEBSV1.CreateAccountingEntryList.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 at the time of checking the RouteToCAVS property.
Routing.AccountingEntryEBSV1.CreateAccountingEntryList.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.

Table 13–12 (Cont.) CreateAccountingEntryListLogisticsReqABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
ABCSExtension.PreXformABMt oEBM	True/false	This property is used as an extension point before transforming application business message (ABM) to enterprise business message (EBM) to EBM. It determines invocation of service at the extension point is to be made or not depending on the property value.
ABCSExtension.PreInvokeEBS	True/false	This property is used as an extension point after transforming ABM to EBM and before Invoking the EBS. It determines invocation of service at the extension point is to be made or not depending on the property value.
Transformation.EnableExtensio ns	True/false	This property should be set to true when customers want to customize the attribute mapping specified in xsl.

13.1.13 CreateAccountingEntryListEbizProvABCSImpl

Table 13–13 CreateAccountingEntryListEbizProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	EBIZ_01	It is the responsibility of the customers to set the SystemID in EBMHeader to which the request should be sent in the enterprise business service (EBS). If the SystemID is not set, the ProviderABCS routes the message to this DefaultSystemID picked from the config file.
Routing.CreateAccountingEntr yListEbizDBAdapterService.Ro uteToCAVS	True/false. Default=false	This property indicates whether the message should be sent to the target application or to Composite Application Validation System (CAVS). If this property is set to true, the message is routed to CAVS, else it is routed to target application through adapter service. The URI of partner link is dynamically decided through a java activity based on this property.
Routing.CreateAccountingEntr yListEbizDBAdapterService.EBI Z_01.EndpointURI	/	If the RouteToCAVS property is set to false, the URI of the partner link is dynamically derived by the java activity from this property. This property should hold the endpoint URI of the provider application or the adapter service connected to the provider application.
Routing.CreateAccountingEntr yListEbizDBAdapterService.CA VS.EndpointURI	/	If the RouteToCAVS property is set to true, the URI of the simulator is dynamically derived by the java activity from this property.
Routing.AccountingEntryRespo nseEBSV1. CreateAccountingEntryListRes ponse.RouteToCAVS	True/false	RouteToCAVS property decides whether the Response message from the provider application should be sent to the requester application or to CAVS based on the Environment Code value. If RouteToCAVS is set to true, EnvironmentCode is set to CAVS and then the simulator URI is picked up from Routing.AccountingEntryResponseEBSV1.CreateAccountingEntryListR esponse.CAVS.EndpointURI. If RouteToCAVS is set to false, EnvironmentCode is set to the value of AccountingEntryListResponse.MessageProcessingInstruction.Environm entCode and If this value is NULL, it is set to PRODUCTION by default.
Routing.AccountingEntryRespo nseEBSV1.CreateAccountingEnt ryListResponse.CAVS.Endpoint URI	/	/
Routing.AccountingEntryRespo nseEBSV1.AccountingEntryList Response. MessageProcessingInstruction.E nvironmentCode	PRODUCTION	/

Table 13–13   (Cont.) CreateAccountingEntryListEbizProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
ABCSExtension.PreXformEBMtoABM	True/false	This property is used as an extension point before transforming enterprise business message (EBM) to application business message (ABM). It determines invocation of service at the extension point is to be made or not depending on the property value.
ABCSExtension. ABCSExtension.PreInvokeABS	True/false	This property is used as an extension point after transforming EBM 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 the property value.
Transformation.EnableExtensions	True/false. Default=false	This property should be set to true when customers want to customize the attribute mapping specified in xsl.

13.1.14 SyncSupplierPartyListEbizReqABCSImpl

Table 13–14   SyncSupplierPartyListEbizReqABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	EBIZ_01	It is the responsibility of the application to send the SystemID from which the request is being sent. If any requester application fails to send this SystemID, AIA picks the default SystemID from this config property.
ABCSExtension.PreXformABMtoEBMSupplierPartyListABM	True/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.SupplierPartyEBS.SyncSupplierPartyList.RouteToCAVS	True/False	EnvironmentCode in the Header population is derived based on this value. If this property value is set to true, the EnviromentCode value is set to CAVS If this property value is set to false, the EnviromentCode is set to the value of Routing.SupplierPartyEBS.SyncSupplierPartyList.MessageProcessingInstruction.EnvironmentCode property from the config file. If Routing.SupplierPartyEBS.SyncSupplierPartyList.MessageProcessingInstruction.EnvironmentCode property is not set, the EnvironmentCode is set to PRODUCTION by default.
Routing.SupplierPartyEBS.SyncSupplierPartyList.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 Composite Application Validation System (CAVS) simulator.
Routing.SupplierPartyEBS.SyncSupplierPartyList. 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 at the time of checking the RouteToCAVS property.
Transformation.EnableExtensions	True/False	This property should be set to true when customers want to customize the attribute mapping specified in xsl.
ABCSExtension.PreInvokeEBSSupplierPartyEBM	/	This property is used as an extension point after transforming application business message (ABM) to enterprise business message (EBM) and before invoking the EBS. It determines invocation of service at the extension point is to be made or not depending on the property value.
RESPONSIBILITY	Responsibility as defined in E-business Suite	The responsibility is used to set the context when retrieving the supplier data from E-Business Suite. The user is obtained from a domain value map based on the language settings and the responsibility is obtained from the Configuration file.

Table 13–14   (Cont.) SyncSupplierPartyListEbizReqABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
BypassAddressIDXref	No/No	This property was introduced for an enhancement for Retek team. This property when set to Yes, it by-passes the cross-reference SUPPLIERPARTY_ADDRESS_ID.
ABCSExtension.PostXformAB MtoEBMSupplierPartyListABM	True/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.
ABCSExtension.PostInvokeEBS SupplierPartyEBM	True/False	This property is used as an extension point after transforming ABM to EBM and before invoking the enterprise business service (EBS). It determines invocation of service at the extension point is to be made or not depending on the property value.

13.1.15 SyncSupplierPartyListLogisticsProvABCSImpl

Table 13–15   SyncSupplierPartyListLogisticsProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Default.SystemID	OTM_01	It is the responsibility of the customers to set the SystemID in EBMHeader to which the request should be sent in the EBS. If the SystemID is not set, the ProviderABCS routes the message to this DefaultSystemID picked from the config file.
ABCSExtension.PreProcessAB M	True/False	This property is used as an extension point before enterprise business message (EBM) is transformed to application business message (ABM). It determines invocation of service at the extension point is to be made or not depending on the property value.
ABCSExtension.PreProcessEBM	True/False	This property is used as an extension point after transforming EBM 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.
Routing.LogisticsWebService.R outeToCAVS	True/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 is routed to CAVS, else it is routed to target application through adapter service. The URI of partner link is dynamically decided through a java activity based on this property.
Routing.LogisticsWebService.C AVS.EndpointURI	/	If the RouteToCAVS property is set to true, the URI of the simulator is dynamically derived by the java activity from this property.
Routing.LogisticsWebService.O TM_01.EndpointURI	/	If the RouteToCAVS property is set to false, the URI of the partner link 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 the provider application.
Routing.SupplierPartyResponse EBS.SyncSupplierPartyList.Rou teToCAVS	True/False	RouteToCAVS property decides whether the Response message from the provider application should be sent to the requester application or to CAVS based on which Environment Code is set while populating ResponseEBM Header.
		If RouteToCAVS is set to true, EnvironmentCode is set to Composite Application Validation System (CAVS) and then the simulator URI is picked up from Routing.SupplierPartyResponseEBS.SyncSupplierPartyList.CAVS.Endpoin tURI.
		If RouteToCAVS is set to false, EnvironmentCode is set to the value of Routing.CurrencyExchangeResponseEBS.SyncCurrencyExchangeList Response.MessageProcessingInstruction.EnvironmentCode
		If this value is NULL, EnvironmentCode is set to PRODUCTION by default.
Routing.SupplierPartyResponse EBS.SyncSupplierPartyList.CA VS.EndpointURI	/	/

Table 13–15 (Cont.) SyncSupplierPartyListLogisticsProvABCSImpl Configuration Properties

Property Name	Value/Default Value	Description
Routing.SupplierPartyResponse EBS.SyncSupplierPartyList.Mes sageProcessingInstruction.Envir onmentCode	PRODUCTION	/
OTM_01.CONTACT_DOMAIN	GUEST	Domain name of OTM where Contact is created. OTM uses this Contact to send Transmission Report to FMW
OTM_01.CONTACT_GID	SUPPLIER_ESID	This contact is created in OTM to send TransmissionReport to FMW.
LocationRef.Address.CountryC ode	USA	This property is used to set the country code for supplier and site.
		Since this is mandatory for creating a Location in OTM and Supplier does not have any address associated, the country code is obtained from the Configuration File. For supplier sites, it is taken from the site address.
DOMAIN.NAME	GUEST	This property is used to get the Domain Name of the Suppliers, which is used as Domain Name in OTM.
SYNCPURCHASINGSITES	True/false	This property is checked to decide whether Supplier Sites should also be synchronized to OTM.
		If it is set to true, the Supplier Sites should be created as Service Providers / Locations in OTM.
ABCSExtension.PostProcessAB M	True/false	This property is used as an extension point before enterprise business message (EBM) is transformed to application business message (ABM). It determines invocation of service.

13.2 Handling Errors

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

13.3 Enterprise Business Object Implementation Maps

For more information about how services are mapped, see *EBO Implementation Maps* (EIMs) on My Oracle Support, article ID: 881022.1.