BEA WebLogic Adapter for Oracle® Applications

User Guide

Release 7.0.3
Document Date: April 2003

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA WebLogic Adapter for Oracle Applications **User Guide**

Part Number	**Date**
N/A	April 2003

About This Document

This document describes the BEA WebLogic Adapter for Oracle Applications and how to use it for developing events and services for integration with other EIS applications

This document covers the following topics:

- Chapter 1, "Introduction," describes a means to exchange real-time business data between Oracle Applications systems and other application, database, or external business partner systems.
- Chapter 2, "Creating Schemas," describes how to use the BEA WebLogic Adapter for Oracle Applications to access and integrate with Oracle Applications using the Oracle JDBC driver and AQ API.
- Chapter 3, "Creating Application Views," describes how to create application views, add services and events to application views, and test application views.
- Chapter 4, "Using Tracing," describes how to generate different kinds of trace logs to diagnose problems and provide audit trails.
- Appendix A, "XML Business Functionality in the BEA WebLogic Adapter for Oracle Applications," describes the business functionality of the XML supported by the BEA Oracle Applications Adapter.

What You Need to Know

This document is intended for system integrators with programming backgrounds and an understanding of the Oracle Applications product in at least some particular application space.

Users of this guide are expected to have the following skill set in order to understand the context of this document:

- General knowledge of the Oracle Applications product environment, including Oracle Applications Server, Oracle Applications GUI, Oracle Applications Tools, and configuration of Oracle Applications Server tasks.
- General knowledge of Oracle Applications enterprise application integration (EAI) concepts including creation and modification of Oracle Applications Business Services and using Oracle Applications Tools.
- Specific Oracle Applications business application knowledge.
- Knowledge of Oracle Applications processes and data models for the required application area.
- General knowledge of BEA WebLogic Integration architecture.
- General knowledge of client-server concepts.

Extensive knowledge of Oracle Applications is not required, but may be helpful in learning about the Oracle Applications Adapter. For system integrators concerned with the development of a client-server interface between Oracle Applications and other applications, this user guide addresses the Oracle Applications integration aspects. It does not cover any other applications or application wrappers.

Related Information

The BEA corporate Web site provides all documentation for WebLogic Server and WebLogic Integration. For information about these products, go to `http://e-docs.bea.com`. Documents that you may find helpful when installing the BEA WebLogic Adapter for Oracle are:

- BEA WebLogic Adapter for Oracle Applications Installation and Configuration Guide
- BEA WebLogic Adapter for Oracle Applications Release Notes
- BEA WebLogic Server installation and user documentation, which is available at the following URL:

`http://edocs.bea.com/more_wls.html`

- BEA WebLogic Integration installation and user documentation, which is available at the following URL:

  `http://edocs.bea.com/more_wli.html`

Oracle Applications Documentation

Oracle Applications manuals are available from Oracle Inc. and are provided typically online or on CD-ROM. The following topics, based on the Oracle Applications Books for Version 11.x, Oracle Database 8i and 9i of Oracle Applications, are applicable:

- "Overview" in *Oracle Application Concepts*
- "Database" in *Oracle 8i/9i Users Guide*
- "Transports and Interfaces" in *Network and Security Guide*
- "Concepts" in *Oracle 9i Concepts*
- "Administration" in *Oracle Applications System Administration Guide*
- "Business Processes and Rules" in *Oracle Workflow Guide*
- BEA Application Explorer documentation
- WebLogic Server documentation
- WebLogic Integration documentation

Contact Us!

Your feedback on the BEA WebLogic Adapter for Oracle Applications documentation is important to us. Send us e-mail at `docsupport@bea.com` if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA WebLogic Adapter for Oracle Applications Controlled Availability documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Adapter for Oracle Applications Controlled Availability release.

If you have any questions about this version of BEA WebLogic Adapter for Oracle, or if you have problems installing and running BEA WebLogic Adapter for Oracle, contact BEA Customer Support through BEA WebSupport at `www.bea.com`. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

Convention	Item

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Examples*: LPT1 SIGNON OR
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

Convention	Item

. . .	Indicates one of the following in a command line:
	■ That an argument can be repeated several times in a command line
	■ That the statement omits additional optional arguments
	■ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Table of Contents

3. Creating Application Views

CHAPTER

1 Introduction

This section describes how BEA WebLogic Adapter for Oracle Applications provides a means to exchange real-time business data between Oracle E-Business Suite systems and other application, database, or external business partner systems using BEA WebLogic Integration and XML messages. (The Oracle E-Business Suite was formerly known as Oracle Applications, and is referred to that way in this manual.)

This section includes the following topics:

Resource Adapters

The BEA WebLogic Adapter for Oracle Applications is a *resource adapter*. Resource adapters connect one application to another when those applications are not originally designed to communicate with each other. Adapters are bidirectional; they can send requests to the Enterprise Information System (EIS), as well as receive notification of events occurring in the EIS. When the adapter receives event notification, it is referred to as an *event adapter* or *event*. When it makes requests of the EIS, it is referred to as a *service adapter* or *service*.

The *Oracle Applications event adapter* is a specific type of adapter that is triggered when an event occurs in an Oracle Applications environment. The information for the event is then used to build an XML record and forwarded to any specified service for further processing.

Event adapters receive an EIS-specific message from an EIS when an event occurs. They are consumers of event messages and may or may not provide a response.

Event adapters perform the following functions:

- Receive event notification from an EIS client.
- Transform the EIS-specific message format into an XML format. The XML format conforms to the XML schema for the event. The schema is based on metadata in the EIS.

The *Oracle Applications service adapter* is a generic service capable of processing requests for Oracle Applications interface tables, or for Advance Queuing (AQ) queues embedded in XML requests, and forwarding them to a backend Oracle Applications system. The resulting response information is then brought back and processed by the Oracle Applications service adapter for further routing.

Service adapters receive an XML request document from a client and call a specific function in the target EIS. They are consumers of request messages, and depending on the request, may or may not provide a response. With the asynchronous nature of the service adapter, the client application issues a service request and then proceeds with its processing. The application does not wait for the response.

Service adapters perform the following functions:

- Receive service requests from an external client.
- Transform the XML request document into the EIS-specific format. The request document conforms to the request XML schema for the service. The schema is based on metadata in the EIS.
- Transform the response from the EIS-specific data format to an XML document that conforms to the response XML schema for the service. The schema is based on metadata in the EIS.

Key features of the BEA WebLogic Adapter for Oracle Applications include support for:

- Bidirectional message interaction with Oracle Applications. Inbound data is delivered by the service adapter. Outbound messages are transmitted by the event adapter.
- BEA Application Explorer, which uses metadata on your Oracle Applications system to build application views (XML schemas for database events and SQL requests or responses) used by the business process workflows.
- Service (Oracle inbound) and event (Oracle outbound) adapter integration operations with Oracle Applications presenting XML schemas to the business process workflows.
- Message Format Language

Application Access and Interaction

Applications that access Oracle Applications data when an Oracle Applications business event occurs use WebLogic Integration application views, events, and BEA business process management functionality to receive messages from Oracle Applications using the adapter.

Applications that interact with Oracle Applications to cause a new Oracle Applications business event use WebLogic Integration application views, services, and BEA business process management functionality to send messages to Oracle Applications using the adapter.

Executing Oracle Applications Functions

You can use the BEA WebLogic Adapter for Oracle Applications to invoke an Oracle Applications business process such as add, update, or delete account information. You can also use the adapter as part of an integration effort to connect Oracle Applications with non-Oracle Applications systems. The adapter is bi-directional, meaning that it can detect an event in Oracle Applications either by receiving an XML document on an Oracle Advanced Queuing (AQ) queue or by polling tables in an Oracle database.

The adapter can be the cause of an Oracle Applications business event by either passing an XML document into Oracle Applications using an Oracle AQ queue or by using the Oracle JDBC driver to update an interface table.

Accessing Data Stored in Oracle Applications

Oracle Applications provides several methods for moving data into and out of base tables. These methods include using the interface tables, the iProcurement Connector, the XML Gateway, and table base access. No single option is optimal for moving transactional information to and from Oracle Applications.

The BEA WebLogic Adapter for Oracle Applications enables you to import data into Oracle Applications using Oracle-supplied interface tables. Instead of transforming input event documents into standards-based XML structures, the event document is transformed to match the appropriate interface table. Interface tables are themselves conduits to the base tables. Oracle concurrent programs (initiated by the user) move the data from adapter-specific interface tables to their target base tables and/or error handling tables.

Prior to loading the interface table, the BEA WebLogic Adapter for Oracle Applications verifies the XML structure being processed for the appropriate field and data type information. After the input information is verified, the record is loaded into the interface table.

For a list of Oracle open interface tables and their associated modules, see the following tables.

Oracle Financials Interface Tables

Table 1-1 Oracle General Ledger

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
JournalEntries	Inbound	Table	GL_INTERFACE
BudgetAmounts	Inbound	Table	BUDGET_INTERFACE
DailyRates	Inbound	Table	GL_DAILY_RATES_INTERFACE

Table 1-2 Oracle Accounts Payable

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
VendorInvoices (from Invoice Gateway, EDI Gateway, other sources	Inbound	Table	AP_INVOICES_INTERFACE AP_INVOICE_LINES_INTERFACE
CreditCardTransactions	Inbound	Table	AP_EXPENSE_FEED_LINES

Table 1-3 Oracle Cash Management

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
BankStatements	Inbound	Table	CE_STATEMENT_HEADERS_INT_ALL CE_STATEMENT_LINES_INTERFACE

Table 1-4 Oracle Receivables

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
Customers	Inbound	Table	RA_CUSTOMER_INTERFACE RA_CUSTOMER_PROFILES_INTERFACE RA_CONTACT_PHONES_INTERFACE RA_CUSTOMER_BANKS_INTERFACE
CustomerInvoices	Inbound	Table	RA_INTERFACE_LINES RA_INTERFACE_SALESCREDITS RA_INTERFACE_DISTRIBUTIONS
ReceiptsFromBanks	Inbound	Table	AR_PAYMENTS_INTERFACE_ALL

Table 1-5 Oracle Fixed Assets

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
CreateAssets (mass additions)	Inbound	Table	FA_MASS_ADDITIONS

Oracle Manufacturing/Distribution

Table 1-6 Oracle Inventory

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
Transactions	Inbound	Table	MTL_TRANSACTIONS_INTERFACE MTL_SERIAL_NUMBERS_INTERFACE MTL_TRANSACTION_LOTS_INTERFACE
DemandInterface	Inbound	Table	MTL_DEMAND_INTERFACE
OnHandBalances	Outbound	View	MTL_ITEM_QUANTITIES_VIEW
UserDefinedSupply	Inbound	Table	MTL_USER_SUPPLY
UserDefinedDemand	Inbound	Table	MTL_USER_DEMAND
Replenishment	Inbound	Table	MTL_REPLENISH_HEADERS_INT MTL_REPLENISH_LINES_INT
Item	Inbound	Table	MTL_SYSTEM_ITEMS_INTERFACE MTL_ITEMS_REVISIONS_INTERFACE
CustomerItem	Inbound	Table	MTL_CI_INTERFACE
CustItemCrossReferences	Inbound	Table	MTL_CI_XREFS_INTERFACE

Table 1-7 Oracle Engineering / Oracle Bills of Material

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
MFGCalendar	Outbound	View	BOM_CALENDAR_MONTHS_VIEW
BillofMaterial	Inbound	Table	BOM_BILL_OF_MTLS_INTERFACE BOM_INVENTORY_COMPS_INTERFACE BOM_REF_DESGS_INTERFACE BOM_SUB_COMPS_INTERFACE MTL_ITEMS_REVISIONS_INTERFACE
Routings	Inbound	Table	BOM_OP_ROUTINGS_INTERFACE BOM_OP_SEQUENCES_INTERFACE BOM_OP_RESOURCES_INTERFACE MTL_RTG_ITEM_REVS_INTERFACE
ECO (engineering change orders)	Inbound	Table	ENG_ENG_CHANGES_INTERFACE ENG_ECO_REVISIONS_INTERFACE ENG_REVISED_ITEMS_INTERFACE BOM_INVENTORY_COMPS_INTERFACE BOM_REF_DESGS_INTERFACE BOM_SUB_COMPS_INTERFACE

Table 1-8 Oracle Cost Management

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
ItemCostInquiry	Outbound	View	CST_INQUIRY_TYPES CSTFQVIC (View Item Cost Information)
MFGCostReporting	Outbound	View	CST_REPORT_TYPES CSTRFICR (Inventory Valuation Report)

Table 1-8 Oracle Cost Management

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
PeriodicCost (only for the 1st opened period)	Inbound	Table	CST_PC_ITEM_COST_INTERFACE CST_PC_COST_DET_INTERFACE

Table 1-9 Oracle Master Scheduling/MRP and Oracle Supply Chain Planning

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
Forecast	Inbound	Table	MRP_FORECAST_INTERFACE
ForecastEntries	Inbound	Process PL/SQL Table	T_FORECAST_INTERFACE T_FORECAST_DESIGNATOR
MasterSchedule	Inbound	Table	MRP_SCHEDULE_INTERFACE
MasterScheduleRelief	Inbound	Table	MRP_RELIEF_INTERFACE
PlannerWorkbench	Outbound	Process	Stored Procedure MRPPL06, or WIP_JOB_SCHEDULE_INTERFACE PO_REQUISITIONS_INTERFACE PO_RESCHEDULE_INTERFACE
ProjectedRequirements	Outbound	Table	MRP_RECOMMENDATIONS
ProjectedSupply	Outbound	Table	MRP_GROSS_REQUIREMENTS

Table 1-10 Oracle Master Scheduling/MRP and Oracle Supply Chain Planning

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
OrderImport	Inbound	Table	SO_HEADERS_INTERFACE_ALL SO_HEADER_ATTRIBUTES_INTERFACE SO_LINES_INTERFACE_ALL SO_LINE_ATTRIBUTES_INTERFACE SO_LINE_DETAILS_INTERFACE SO_SALES_CREDITS_INTERFACE SO_PRICE_ADJUSTMENTS_INTERFACE
DeliveryBasedShip Confirm Open Interface	Inbound	Table	WSH_DELIVERIES_INTERFACE WSH_PACKED_CONTAINER_INTERFACE WSH_PICKING_DETAILS_INTERFACE WSH_FREIGHT_CHARGES_INTERFACE

Table 1-11 Oracle Purchasing

Interface Name	Data Flow Direction	Table, View, or Process	Table, View, or Module Name
Requisitions	Inbound	Table	PO_REQUISITIONS_INTERFACE PO_REQ_DIST_INTERFACE
RequisitionReschedule	Inbound	Table	PO_RESCHEDULE_INTERFACE
PurchasingDocuments	Inbound	Table	PO_HEADERS_INTERFACE PO_LINES_INTERFACE
Receiving	Inbound	Table	RCV_HEADERS_INTERFACE RCV_TRANSACTIONS_INTERFACE

Table 1-12 Oracle Quality

Interface Name	Data Flow Direction	Table, View or Process	Table, View, or Module Name
CollectionImport	Inbound	Table	QA_RESULTS_INTERFACE
DynamicCollectionPlanView	Outbound	View	Q_COLLECTION_PLAN_NAME_V
DynamicCollectionImportView	Inbound	View	Q_COLLECTION_PLAN_NAME_IV

Table 1-13 Oracle Work in Process

Interface Name	Data Flow Direction	Table, View, or Module Name
WIPMoveAdapter.	Inbound	WIP_MOVE_TXN_INTERFACE
Resource	Inbound	WIP_COST_TXN_INTERFACE
WIPWorkOrderAdapter	Inbound	WIP_JOB_SCHEDULE_INTERFACE
Material	Inbound	MTL_TRANSACTIONS_INTERFACE MTL_SERIAL_NUMBERS_INTERFACE MTL_TRANSACTION_LOTS_INTERFACE
WIPScheduling	Outbound/ Inbound	Stored Procedures LOAD_INTERFACE and LOAD_WIP and WIP_SCHEDULING_INTERFACE
Stored Procedures	LOAD_INTERFACE and LOAD_WIP and	WIP_SCHEDULING_INTERFACE

The BEA WebLogic Adapter for Oracle Applications enables you to take advantage of Oracle-supplied gateways into and out of Oracle Applications. You can use the adapter to transfer event information to and from Oracle Applications using Open Applications Group Specification (OAGIS)-formatted OAGIS XML, commonly known as OAG BODs. These documents are then processed by the iProcurement Connector for input into Oracle Applications or generated by the iProcurement

Connector for outbound information. This type of integration is ideal if your organization currently uses this technology, because it allows quick integration of additional processes through an existing system.

The Oracle AQ system passes and retrieves OAGIS-formatted XML documents to and from the iProcurement Connector.

In many cases the external system generates the following types of event documents:

- Documents that do not support the OAGIS formats.
- Documents that do support the OAGIS formats but may have extensions or require data enrichment or cleansing.

Perform data enrichment or cleansing prior to loading the data into Oracle Applications.

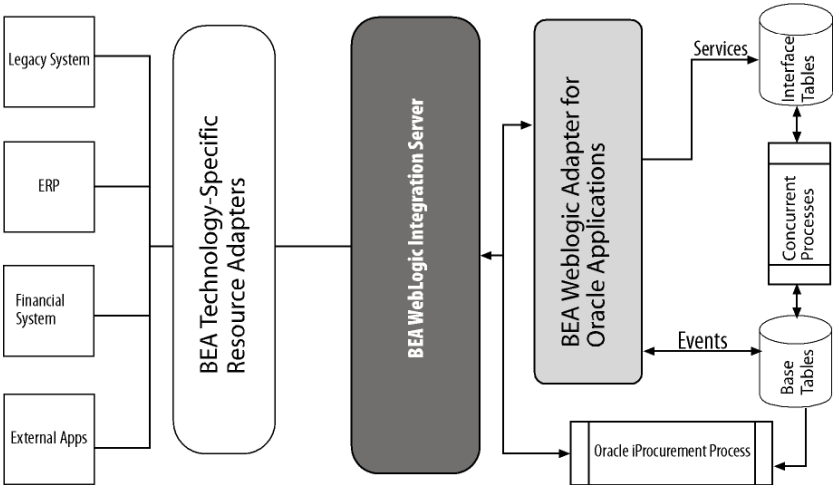
Oracle Applications Enterprise Application Integration Architecture

Oracle Applications provides for integration with other applications and systems using the Oracle Applications Enterprise Application Integration (EAI) framework and the Business Integration Manager facility. The BEA WebLogic Adapter for Oracle Applications uses the Oracle Applications-provided framework and leverages various supplied integration access methods in order to provide the greatest amount of flexibility and functionality while working within the Oracle Applications framework.

Integration access methods supported by the BEA WebLogic Adapter for Oracle Applications include:

- Oracle Advanced Queuing
- Oracle Interface Tables
- Base Table access

Figure 1-1 Oracle Applications Integration Architecture



Using the BEA Application Explorer

The BEA Application Explorer uses an explorer metaphor for browsing the Oracle Applications system for interface table metadata and for presenting schemas for use with OAGIS standard XML. The function of the explorer is to create service and event schemas for the associated interface table, OAG BOD, or database table.

When running a service using the interface tables, you create BEA schemas for the service against an Oracle Applications interface table. A schema corresponds to the table layout for the specific interface table.

When running an event or service using OAGIS XML, you use the Application Explorer to create the BEA schema definitions for the desired XML document. In this case, you can create both event and service schemas for the particular document.

CHAPTER

2 Creating Schemas

WebLogic Integration's application views require schemas to validate the XML documents generated and consumed by the service and event adapters. Specifically, an application view's services use schemas to validate the XML input into a service (a request document) and to validate the XML output from a service (a response document). An application view's events use schemas to validate the XML output from an event and event document.

This section describes how to use the BEA WebLogic Adapter for Oracle Applications to access and integrate with Oracle Applications using the Oracle JDBC driver and AQ API. It includes the following topics:

Creating Schemas for Services

The BEA WebLogic Adapter for Oracle Applications enables the integration of transactions to and from Oracle Applications using interface tables, iProcurement Connector, and database tables. Using the adapter, you can access and integrate Oracle Applications using the Oracle JDBC driver and Advanced Queuing (AQ) API.

When running a service using the BEA WebLogic Adapter for Oracle Applications, you must first create BEA request and response schemas for the service. Use the BEA Application Explorer to generate these schemas.

The service request begins with the receipt of a service request document and, in most cases, ends with an XML response document indicating the result of the business function execution.

The following topics describe the steps for creating a schema.

Starting the BEA Application Explorer

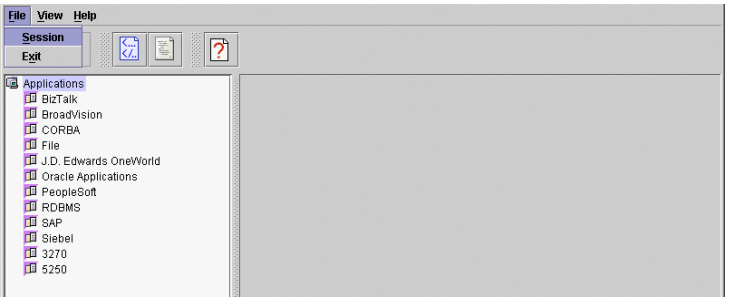
Start the BEA Application Explorer. For more information on the BEA Application Explorer, see the *BEA Application Explorer Installation and Configuration Guide*.

Figure 2-1 BEA Application Explorer Initialization Window

Establishing the Working Directory

After you open the BEA Application Explorer, you can establish the directory associated with BEA WebLogic Server to use when importing service and event XML schemas into the application view repository.

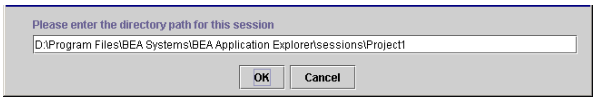
Figure 2-2 BEA Application Explorer - Establishing a Working Directory Window



1. Choose File→Session.

The Enter Session Path dialog box opens.

Figure 2-3 Enter Session Path Dialog Box



2. Enter a folder name.

   In this example, `D:\Program Files\BEA Systems\BEA Application Explorer\sessions\Project1` is the BEA Application Explorer's working directory—the location in which schemas are placed when the BEA Application Explorer generates them.

   The full name of the location in which schemas are saved is:

   `base_directory``\OracleApps\`*`connection_name`*

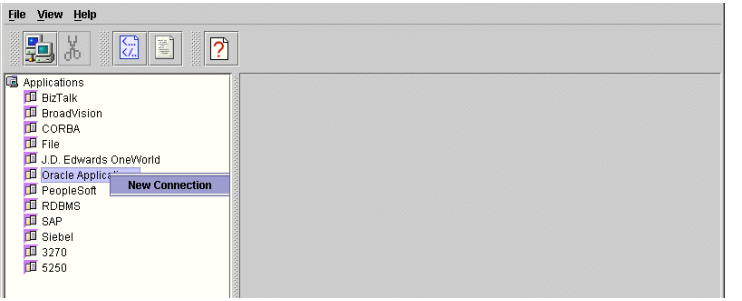
   Here, *`base_directory`* is the directory path for the session, and *`connection_name`* is the name of the connection to Oracle Applications.

   The session path appears at the bottom of the BEA Application Explorer window.

Establishing a Connection to Oracle Applications

The BEA Application Explorer provides an easy method for creating connections to an Enterprise Information System (EIS).

Figure 2-4 BEA Application Explorer - Establishing a New Connection Window



To create a connection:

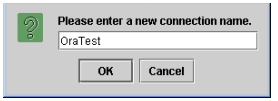
1. Right-click the Oracle Applications node.

   As there are no previous connections in this example, only the New Connection option appears.

2. Select New Connection.

   The Input Dialog box displays.

Figure 2-5 Input Dialog Box - Entering a New Connection Name



   a. In the Input dialog box, enter a name for the Oracle Applications connection, for example, `OraTest`.

   b. Click OK. The information is saved and a new subdirectory under the OracleApps subdirectory of your session path is created.

   The following window, Oracle Applications Logon, prompts you for information for accessing your Oracle Applications system.

Figure 2-6 Oracle Applications Logon Window

3. Enter the parameters that establish the connection between the Oracle Applications client and the Oracle Applications system:

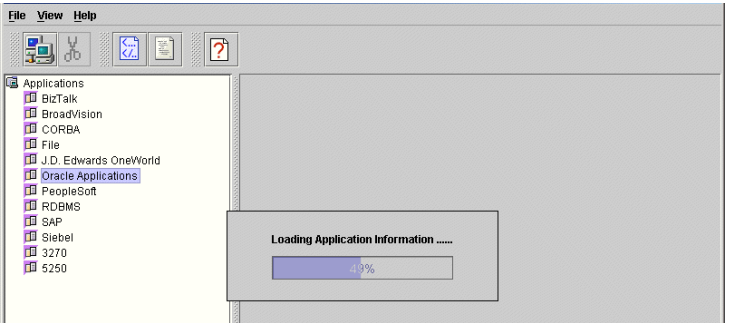
   - Host: The name of the server on which Oracle Applications is running or its IP address.
   - Port: The port number on which the server is listening.
   - SID: The database system ID for the instance of Oracle Applications.
   - User: A valid user ID for Oracle Applications.
   - Password: The password associated with the user ID.

For more information on these parameters, see your Oracle Applications documentation, or ask your Oracle Applications system administrator.

4. Click OK.

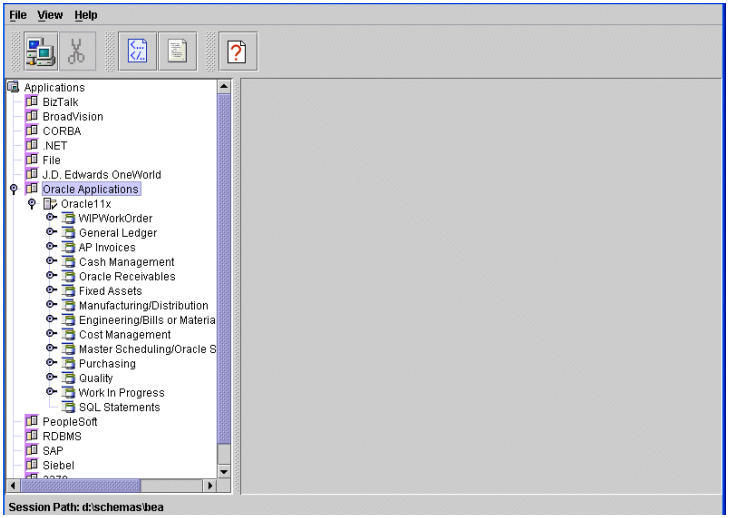
   The BEA Application Explorer displays a progress message, as shown in the following figure.

Figure 2-7 BEA Application Explorer - Extraction Progress Message



After loading the application information, the BEA Application Explorer displays a tree with a node showing the new connection (in this example, `OraTest`).

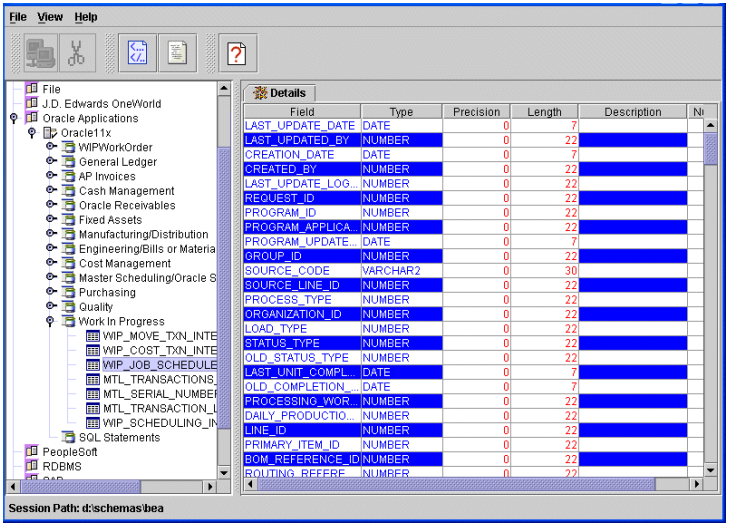
Figure 2-8 BEA Application Explorer - New Connection



5. Click the icon to the left of the connection name to display the Oracle Applications interface tables. You can expand and explore all the interface tables available.

6. Click a specific interface table in the left pane to display detailed information about that table in the right pane.

Figure 2-9 BEA Application Explorer - Business Function Details Window



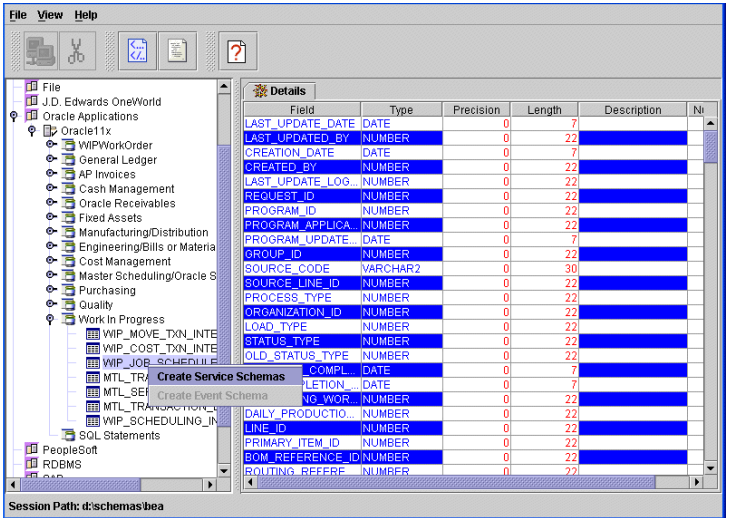
Creating the Service Schemas

The BEA Application Explorer creates the following WebLogic Integration schemas:

- Service XML Request Schema
- Service XML Response Schema

It also generates the appropriate entries in the `manifest.xml` file. This file contains connection and configuration information for the Oracle Applications server on which the schemas were created, as well as pointers to the schemas themselves.

Figure 2-10 BEA Application Explorer - Creating Service Schemas

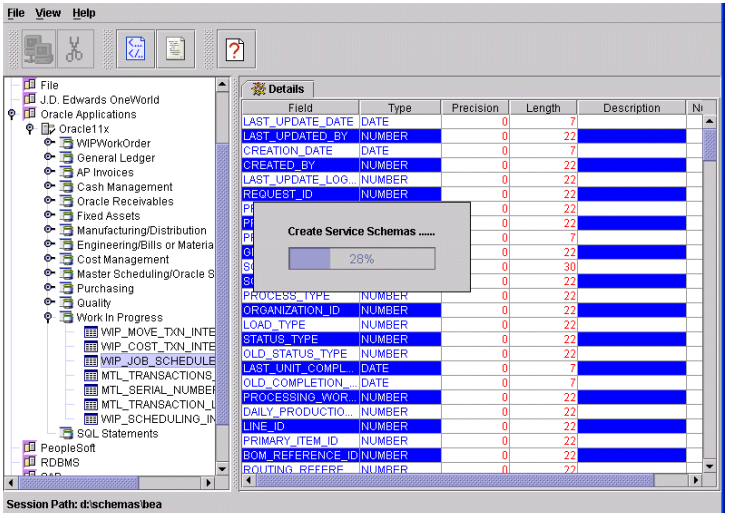


1. In the BEA Application Explorer, right-click the desired interface table and select Create Service Schemas (this option includes request and response schemas).

   The BEA Application Explorer accesses the Oracle Applications system and builds schemas, which are then published to the WebLogic Integration repository.

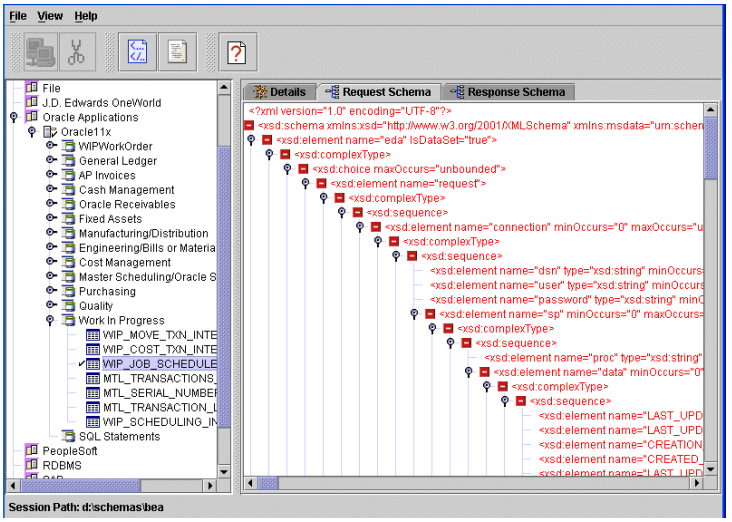
   A progress message indicates that the schemas are being created, as shown in the following figure.

Figure 2-11 BEA Application Explorer - Service Schema Progress Message



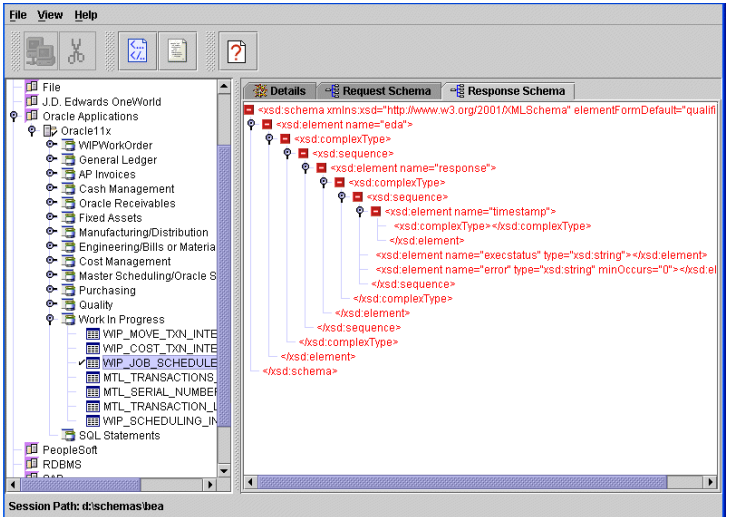
After the schemas have been created, you can view the request schema.

Figure 2-12 BEA Application Explorer - Request Schema Window



2. Select the Request Schema tab at the top of the right pane to view the schema.

Figure 2-13 BEA Application Explorer - Response Schema Window

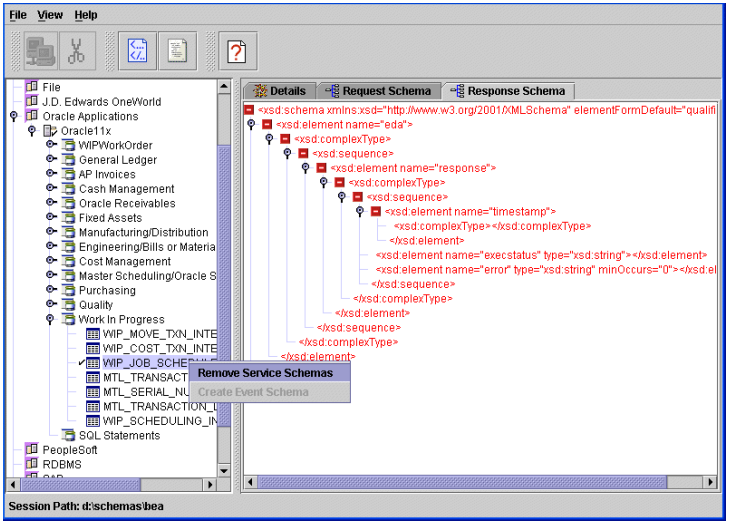


3. Select the Response Schema tab at the top of the right pane to view the schema.

Removing a Service Schema

You can remove a service schema. The BEA Application Explorer determines if the schemas are present, and if they are, displays the option to remove them.

Figure 2-14 BEA Application Explorer - Removing Service Schemas Option

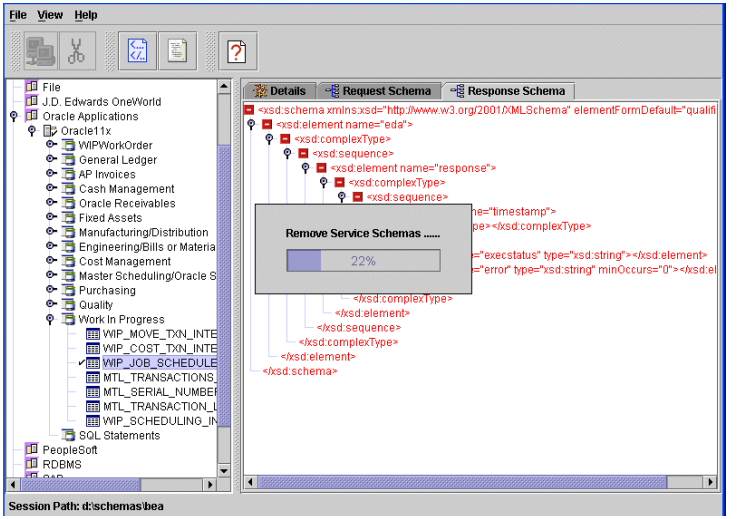


To remove a service schema:

1. Right-click the desired interface table to display the Remove Service Schemas option (request and response schemas).
2. Select Remove Service Schemas.

The BEA Application Explorer displays a progress message while it removes the schemas from `the manifest.xml` file.

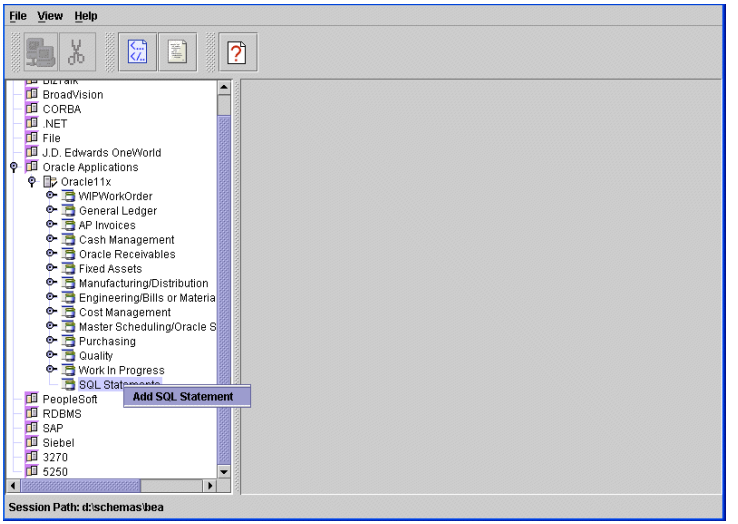
Figure 2-15 BEA Application Explorer - Removing Service Schemas



Creating Schemas for Events

Schemas for events are generated under the SQL Statement node.

Figure 2-16 BEA Application Explorer - Add SQL Statement

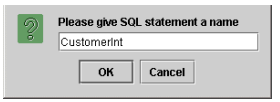


To start the process of generating an event schema:

1. Right-click the SQL Statement node.
2. Select Add SQL Statement.

   The SQL statement Input box appears.

Figure 2-17 SQL Statement Name Input Box



3. Enter a name for the schema group.

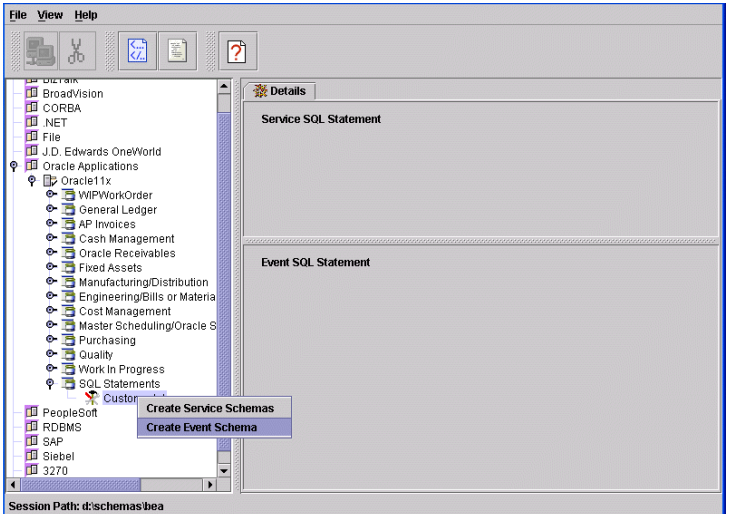
   It is good practice to specify a name that describes the service. For example, `CustomerInt` would represent an event on the Customer Interface table.

   Make a note of this name, as you will need it when you add an event to an application view, as described in "Adding an Event to an Application View" in Chapter 3, "Creating Application Views." Otherwise, you will only be able to access this name by running the BEA Application Explorer again.

4. Click OK.

   After the SQL Statement node is built, you are ready to build schemas.

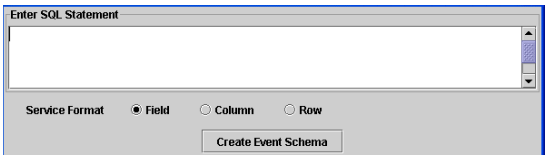
Figure 2-18 BEA Application Explorer - Create Event Schema Option



5. To generate the schemas, right-click the SQL Statement and select the Create Event Schemas option.

   The Create Event Schema window opens.

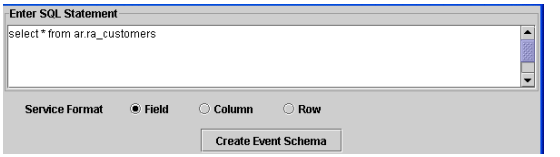
Figure 2-19 Create Event Schema Window



6. Enter the SQL statement to be used by the application view event.

   The BEA WebLogic Adapter for Oracle Applications listener enables the creation of events that utilize complex SQL statements, including JOINs, WHEREs, and other verbs. When entering the SQL statement, be sure to use the SQL generating the input record for the event.

Figure 2-20 SQL Statement Entered

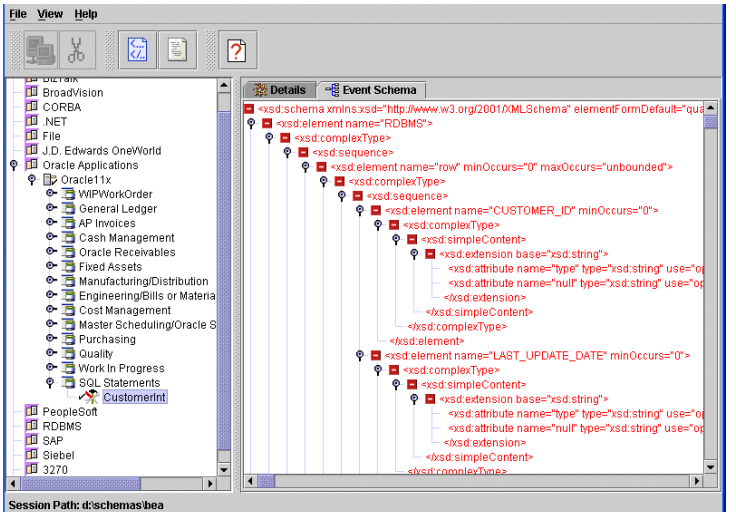


7. Select the appropriate service format for the returning result set.
8. Click Create Event Schema.

   If there is a problem executing the SQL statement, the error message from Oracle is displayed in an error window.

   In the right pane, the Details tab displays the SQL statement used, and the Events tabs displays the event schema, as shown in the following figure.

Figure 2-21 BEA Application Explorer - Schema Display

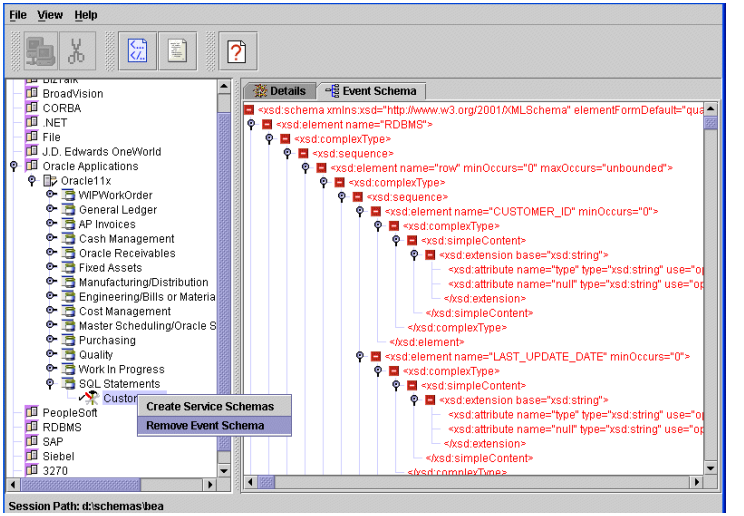


The schema is now generated and ready to use.

Removing Schemas

You can remove schemas.

Figure 2-22 BEA Application Explorer - Removing Schema Option



To remove a schema from an SQL Statement:

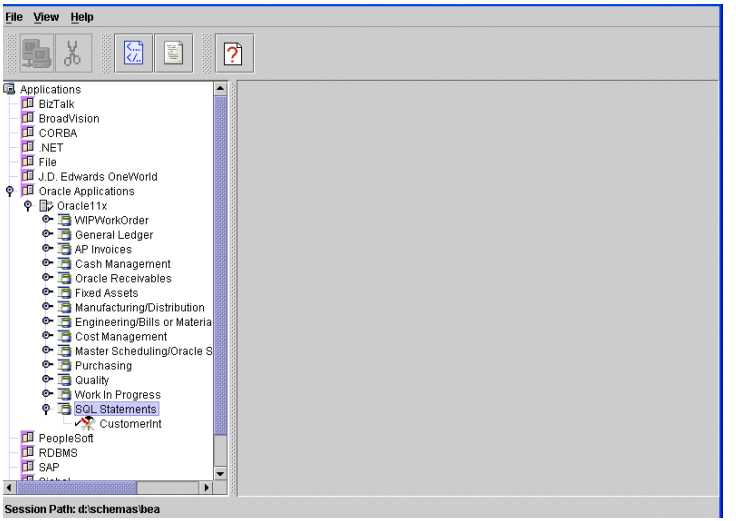
1. Right-click the desired SQL Statement
2. Select a remove schema option.

As shown in the previous figure, the event schema is available to be removed. The schema is removed, and the `manifest.xml` file is updated. The right pane will no longer display the tabs for the schema.

Viewing Other Files

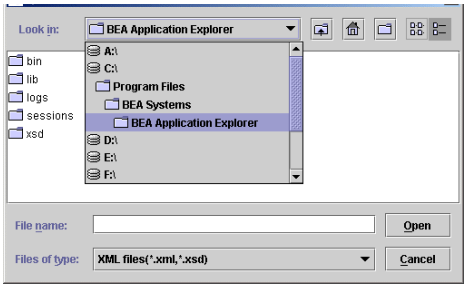
You can view XML and non-XML text files in addition to the created schemas.

Figure 2-23 BEA Application Explorer - View Menu



1. To view other files, choose View→View XML or View→View Text.

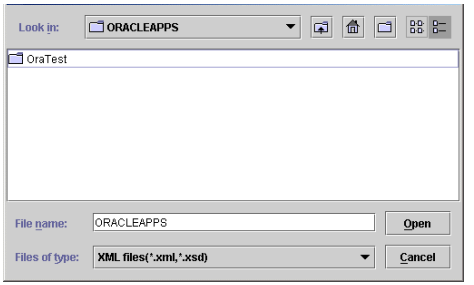
Figure 2-24 Open Box - Browsing to the Desired File



2. Browse to the desired file.

3. To navigate between directories, select the desired subdirectory from the drop-down list or use one of the navigation buttons in the top right of the window.

Figure 2-25 Open Box - Locating manifest.xml File



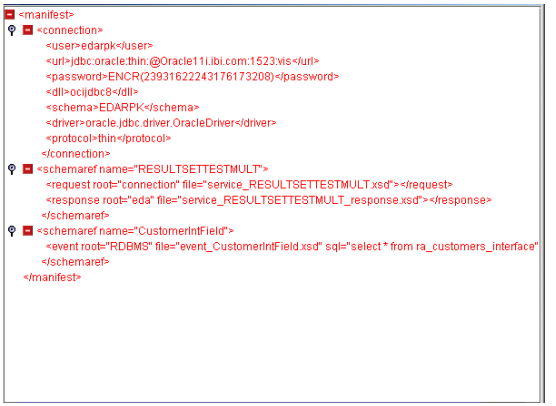
4. Locate `manifest.xml`.

   This file contains information about the schemas used for different events and services, as well as connection information for the EIS.

5. Select the file and click Open.

   The following sample `manifest.xml` file contains the connection, configuration, and schema information.

Figure 2-26 Sample manifest.xml File



3 Creating Application Views

When you define an application view, you create an XML-based interface between WebLogic Server and a particular Enterprise Information System (EIS) application within your enterprise. Once you create an application view, a business analyst can use it to create business processes that use the application. With the BEA WebLogic Adapter for Oracle Applications, you can create any number of application views, each with any number of services and events.

This section describes how to create application views, add services and events to application views, and test application views. It includes the following topics:

Defining an Application View

1. Log on to the WebLogic Integration Application View Console.

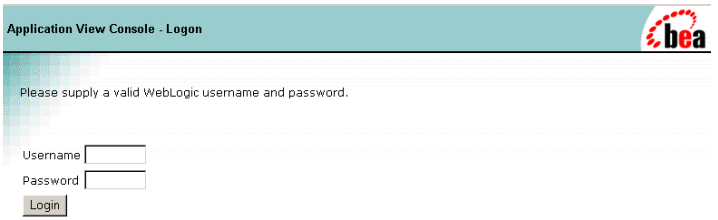
For more information, see "Defining an Application View" in *Using Application Integration*:

- For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/2usrdef.htm`
- For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/2usrdef.htm`

2. Access the Application View Console logon page at http://*host*:*port*/wlai.

   Here, `host` is the IP address or DSN of the WebLogic Server, and `port` is the port number on which the server is listening. The default port number at install time is 7001.

Figure 3-1 Application View Console Logon Window

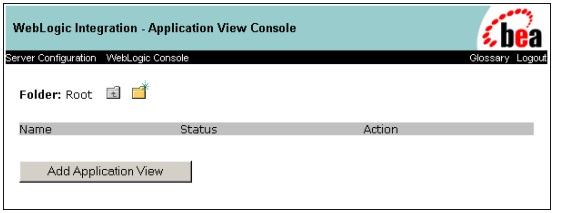


3. If prompted, add a user name and password.

   Note: If the user name is not `system`, it must be included in the `adapter` group. For more information on adding the administrative server user name to the `adapter` group, see the *BEA WebLogic Adapter for Oracle Applications Installation and Configuration Guide*.

4. Click Login.

Figure 3-2 Application View Console Window



5. On the console, click Add Application View to create a new application view for the BEA WebLogic Adapter for Oracle Applications.

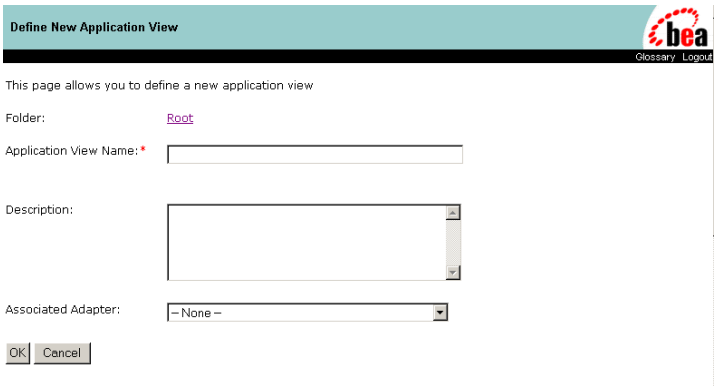
   An application view enables a set of business processes for the adapter's target EIS application.

   For more information, see "Defining an Application View" in *Using Application Integration*:

   - For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/2usrdef.htm`
   - For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/2usrdef.htm`

   The Define New Application View window opens.

Figure 3-3 Define New Application View Window



6. Enter the required information.

   The folder is the current active folder in which you are working.

a. In the Application View Name field, enter a name that describes the set of functions performed by this application.

   Each application view name must be unique to its adapter.

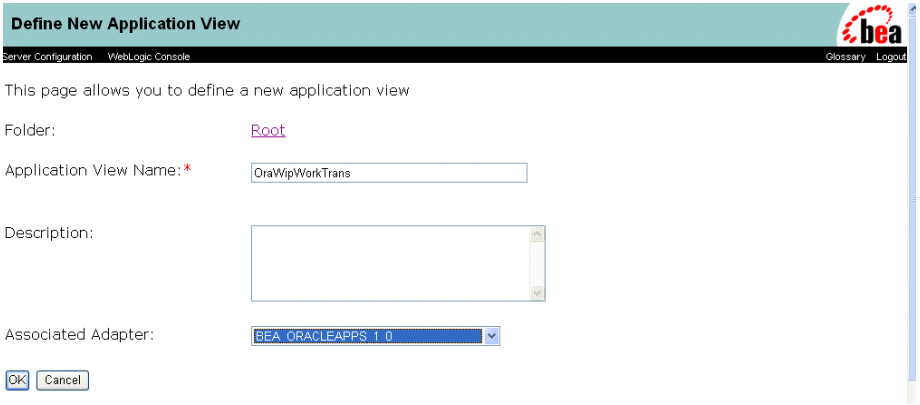
   Valid characters are a-z, A-Z, 0-9, and _ (underscore).

b. In the Description field, enter any relevant notes. Users view these notes when they use this application view in workflows, using business process management functionality.

c. From the Associated Adapter list, select the `BEA_ORACLEAPPS_1_0` adapter for use in creating this application view.

The following is an example of a completed page.

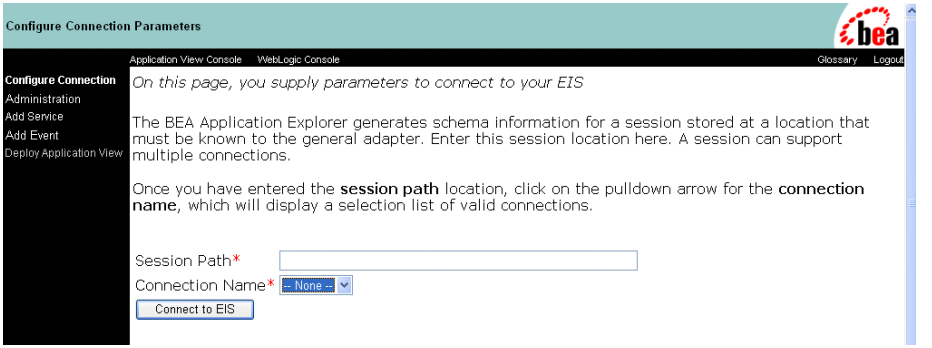
Figure 3-4 Define New Application View Window with Information Completed



7. Click OK.

   The Configure Connection Parameters window opens.

Figure 3-5 Configure Connection Parameters Window



8. In the Configure Connection Parameters window, efine the location of the schema definitions for the service or event.

   This information is necessary for the application view to interact with the target EIS.

   Enter this information only once per application view.

   The parameters required to connect to Oracle Applications are:

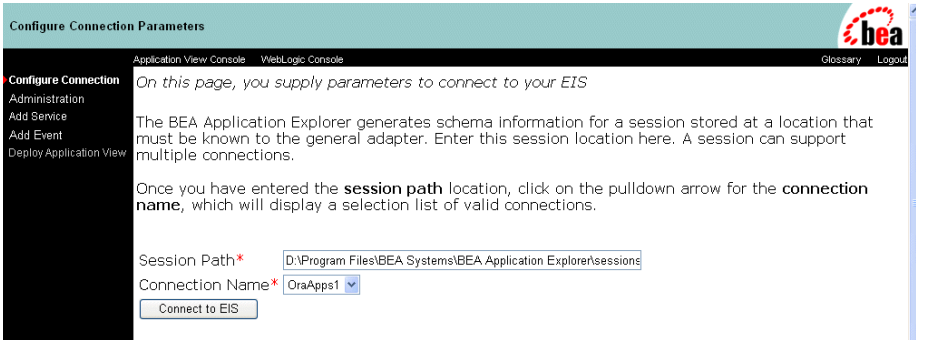
   - Session Path. The location in which the BEA Application Explorer placed the schemas and `manifest.xml` files. The default Windows location is *`install_drive:`*`\Program Files\BEA Systems\BEA Application Explorer\sessions\default`.

     The default UNIX location is */`*`install_path`*`/sessions/default`.

   - Connection name. The name of the connection used for creating schemas.

   The following figure shows a sample completed Configure Connection Parameters window:

Figure 3-6 Configure Connection Parameters Window with Information



9. Click Connect to EIS.

   The Application View Administration window opens. You can now add a service or event to the application view.

Adding a Service to an Application View

After you have satisfied the following requirements

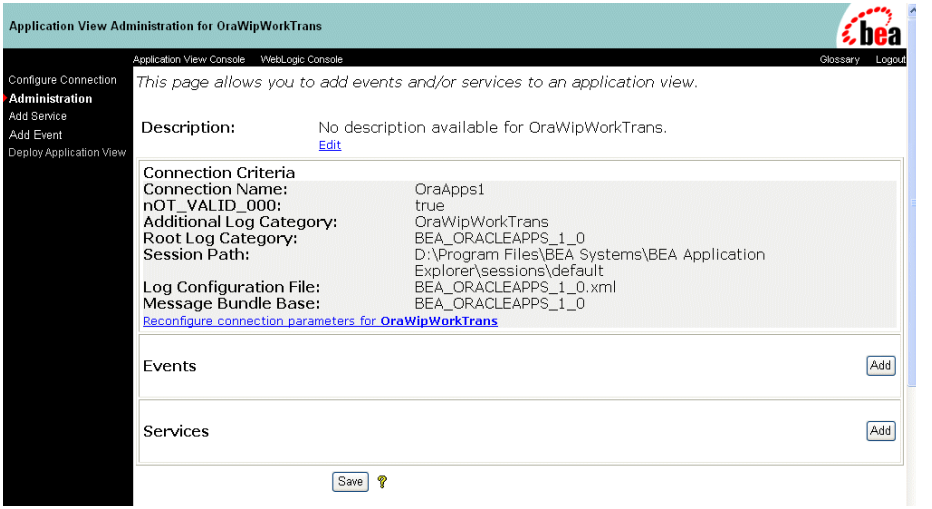
- Created a service schema using the BEA Application Explorer, as described in Creating Schemas for Services in Chapter 2, "Creating Schemas"
- Created and configured an application view, as described in "Defining an Application View" on page 3-1

you can add a service to the application view by creating and configuring a service adapter:

1. Open the Application View Console.
2. Click Administration.

   The Application View Administration window opens.

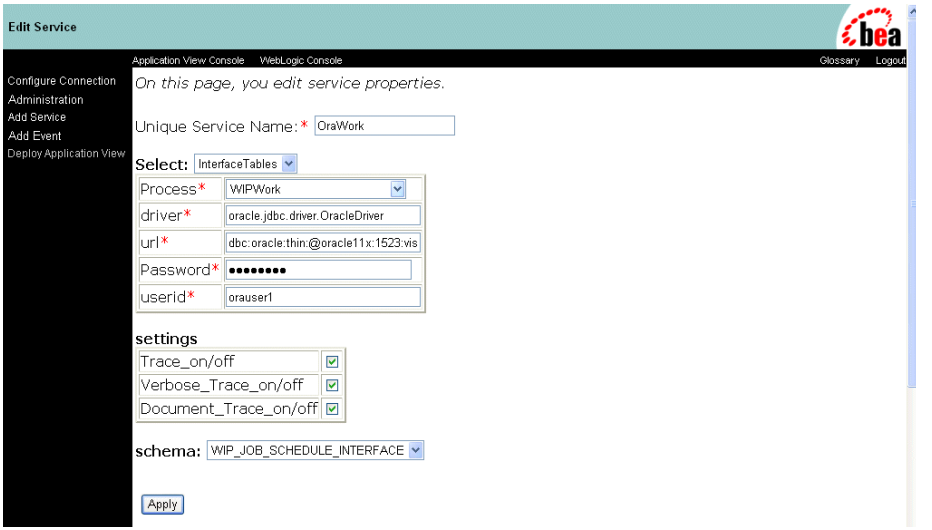
Figure 3-7 Application View Administration Window



3. Add services by clicking Add on the lower right side of the window in the Services section or by clicking Add Service from the left panel.

   The Add Service window opens.

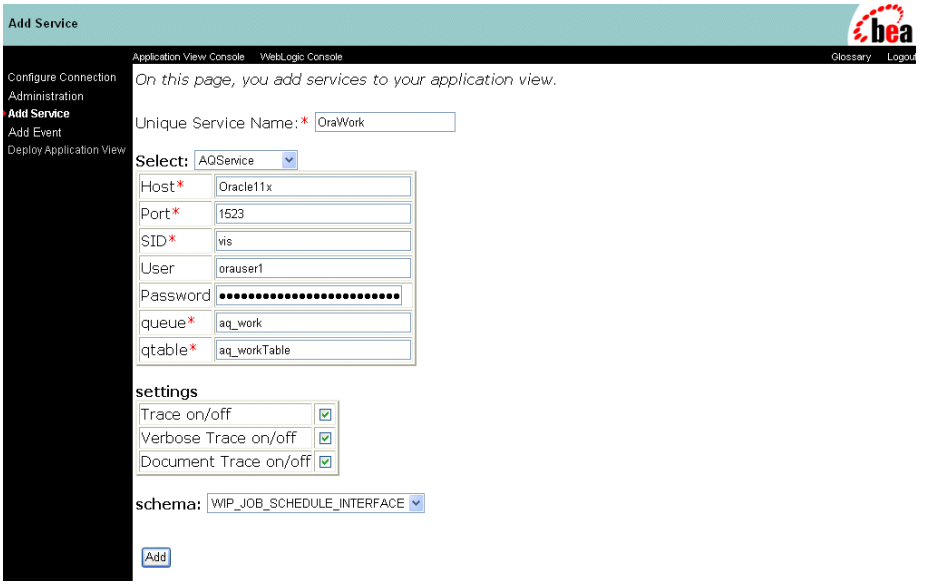
Figure 3-8 Add Service Window for Interface Tables



4. Select the type of service interaction with the Oracle Applications system from the Select drop-down list.

   Note that in the preceding figure, InterfaceTables is selected; in the following figure, AQService is selected.

Figure 3-9 Add Service Window for Oracle AQ



5. Enter the information requested, as defined in the following tables for Oracle Interface Tables and Oracle AQ:

Parameter	Definition
Unique Service Name	This name must be unique to its application view and should describe the function performed by this service. Valid characters include a-z, A-Z, 0-9, and the _ (underscore) character.
Select	Identifies the type of access being utilized by this service. Select from the following choices: **Interface Tables:** Utilizes Interface tables to pass transactional data into Oracle Applications. **AQService:** Utilizes Oracle Advanced Queuing (AQ) queues to pass data, usually XML, to Oracle Applications.

Parameter	Definition
Process	Is the name of the transaction process. For details, see "Oracle Financials Interface Tables" on page 1-4.
driver	Is the name of the Oracle JDBC driver.
url	Is the URL for the connection to Oracle.
Password	Is the password associated with the specified user ID.
userid	Is the Oracle Database ID to be used to access the Oracle Applications system.
Trace_on/off	Generates a basic trace that displays the input XML (up to 300 bytes) before parsing, and shows the request being processed. For more information about tracing, see Chapter 4, "Using Tracing."
Verbose_Trace_on/off	Generates a trace that displays configuration parameters used by the adapter. For more information about tracing, see Chapter 4, "Using Tracing."
Document_Trace_on/off	Generates a trace that displays the input document after it was analyzed and the response document being returned. For more information about tracing, see Chapter 4, "Using Tracing."
schema	From the drop-down list, select the name of the schema that defines the service you are adding.

The following table describes the service parameters for Oracle AQ.

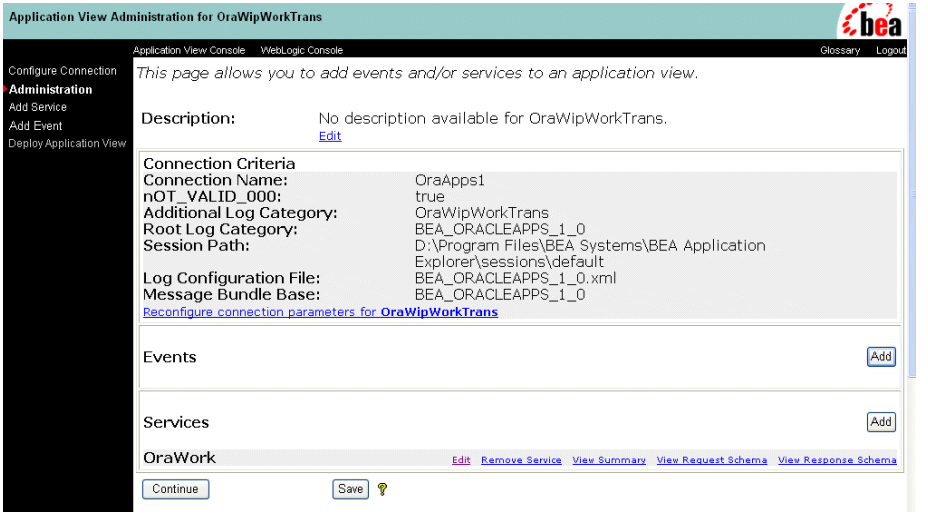
Parameter	Definition
Unique Service Name	This name must be unique to its application view and should describe the function performed by this service. Valid characters include a-z, A-Z, 0-9, and the _ (underscore) character.
Select	Identifies the type of access being utilized by this service. Select from the following choices: **Interface Tables:** Utilizes Interface tables to pass transactional data into Oracle Applications. **AQService:** Utilizes Oracle Advanced Queuing (AQ) queues to pass data, usually XML, to Oracle Applications.
Host	Is the host machine name, or IP address, for the Oracle Applications system.
Port	Is the connection port for accessing the Oracle Applications database.
SID	Is the System ID for the Oracle Applications database.
User	Is the Oracle Database ID to be used to access the Oracle Applications system.
Password	Is the password associated with the specified user ID.
queue	Is the Oracle Advanced Queuing queue name.
qtable	Is the table name on which the specified AQ queue resides.
Trace on/off	Generates a basic trace that displays the input XML (up to 300 bytes) before parsing, and shows the request being processed. For more information about tracing, see Chapter 4, "Using Tracing."
Verbose Trace on/off	Generates a trace that displays configuration parameters used by the adapter. For more information about tracing, see Chapter 4, "Using Tracing."
Document Trace on/off	Generates a trace that displays the input document after it was analyzed and the response document being returned. For more information about tracing, see Chapter 4, "Using Tracing."

Parameter	Definition
schema	From the drop-down list, select the name of the schema that defines the service you are adding.

6. After entering the information, click Add.

   When the service is added, it appears in the Services section of the Application View Summary window, as shown in the following figure.

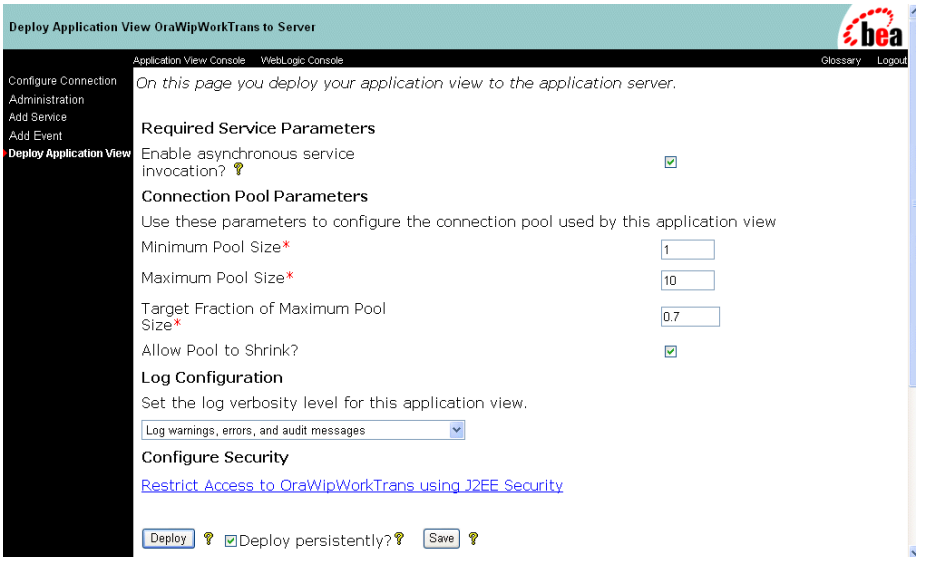
Figure 3-10 Application View Summary Window



7. Click Continue to deploy the application view.

   The Deploy Application View window opens.

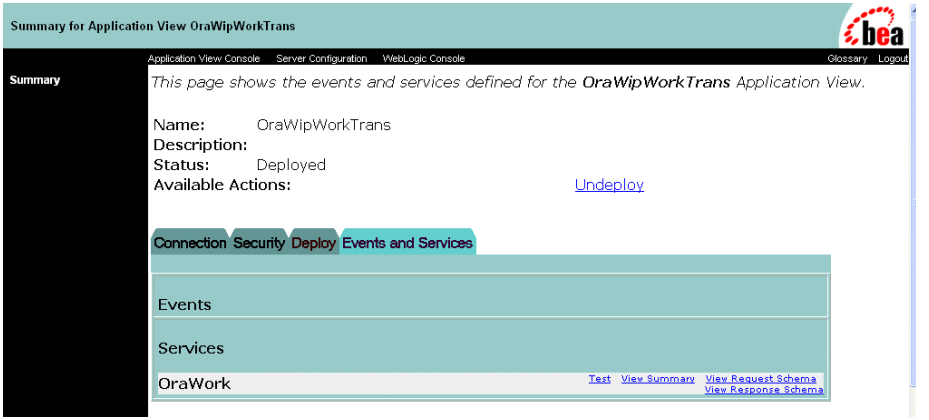
Figure 3-11 Deploy Application View Window for a Service



8. Click Deploy.

   The Application View Summary window opens.

Figure 3-12 Application View Summary Window for a Service



From this window you can test the service, as described in "Testing a Deployed Service" on page 21.

Adding an Event to an Application View

The BEA WebLogic Adapter for Oracle Applications events support the polling of relational tables (OracleAppsTables) and Advanced Queuing (OracleAQ) queues for incoming data inserted by any process, trigger, or method in Oracle Applications.

A table listener captures the incoming data from the specified tables and performs operations based on the contents of the rows. An AQ event listener retrieves the contents of the AQ queue and passes it as the event document.

The Oracle Applications event adapter, when triggered, reads one or more rows, or messages, from the table, or queue, and creates an XML document representing the data in each. Standard business logic facilities are then applied to the constructed XML documents, which include transformation, validation, security management, and application processing.

Note: The Oracle Applications event adapter requires the JDBC driver for the database being monitored. Please contact your DBMS vendor to obtain the appropriate JDBC driver.

Before you can add an event to an application view, you must have first:

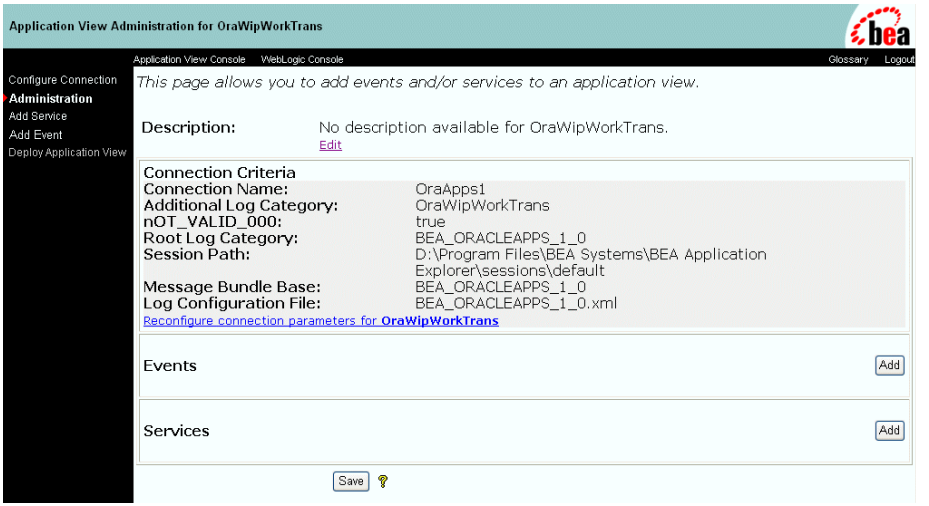
- Created an event schema using the BEA Application Explorer, as described in Creating Schemas for Events in Chapter 2, "Creating Schemas."
- Created and configured an application view, as described in "Defining an Application View" on page 3-1.

To add an event to an application view:

1. Open the Application View Console.
2. Click Administration.

   The Application View Administration window opens.

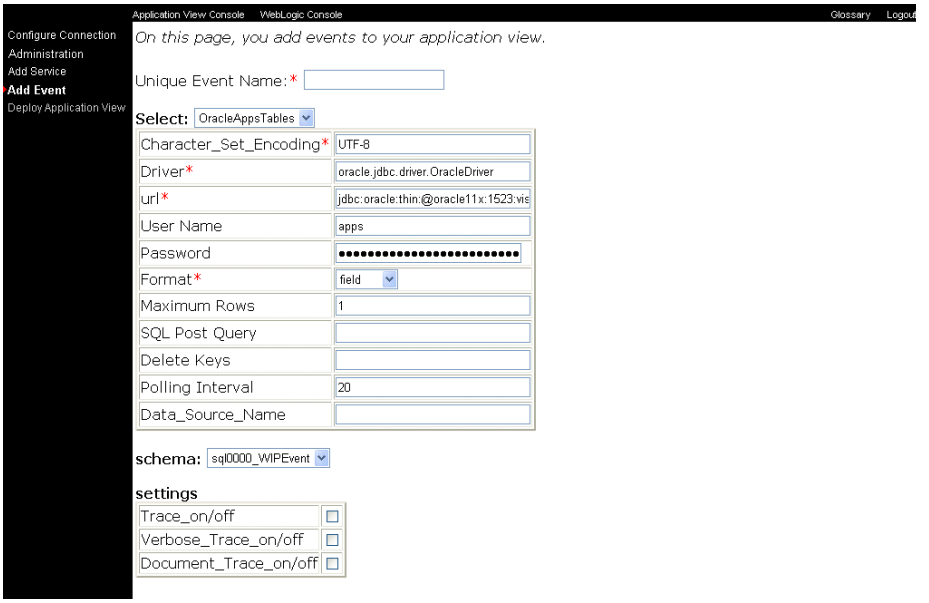
Figure 3-13 Application View Administration Window for an Event



3. Click Add in the Events row.

   The Add Event window opens.

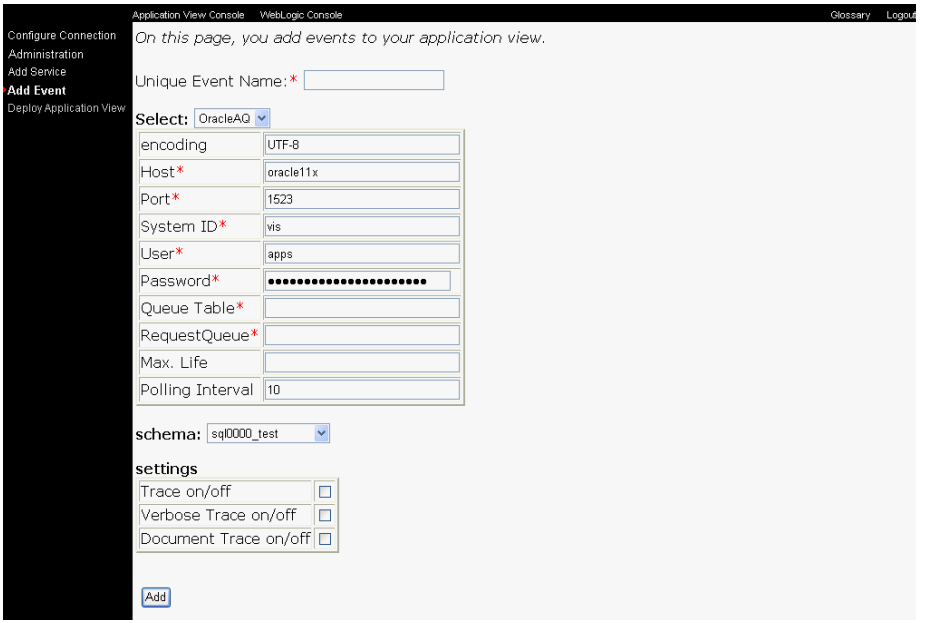
Figure 3-14 Add Event Window for Oracle Application Tables



4. Select the type of service interaction with the Oracle Applications system from the Select drop-down list.

   Note that in the preceding figure, OracleAppsTables is selected; in the following figure, OracleAQ is selected.

Figure 3-15 Add Event Window for Oracle AQ



5. Enter the information requested, as defined in the following tables for Oracle Applications Tables and Oracle AQ.

   Note: For Oracle Applications Tables, be sure to enter a value for the SQL Post Query parameter, unless you want the event adapter to delete the records that it reads. For more information, see the description of SQL Post Query in the following table.

Parameter	Description
Unique Event Name	This name must be unique to its application view. Valid characters include a-z, A-Z, 0-9, and the _ (underscore) character.
Character_Set_Encoding	The character encoding system to use. It defaults to UTF-8.

Parameter	Description
Driver	Is the JDBC Oracle Driver (`oracle.jdbc.driver.OracleDriver`).
url	Is the URL for the Oracle database and service. For example: `jdbc:oracle:thin:@Oracle11i.ibi.com:1521vis`
User Name	Is the Oracle Database ID to be used to access the Oracle Applications system.
Password	Is the password used with the user name to access the Oracle Application.

Format	Choose one of the following:
	■ **column.** The data produced is returned field by field, and each field is enclosed in <column> tags. The column tag has an attribute whose value is the name of the field; for example,
	`<row>`
	`<column name="ID">1000</column>`
	`<column name="First_Name">Scott</column>`
	`</row>`
	■ **field.** The data produced is returned field by field, and each field is enclosed in a tag that bears the field name; for example,
	`<row>`
	`<ID>1000></column>`
	`<FIRST_NAME>Scott</column>`
	`</row>`

Maximum Rows	Number of data rows to be retrieved from the table in a single operation. For example, if five were specified, then up to five rows are read and processed in a single operation. In most circumstances, you should not allow this parameter to exceed the number of parallel threads available for execution. The number of events created in a single polling interval is dependent on the setting for Maximum Rows, and the number of new rows added to the table since the last time the table was polled. For example, if Maximum Rows is set to 5, and 23 new rows are found to have been added when the table is polled, four events containing five rows, and one event containing three rows are created.

Parameter	Description

SQL Post Query	One or more SQL statements that are executed after each new record has been read from the table. If you specify more than one statement, terminate each with a semicolon (;).
	If you do not specify a value for SQL Post Query, it defaults to the following statement, which erases all the table's data once the table has been read:
	`DELETE field1,field2, ... FROM table`
	You can choose to retain the table's data once it has been read by specifying a value for this parameter, as shown in the examples that follow.
	If the SQL SELECT statement that you specified for this event (when you created the event's schema in the BEA Application Explorer) included a date column or long column, then you must provide a value for either the SQL Post Query or Delete Keys parameter, and the value you provide must not contain a date column or long column.
	Note that the SQL Post Query and Delete Keys parameters are mutually exclusive, as Delete Keys applies to the default DELETE statement, and SQL Post Query overrides the default DELETE statement. Provide a value for one or the other, but not for both.
	There are two field operators, `?` and `^`, that you can use in a post-query SQL statement:
	■ `?field` is evaluated at run time as `field = value`
	The `?` operator is useful in UPDATE statements:
	`UPDATE table WHERE ?field`
	For example, the following statement
	`UPDATE Stock_Prices_Temp WHERE ?RIC`
	might be evaluated at run time as:
	`UPDATE Stock_Prices_Temp WHERE RIC = 'PG'`
	■ `^fieldname` is evaluated at run time as `value`
	The ^ operator is useful in INSERT statements:
	`INSERT INTO table VALUES (^field1, ^field2, ^field3, ... )`
	For example, the following statement
	`INSERT INTO Stock_Prices_Temp VALUES (^RIC, ^Price, ^Updated)`
	might be evaluated at run time as:
	`INSERT INTO Stock_Prices_Temp VALUES ('PG', 88.62, '2003-03-18 16:24:00.0')`

Parameter	Description
Delete Keys	A comma-separated list of keys used in the default DELETE statement. DELETE operates on keys, so specify the table's key columns. If the SQL SELECT statement that you specified for this event (when you created the event's schema in the BEA Application Explorer) included a date column or long column, then you must provide a value for either the Delete Keys or SQL Post Query parameters, and the value you provide must not contain a date column or long column. Note that the Delete Keys and SQL Post Query parameters are mutually exclusive, as Delete Keys applies to the default DELETE statement, and SQL Post Query overrides the default DELETE statement. Provide a value for one or the other, but not for both.
Polling Interval	Interval in seconds at which the table is monitored for new rows. If this parameter is not configured, the default value is two seconds.
Data_Source_Name	The Data Source JNDI name for the JDBC connection pool to use for connecting to the RDBMS system. If a value is present, the adapter will use the connection pool to connect to the RDBMS. If no value is specified, connection will use the Driver, URL, UserId and Password specified in the service. For more information about setting up a data source, see the *BEA WebLogic Adapter for RDBMS User Guide.*
Schema	From the drop-down list, select the schema that defines the event you are adding. An event's schema name is the name that was specified for the corresponding SQL statement in the BEA Application Explorer, prefixed with "`sql0000_`". For example, if in the BEA Application Explorer you had specified the name `TEST` for an SQL event schema, the Schema drop-down list will display the schema name `sql0000_TEST`.
Trace on/off	Generates a basic trace that displays the input XML (up to 300 bytes) before parsing, and shows the request being processed. For more information about tracing, see Chapter 4, "Using Tracing."
Verbose Trace on/off	Generates a trace that displays configuration parameters used by the adapter. For more information about tracing, see Chapter 4, "Using Tracing."
Document Trace on/off	Generates a trace that displays the input document after it was analyzed and the response document being returned. For more information about tracing, see Chapter 4, "Using Tracing."

The following table describes the event parameters for Oracle AQ.

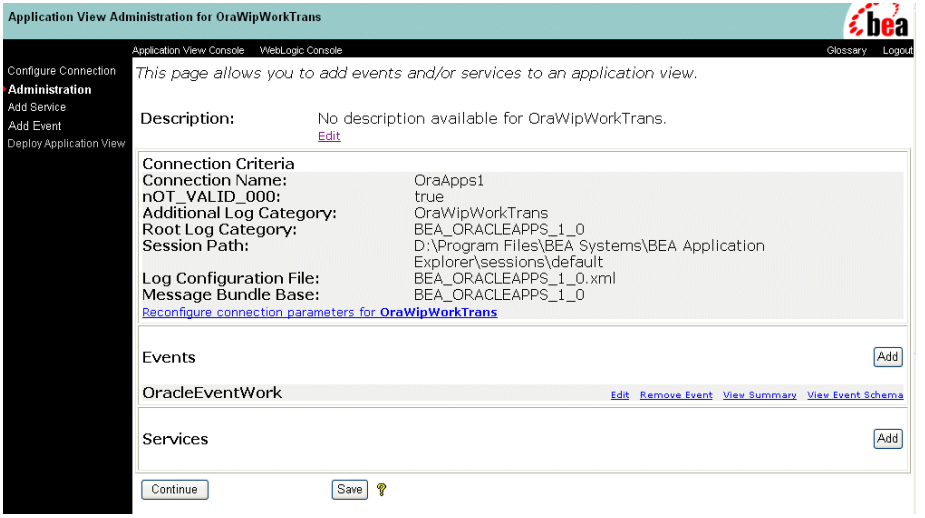
Parameter	Description
Unique Event Name	This name must be unique to its application view. Valid characters include a-z, A-Z, 0-9, and the _ (underscore) character.
encoding	The character encoding system to use. It defaults to UTF-8.
Host	Is the host machine name, or IP address, for the Oracle Applications system.
Port	Is the connection port for accessing the Oracle Applications database.
System ID	Is the System ID for the Oracle Applications database.
User	Is the Oracle Database ID to be used to access the Oracle Applications system.
Password	Is the password used with the user name to access the Oracle Application.
Queue Table	Is the table name the AQ queue specified resides on.
RequestQueue	Is the Oracle Advanced Queuing queue name.
Max. Life	The maximum time that a request may take to complete. This is used to prevent runaway requests: any request taking longer than this time period to complete will be terminated.
Polling Interval	Interval in seconds at which the table is monitored for new rows. If this parameter is not configured, the default value is two seconds.
Schema	From the drop-down list, select the name of the schema (for example, Oracle) that contains connection and other related information about the service you are adding. An event's schema name is the name that was specified for the corresponding SQL statement in the BEA Application Explorer, prefixed with "`sql0000_`". For example, if in the BEA Application Explorer you had specified the name `TEST` for an SQL event schema, then the Schema drop-down list will display the schema name `sql0000_TEST`.
Trace on/off	Generates a basic trace that displays the input XML (up to 300 bytes) before parsing, and shows the request being processed. For more information about tracing, see Chapter 4, "Using Tracing."
Verbose Trace on/off	Generates a trace that displays configuration parameters used by the adapter. For more information about tracing, see Chapter 4, "Using Tracing."

Parameter	Description
Document Trace on/off	Generates a trace that displays the input document after it was analyzed and the response document being returned. For more information about tracing, see Chapter 4, "Using Tracing."

6. Click Add.

   The Application View Administration window opens.

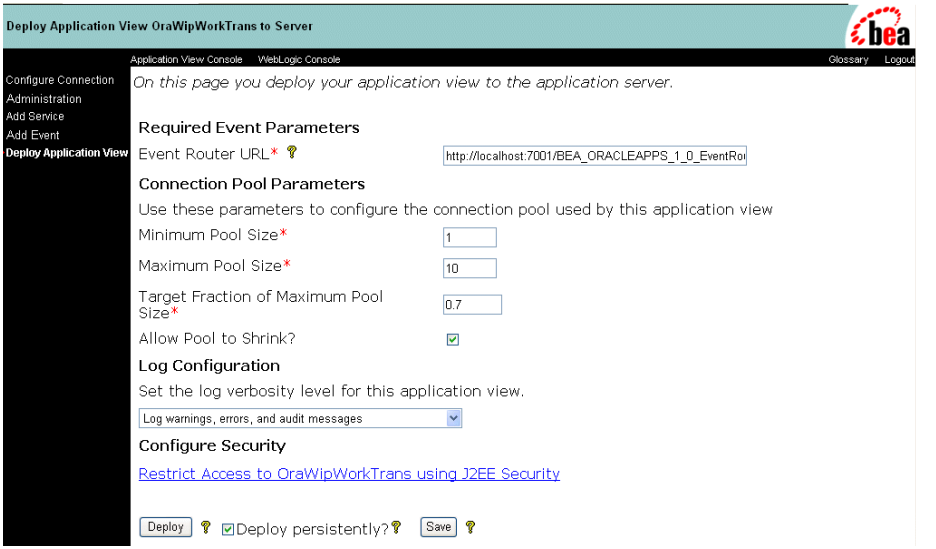
Figure 3-16 Application View Administration - Oracle Applications Event Adapter



7. Click Continue.

   The Deploy Application View window opens.

Figure 3-17 Deploy Application View Oracle Events to Servers



8. Click Deploy to deploy the application view.

Testing a Deployed Service

Test the application view service to evaluate whether or not it interacts properly with the adapter.

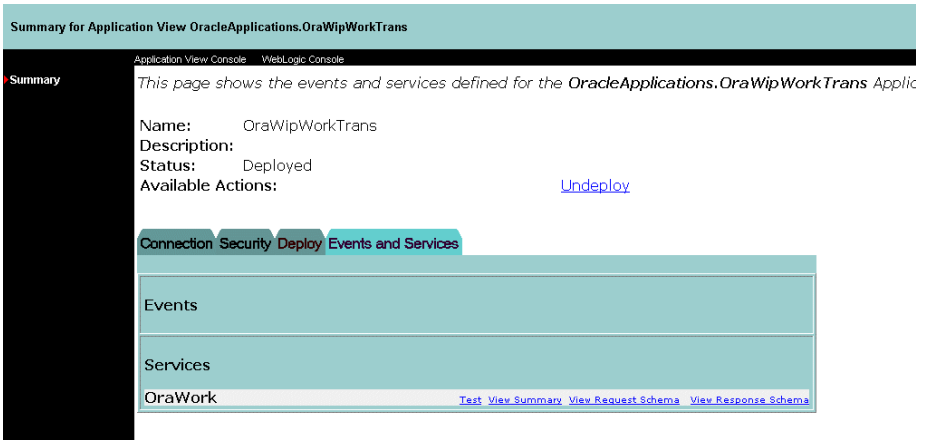
Note: Test an application view after you create and deploy an application view that contains a service.

1. Open the Application View Summary window.

   For information on opening this window, see "Adding a Service to an Application View" on page 3-6.

   The following figure shows a defined service for the application view.

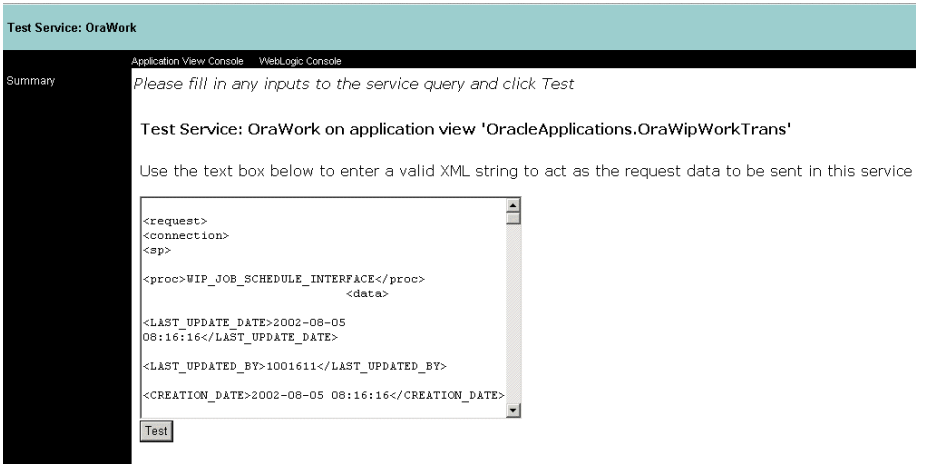
Figure 3-18 Summary for Application View Window - Testing Application View Services



2. In the Services area, locate the service and click Test.

   The Test Service page opens.

Figure 3-19 Testing Service Window - Entering XML



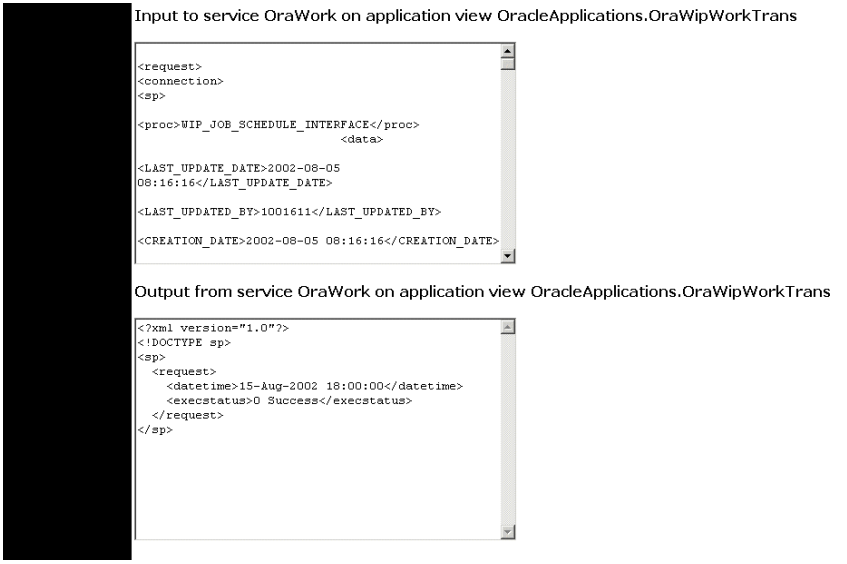
3. Enter the XML that invokes the service request.

   Note: This example is for a service running against Oracle Applications Interface Tables.

4. Click Test to test the service.

The request and response documents appear, as shown in the following figure.

Figure 3-20 Test Results Window



The following code listing is a sample of XML that can be submitted for the Wip Work Transaction. In the example, a WIP Work Order is being created in Oracle Applications.

Listing 3-1 Sample XML for Wip Work Transaction

```
<request>
<connection>
<sp>
   <proc>WIP_JOB_SCHEDULE_INTERFACE</proc>
   <data>
      <LAST_UPDATE_DATE>2002-08-05 08:16:16</LAST_UPDATE_DATE>
      <LAST_UPDATED_BY>1001611</LAST_UPDATED_BY>
      <CREATION_DATE>2002-08-05 08:16:16</CREATION_DATE>
      <CREATED_BY>1001611</CREATED_BY>
      <LAST_UPDATE_LOGIN>NULL</LAST_UPDATE_LOGIN>
      <REQUEST_ID></REQUEST_ID>
      <PROGRAM_ID></PROGRAM_ID>
      <PROGRAM_APPLICATION_ID></PROGRAM_APPLICATION_ID>
      <PROGRAM_UPDATE_DATE></PROGRAM_UPDATE_DATE>
      <GROUP_ID>WIP_JOB_SCHEDULE_INTERFACE_S.NEXTVAL</GROUP_ID>
```

```
<SOURCE_CODE></SOURCE_CODE>
<SOURCE_LINE_ID></SOURCE_LINE_ID>
<PROCESS_TYPE></PROCESS_TYPE>
<ORGANIZATION_ID>NULL</ORGANIZATION_ID>
<LOAD_TYPE>1</LOAD_TYPE>
<STATUS_TYPE>3</STATUS_TYPE>
<OLD_STATUS_TYPE></OLD_STATUS_TYPE>
<LAST_UNIT_COMPLETION_DATE></LAST_UNIT_COMPLETION_DATE>
<OLD_COMPLETION_DATE></OLD_COMPLETION_DATE>
<PROCESSING_WORK_DAYS></PROCESSING_WORK_DAYS>
<DAILY_PRODUCTION_RATE></DAILY_PRODUCTION_RATE>
<LINE_ID></LINE_ID>
<PRIMARY_ITEM_ID>155</PRIMARY_ITEM_ID>
<BOM_REFERENCE_ID></BOM_REFERENCE_ID>
<ROUTING_REFERENCE_ID></ROUTING_REFERENCE_ID>
<BOM_REVISION_DATE></BOM_REVISION_DATE>
<ROUTING_REVISION_DATE></ROUTING_REVISION_DATE>
<WIP_SUPPLY_TYPE>7</WIP_SUPPLY_TYPE>
<CLASS_CODE>Discrete</CLASS_CODE>
<LOT_NUMBER></LOT_NUMBER>
<LOT_CONTROL_CODE></LOT_CONTROL_CODE>
<JOB_NAME></JOB_NAME>
<DESCRIPTION></DESCRIPTION>
<FIRM_PLANNED_FLAG></FIRM_PLANNED_FLAG>
<ALTERNATE_ROUTING_DESIGNATOR></ALTERNATE_ROUTING_DESIGNATOR>
<ALTERNATE_BOM_DESIGNATOR> </ALTERNATE_BOM_DESIGNATOR>
<DEMAND_CLASS></DEMAND_CLASS>
<START_QUANTITY>100</START_QUANTITY>
<OLD_START_QUANTITY></OLD_START_QUANTITY>
<WIP_ENTITY_ID></WIP_ENTITY_ID>
<REPETITIVE_SCHEDULE_ID></REPETITIVE_SCHEDULE_ID>
<ERROR></ERROR>
<PARENT_GROUP_ID></PARENT_GROUP_ID>
<ATTRIBUTE_CATEGORY></ATTRIBUTE_CATEGORY>
<ATTRIBUTE1></ATTRIBUTE1>
<ATTRIBUTE2></ATTRIBUTE2>
<ATTRIBUTE3></ATTRIBUTE3>
<ATTRIBUTE4></ATTRIBUTE4>
<ATTRIBUTE5></ATTRIBUTE5>
<ATTRIBUTE6></ATTRIBUTE6>
<ATTRIBUTE7></ATTRIBUTE7>
<ATTRIBUTE8></ATTRIBUTE8>
<ATTRIBUTE9></ATTRIBUTE9>
<ATTRIBUTE10></ATTRIBUTE10>
<ATTRIBUTE11></ATTRIBUTE11>
<ATTRIBUTE12></ATTRIBUTE12>
<ATTRIBUTE13></ATTRIBUTE13>
<ATTRIBUTE14></ATTRIBUTE14>
<ATTRIBUTE15></ATTRIBUTE15>
```

```
      <INTERFACE_ID></INTERFACE_ID>
      <LAST_UPDATED_BY_NAME></LAST_UPDATED_BY_NAME>
      <CREATED_BY_NAME></CREATED_BY_NAME>
      <PROCESS_PHASE>2</PROCESS_PHASE>
      <PROCESS_STATUS>1</PROCESS_STATUS>
      <ORGANIZATION_CODE>M1</ORGANIZATION_CODE>
      <FIRST_UNIT_START_DATE>2002-08-05</FIRST_UNIT_START_DATE>
      <FIRST_UNIT_COMPLETION_DATE>2002-08-05</FIRST_UNIT_COMPLETION_DATE>
      <LAST_UNIT_START_DATE>2002-08-05</LAST_UNIT_START_DATE>
      <SCHEDULING_METHOD></SCHEDULING_METHOD>
      <LINE_CODE></LINE_CODE>
      <PRIMARY_ITEM_SEGMENTS></PRIMARY_ITEM_SEGMENTS>
      <BOM_REFERENCE_SEGMENTS></BOM_REFERENCE_SEGMENTS>
      <ROUTING_REFERENCE_SEGMENTS></ROUTING_REFERENCE_SEGMENTS>
      <ROUTING_REVISION></ROUTING_REVISION>
      <BOM_REVISION></BOM_REVISION>
      <COMPLETION_SUBINVENTORY ></COMPLETION_SUBINVENTORY>
      <COMPLETION_LOCATOR_ID></COMPLETION_LOCATOR_ID>
      <COMPLETION_LOCATOR_SEGMENTS></COMPLETION_LOCATOR_SEGMENTS>
      <SCHEDULE_GROUP_ID></SCHEDULE_GROUP_ID>
      <SCHEDULE_GROUP_NAME></SCHEDULE_GROUP_NAME>
      <BUILD_SEQUENCE ></BUILD_SEQUENCE >
      <PROJECT_ID></PROJECT_ID>
      <PROJECT_NAME></PROJECT_NAME>
      <TASK_ID></TASK_ID>
      <TASK_NAME></TASK_NAME>
      <NET_QUANTITY>100</NET_QUANTITY>
      <DESCRIPTIVE_FLEX_SEGMENTS></DESCRIPTIVE_FLEX_SEGMENTS>
      <PROJECT_NUMBER></PROJECT_NUMBER>
      <TASK_NUMBER></TASK_NUMBER>
      <PROJECT_COSTED></PROJECT_COSTED>
      <END_ITEM_UNIT_NUMBER></END_ITEM_UNIT_NUMBER>
      <OVERCOMPLETION_TOLERANCE_TYPE></OVERCOMPLETION_TOLERANCE_TYPE>
      <OVERCOMPLETION_TOLERANCE_VALUE></OVERCOMPLETION_TOLERANCE_VALUE>
      <KANBAN_CARD_ID></KANBAN_CARD_ID>
      <PRIORITY>2</PRIORITY>
      <DUE_DATE>2002-08-05</DUE_DATE>
      <ALLOW_EXPLOSION>Y</ALLOW_EXPLOSION>
      <HEADER_ID></HEADER_ID>
      <DELIVERY_ID></DELIVERY_ID>
      <COPRODUCTS_SUPPLY></COPRODUCTS_SUPPLY>
      <DUE_DATE_PENALTY></DUE_DATE_PENALTY>
      <DUE_DATE_TOLERANCE></DUE_DATE_TOLERANCE>
   </data>
</sp>
</connection>
</request>
```

Note that in this example the Oracle Applications Sequence stored for the Group ID is used, as seen in the following statement:

```
<GROUP_ID>WIP_JOB_SCHEDULE _INTERFACE_S.NEXTVAL</GROUP_ID>
```

While not every field is mandatory, the Oracle documentation should be consulted for a list of required fields as well as basic field definitions.

The BEA Application Explorer can be used to generate basic templates for transactions. These templates should be used as a guide when constructing XML to be used for Oracle transactions.

The adapter returns any post-validated XML showing errors and warning. After errors are generated, you have up to two options for fixing rejected records. The Oracle Application System graphical user interface may be used to delete records that do not pass validation; XML may be resubmitted through the Oracle Applications Adapter. The second option is to fix the record within the Oracle System graphical user interface and, if required, initiate the concurrent process. The concurrent processes cause the records to be validated and moved from interface tables to base tables.

If the test fails, a response document displays the appropriate error messages. Correct the error and resubmit the request.

If the test does not fail, the application view service is successfully deployed and tested. You can write custom code to exploit the adapter or create a workflow using WebLogic Integration Studio.

For more information, see "Using Application Views in the Studio" in *Using Application Integration*:

- For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/3usruse.htm`
- For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/3usruse.htm`

Testing a Deployed Event

Test the application view event to evaluate whether or not it interacts properly with the adapter.

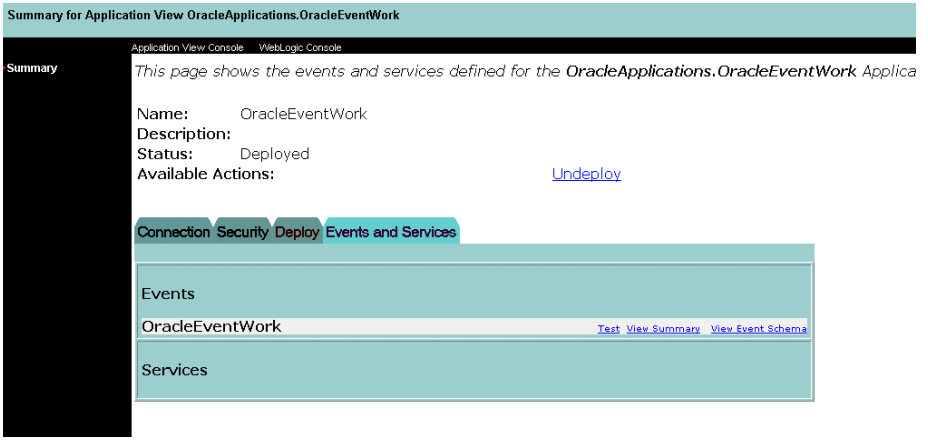
Note: Test an application view after you create and deploy an application view that contains a event.

1. Open the Application View Summary window.

   For information on opening this window, see "Adding an Event to an Application View" on page 3-13.

   The following figure shows a defined event for the application view.

Figure 3-21 Summary for Application View Window - Testing Application View Event

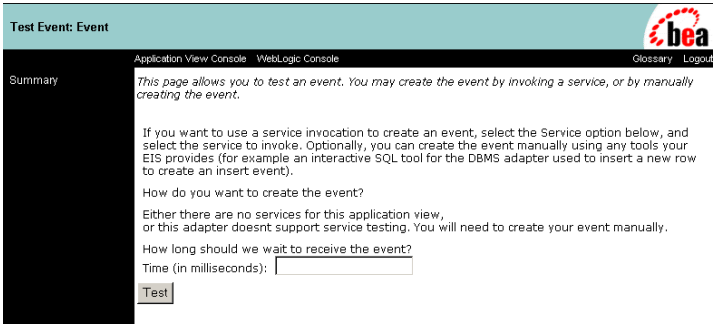


2. In the Events area, locate the event and click Test.

   The Test Event page opens.

   Note: This example is for an event running against Oracle Applications Tables.

Figure 3-22 Test Event Window

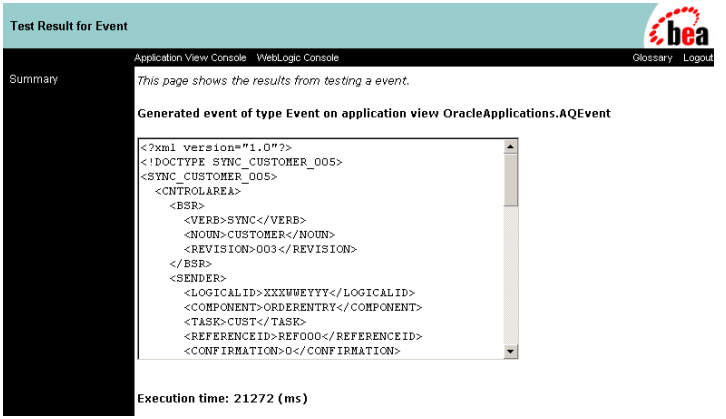


3. To initiate the event for the Wip_Work Order, enter a time in milliseconds.
4. Click Test.

   The test waits for an incoming event and then creates a work order from within the Oracle Applications system. For more information, see the *Oracle Applications Work in Progress* guide.

   When the event occurs, the event test result document appears, as shown in the following figure.

Figure 3-23 Event Test Results Window

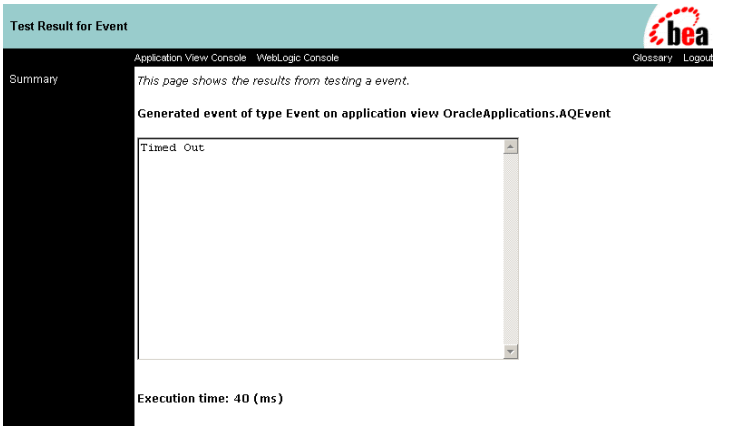


The application view event has now been successfully deployed and tested. You can write custom code to exploit the adapter or create a workflow using Studio. For more information, see “Using Application Views in Business Process Management” at the following URL:

`http://e-docs.bea.com/wlintegration/v2_1/aiuser/3usruse.htm`

Note: If the event is not received in time, a Timed Out message is returned, as shown in the following figure.

Figure 3-24 Timed Out Message



Using a Service or Event in a Workflow

This section contains specific examples of using services and events in business process management workflows. A business process management workflow is the process that links an event in one EIS to the services in another EIS.

You can build complex workflows that use multiple events and services, as well as transformations and business logic. This section does not describe the complete functionality of the business process management. For more information, see "Using Application Views in Business Process Management" at the following address:

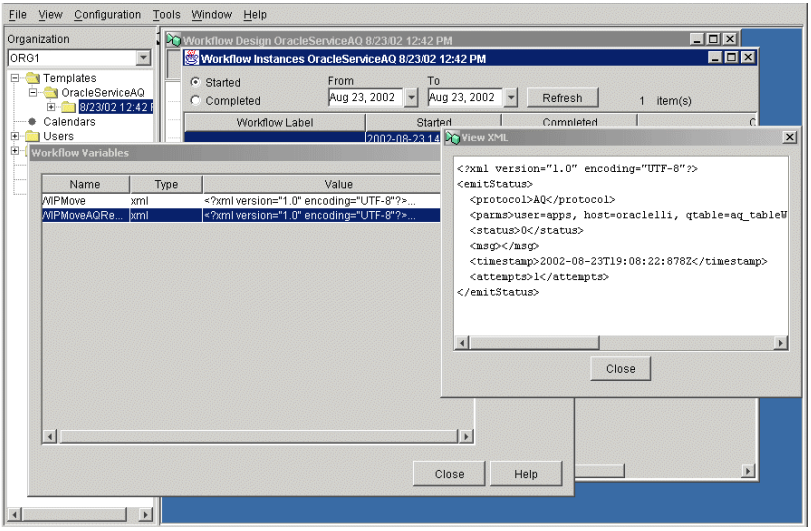
`http://edocs.bea.com/wlintegration/v2_1/aiuser/3usruse.htm`

Example: Using a Service in a Workflow

In this example, a workflow is built using a static document called WIPMove. This document is then used as input to the Oracle Applications by being placed in an Advanced Queuing (AQ) queue for processing. The resulting document is placed in the WIPMoveAQResponse. Since the connection to Oracle Applications is through an AQ queue, the response document details the interaction with AQ, not Oracle Applications.

The following figure illustrates the service request in a process workflow: In this workflow, the input document is defined as an action of the Start node. The Task1 action defines the use of the XML document passed to it from the Start node. The Task1 sends it to the Oracle Applications AQ service.

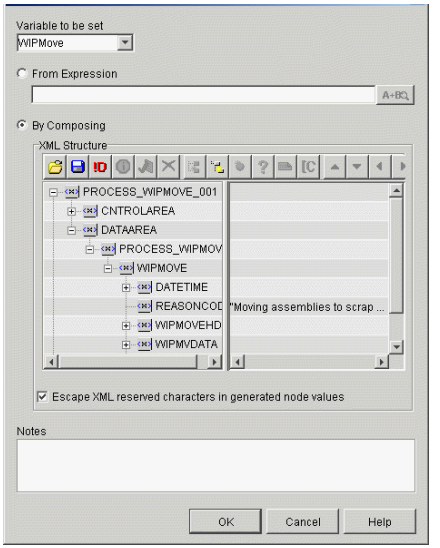
Figure 3-25 Service Request in a BPM Workflow



The Oracle Applications Request

The following figure shows the static document created for the workflow. In this case, the document is a WIPMove intended to mark a discrete job as being moved from one step in the manufacturing process to the next. In a real world workflow, the document would typically be the result of an event in another EIS, with transformations and enrichment steps performed.

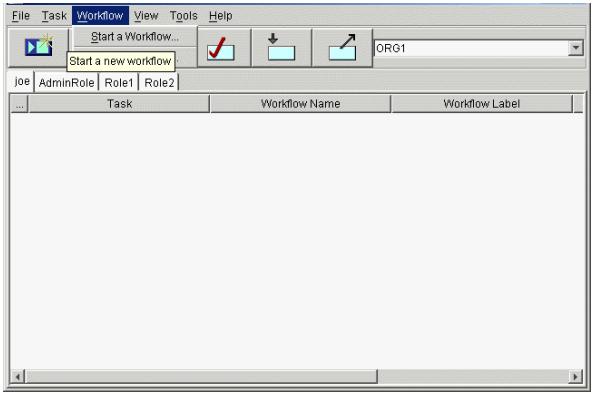
Figure 3-26 Static Workflow Document Window



Starting the Workflow

After the workflow is built, Worklist is used to start the process manually. Worklist is required because the input document is static and not the result of an event from another EIS.

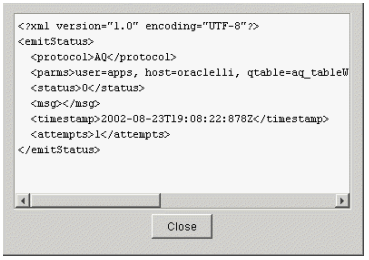
Figure 3-27 WebLogic Integration - Starting the Workflow



Oracle AQ Response

When the workflow is started, the Task1 step takes the document and sends it to the Oracle AQ queue. Upon completion of the task, the workflow places the response XML document in the WIPMoveAQResponse variable, as shown in the following figure.

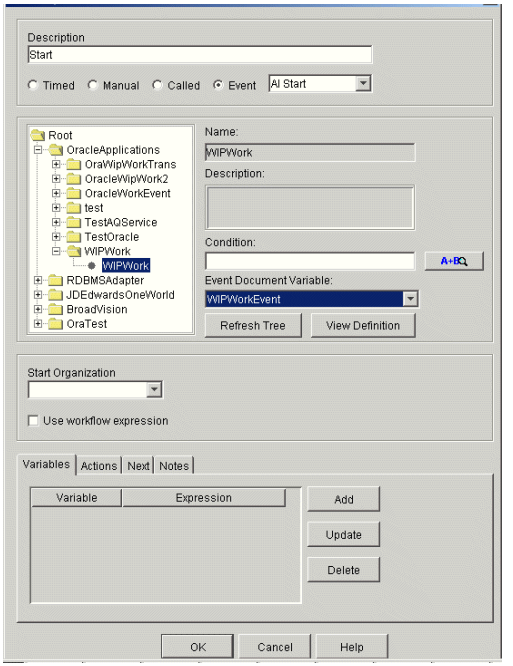
Figure 3-28 Response XML Document Window



Example: Using an Event in a Workflow

Events are used to start process workflows and are set up in a manner similar to services. In this example, the Oracle Applications Event starts the workflow.

Figure 3-29 Start Properties Dialog Box - Workflow Event Description



To use an event in a workflow:

1. Click the Event radio button in the Start Properties dialog box.
2. Select AI Start from the drop-down list.

   A list of available application views appears.
3. Navigate to the desired event; in this example, OracleApplications/WIPWork.
4. Double-click the event.

   The inbound XML event document is assigned a name; in this case it is WIPWorkEvent.

   This document is passed to later steps in the workflow.

Since this workflow is event driven, Worklist is not required to start it. Instead, the workflow is started by an event in the Oracle Applications system. Once the Oracle Applications Event occurs, the workflow captures it. The event XML is in a form that is dependent on the Oracle Applications output. The sample event document is shown in the following figure.

Figure 3-30 Sample Event Document

CHAPTER

4 Using Tracing

Tracing is an essential feature of an adapter. Most adapters integrate different applications and do not interact with end users while processing data. Unlike a front-end component, when an adapter encounters an error or a warning condition, the adapter cannot stop processing and wait for an end user to respond.

Moreover, many business applications that are connected by adapters are mission-critical. For example, an adapter might maintain an audit report of every transaction with an Enterprise Information System (EIS). Consequently, adapter components must provide both accurate logging and auditing information. The adapter tracing and logging framework is designed to accommodate both logging and auditing.

This section describes tracing for services and events. It contains the following topics:

Levels and Categories of Tracing

Tracing is provided by both the BEA adapter framework and by the BEA WebLogic Adapter for Oracle Applications. The BEA WebLogic Integration framework provides five distinct levels of tracing:

Table 4-1

Level	Indicates
AUDIT	An extremely important log message related to the business processing performed by an adapter. Messages with this priority are always written to the log.
ERROR	An error in the adapter. Error messages are internationalized and localized for the user.
WARN	A situation that is not an error, but that could cause problems in the adapter. Warning messages are internationalized and localized for the user.
INFO	An informational message that is internationalized and localized for the user.
DEBUG	A debug message, that is, information used to determine how the internals of a component are working. Debug messages usually are not internationalized.

The adapter framework provides three specialized categories of tracing:

Table 4-2

Level	Indicates
Basic Trace	Basic traces. Displays the input XML (up to 300 bytes) before parsing, and shows the request being processed. The default setting is off.
Verbose Trace	More extensive traces. Displays configuration parameters used by the adapter. The default setting is off.

Table 4-2

Level	Indicates
Document Trace	Displays the input document after it was analyzed and the response document being returned. Because some documents are very large, this trace category can severely affect performance and memory use. The default setting is off.

Note: To obtain the appropriate trace, both the level and the category must be declared. In a debug situation, BEA Customer Support will request (minimally) a Basic and a Verbose trace.

Tracing and Performance

The additional trace capabilities provided by the adapter are not strictly hierarchic; rather they are categorized. These traces are designed to provide debugging help with minimum effect on performance. All internal adapter traces are controlled through the additional tracing settings, and all additional settings route their output to the standard debug setting.

If you configure the adapter for additional settings and do not configure standard trace settings, the traces are generated but never appear in output. This affects performance, as the production of the trace continues even though you receive no benefit of the additional trace information.

Creating Traces for Services and Events

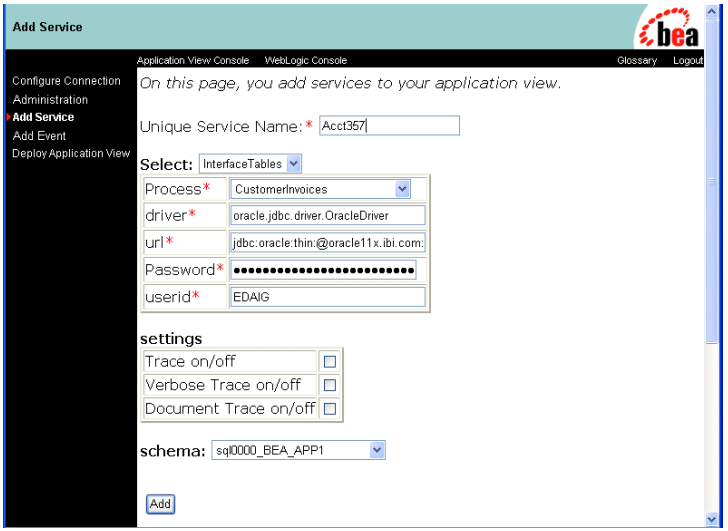
The following topics discuss the steps required to create traces to diagnose adapter problems.

Creating Traces for a Service

To create traces for a service:

1. Create or modify the service.
2. Ensure that all of the adapter parameters are entered correctly.

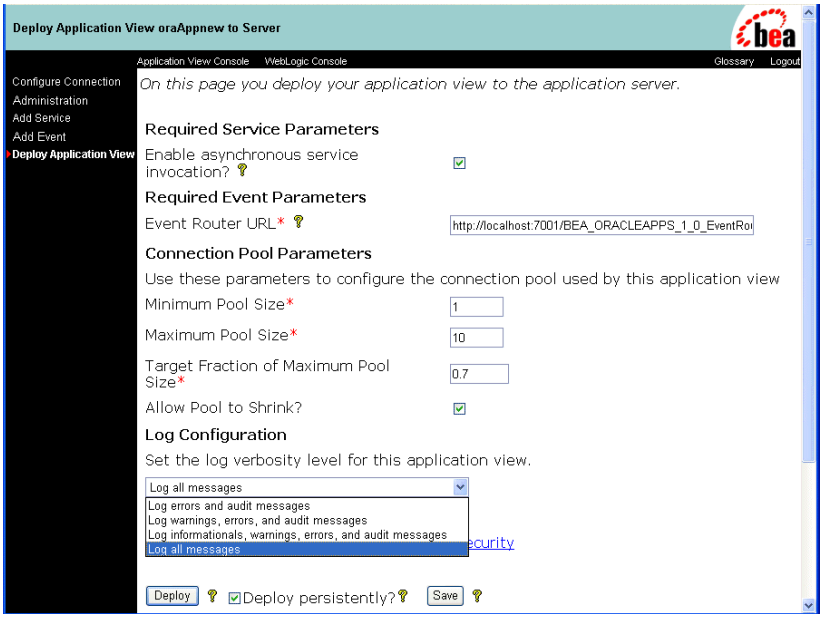
Figure 4-1 Add Service Window



3. Select the appropriate schema from the drop-down list.
4. Select the appropriate trace levels as described in Table 4-2: Trace, Verbose trace, and Document trace.
5. Click Add to continue to the next configuration pane.
6. Click Continue to move to the next configuration pane.

   The Deploy Application View window opens.
7. Navigate to the Log Configuration area and select the desired trace level.

Figure 4-2 Deploy Application View window



This pane enables you to select the trace level for the BEA WebLogic Integration framework.

For maximum tracing, select Log all Messages. This is recommended to obtain optimum debugging information for BEA support personnel.

Note: This causes all generated messages to be written to the log. You must select the desired category as defined in Table 4-2 in the adapter to generate the required messages.

8. Click Deploy (or Save) to set the trace settings and deploy the application view.

   Traces are created the next time the service is invoked.

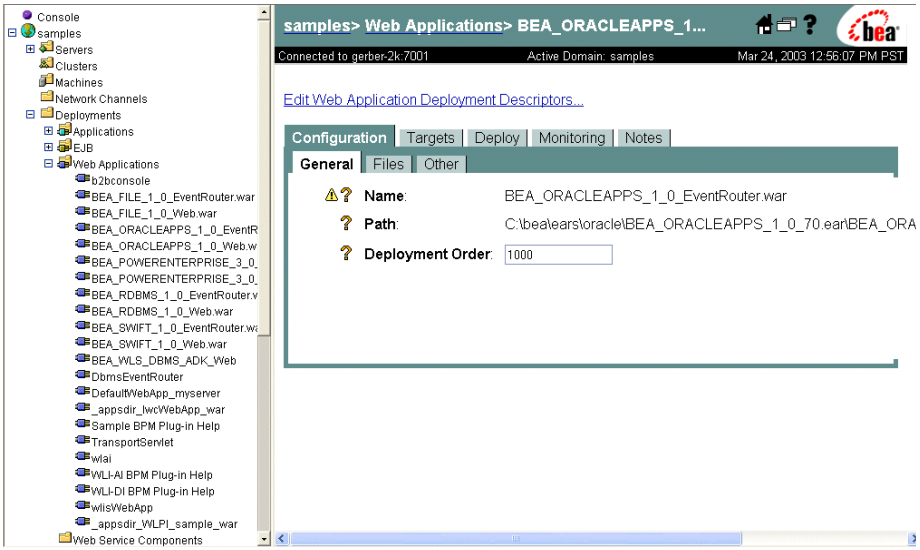
   Traces are output to a file named BEA_ORACLEAPPS_1_0.log in the WebLogic Domain home directory.

Creating or Modifying the WebLogic Framework Tracing Level for an Event

To create or modify the WebLogic framework tracing level for an event:

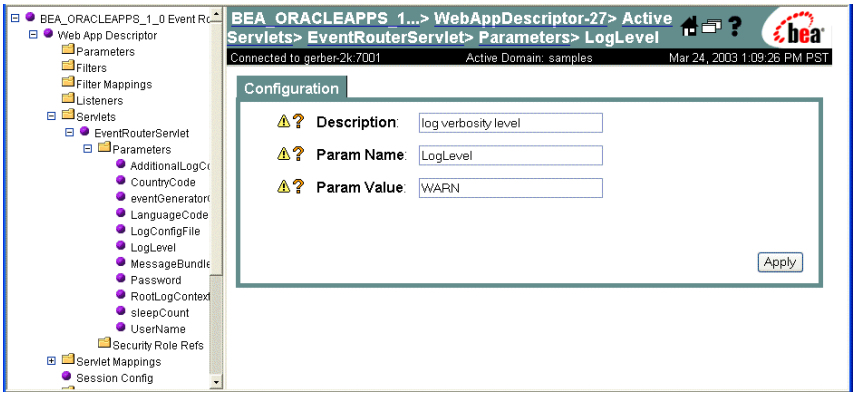
1. Logon to the BEA WebLogic Server Console.

Figure 4-3 WebLogic Server Console



2. Select Web Applications.
3. Select the EventRouter corresponding to the adapter that will be traced. For example, if you require traces for an Oracle Applications event, select BEA_ORACLEAPPS_1_0_EventRouter.war.
4. Click Edit Web Application Deployment Descriptors.
5. When the following window opens, select Servlets.
6. In the folder below Servlets, select EventRouterServlet.
7. Select Parameters.
8. Select LogLevel.

Figure 4-4 WebLogic Server Console: Configuration



This pane enables you to select the trace level for the BEA WebLogic Integration framework.

For maximum tracing, enter DEBUG. This is recommended to obtain optimum debugging information for BEA support personnel.

The following levels are valid:

Table 4-3

Level	Indicates
AUDIT	An extremely important log message related to the business processing performed by an adapter. Messages with this priority are always written to the log.
ERROR	An error in the adapter. Error messages are internationalized and localized for the user.
WARN	A situation that is not an error, but that could cause problems in the adapter. Warning messages are internationalized and localized for the user.
INFO	An informational message that is internationalized and localized for the user.
DEBUG	A debug message, that is, information used to determine how the internals of a component are working. Debug messages usually are not internationalized.

9. Click Apply to save the newly entered trace level.
10. Click BEA_ORACLEAPPS_1_0EventRouter.
11. Click Persist to apply the logging changes.

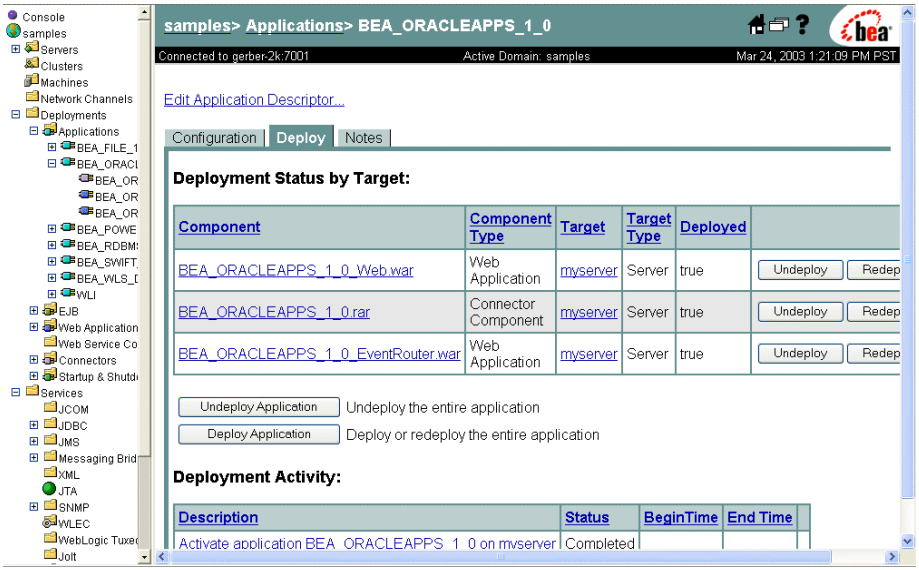
    This change need only be made once.

    It is set for all events associated with a given adapter.
12. Return to the WebLogic Server Console.
13. Select Applications from the WebLogic Server Console.
14. Select the adapter whose EventRouter you have modified in the previous steps.
15. Select the Deploy tab in the right pane.

    The right pane displays the following adapter components:

    - BEA_ORACLEAPPS_1_0.rar
    - BEA_ORACLEAPPS_1_0.web.rar
    - BEA_ORACLEAPPS_1_0_EventRouter.war.

Figure 4-5 WebLogic Server Console: Redeploy



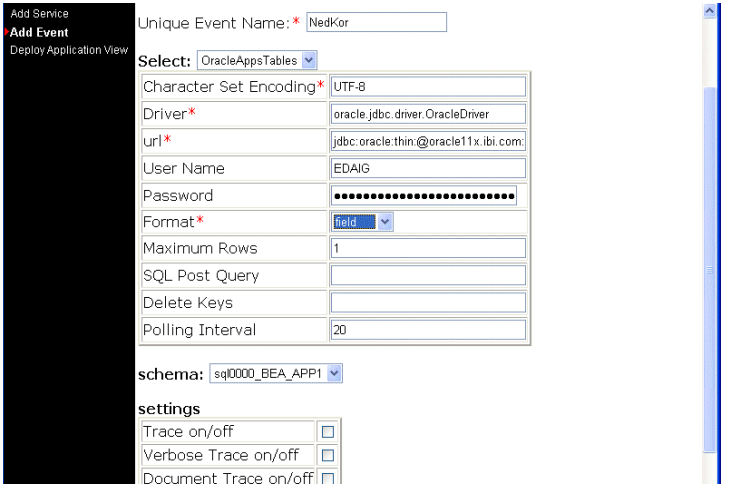
16. Redeploy the EventRouter by clicking the Redeploy button to the right of BEA_ORACLEAPPS_1_0_EventRouter.war.

Creating Adapter Logs for an Event

To create adapter logs for an event:

1. Create or modify the event.
2. Ensure that all of the adapter parameters are entered correctly.

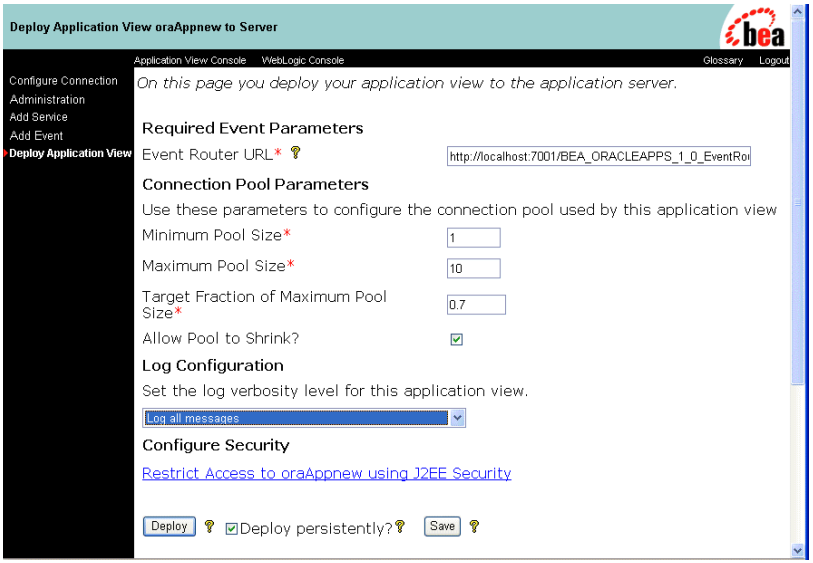
Figure 4-6 Add Event Window



3. Select the appropriate schema from the drop-down list.
4. Select the appropriate trace levels as described in Table 4-2: Trace, Verbose trace, and Document trace.
5. Click Add to continue to the next configuration pane.
6. Click Continue to move to the next configuration pane.

   The Deploy Application View window opens.

Figure 4-7 Deploy Application View Window



7. Navigate to the Log Configuration area and select the desired trace level.

   This pane enables you to select the trace level for the BEA WebLogic Integration framework.

   For maximum tracing, select Log all Messages. This is recommended to obtain optimum debugging information for BEA support personnel.

8. Click Deploy (or Save) to set the trace settings and deploy the application view.

   Traces are created the next time the event occurs.

   Traces are output to a file named BEA_ORACLEAPPS_1_0.log in the WebLogic Domain home directory.

A XML Business Functionality in the BEA WebLogic Adapter for Oracle Applications

This section describes the business functionality of the XML supported by the BEA WebLogic Adapter for Oracle Applications. It includes the following topics:

BEA WebLogic Adapter for Oracle Applications and Oracle Functionality

The BEA WebLogic Adapter for Oracle Applications supports Oracle Advanced Queuing for WebLogic Integration services and events. This provides direct bidirectional access to the Oracle Internet Procurement Connector for direct Open Application Group (OAG) formatted XML and also to the Oracle XML Gateway, which supports all DTD-based XML standards. The majority of Oracle prebuilt

messages delivered with the Oracle e-Business Suite are pre-mapped using the Open Application Group (OAG) standard. Any Oracle prebuilt message may be re-mapped to any standard of choice using the XML Gateway Message Designer.

The Oracle Internet Procurement (OIP) Connector provides for easy integration between the Oracle Internet Procurement application and external systems. The OIP Connector is based on the open, industry-standard Open Application Group Internet Standard (OAGIS) XML. For more information on the OAGIS XML, please see the Open Application Group Web site at `http://www.openapplications.org`. The standard allows OIP Connector interaction with middleware systems adhering to the OAGIS standard.

Oracle XML Gateway is a set of services that allows for easy integration with the Oracle e-Business Suite to support non-standard XML messaging. The Oracle e-Business Suite utilizes the Oracle Workflow Business Event System to publish and subscribe to application business events to automatically trigger the creation and consumption of XML messages.

Oracle XML Gateway consumes events raised by the Oracle e-Business Suite, and subscribes to inbound events for processing. Oracle XML Gateway uses the message propagation feature of Oracle Advanced Queuing to integrate with the Transport Agent to deliver or receive messages to and from business partners.

XML Business Functionality Descriptions

The following section describes the business functionality of the XML supported by the BEA WebLogic Adapter for Oracle Applications.

Post Journal

Journal is an XML business document that transmits data necessary to create a journal entry from any sub ledger business application to a general ledger application. This scenario assumes that the details of the financial transactions are kept in the sub ledgers and that the drill back mechanism from the general ledger component to the sub ledger components is addressed by this XML. Many applications in the enterprise environment create data that causes changes in the account balances of a general ledger

application (such as Benefits, Costing, Human Resources and Payroll, Inventory, Manufacturing, Production, and Treasury). By no means is this a complete list of all the activities which may generate a journal entry. Many other tasks that occur within the enterprise applications cause the creation of a General Ledger journal entry. For example, the adjustment of inventory value is a task that occurs within Inventory.

Journal supports the summarization of accounting activities from sub ledgers to the general ledger. This summary is used to express account balances, which are used to report on the financial condition of the enterprise. A Journal XML consists of the Journal Entry Header information which includes the type of journal being entered, a Journal Entry Detail Line (typically, there are at least two occurrences given the common accounting rule of every debit requiring an equally balancing credit), and an account code, which consists of a number of separate elements. Some of these elements can be linked to specified XML fields; for example, the nominal account key (GLNOMACCT) or the cost center (COSTCENTER).

Confirm

The purpose of the Confirm XML is to add a layer of software application to the software application exception capability to the integration solution. The Confirm XML does this by providing a mechanism for the business software layers to communicate to each other in addition to the confirmation mechanisms that may be provided with any middleware tools involved in the solution.

In instances when the receiving application must communicate its native informational messages to an integrated partner application, it is important to ensure that the applications have a mechanism for speaking their own language to each other.

The use of the Confirm XML is monitored using a confirmation indicator set by the sending business software application. This is accomplished by setting the Confirmation Indicator (CONFIRMATION) in the Control Area of the originating XML. The Confirmation Indicators are defined as follows:

- 0 = Do not send back a confirmation XML.
- 1 = Send back a confirmation XML only if an error has occurred.
- 2 = Send a confirmation XML regardless of whether an error has occurred.

Confirm XML:

1. Ensures that the message was received and understood within the application or component
2. Communicates error conditions at the application level (such as when a field is missing, a customer does not exist, and so on).
3. Communicates that a critical update was successful (or unsuccessful), such as an update to inventory, credit balance, or ledger balance.

Process PO

A Purchase Order is an XML business document that an organization issues to request delivery of goods or services for specific dates and locations.

The Process PO XML is used to transmit a purchase order to a supplier's order management application. The Process PO is the task of sending the electronic form of a purchase order document to a supplier. This is designed as an external purchase order.

Data such as name, currency, payment method, bill-to and ship-to addresses, and payment terms, along with information relevant to the supplier, is transmitted to the supplier to engage an order for goods and services.

Acknowledge PO

The Acknowledge PO XML is a document used to acknowledge receipt of the Purchase Order and to reflect any changes. This is designed as an external purchase order.

Commonly, the acknowledgment is generated by an order management application and transmitted to a purchasing or procurement application. Acknowledgement (when the entire document is accepted, rejected, or modified) represents feedback from the supplier concerning the original purchase order received.

Load Receivable

Receivable is an XML business document that transmits data to create a receivable open item in Oracle receivables from the billing information generated in an order management application. Receivable may also update the general ledger, depending on the specific architecture of the accounting application.

The scope of receivables is to create an XML to recognize customer obligations. Specific transactions include:

- Sales Invoice
- Credit Memo
- Debit Memo
- Charge Back

Receivable XML may also be used for transactions that do not originate from an order management application.

The receivable application may be a direct sub-ledger of the general ledger. Updates to G/L balances occur using the receivable module; therefore, the receivable XML contains both receivable and general ledger transaction information.

The other environment may also exist when general ledger updates occur directly from the Order Management application. The reconciliation between the receivable and general ledger is a function within the financial applications rather than of the integration space. This model allows the G/L balances to be updated in either detail or summarized form.

The role of the receivable application includes functions such as:

- Allowing Cash Application
- Dunning
- Dispute management

Relevant data that identifies the receivable includes header information (totals, identifier), partner location and contacts, invoice payment terms, and tax information.

Load Payable

Payables is an XML business document that transmits data from the purchasing information generated in a purchasing application to create a payable open item in a payables application. This may also update the general ledger, depending on the specific architecture of financial applications.

Payables indicates that the supplier's invoice is ready to be paid and has been approved before the information moves to the accounts payable application. An approved invoice is also known as a voucher. The application later defines invoices that are matched within the accounts payable application in a separate business service request.

Some financial applications have the general ledger and accounts payable databases tightly integrated where updates to the accounts payable application are automatically reflected in general ledger balances. Payables transmits all information needed for both the accounts payable and the general ledger. Other applications allow the general ledger balances to be updated separately from the accounts payable. In this case, the payables and journals XML accomplishes this scenario.

Sync Customer

The purpose of the Sync Customer XML is to keep customer information that exists on separate databases synchronized. The Sync Customer XML allows the adding of new customers and the modification of previously established customers. These customer records are used to reference invoices in a billing system.

Sync Supplier

The Sync Supplier XML facilitates keeping supplier information that exists on separate data bases synchronized. The Sync Supplier allows the adding of new suppliers and the modification of previously established suppliers.

This record consists of relevant data to a partner (supplier name and address, contact information associated with the partner, and references of supporting documents).

Load LdgrBudget

The LDGRBUDGET is an XML business document that transmits budget amounts from all possible source applications throughout an enterprise to a general ledger or budget application. Usually, budget data is created using a spreadsheet by each organization in an enterprise. Once budget data is ready, LDGRBUDGET XML facilitates the transfer of budget amounts against a revenue or expense account codes to a general ledger application for controlling expenditures and creating variance and analysis reports.

Get PO and Show PO

The Get PO is an XML business service request that enables a business application module to request information concerning a specific purchase order from Oracle Purchasing. The reply to this request is the Show PO XML business request. There are a variety of business applications in several environments that may use this capability. For example, an MRP application may use this capability to ask for information from Oracle Purchasing, or a Plant Data Collection application may also use it to request information from Oracle Purchasing. This XML does not usually cause updates to occur.

As a Get PO request is made, the Show PO XML supplies purchase order information to another business application module. This request is also used as push notification of an event. There are many possible business applications in several environments that use this capability. For example:

1. Oracle Purchasing could use this request to send information to a Plant Data Collection application.
2. Other modules such as MRP, Inventory, or Manufacturing could use this to obtain order information.
3. The PO application can notify the MRP/Inventory application when a vendor gives or changes a promise to deliver.

Receive PO

The Receive PO is an XML business service request that supplies the information that the Oracle Purchasing module requires to assign a receipt posting tag to a purchase order. Oracle Purchasing uses the receiving and inspection information supplied by the Receive PO XML to ensure that the organization only accepts and pays for the items ordered, received, or inspected, depending on the identified matching rule. A variety of business applications in several scenarios may use this capability. For example:

- Receipt of information from other non-Oracle systems.
- Receipt of barcoded and other receiving information from scanners.
- Advance Shipment Notices (ASNs) sent from suppliers.

As the Receive PO XML supplies the information, Oracle's Receiving Open Interface maintains the integrity of the new data as well as the receipt data already in Oracle Purchasing.

Get Prodorder

Prodorder is an XML business document that details the work order information. This document may be related to three business service requests: Get Prodorder XML, Show Prodorder XML and Receive Prodorder XML. This enables the organization to import discrete job and repetitive schedule information, discrete job operations, and material, resource and scheduling information from any source.

Show Prodorder and Receive Prodorder

The Get ProdOrder XML enables a business application module to request specific work order information from another business application module. Sources may include handheld devices, other manufacturing execution systems, planning systems, and order entry systems. The reply to this is the SHOW PRODORDER XML. Information on the following items may be used to request a work order: the pre-determined bill of material structure, lot, or serial information about the item; the operation in the routing at which to change the work order; and the accounting information that can optionally accompany a change in the order.

The Show ProdOrder is an XML business document that supplies work order information to another business application module. The environment for this request can be within the enterprise or outside the enterprise. On the other hand, the Receive Prodorder XML is a business service request that supplies information which the ERP system requires to do receipt posting against a work order.

Transfer Item

The Transfer Item XML is a business service request that enables organizations to do sub-inventory or direct inter-organization transfers in Oracle inventory. As a company defines multiple inventories, warehouses, and manufacturing facilities as distinct organizations, the Transfer Item XML allows for the efficient performance of transfer of one or more items in a single transaction. This also allows the transfer of partial quantities of the same item to different sub-inventories and locations.

Get Issueinfo and Show_IssueInfo

IssueInfo is an XML business service request for information against an order, from the ERP system into a plant data collection system to confirm the item issue transaction. The environment for this integration is from plant data collection systems to the Oracle manufacturing modules. Information concerning the work order to which the material item is being issued, the lot or serial number for the work order item, and the operation in the routing of the production item is transmitted to request an item issue.

The purpose of the Show IssueInfo XML business service request is to supply to another system item issue information against an order to confirm the issue transaction against that order. Examples of order types would be a work order, a sales order, a service order, or a maintenance order. This XML may be used individually, or as part of a larger interface scenario. Oracle Manufacturing (including Inventory and MRP) - to interface with Oracle or non-oracle Order entry systems, work in process systems, and the plant maintenance systems using this IssueInfo XML - allows for the issuance against an order.

Confirm Issue

Issue is an XML business service request used to notify Oracle Manufacturing of the issue of required material to a work order for making a product. This XML is also used to notify Oracle Manufacturing of the return of material from a work order or job order back into inventory. The business environments most likely to require this capability are any type of manufacturing scenario.

The Issue XML communicates what item is being issued, where it is being issued from, which processing operation it is being issued to, what quantity was issued, and at what time this event occurred. In the case of a return, this communicates what item is being returned, which processing operation it is being returned from, which sub-inventory location it is being returned to, the quantity being returned, and the time at which this event occurred. This XML commonly causes updates to occur.

Issue MiscItem and Receive MiscItem

The MiscItem XML is another business service request that reflects an unplanned issue or receipt of an item to or from a miscellaneous location. Possible reasons for issuance include somebody broke the material, or the material is defective and needs replacing, or the material is used up and needs replenishment. Miscellaneous transactions may be issued from a plant data collection system or an inventory system to Order Management or Manufacturing modules. The MiscItem XML may be used to issue materials to groups that are not inventory, receiving or work in process such as research and development group. This can also assist in the issuance of items to individuals or projects or to issue damaged items to expense accounts such as scrap. The MiscItem XML may also be used to notify a corresponding business application to reflect an unplanned receipt of an item to a miscellaneous location. This is primarily designed for supplies items or MRO items.

Get Countinfo and Show Countinfo

Countinfo is an XML business service request to enable a business application to request and show inventory on-hand quantities from or to an Oracle Inventory system. The Countinfo XML enables a business application to request on-hand quantity information from an ERP system. This may also be used to show inventory count

information to enable Oracle Inventory to send inventory count information to a PDC or other application system. This request may be used as a response to a get countinfo request or as a push notification of an event. There are many possible business applications in several environments that may use this capability.

Update Invencount

Invencount is an XML business service request that transmits an inventory count to Oracle Inventory from the actual physical inventory location. This count may be a cycle count or a physical count. This XML may also apply to planned or unplanned inventory counts. The Invencount XML allows the user to transmit physical inventory count information that can e used to reconcile system-maintained item on-hand balances with the actual counts of inventory. Accurate system on-hand quantities are essential for managing supply and demand, maintaining high service levels, and planning production.

Get Credit, Show Credit, and Update_credit

Credit is an XML business service request used to either Request, Show, or Update credit. This also includes Change Status request used to keep order, shipment and open items amounts current. This contains all of the information necessary to make a decision to give credit to a customer. The Request is made for the Order Management to request credit data for a customer from Oracle Receivables. This does not imply any update, it is only an inquiry function. The Show Credit XML is used as a response back to the order management application. The Update Credit XML may be used in both directions between the order management and the accounts receivable application. Its purpose is to keep order, shipment and open item amounts current. The Update Credit XML also transmits changes in the accounts receivable open item balances to the credit management function of the customer order management application.

Change Status

The Change Status XML request is used to update the order management application with any changes in business status for a particular customer. The purpose of this request is to notify the customer order management application that the overall credit status of a customer has changed or status on specific order(s) are to be changed.

Load Projacctg

Projacctg is an XML business service request that enables all relevant sub-systems (such as Accounts Payable for payment of materials used, Accounts Receivable for invoicing of billing projects, Budget for validation, Purchasing to report committed costs, Assets to record capitalized projects and Human Resources system to update employee information) that submit single sided transactions to send information to Oracle Project Accounting. Applications which produce double-sided transactions would use other XML business service requests to update the project accounting application. For example, order management or purchasing applications would use the Load Receivable or Load Payable XML. The Projacctg XML assists in providing project-oriented companies with a flexible approach to define and structure projects, tasks and budgets by which to monitor project status. Integration with other applications allows effective accounting for costs or to process revenue and invoices.

Sync Projinfo

The purpose of the Sync Projinfo XML business service request is to enable all relevant sub-systems that submit transactions to Oracle Project Accounting to maintain valid values for the key project fields. The target applications for this update would include, but not necessarily be limited to: Accounts Payable, Accounts Receivable, Budget, Order Management, Purchasing, Time and Labor, Travel and Expense.

Get Wipconfirm and Show Wipconfirm

Get WIP Confirm is an XML business service request that enables the requesting of data necessary to perform a confirmation of the movement of WIP (Work in Process). This does not commonly cause updates to occur. This template may be used individually, or as part of a larger interface scenario with a plant data collection system or a shop floor control system. Relevant data to get confirmation of the movement of WIP include: information concerning the specific work order or job order and the WIP operation or step in a routing from which the product is being completed into inventory or at which to return back into manufacturing, accounting information collected in the WIP Completion and Return transaction, information concerning the production order in the transaction as reference to an existing job order, and information concerning the resources associated with a particular WIP operation or sub-operation within a routing.

As a response to the Get WIP Confirm XML, the XML template Show WIP Confirm is used.

Update Wipconfirm

Update WIP Confirm is an XML business service request that notifies Oracle Manufacturing of the completion of an end product in the production process and movement of that product to a finished goods inventory. This XML also notifies a Oracle Manufacturing of the return of end product from a finished goods inventory back into the production process. The business environments most likely to require this capability are any type of manufacturing scenario. This communicates which processing step the product is coming from, the quantity moving, the inventory organization or sub-inventory location it is moving to, and the time at which this event occurred. In the case of a return, the request communicates which inventory location the product is coming from, the quantity moving, the processing step it is moving to, and the time at which this event occurred.

Get PickList

The Get PickList is an XML business service request that enables a request for the retrieval of a single Pick Slip from an ERP system. A pick slip or pick list is an internal shipping document pickers use to locate items to ship for an order. The reply to this

request is the Show PickList. Individual lines from a Pick Slip are not selectable with this request. Only the complete document is selected and returned. Information relevant to the request include: a picking document that is generated in an ERP shipping module or Oracle Order Management.

List PickList

List Picklist is another XML business service request that provides a list of Pick Slips from an ERP system to another application. This may be initiated in response to a GETLIST PICKLIST request or upon some business event. When a receiving application receives this XML, the information can be used as is or it may be used to initiate a selection of a specific Pick slip through the GET PICKLIST request. The processing is designed to provide multiple occurrences of summary data. This does not usually cause updates to occur.

The LIST XML is used for the receiver of the GETLIST XML to respond with the results of the search that was initiated by the GETLIST. Each of the records being returned is contained within an instance of the LIST_PICKLIST element underneath the DATAAREA Element.

The attributes associated with the LIST XML are as follows:

- rsstart attribute is a number that indicates the starting record for the section of the resulting set returned in the list message. This value should always match the rsstart value in the originating GetList XML.
- rscount attribute is a number that indicates the number of records returned in the message. The subsequent request for additional records should have a rsstart value of rscount + 1.
- rstotal attribute is a number that indicates the total number of records in the result set.
- rscomplete attribute is a Boolean that indicates that the list provided exhaust the possible values.
- rsref attribute is a string that represents the implementation-specific result set identifier for subsequent requests.

Show PickList

The purpose of the SHOW PICKLIST XML is to show the details of an individual Pick Slip from an ERP system. This may be sent in response to a GET PICKLIST or it may be initiated upon some business event. This does not usually cause updates to occur. The picking document, the lines on a specific picking document, and details about a line item on a Pick Slip (such as date and time of loading and shipping) that are generated in an ERP shipping or order management system are some of the data that appears on the request.

Update PickList

Update Picklist is an XML business service request that updates the details of an individual Picking List from a plant level to an ERP system. A pick slip or pick list is an internal shipping document pickers use to locate items to ship for an order. This usually causes updates to occur.

Get Personnel and Show Personnel

Personnel is an XML business service request that may be used to either Get Personnel to request personnel data for a resource or Show Personnel to provide personnel data for a resource to a requesting business application. This facilitates integration between a human resource system to manufacturing systems such as shop floor and plant data collection. Personnel information such as employee name, cost center, division, employee category, status, job code, shift and wage group are required to manufacture, cost and schedule products.

Update Persontime

The UPDATE PERSONTIME is an XML business service request that updates work time information for an employee from a data collection application to an ERP Human Resource application. This causes updates to occur. This may pass on data to update personnel information, employee category, employee status, overtime, and the quantity of employees' reporting hours, days, and so on.

Sync Field

Sync Field is another XML business service request that enables the validation of data that exists on separate application's databases. This request can cause on-line validation to occur or may be a single tool for synchronizing data. In Oracle Financials, for example, this is very useful when a company engages in consolidation or inter-company transactions of which each balancing company resides in separate databases.

Sync Personnel

The XML called Sync Personnel is a business service request that enables the synchronization of employee data that exists on separate databases between manufacturing and human resource applications. The Sync Personnel allows the adding of new employees and their relevant data as well as the modification of previously established employees. The Sync Personnel is used to facilitate the maintenance of human resource data in a manufacturing work force planning module, in Oracle this may be under Master Scheduling module. This enables the workforce planning module to use current personnel information when creating finite production schedules. This can also be used by Oracle Project Accounting or a work order management application to assign qualified personnel or to perform resource planning. Employee data such as employee name, category, job code, position, shift and competency are ut a few data that need to be synchronized between a manufacturing and a human resource application.

Sync Wrkschdule

Sync Wrkschdule is an XML business service request that enables the synchronization of Work Schedule data that exists on separate databases. The Sync Wrkschdule allows the adding of new Work Schedules as well as the modification of previously established Work Schedules. This causes updates to occur and may be used as part of a large integration scenario or as a single tool for synchronizing data. This is designed primarily to enable synchronization of data in a Human Resource to Manufacturing Application integration scenario. Personnel work schedules detailed in a human resources system may be imported or exported to or from a manufacturing application such as Oracle Manufacturing.

Sync Mfgtlcode

The **Sync Mfgtlcode** is an XML template that provides manufacturing codes that need to be captured with the labor hours in a time and labor reporting application. The Sync Mfgtlcode can also be used to provide other operational codes (such as medical codes) to a time and labor reporting application. The reason for associating these codes with labor hours is due to informational requirements of the manufacturing and/or financial systems.

The Sync Mfgtlcode may not be useful or necessary in integration scenarios in which one of the following is true:

1. The manufacturing codes observe complex hierarchical relationships or valid combinations. In this scenario, a plant data collection application or shop floor control application would probably be the labor reporting system,
2. Large volume repetitive manufacturing environments or process manufacturing in which none of the fields contained in the Sync Mfgtlcode need to be associated with labor hours by the manufacturing or financial applications.

Sync COA

COA is an XML business service request that distributes general ledger chart of accounts (COA) code identifiers to other applications to store for validation purposes. This scenario assumes that the general ledger "owns" the chart of accounts definition and the instances of data within it. The sub ledger applications have several choices for validation of account numbers and other fields in the complete chart of accounts structure. One of these choices is to synchronize the chart of accounts structure and data from the general ledger application to all of the sub ledgers. The Sync COA allows for validation of account codes and account code combinations used in the subledger to record transactions and eventually post to the account balances in general ledger. The synchronization process can ensure that all of the components in the integration scenario have the same data necessary to communicate.

Sync Exchngrate

The purpose of the **Sync Exchngrate** XML business service request is to enable the passing of updates of currency exchange rates to other applications that have exchange rate tables. In Oracle Financials for example, daily rates and period-end rates tables are maintained in Oracle GL. The sync Exchngrate XML facilitates the updating of exchange rates for other subsystems that require rates to convert or revalue foreign currency denominated transactions.

Add Requisitn

Add Requisitn is an XML business service request that sends demand for goods or services to another business application for consideration of buying or in some way obtaining the requested items. In Oracle, this facilitates the sending of a purchase requisition to trigger the issuance of a purchase order for items requested. This XML usually causes updates to occur and may be used as part of a larger integration scenario or as a single tool for communicating demand. This request may originate from employees or requestor (using a self-service application), or may be triggered by Manufacturing modules such as MRP or Inventory when the reorder point reaches a level that signals the planner to issue a request for materials for production. Data such as supplier name, payment method and information that describes the requested item and its attributes including sub-components or sub-assemblies are sent to another business application (may be Oracle Purchasing) for validation and approval to purchase. The inclusion of a REQUISTNID (requisition ID) ties back a document to its origin, thus providing an identifier that enables drill back audit trail functionality.

Change Requisitn

The **Change Requisitn** XML business service request communicates changes to an existing purchase requisition for goods or services. This change must refer to the original document and/or item requested. The change processing assumes replacement of fields sent, with the exception of REQUISTNID and REQLINENUM fields. If any of the Field Identifiers above require changing, that constitutes a cancellation of the request and/or the addition of another requisition. This may usually cause updates to occur and may be used as part of a larger integration scenario (between an Inventory or MRP system to a purchasing system) or as a single tool for communicating demand.

Cancel Requisitn

The **Cancel Requisitn** XML business service request communicates from one business application to one or more other business applications that a previous requisition or requisition line item is no longer needed. This cancel must refer to the original document and/or item ordered. The user may either cancel the entire requisition or only specific lines on the requisition. To cancel the entire requisition, include only the REQUISITN Header information for the instance of the requisition you wish to cancel. To cancel a line or several lines, each line to be cancelled must be included. This XML commonly causes updates to occur and may be used as part of a larger integration scenario or as a single tool for communicating demand.

Get Requisitn and Show Requisitn

An XML business service request called **Get Requisitn** enables a business application to request information concerning a specific requisition from another business application which is usually a purchasing system. The reply to this XML is the Show Requisitn XML. This Get Requisitn XML does not usually cause updates to occur. It may be used as part of a larger integration scenario or as a single tool for requesting information on existing demands for goods or services. Other users may need requisition information to validate statistics on growing demand for a specific item for planning purposes.

To send information relative to demand for goods or services to another business application or may be a notification vehicle initiated upon an event in a business application, the Show Requisitn XML business service request is used. The Show Requisitn XML usually does not cause updates to occur and may be used as part of a larger integration scenario or as a single tool for communicating demand. All data relevant to a specified purchase requisition (such as Supplier information, information describing the requested items and its attributes, sub-components and sub-assemblies) is sent via the Show Requisitn XML.

Getlist Requisitn

Getlist Requisitn is an XML business service request enabling a business application to request summary information for one or more requisitions from another business application. The Getlist Requisitn also enables the retrieval of information across several documents by using selection fields. An example of this could be requesting all Requisition Lines for a specific ITEM. This type of functionality is limited to the capabilities of the responding application and needs to be determined during the implementation project. This XML does not usually cause updates to occur. It may be used as part of a larger integration scenario (between a manufacturing application and a purchasing application) or as a single tool for requesting information on existing demands for goods or services.

Many of the Data Types and the Field Identifiers contained within this XML may be used to select required information. This is done by requesting specific Field Identifiers or by requesting an entire Data Type. If the Data Type is not using Field Identifiers to select information, but the data within a Data Type is requested to be returned, the Data Type is coded without any Field Identifiers. This signifies to the responding application that all of the data that corresponds to that Data Type is to be included in the response. It is also possible to request that a range of values be returned the first occurrence of the XML is treated as the from and the second optional occurrence is treated as the to value. The maxitems attribute defined below indicates the maximum number of entries to return in the List XML that is used in response to this GETLIST.

List Requisitn

To send information relative to demand for goods and services to another business application the **List Requisitn** XML business service request may be used. This may be in response to a Getlist Requisitn request, or it may be a notification vehicle, initiated upon an event in a business application. The LIST verb describes the behavior of supplying one or several documents in a summary format to the requesting business application. These listings of information may be supplied for requisition documents, or requisition lines and/or requisition sub lines. This XML usually does not cause updates to occur and may be used as part of a larger integration scenario or as a single tool for communicating demand.

The List XML is used for the receiver of the Getlist XML to respond with the results of the search that was initiated by the Getlist. Each of the records being returned is contained within an instance of the LIST_REQUISITN element underneath the DATAAREA Element.

Getlist PO

Getlist PO is an XML business service request that enables a business application to request information containing summary information for one or more Purchase Orders from another business application. The Getlist PO also enables the retrieval of information across several documents by using selection fields. An example of this could be requesting all purchase order lines for a specific item. This type of functionality is limited to the capabilities of the responding application and needs to be determined during the implementation project. This does not usually cause updates to occur. It may be used as part of a large integration scenario or as a single tool for requesting information on existing demands for goods or services. A Plant Data Collection application could use this XML to request information from Oracle Purchasing or a MRP, Inventory or Manufacturing business application could use this to obtain order information.

Many of the Data Types and the Field Identifiers contained within this XML may be used to select required information. This is done by requesting specific Field Identifiers or by requesting an entire Data Type. If the Data Type is not using Field Identifiers to select information, but the data within a Data Type is requested to be returned, the Data Type is coded without any Field Identifiers. This signifies to the responding application that all of the data that corresponds to that Data Type is to be included in the response. It is also possible to request that a range of values be returned the first occurrence of the XML Element is treated as the from and the second optional occurrence is treated as the to value. The maxitems attribute defined below indicates the maximum number of entries to return in the List XML that is used in response to this GETLIST.

List PO

A **List PO** XML business service request is used to send information relative to demand for goods or services to another business application. This may be in response to a Getlist PO request, or it may be a notification vehicle, initiated upon an event in a

business application. The LIST verb describes the behavior of supplying one or several documents in a summary format to the requesting business application. These listings of information may be supplied for Purchase Orders, PO Lines, or PO Sub-Lines. This usually does not cause updates to occur and may be used as part of a larger integration scenario or as a single tool for communicating demand. There are many possible business applications in several environments that may use this capability. For example, a purchasing application could use this XML to send information to a Plant Data Collection application or a MRP, Inventory or Manufacturing business application could use this to obtain order information. The environment for this XML can be within the enterprise or outside the enterprise.

Add PO

The purpose of the XML called **Add PO** is to communicate from one business application to one or more other business applications that a Purchase Order has been added or needs to be added, depending on the business case. The environment for this XML can be within the enterprise or outside the enterprise. There are many possible business applications in several environments that may use this capability. One example of a business integration scenario where this XML could be useful is to communicate back to an Inventory system that a purchase order has been processed and added in the purchasing system. The planner or buyer may use the purchase order information (such as payment terms, miscellaneous charges, the description and price of items ordered, and the dates and quantities for delivery or shipment of ordered products) to analyze and schedule production.

Change PO

The purpose of the **Change PO** XML is to request another business application module such as Oracle Purchasing to make changes to an existing Purchase Order. This change must refer to the original document and/or item ordered. The change processing assumes replacement of fields sent, with the exception of the POID and the POLINENUM fields. If any of the Field Identifiers require changing, that constitutes a cancellation of the request and/or the addition of another Purchase Order.

Cancel PO

The **Cancel PO** is an XML business service request that allows a requestor or planner to communicate from one business application to one or more other business applications that a previous Purchase Order or Purchase Order line is no longer needed. This cancel must refer to the original document and/or item ordered. A planner or requestor may invoke this XML to change the entire purchase order or to change specific purchase order lines. To cancel the entire order, include only the Purchase Order Header information for the instance of the Purchase Order you wish to cancel. To cancel a line or several lines, each line to be cancelled must be included. This XML commonly causes updates to occur and may be used as part of a larger integration scenario or as a single tool for communicating demand. In Oracle Purchasing, a cancellation of a purchase order eventually cancels accounting entries related to the order.

Update Delivery

Delivery of goods from a supplier business partner via the services of a transportation provider (carrier) is an important event in a manufacturing or a purchasing application. As goods are delivered, Purchasing controls the items ordered through receiving, inspection, transfer, and internal delivery. These features control the quantity, quality, and internal delivery of the items delivered. The **Update Delivery** is an XML business service request that enables the update of delivery information for goods or services to another business application module. This XML can be initiated by a business application based on some event and sent to one or more relevant business applications. Possible business applications in several environments that may use this capability include: a Purchasing application to notify an Accounts Payable application of a specific delivery enabling the Accounts Payable application to accurately calculate the amount it needs to pay a business partner, a PO application could use this XML to send information to a Plant Data Collection application, a Purchasing application could use this to notify a MRP, Inventory, or Manufacturing business application that a delivery has occurred and the goods are available for use or inspection, and so on.

Update Inspection

An **Update Inspection** XML business service request supplies inspection information for goods or services to another business application module. This may is initiated by the sending system upon some event occurring. There are many possible business applications in several environments that may use this capability. Examples include: a PO application could use this to send information to a Plant Data Collection application, or vice versa; a MRP, Inventory, Purchasing or Manufacturing business application could use this to communicate inspection information; a Laboratory Information System could send quality information to an Inventory application; or a Quality Control application could send information to a MRP, Inventory, or Purchasing application. Matching of supplier invoices against a purchase order may be defined in Oracle Purchasing or Oracle Payables at a four-way level, that is price and quantity billed must match quantity received (or delivered), ordered and inspected. The information supplied by the Update Inspection XML supports the four-way matching in Oracle. The XML also describes reasons for quantities rejected in the INSPECTION Data Type.

Sync Item

Sync Item is an XML document that supplies information for goods or services to another business application module. This XML may also be initiated by the sending system upon some event occurring. This XML is not for synchronizing ITEM quantities at each inventory location. The Sync Inventory XML is used for this purpose. There are many possible business applications in several environments that may use this capability. For example, a MRP, Inventory, or Manufacturing business application could use this to communicate item information. This XML can be used to synchronize items used in finished goods, raw materials, work-in-process or components in a bill of materials. Data such as attributes of additional units-of-measure for the item, attributes of cost or value of an item and the location the item may be kept are synchronized for a more efficient handling of the items.

Sync Sitelevel

Sync Sitelevel is an XML business service request that enables a mechanism to ensure that the physical location identifiers (physical locations of organizations or the location codes and their meanings) are synchronized between the business applications that require this to communicate clearly. This is particularly critical when only the codes that identify locations are used. Without the meaning of the codes clearly communicated, the integration is not effective. This XML enables the SITELEVEL codes to be synchronized among business applications. This may also be initiated by the sending system upon some event occurring. There are many possible business applications in several environments that may use this capability. For example, a MRP, Inventory, or Manufacturing business application could use this to communicate SITELEVEL information. The environment for this XML can be within the enterprise or outside the enterprise.

Get Prodavil

Get Prodavil is an XML business service request that enables requests of product availability data by an Order Management business application (such as Oracle Management) to an Available to Promise (ATP) or Production business application. The business process scenario is the Order Management application interacting with the Available to Promise or Production application in order to determine availability of a product for the customer. This scenario is commonly referred to as Make to Order or Build to Order. The response to this request is the Show Prodavil. In the case where Finished Goods Inventory resides with the Order Management business application, Order Management may look at its' own Finished Goods Inventory before asking the ATP for product availability. Otherwise, The Order Management application always looks at the Available to Promise. The customer implementation rules or the business applications' capabilities usually determine this processing. In the case where there is a Finished Goods Inventory application on both sides of the integration, it is assumed that the Inventory applications are synchronized. In the case where Order Management is using the Production or Manufacturing Inventory, it is possible that Order Management would look at that Inventory availability before asking the Available to Promise application for product availability. This case is not covered at this time.

This XML usually does not cause updates to occur and may be used as part of a larger integration scenario or as a single tool for requesting product available to promise information. The picture below visualizes one possible use of this XML.

Show Prodavil

The purpose of the **Show Prodavil** XML business service request is to respond to a Get Prodavil request or to initiate the passing of product availability data from a Production or Available to Promise (ATP) business application to an Order Management business application. The business process scenario is the Order Management application interacting with the Available to Promise or Production application in order to determine availability of a product for the customer. This scenario is commonly referred to as Make to Order or Build to Order. This may or may not cause updates to occur and may be used as part of a larger integration scenario or as a single tool for requesting product available to promise information. Data necessary to communicate ATP such as the item, item description, available quantity, date and time of availability is passed from a production application to an order management application.

Update ProductReq

Update Productreq is an XML business service request that enables a business application such as Order Management to reserve a particular quantity of goods or services for a specific date and time. The business process scenario is the Order Management application interacting with the Available to Promise or Production application in order to determine availability of a product for the customer. This scenario is commonly referred to as Make to Order or Build to Order. In Oracle Applications, the Available to Promise is set up in the Order Management system, triggered by data from the manufacturing systems. The Update Productreq XML accomplishes this task in a two step process within this one request:

1. The receiving business application checks to see if an item is available in sufficient quantity by a specific date and time.
2. The receiving business application then reserves that quantity of inventory for that specific date and time combination if the product is available.

If the product requested is not available, the responding application may send one of two responses: either to use a Confirm XML to confirm the denial of the request or use a Show Prodavil to communicate an alternative product availability. This may be ITEM, DATE, or QUANTITY, or a combination of these. This may also be

accompanied with a message in the NOTES field Identifier stating that this is an alternative. If the product requested is available, the responding application sends a Confirm XML business service request to confirm the execution of the request.

Create Prodorder

A **Create Prodorder** XML notifies a Manufacturing Application of the need to make a product or parts in a specific quantity, for a specific need by date. The business environments most likely to require this capability are an Engineer to Order or a Configure to Order manufacturing scenario. This XML business request communicates what the product configuration is and what choices have been made from the configuration. This commonly causes updates to occur. As an order for a specific product is received in Order Management, a production order is created and an available to promise is set. The environment for this XML can be within the enterprise or outside the enterprise.

Cancel Prodreq

Cancel Prodreq is an XML business service request that communicates from one business application to one or more other business applications that a previously requested item is no longer required. This cancel must refer to the original item requested. To cancel the item(s), each item to be cancelled must be included. This XML commonly causes updates to occur. Information required to identify the previously ordered item and to reset the order information gets communicated using this XML.

Sync Inventory

The purpose of the **Sync Inventory** XML is to enable the synchronization of Inventory data that exists on separate Item Master databases. This data is not the master data that describes the attributes of the item such as dimensions, weight, or unit of measure. This is data that describes the ITEM as it exists at a specific location. The primary focus of this XML is to synchronize the quantity of an item by stocking location. This may create new Inventory records. Its purpose is to either create or update existing Inventory records.

Get BOM and Show BOM

Get BOM (Bills of Materials) is an XML business service request that enables an application to request specific Item Bill of Material information from another business application module. The response to the Get BOM is the Show BOM. There are many possible business applications in several environments that may use this capability. For example, an MRP, Inventory, or Manufacturing business application could use this to communicate Item Bill of Material information. The environment for this XML can be within the enterprise or outside the enterprise.

The **Show BOM** XML is used to supply Item Bill of Material information to another business application module. This XML may also be initiated by the sending system upon some event occurring. There are many possible business applications in several environments that may use this capability. For example, an MRP, Inventory, or Manufacturing business application could use this to communicate Item Bill of Material information. Data relevant to the business such as information describing the BOM structure and its contents, information describing the attributes of a specific item or option within a classification are supplied by using the Show BOM XML. An example of an option would be CD-ROM for a laptop computer. Then each of the types of CD-ROM's for the option would be a separate ITEM. An example of an option class would be memory for a laptop. The options could then be 12, 24, or 40 megabytes of RAM. Each of these options would then have separate ITEM identifiers for memory modules that makes up the appropriate amount of memory. For 40 megabytes of RAM, this could be two 16 megabyte memory modules and an 8 megabyte module, or one 16 megabyte and three 8 megabyte memory modules.

Sync BOM

Sync BOM (Bill of Materials) is an XML business service request that communicates to a business application module or system the need to initiate the creation of a Bill of Material structure. This XML may be necessary to address the Make to Order, Assemble to Order, or Mixed Mode business ordering scenarios in a Order Management to Manufacturing application integration scenario. In Oracle for example, Oracle Manufacturing and Oracle Order Management use bills of material to store lists of items that are associated with a parent item and information about how each item is related to its parent. Oracle Manufacturing supports standard, model, option class, and planning bills of material. There are many possible business applications in several environments that may use this capability. For example, an

MRP, Inventory, or Manufacturing business application could use this to communicate the requirement to synchronize a Bill of Material. The synchronization process can ensure that all of the components in the integration scenario have the same data necessary to communicate. The environment for this XML can be within the enterprise or outside the enterprise.

Allocate Activity

Allocate Activity is an XML business service request that enables the update of ACTIVITY information from a production or manufacturing application to a costing application. This is necessary for applications that are based on a Dual Cycle Accounting model. This Dual Cycle Accounting model does not capture the details of the activities that caused entries to be made in the general ledger application, but instead captures them in a separate overall costing application. For Single Cycle accounting systems, the Journals XML is used to ensure that the costing information flows from the Manufacturing Application to the Financial Application. This XML commonly causes updates to occur and may be used as part of a large integration scenario or as a single tool for updating data.

Load Matchdoc

When using Oracle Applications, the purchase order and invoice matching functionality exists in Oracle Accounts Payable. But for other application suites, the matching functionality is made in the purchasing application. The invoice matching process may include several document types, including the following: for a Two way match - Purchase Order and the Invoice; for a Three way match - Purchase Order, Invoice, and the Receipt; and for a Four way match – Purchase Order, Invoice, Receipt, and Inspection results. For the four way match, it is assumed that inspection results have been updated on the Purchase Order for visibility in matching. The **Load Matchdoc** is an XML that is used to keep invoice, purchase order, goods receipt note and inspection ticket information current.

The Load Matchdoc XML is for use both by the accounts payable application and the purchasing application in exchanging the transactions that are required to be matched. When matching takes place in the accounts payable application, the purchasing application must inform the accounts payable application of the purchasing transactions (purchase orders, goods receiving notes and inspection tickets) to which

the invoice (in accounts payable) is to be matched. These integration scenarios have been developed for document matching to occur at the line level within the PO document and the Invoice document. This may be a one to one relationship, or it may be a many to one relationship from Invoice to PO or from the PO to the Invoice. Charges not associated with a specific Invoice line must be matched individually. When matching takes place in the purchasing application, the accounts payable application may have to inform the purchasing application of the supplier invoice to which purchasing transactions (purchase orders, goods receiving notes and inspection tickets) are to be matched if the invoice is initially entered into the accounts payable application. Note that in some situations, invoices are entered directly into the purchase order application or are created by the purchase order application when using evaluated receipt settlement (ERS) and in this instance, it is not necessary to perform the separate integration described under this XML.

Update Matchok and Update Matchfail

After loading matching documents and matching functionality is executed in an accounts payable application, an **Update Matchok** XML or an **Update Matchfail** XML is used to send successful matching notification or failure notification to a purchasing application. The purpose of the Update Matchfail XML is to notify the purchasing application of a matching failure such as a tolerance failure. For example in Oracle Payables, a user can match payables invoices to purchase orders to ensure that only goods ordered, received and inspected are paid. Controls and tolerances are set to specify the range of variance allowable if the amounts or quantities on the invoice are greater than the amounts or quantities on the purchase order or receipt. Oracle Payables then creates distributions and checks that the match is within the defined tolerance. Oracle Payables may use the Update Matchok XML or the Update Matchfail XML to notify the purchasing application. A purchasing application may then use this data to close or adjust the purchase order records or inform the supplier of discrepancies accordingly.

Load_PLInvoice and Load Payable

Load PLInvoice XML is a business document that transmits data to create an unapproved open item in either a payables application or a purchasing application. The scope of the Load PLInvoice (load purchase ledger invoice) indicates that the supplier's invoice has not yet been approved and the invoice is to be used as part of the

invoice matching process. For invoices that are approved for payment, this is handled by a separate XML business service request called **Load Payable XML**. If the matching functionality exists in a purchasing application, the Load PLInvoice XML is used to transmit data into the purchasing application, and only after matching does the PO application use the Load Payable XML to post approved invoices into the payables application. On the other hand, if the invoice matching functionality exists in the accounts payable application, the invoice is entered into accounts payable using the Load PLInvoice XML, and purchasing publishes matching document information to which accounts payable subscribes.

Get Matchdoc and Show Matchdoc

A **Get Matchdoc** XML business service request is used to enable both the accounts payable application and the purchasing application to request the transactions that are required to be matched or request information concerning matching. In both cases, the receiving application uses the **Show Matchdoc** XML to return the requested information. This XML does not usually cause updates to occur. In certain application suites, purchase order and invoice matching functionality exists in the purchasing application, while in other suites this functionality exists in the accounts payable application. If the invoice matching functionality exists in the purchasing application, the invoice is entered into accounts payable, purchasing requests invoice information using Get Matchdoc XML and accounts payable provides invoice information using a Show Matchdoc XML. If the invoice matching functionality exists in the accounts payable application, the invoice is entered into accounts payable, purchasing requests matching document information using Get Matchdoc XML and accounts payable provides matching document information using Show Matchdoc XML.

Getlist Picklist

The purpose of the Getlist Picklist XML business service request is to enable a business application to request summary information for one or more Picking Slips from an ERP system. If a list of documents is requested, that list is used in order for a selection and GET request of a specific picking list to be made, if necessary. This XML does not usually cause updates to occur. For example, this XML may be used by a Plant Data Collection system to request for summary information for a Pick slip from an Order Management system. Requestor may give specific field identifiers or request an entire data type, depending on the need for information.

Update Inventory

Update Inventory is an XML business service request that enables the update of Inventory data that exists on separate Item Master databases. This data is not the master data that describes the attributes of the item such as dimensions, weight, or unit of measure. This is data that describes the ITEM as it exists at a specific location. The Update Inventory XML assumes that one Inventory Application is the "owner" of the data. Either the owner of the data or the non-owner of the data may initiate this XML. The Inventory Application which is the primary owner of the data is determined by which one updates the financial records to be posted to the general ledger. The non-owner inventory does not update the financial records. To do so may result in duplicate postings or other errors in the financial record keeping. All Inventory events which may affect the financial records need to be sent to the owner inventory either through the Update Inventory XML or the Sync Inventory XML. This does not create new Inventory records. Its purpose is to update existing Inventory records.

Getlist Countinfo and List Countinfo

A **Getlist Countinfo** XML enables a business application to request several occurrences of summary Inventory Count information from an ERP system. This may be used for cycle counting or for physical inventory counts. The response to this request is the **List Countinfo** XML. This may be used individually, or as part of a larger interface scenario such as a Plant Data Collection and an Inventory system. The effective date and time, document ID or document type and the item may be used to get count information. This XML enables range selections. This is accomplished by including two separate occurrences of a Field Identifier. The first occurrence is the "FROM" selection and the second, or duplicate occurrence of the Field Identifier is the "TO" in the range to be selected. If a second Field Identifier is not included, the selection continues until the data no longer applies to the selection or the MAXITEMS is reached. All of the inventory documents may be requested by sending a value of "ALL" in the DOCUMENTID field.

A **List Countinfo** XML enables an Inventory application system to send multiple occurrences of summary or detail inventory count information to a Plant Data Collection or other application system. Data such as item description, quantity and serial number are examples of data that may be returned in the List Countinfo XML response.

Cancel Prodorder

Cancel Prodorder is an XML business service request that notifies a Manufacturing Application of the need to cancel a previous order to make a product in a specific quantity, for a specific need by date. This XML may be used to cancel an entire Production Order, or a specific line on the production order. A cancel must refer to the original document and/or item ordered. To cancel the entire order, include only the header information for the instance of the Production Order that needs to be to cancelled. To cancel a line or several lines, each line (as indicated on the predetermined Bill of Material structure) to be cancelled must be included in the request. An Order Management system may send this request to a manufacturing system (in Oracle this is done under the Master Scheduling module) to cancel previously recorded Work or Job Order. Optionally, a user may also include information concerning the operation in the routing at which to change the production order in the Work in Process module.

Sync Salesorder

The Sync Salesorder is an XML that facilitates the synchronization of sales or customer order information kept on separate databases throughout an enterprise. The Sync Salesorder allows the adding of new sales orders and the modification of previously established sales orders.

Add Salesorder

For organizations with automated sales force systems, customer orders may be communicated to an Order Management system using the Add Salesorder XML business document. The Add Salesorder XML communicates from one business application to one or more other business applications that a Sales Order has been added or needs to be added, depending on the business case. interface scenario. Data pertinent to an order such as customer information and address (including the Bill To and Ship To addresses), sales person information, payment terms, any miscellaneous charges, an accounting distribution, the item or product ordered along with the quantity, price and other descriptive information and the schedule of delivery may be communicated effectively using the Add Salesorder XML.

Cancel Salesorder

In order to communicate from one business application to one or more other business applications that a previous Sales Order, line, or schedule is no longer needed, the **Cancel Salesorder** XML may be used. This cancel must refer to the original document, item, and schedule. To cancel the entire order, include only the Salesorder Header information for the instance of the Salesorder to be cancelled. To cancel sales order lines and/or salesorder schedules, each line or schedule to be cancelled must be included in the occurrence of the XML with the SOLINENUM and SOSLINENUM identifiers specified respectively. If a schedule is to be cancelled, the line that the schedule refers to must be included or the schedule can not be found. This XML may be used by a salesperson who needs to cancel a pre-recorded order in an Order Management system.

Change Salesorder

A Change Salesorder XML is a business service request that is used to request that another business application component make changes to an existing Sales Order. This change must refer to the original document and/or item ordered. The change processing assumes replacement of fields sent, with the exception of the fields for the sales order ID, the sales order line number, and the sales order schedule line number. If any of these Field Identifiers require changing, that constitutes a cancellation of the request and/or the addition of another Sales Order. This commonly causes updates to occur. A salesperson may decide to make changes to the orders already recorded in an Order Management system. This allows for flexibility in a system.

Get Salesorder and Show Salesorder

Get Salesorder is another XML business service request that enables a business application module to request information concerning a specific sales order from another business application. The reply to this XML is the Show Salesorder. There are several possible business applications in several environments that may use this capability. For example, a Sales Automation application may use this XML to ask for information from a Order Management application. Request may be made by specifying the sales order ID and may optionally request for shipment and line details.

The **Show Salesorder** XML supplies Sales Order Information to another business application module such as a sales automation system. This request may be used as a response to a Get Salesorder request or as a push notification of an event. This XML does not usually cause updates to occur. Examples of what may be returned or shown include customer information, address, sales person information, and if necessary information on each order line and schedule. A sales person may also use the data returned by a Show Salesorder XML to know the commission amount to be appropriately credited for the order, assuming that the commissions have already been calculated.

Getlist Salesorder and List Salesorder

To enable a business application module to request information containing summary information for one or more sales orders from another business application, such as Oracle Order Management, the **Getlist Salesorder** XML request may be used. The response to this request is the List Salesorder. The Getlist Salesorder also enables the retrieval of information across several documents by using selection fields. An example of this could be requesting all sales order lines for a specific item. This type of functionality is limited to the capabilities of the responding application and needs to be determined during the implementation project. This does not usually cause updates to occur. It may be used as part of a large integration scenario or as a single tool for requesting information on existing demands for goods or services. For example, a Sales Automation application may use this XML to ask for information from an order management application.

A **List Salesorder** XML is used to proactively send a listing of summary information about sales orders to one or more other applications. This does not usually cause updates to occur. It may be used as part of a large integration scenario or as a single tool for sending information concerning existing demands for goods or services. For example, an order management application may use this to respond to a request for information from a Sales Automation application.

Sync PO

Sync PO is an XML business document that facilitates keeping purchase order information synchronized on separate databases throughout an enterprise. The Sync PO XML allows the adding of new purchase orders and the modification of previously

established purchase orders. There are many possible business applications in several environments that may use this capability. One example of a business integration scenario where this XML could be useful is between a purchasing system and a Material Resource Planning (MRP) system under a manufacturing scenario. The environment for this XML can be within the enterprise or outside the enterprise.

Sync Routing

Routing is the process an order must take in order to produce the finished goods. The **Sync Routing** XML is used to communicate to a business application component or system the need to create a new Routing or to update an existing Routing structure. This XML may be necessary to address the Make to Order, Assemble to Order, and Finished Goods business ordering scenarios in a Logistics to Manufacturing application integration scenario. There are many possible business applications in several environments that may use this capability. For example, an MRP, Inventory, or Manufacturing business application could use this to communicate the requirement to synchronize the Routing necessary to build finished goods. This XML may be used individually, or as part of a larger interface scenario. Routings between a Configuration Management system and a manufacturing system or with a Finite Scheduling system are examples where the Sync Routing XML can be used. The environment for this XML can be within the enterprise or outside the enterprise.

Get Routing and Show Routing

The purpose of the **Get Routing** XML is to communicate to a business application module or system a request for an existing routing structure to be returned in a Show Routing XML. This may be used individually, or as part of a larger interface scenario.

Show Routing is an XML business service request that communicates to a business application module or system the relevant information about a specific routing. This XML may be used individually, or as part of a larger interface scenario. A Show Routing may be used to request a routing to be sent between a Routing and Configuration Management (or Oracle Bill of Materials module) system and a Manufacturing Execution System (Oracle Work in Process). This same scenario could exist between a Routing and Configuration Management system and a Finite Scheduling system. Relevant data such as the series of operations that create the routing, description of a particular item within a routing structure, a grouping of

operations for the routing as well as a sequencing of operations, relationships between operations, the people needed within an operation and the description of the step within an operation for a specific routing are examples of information that may be communicated.

Getlist Routing and List Routing

A **Getlist Routing** XML is used to communicate to a business application component or module a request for a summary list of a routing structure or structures to be returned in a List Routing XML. This XML may be used individually, or as part of a larger interface scenario. Example could be a Getlist Routing to request a list of possible routings to be sent via a List Routing between a Configuration Management system and a Manufacturing Execution System. This same scenario could exist between a Configuration Management system and a Finite Scheduling system.

The purpose of the **List Routing** XML is to communicate one or more summary listings of routing information to another business application component. This may be the result of a Getlist request between a Bill of Material system and a Work in Process system or it may be initiated by some other business event.

Getlist BOM and List BOM

Getlist BOM (Bill of Material) is an XML that enables an application or component to request a summary list of Bill of Material information from another business application or component. The response to the Getlist BOM is the List BOM. The Getlist BOM also enables the retrieval of information across several documents by using selection fields. An example of this could be requesting all Bills of Material for a specific ITEM. This type of functionality is limited to the capabilities of the responding application and needs to be determined during the implementation project. There are many possible business applications in several environments that may use this capability. For example, an MRP, Inventory, or Manufacturing business application could use this to communicate Item Bill of Material information.

The **List BOM** XML is used to communicate one or more summary listings of BOM information to another business application component. This may be the result of a Getlist BOM XML or it may be initiated by some other business event. This XML may return information that generally describes the Bill of Material Structure and its

contents, the attributes of a specific item, the attributes of a specific OPTION for an ITEM and information that describes the class of OPTION for a particular Product or Item.

Get Item and Show Item

A **Get Item** XML enables a business application module to request information concerning a specific ITEM from another business application. The reply to this XML is the Show Item XML. There are many possible business applications in several environments that may use this capability. For example, an MRP, Inventory, or Manufacturing business application could use this to request item information. With Oracle applications, the Master Scheduling/MRP system uses item information to do forecasting of resources, Purchasing uses item information as basis for material replenishment and Inventory maintains these item information in its Master Item table.

To supply ITEM information to another business application module the Show Item XML can be used. This request may be used as a response to a GET ITEM request or as the result of some other business event. This XML does not usually cause updates to occur. There are many possible business applications in several environments that may use this capability. For example, item information may be communicated to a Purchasing Application from the Master Files kept in a MRP or Inventory system. The environment for this XML can be within the enterprise or outside the enterprise. Item information may include: general description of an item and its attributes, attributes of additional units-of-measure for the Item, attributes of cost or value for the Item and the location the item may be kept and the attributes of that location in relation to the item.

Getlist Item and List Item

Item is anything made, purchased, or sold including components, sub-assemblies, finished goods or supplies. A **Getlist Item** XML business document enables a business application module to request summary information concerning an item or items from another business application. For example, an MRP, Inventory, or Work in Process application could use this to request item information from a purchasing application. This type of functionality is limited to the capabilities of the responding application and needs to be determined during an implementation project. Items are defined under Oracle Inventory and item information is used by other modules such as MRP,

Inventory and Work in Process. The response to this request is the List Item XML. This does not usually cause updates to occur. The environment for this XML can be within the enterprise or outside the enterprise.

The List Item XML enables a business application module to respond to a Getlist Item request or to proactively send a listing of summary information about items to one or more other applications. Attributes of an item such as the description of the packing material to be used to package the item at its stocking unit of measure, the physical characteristics of the item, the shipping material to be used to ship the item and other attributes that generally describe the item defined in Oracle Inventory, may be listed for use by another application.

Getlist Prodorder and List Prodorder

Prodorder is an XML business document that details the manufacture of a specific quantity of assembly, using specific materials and resources, in a limited time. For other applications, this may be termed as Work Order, Discrete Job or Assembly Order. With the **Getlist Prodorder** XML, a business software component such as a Receiving or Plant Data Collection system is able to request summary production order information from a Production system or Work in Process application. This XML also enables the retrieval of information across several documents by using selection fields. An example of this could be requesting all production order lines for a specific item.

The purpose of the **List Prodorder** XML request is to enable a business software component to respond to a request or to proactively send a listing of summary information about production orders to another business software component. This XML does not usually cause updates to occur. It may be used as part of a large integration scenario or as a single tool for requesting information on existing demands for goods or services. Information on individual items on the pre-determined Bill of Material structure, lot or serial number information about the final assembly defined in the production order, and the accounting information that can optionally accompany the production order may be listed.

Sync Prodorder

The purpose of the **Sync Prodorder** XML request is to notify a manufacturing application of a change in the need to make a product. Such changes can in quantity or in need by date. The business environments most likely to require this capability are an engineer to order or a configure to order manufacturing scenario. This XML can communicate a new revision or change in configuration of the product being made.

Sync Engchgordr

Engineering Change Orders (ECOs) enable control of new item revisions and bill of material changes. With one ECO, several bill of material changes can be grouped that affect one or more bills. Using Oracle Engineering, ECOs can be defined for all types of items and bills, including manufacturing and engineering items, bills and their components, planning, model, option class, and standard items, primary and alternate bills of material.

Sync Engchordr is an XML business service request that communicates to a business application module or system the need to initiate the creation of an Engineering Change Order. This XML may be necessary to address the make to order, assemble to order, or mixed mode business ordering scenarios in an order management to manufacturing application integration scenario. There are many possible business applications in several environments that may use this capability. For example, a PDM, MRP, inventory, or manufacturing business application could use this to communicate the requirement to synchronize an Engineering Change Order.

Get Engchgordr and Show Engchordr

Engineering Change Orders (ECOs) enable control of new item revisions and bill of material changes. With one ECO, several bill of material changes can be grouped that affect one or more bills. Using Oracle Engineering, ECOs can be defined for all types of items and bills, including manufacturing and engineering items, bills and their components, planning, model, option class, and standard items, primary and alternate bills of material.

Get Engchordr is an XML business service request that communicates to a business application module or system the need to produce a Show Engchordr XML business document for the Engineering Change Order specified in the message. This XML may be necessary to address the make to order, assemble to order, or mixed mode business ordering scenarios in an order management to manufacturing application integration scenario.

To communicate to a business application module or system the sending systems representation of a specified Engineering Change Order, the Show Engchordr XML is used. The engineering change order is an instruction from design engineering or integrated product team that the approved design has been agreed by all stakeholder departments and is the valid method of manufacture on a given date and model unit.

Confirm Engchordr

Engineering Change Orders (ECOs) enable control of new item revisions and bill of material changes. With one ECO, several bill of material changes can be grouped that affect one or more bills. Using Oracle Engineering, ECOs can be defined for all types of items and bills, including: manufacturing and engineering items, bills and their components, planning, model, option class, and standard items, primary and alternate bills of material.

The purpose of the **Confirm Engchordr** XML is to communicate to a business application module or system that the synchronization of a specified engineering change order has been completed successfully. This XML may be necessary to address the make to order, assemble to order, or mixed mode business ordering scenarios in a order management to manufacturing application integration scenario.

Sync Dspchlist

A dispatch list is a list of tasks to be done by a manufacturing execution system in order to fulfill production orders. After production orders are sent out, master scheduling creates the dispatch list. Several applications (such as finite scheduling and ERP) provide a dispatch list to a manufacturing system.

Sync Dspchlist is an XML business service request used to synchronize dispatch list (finite schedule) information. This synchronizes information about the entire WIP transaction, information concerning the specific WIP operation, or step in a routing

and information concerning the resources associated with a particular WIP operation. The synchronization process can ensure that all of the components in the integration scenario have the same data necessary to communicate.

Get Dspchlist and Show Dspchlist

A dispatch list is a list of tasks to be completed by a manufacturing execution system in order to fulfill production orders. After production orders are sent out, master scheduling creates the dispatch list. Several applications (such as finite scheduling and ERP) provide a dispatch list to a manufacturing system.

The purpose of the Get Dspchplist XML business service request is to enable a business application module to request this information from another business application. The reply to this XML is SHOW. This XML does not usually cause updates to occur. This XML may be used individually, or as part of a larger integration scenario. For example, a manufacturing execution system or work in process module may get dispatch list information from a production planning module.

Show Dspchlist is an XML business document that communicates to a business application module, such as production planning, the sending systems representation of dispatch list (finite schedule) information. This XML may be used as a response to a Get Dspchlist request or as a push notification of an event. Information about the entire WIP transaction, specific WIP operation, or step in a routing and information concerning the resources associated with a particular WIP operation may be sent through the XML.

Update Dspchlist

Update Dspchlist is an XML business service request that allows a user to update or make changes to an existing dispatch list (finite schedule) information. A dispatch list is a list of tasks to be completed by a manufacturing execution system in order to fulfill production orders. After production orders are sent out, master scheduling creates the dispatch list. Several applications (such as finite scheduling and ERP) provide a dispatch list to a manufacturing system. This XML causes updates to occur.

Sync Maintorder

Sync Maintorder is an XML that ensures that all business software components in a specific integration instance have the current maintenance order information. This XML is commonly used to publish the need to create or update a Maintenance Order in a publish and subscribe integration environment. One possible scenario is the synchronization of Maintenance Order between field devices, service trucks, and so on, with a Computerized Maintenance Management System (CMMS). The synchronization process can ensure that all of the components in the integration scenario have the same data necessary to communicate. This data may include a description of the maintenance order, any accounting distribution information associated with the order, safety information related to work, location, and equipment, information on failure, cause and remedy, resources to perform the maintenance order, labor and tooling requirements to perform the operation, and information on the specific maintenance operation to be performed.

Create Maintorder

The purpose of the Create Maintorder XML document is to publish to a business application component or system the need to create or update a maintenance order. This XML is commonly used to in a publish and subscribe integration environment. This XML may be used individually, or as part of a larger interface scenario. For example, creation of a maintenance order may be sent from a field device to a Computerized Maintenance Management System (CMMS).

Update Maintorder

The purpose of the Update Maintorder XML document is to communicate any updates or changes necessary in an existing maintenance order. The environment for this XML can be within the enterprise or outside the enterprise. Maintenance orders detail the necessary maintenance operations for service equipment and tools used in the manufacturing scenario. Updates or changes may be made with the maintenance order information such as labor or craft resources, tool resources, operation, labor, and material resources or header information describing the maintenance order.

Cancel Maintorder

Cancel Maintorder is an XML business service request that communicates from business application component such as a field device to one or more other business applications (that is, a computerized maintenance management system) that a previous maintenance order is no longer needed. This cancel must refer to the original document and/or maintenance order posted.

Get Maintorder and Show Maintorder

The purpose of the Get Maintorder XML is to enable a business applications module to request order information from another business application. The response to this XML is SHOW. For example, a field device may want to get relevant information such as the resources to be used in the conduct of the maintenance operation, the operations that need to be performed under the order, and accounting distribution that may be necessary to relate the expenses of the maintenance to a particular cost center or department from a Computerized Maintenance Management system (CMMS).

In response, the CMMS creates a Show Maintorder XML to communicate data that meets the selection criteria posted under a Get Maintorder XML. The sending application sends what is requested in the Get Maintorder XML, if applicable, or alternatively, sends information that matches the data within the specified data type. A Show Maintorder XML communicates to a business application module or system the sending systems representation of maintenance order information.

Getlist Maintorder List Maintorder

A Getlist Maintorder XML enables a business application module to request information containing summary information for one or more maintenance orders. The response to this request is the LIST. The environment for this XML can be within the enterprise or outside the enterprise. The Getlist Maintorder also enables the retrieval of information across several documents by using selection fields. An example of this could be requesting all RESOURCE data type occurrences for a specific OPERATION. This type of functionality is limited to the capabilities of the responding application and needs to be determined during the implementation project. This XML does not usually cause updates to occur.

A List Maintorder XML publishes one or more summary listings of maintenance order information to other business application component. This may be in response to a Getlist Maintorder request or to proactively publish a listing of summary maintenance order information for a business event. When a receiving application receives this XML, the information can be used as is or it may be used to initiate a selection of a specific maintenance order through the Get Maintorder XML request. The processing is designed to provide multiple occurrences of summary data.

Sync UOMGroup

A UOMGROUP is a set of item-independent relationships that describe how an alternate units-of-measure is related to the stocking unit-of-measure or to other alternate units-of-measure. The environment for this XML can be within the enterprise or outside the enterprise. The purpose of the Sync UOMGROUP XML is to supply a set of unit-of-measure relationships to another business application module. This XML addresses the need for applications to exchange item-independent alternative UOM information beyond the stocking UOM. Item-independent implies that the materials management component or other subscribing applications manage the item's UOM as an entirely separate grouping that are later associated with one or more items.

This XML supports unit-of-measure (UOM) information flows associated with items being managed as inventory. The materials management, inventory control, warehousing and receiving, or shipping departments within an organization require the packaging relationship details between UOMs and the handling characteristics for each UOM to effectively manage the flow and storage of inventory. While the stocking level UOM provides for basic inventory accounting functionality, it may not be the most efficient, common, or natural UOM for manufacturing, purchasing, selling, or handling the item. The item may be packaged into standardized or supplier-specific bulk quantities for import or export, wholesale, or retail sales (for example, each, box, case, pallet, and so on).

Get UOMGroup and Show UOMGroup

A UOMGROUP is a set of item-independent relationships that describe how an alternate units-of-measure is related to the stocking unit-of-measure or to other alternate units-of-measure.

The purpose of the Get UOMGROUP XML is to request an existing UOMGROUP structure or structures from a business application component or module to be returned in a Show UOMGROUP XML. There are many possible business applications in several environments that may use this capability. For example, an MRP, inventory, or manufacturing business application could use this to request alternate UOM information for one or more items.

The Show UOMGROUP XML transfers the UOM relationships independent of the item definition. This XML supplies Unit-of-Measure Group relationship information to another business application module. This request may be issued as a response to a Get UOMGROUP request or as the result of some other business event. This XML addresses the need for applications to exchange item-independent alternative UOM information beyond the stocking UOM. Item-independent implies that the materials management component or other subscribing applications manage the item's UOM as an entirely separate grouping that are later associated with one or more items. The environment for this XML can be within or outside the enterprise.

Getlist UOMGroup and List UOMGroup

A UOMGROUP is a set of item-independent relationships that describe how an alternate units-of-measure is related to the stocking unit-of-measure or to other alternate units-of-measure.

The purpose of the Getlist UOMGROUP XML is to request a summary list of a UOMGRPHDR structure or structures from a business application component or module to be returned in a List UOMGROUP XML. The environment for this integration is application-to-application (A2A) within a single enterprise, or extended across enterprises, where there is a single master application that manages the assignment of all item identifiers. There are many possible business applications in several environments that may use this capability. For example, an MRP, inventory, or manufacturing business application could use this to request alternate UOM information for one or more items.

A List UOMGROUP XML supplies unit-of-measure group summary information to another business application module. This may be the result of a Getlist UOMGROUP request or some initiated by some other business event. When a receiving application receives this XML, the information can be used as is or it may be used to initiate a selection of a specific UOM group through the Get UOMGroup request. The processing is designed to provide multiple occurrences of summary data. The List XML is used for the receiver of the Getlist XML to respond with the results of the

search that was initiated by the Getlist. Data such as those describing the packing material to be used to package the UOM, the inventory stocking unit of measure that can be tracked by an inventory control application, physical characteristics of the UOM (that is, height, volume, width), description of the shipping material to be used, are listed in summary by a List UOMGROUP XML to respond to a particular GetList XML.

Sync Catalog

Sync Catalog is an XML document that communicates the need to initiate the creation of catalog information, as well as update existing catalog information, to a business application module or system. In communicating catalog information, the Sync Catalog may cause other information to be coordinated including item identifiers, specifications, pricing information agreed (that is, purchase agreements, price lists), availability and delivery information, and related items and accessories. This XML may be necessary to address the make to order, ordering scenarios in an Order Management to Manufacturing application integration scenario.

The existing definition does not attempt to model configuration rules, and is therefore not sufficient to model configure to order environments that require cross option validation. Assemble to order environments that can be modeled using a marketing bill of material are addressed. There are many possible business applications in several environments that may use this capability. For example, a Manufacturing, distributor or reseller business application could use this to communicate the requirement to synchronize a Catalog.

It may also be necessary to support Component Supplier Management (CSM) scenarios. In this scenario, a company provides a service of sourcing and codifying the products of many companies and publishing a consolidated catalog. A customer purchases the product from the Catalog provider. They have the capability to do comparison shopping from the catalog.

Get Catalog and Show Catalog

The purpose of the GET CATALOG XML is to enable a business application module or system to request catalog information. In communicating Catalog information, the Get Catalog XML may cause other information to be coordinated including, but not

limited to Item Identifiers, Specifications, Pricing Information agreed (that is, Purchase Agreements, Price Lists), Availability and Delivery Information and related items and accessories. This XML may be necessary to address the Make to Order or ordering scenarios in an Order Management to Manufacturing application integration scenario. There are many possible business applications in several environments that may use this capability. For example, a Manufacturing, distributor, or reseller business application could use this to get Catalog information.

It may also be necessary to support Component Supplier Management (CSM) scenarios. In this scenario, a company provides a service of sourcing and codifying the products of many companies and publishing a consolidated catalog. A customer purchases the product from the Catalog provider. They have the capability to do comparison shopping from the catalog.

A Show Catalog is an XML response to a Get Catalog request. The environment for this XML can be within the enterprise or outside the enterprise, including trading partners. The purpose of the Show Catalog XML is to supply a business application module or system with requested catalog information. Catalog information may include: fields that describe the catalog, identity of the publisher of the catalog, any category defined in the system to group specific items, description or listing of the items identified under the catalog, the specifications and specification value of a given item. The sending application sends what is requested in the Get Catalog XML, if applicable, or alternatively, sends information that matches the data in each of the data types.

Sync Itemclass

The Item Classification XML may be used in both a business to business context and an application integration context. An example of this in an application integration context might be between a Product data management system and a procurement system to access outsourcing opportunities for items in a similar classification. Another example in a business to business context may be a component supplier management company letting a supplier know the marketing classification scheme it wishes to use in its catalog.

Sync Itemclass is an XML that communicates to a business application module or system the need to synchronize the classification and specification schemes. This may be necessary to address Manufacturers, Distributor Resellers business ordering

scenarios in an Order Management to a Manufacturing application integration scenario. It may also be necessary to address Component Supplier Management scenarios.

There are many possible business applications in several environments that may use this capability. For example, a PDM, MRP, Inventory, or Manufacturing business application could use this to communicate the requirement to synchronize Item classification schemes. Examples may include, (but are not limited to): marketing classifications, production classifications, procurement classifications and shipping classifications.

Get Itemclass and Show Itemclass

The Item Classification XML may be used in both a business to business context and an application integration context. An example of this in an application integration context might be between a Product data management system and a procurement system to access outsourcing opportunities for items in a similar classification. Another example in a business to business context may be a component supplier management company letting a supplier know the marketing classification scheme it wishes to use in its catalog.

Get Itemclass is an XML business document that enables a business application module or system to request information concerning classification and specification schemes. The environment for this XML can be within the enterprise or outside the enterprise, including trading partners. This XML may be necessary to address Manufacturers, Distributor Resellers business ordering scenarios in an Order Management to Manufacturing application integration scenario. It may also be necessary to address Component Supplier Management scenarios.

There are many possible business applications in several environments that may use this capability. For example, a Catalog Management System could use this to communicate the requirement to get Item classification schemes information from Product Data Management, Materials Resource Planning, Inventory, or Manufacturing business applications. The CLSSSCHMID Field Identifier is used as a selection field. This is the only Field Identifier value that should be sent in this GET request.

A Show Itemclass XML is used to supply a business application module or system with information concerning classification and specification schemes. This XML may be necessary to address Manufacturers, Distributor Resellers business ordering scenarios in a Order Management to Manufacturing application integration scenario.

It may also be necessary to address Component Supplier Management scenarios. This XML returns information such as classification scheme description, data that describes an element or breakdown of the requested classification scheme, item category and item category assignments.

Sync ITEMXREF

The Item cross-reference XML may be used in both a business to business context and an application integration context. An example of this in a business to business context might be a customer letting a supplier know the item numbering for a given European Article Number. An example of this in an application integration context might be between a Product Data Management system being the source system for Harmonized Schedule B numbers that are used by the transportation management system.

Sync ITEMXREF is an XML document that communicates to a business application module or system the need to synchronize an Item cross-reference. Cross-references may be to other item identifiers to the same form fit and function, as well as references to item identifiers of other items (form fit and function). For this XML, item relationships are used to refer to where the “to item” identifier, identifies a different form, fit and function to the “from item” identifier. It should be noted that the item identifier that is “Primary” in one system may be a secondary identifier in another system.

For example, in the Application Integration space, the manufacturing system may regard the “Item Number” as the primary identifier. The Order Management System may regard the Catalog number as the primary identifier. A company that manufactures hand held multi-meters may identify a given item in manufacturing with a 12 digit numeric code, 5432 123 12345. The marketing and sales department may refer to the same item by its catalog number of FL 30/4.

In the Business to Business case a supplier of hand held multi-meters may market their products through a catalog provider. The supplier has an item identifier with a corresponding party specific cross reference to the catalog providers identifier for the same item. The catalog provider has a item identifier for hand held multi-meters and a corresponding party specific cross reference to the suppliers item number. An example of this in a business to business context might be a customer letting a supplier know the valid substitutes that a supplier may supply to fulfill an order for a specific item number. An example of this in an application integration context might be between a Product Data Management (PDM) system and an Order Management system for accessories that may be offered to a customer with the sales of a major item. For

example, if a designer of a video camera has designed it to work with accessories such as tripod, extended life battery pack, external microphone and head cleaner, the video may be designed to have the spares replaced by the consumer such as lens cover, strap and handle. The video camera may need the following consumable items on a recurring basis: video cassettes, batteries or lens cleaners. The relationship between these items and the video camera may need to be represented to the Customer in Web Based ordering or Customer Service Representative (CSR), in desk based order entry.

Get ITEMXREF and Show ITEMXREF

The purpose of the Get ITEMXREF XML is to enable a business application module or system to request information concerning an Item cross-reference. Cross-references may be to other item identifiers to the same form fit and function, as well as references to item identifiers of other items (form fit and function). For this XML, item relationships are used to refer to where the “to item” identifier, identifies a different form, fit and function to the “from item” identifier. It should be noted that the item identifier that is “Primary” in one system may be a secondary identifier in another system. For example, in the Application Integration space, the manufacturing system may regard the “item Number” as the primary identifier. The Order Management System may regard the Catalog number as the primary identifier. A company that manufacture hand held multi-meters may identify a given item in manufacturing with a 12 digit numeric code, 5432 123 12345. The marketing and sales department may refer to the same item by its catalog number of FL 30/4.

This XML may be necessary to address Manufacturers, Distributor Resellers business ordering scenarios in a Order Management to Manufacturing application integration scenario. It may also be necessary to address Component Supplier Management scenarios. The Catalog Management System quotes a set of cross references that are not currently in the catalog. The cross-references may be either alternate identifiers of the products in the Catalog such as EAN Numbers and UPC Codes or they may be related items such as accessories, spares, or consumables. The Cross Reference document may flow from either the purchasing system of the selling system to let the Catalog Management System be aware of the identifiers used by all buying and selling parties.

A Show ITEMXREF XML supplies a business application module or system with information concerning an Item cross-reference. The XML may be used to relate item identifiers for item identifiers that identify different items (form fit and function). The Relationship types may also be universal. The sending application sends what is

requested in the GET, if applicable, or alternatively, sends information that matches the data in the specified Data Type. The Show ITEMXREF XML allows the user to cross-refer either between classifications, or between items.

Sync Pricelist

A Price List is a list containing the basic selling price per unit for a group of items, item categories, or services offered. Price lists are essential to ordering products because each item entered on an order must have a price. Most Order Management systems contain basic list information and one or more pricing lines, pricing attributes, qualifiers, and secondary price lists. Basic information for Oracle Order Management includes the price list name, effective dates, currency, pricing controls, rounding factor, and shipping defaults such as freight terms and freight carrier.

Sync Pricelist is an XML document that communicates to a business application module or system the need to synchronize product price list information. This allows the publication from the order management system of the Price List that accompanies the catalog. This XML may be necessary to address the Make to Order, Assemble to Order, or Mixed Mode business ordering scenarios in an Order Management to a Manufacturing application integration scenario.

There are many possible business applications in several environments that may use this capability. For example, a Manufacturing, distributor or reseller business application could use this to communicate the price change or request a price list. It may also be necessary to support Component Supplier Management (CSM) scenarios. In this scenario, a company provides a service of sourcing and codifying the products of many companies and publishing a consolidated catalog. A customer purchases the product from the Catalog provider. They have the capability to do comparison shopping from the catalog.

The primary business process the Pricelist XML supports are to provide an instruction from a supplier to a customer that the supplier’s list price for the suppliers product needs updating. It may also be an instruction from Marketing Systems to update order management systems. It may be an instruction from Order Management systems to Sales force automation systems to update Price Lists.

Get Pricelist and Show Pricelist

A Price List is a list containing the basic selling price per unit for a group of items, item categories, or services offered. Price lists are essential to ordering products because each item entered on an order must have a price. Most Order Management systems contain basic list information and one or more pricing lines, pricing attributes, qualifiers, and secondary price lists. Basic information for Oracle Order Management includes the price list name, effective dates, currency, pricing controls, rounding factor, and shipping defaults such as freight terms and freight carrier.

The Get Pricelist XML business document enables a business application module or system to request information concerning new or existing product price lists. The environment for this XML can be within the enterprise or outside the enterprise, including trading partners. This XML may be necessary to address the Make to Order, Assemble to Order, or Mixed Mode business ordering scenarios in an Order Management to Manufacturing application integration scenario.

There are many possible business applications in several environments that may use this capability. For example, a Manufacturing, distributor or reseller business application could use this to request a price list. The Catalog Management System quotes a price list that does not exist in the Catalog Management System causing the Catalog Management System to request the price list. It may also be necessary to support Component Supplier Management (CSM) scenarios. In this scenario, a company provides a service of sourcing and codifying the products of many companies and publishing a consolidated catalog. A customer purchases the product from the Catalog provider. They have the capability to do comparison shopping from the catalog. Selection of price list to be returned may be made by providing a Pricelist ID using the Get Pricelist XML.

The purpose of the Show Pricelist XML is to supply a business application module or system with information concerning new or existing product price lists. This allows for the publication from the order management system of the Price List that accompanies the catalog.

The following types of data may be returned in a Show Pricelist XML:

- Price List Lines—the list of items or commodities and a base price for the items
- List price breaks—the prices and modifiers to the price for buying a given quantity or value of an item or item category on a price list line

- List Price Breaks—the prices and modifiers to the price for buying a given value of any product
- Price List Qualifiers—qualifies the Price Lists that may be used to price an item or item classification, in a given catalog, on a given date or for a given customer.

Sync Itemspecs

The Item Specification XML may be used in both a business to business context and an application integration context. An example of this in an application integration context might be between a Product Data Management (PDM) system and a Procurement system to access outsourcing opportunities for items with similar specifications.

Sync ITEMSPECS is an XML document that communicates to a business application module or system the need to synchronize the specification of items within a catalog. The Item specification describes items in a catalog. The environment for this XML can be within the enterprise or outside the enterprise, including trading partners.

This XML may be necessary to address Manufacturers, Distributor Resellers business ordering scenarios in a Order Management to Manufacturing application integration scenario. It may also be necessary to address Component Supplier Management scenarios. There are many possible business applications in several environments that may use this capability. For example, a Product Data Management, Material Resource Planning, Inventory, or Manufacturing business applications could use this to communicate the requirement to synchronize Item specifications. Examples may include, (but are not limited to): Engineering systems communicating specification information to a purchasing system or a Supplier to a CSM company letting the CSM know the specifications of the items in the supplier's catalog.

Get Itemspecs and Show Itemspecs

The purpose of the Get ITEMSPECS XML is to enable a business application module or system to request information concerning the specification of items within a catalog. The Item specification describes items in a catalog. The environment for this XML can be within the enterprise or outside the enterprise, including trading partners. This XML may be necessary to address Manufacturers, Distributor Resellers business ordering scenarios in a Order Management to Manufacturing application integration scenario.

It may also be necessary to address Component Supplier Management scenarios. An example scenario for the Get ITEMSPECS XML could be that the specification documents quoted in the catalog are not yet in the Catalog Management System causing the Catalog Management System to request the item specifications from the Product Data Management system (PDM).

Using a Show ITEMSPECS XML, the PDM system publishes a set of Specifications for the items in the catalog. The purpose of the Show ITEMSPECS XML is to supply a business application module or system with information concerning the specification of items within a catalog. The Item specification describes items in a catalog. This XML may be necessary to address Manufacturers, Distributor Resellers business ordering scenarios in a Order Management to Manufacturing application integration scenario. It may also be necessary to address Component Supplier Management scenarios.

The sending application sends what is requested in the GET, if applicable, or alternatively, sends information that matches the data in the following data types:

- Feature Value information - defines the valid list of values for a given feature. An example of a feature value for a feature might be, operating voltage 110/240V AC.
- Feature Value Assignment information - describes the assignment of specifications or the values of specifications to a given item, or item category. An example of Feature Value Assignment information might be that a FL 856 hand held multi-meter may have an operating voltage of 110/240V AC.

Sync RFQ

Request for Quotation (RFGQ) is a document used to solicit supplier quotations for goods or services needed. Possible sources of RFQs would be Purchasing, Master Scheduling or MRP, Inventory, or Work in Process systems. Another scenario of which the Sync RFQ XML refers to is the integration of a buyer's business software components to supplier's business software components.

Sync RFQ XML document ensures that all business software components in a specific integration instance have the current Request for Quotation information. Sync RFQ XML may be used in an integration scenario between a Buyer's Purchasing system and a Seller's Order Management system. The workflow sequence for custom, built to order involves more activity than that of finished goods. The buyer's focus involves

finding the best price for an item or quantity of items that are not currently available. A long-term business relationship is often involved leading to expectations that must be addressed within the quote. Customized products require a higher level of specification. Individual supplier inquiries may be answered formally for all participants to insure competitive parity. Conversely, buyers may have questions related to an individual quote that lead to a change. Sync RFQ XML is used when the RFQ data persists in multiple systems.

Add RFQ

Request for Quotation (RFC) is a document used to solicit supplier quotations for goods or services needed. Possible sources of RFQs would be Purchasing, Master Scheduling or MRP, Inventory, or Work in Process systems. Another scenario of which the Add RFQ XML refers to is the integration of a buyer's business software components to supplier's business software components.

Add RFQ is an XML used to communicate from one business application to one or more other business applications that additional data related to a Request for Quotation has been added or needs to be added, depending on the business case. The environment for this XML can be within the enterprise or outside the enterprise. A Buyer's Purchasing System may use the Add RFQ XML to add a Request for Quotation to a Supplier's Order Management system.

The Add RFQ XML is used in pre-purchase activities for Quantities of Finished Items and Custom Build to Order Items. Add RFQ XML releases a request to one or more partners providing deadlines for response and item details. Add RFQ XML is used when the recipient takes ownership of the RFQ source data.

Change RFQ

Request for Quotation (RFC) is a document used to solicit supplier quotations for goods or services needed. Possible sources of RFQs would be Purchasing, Master Scheduling or MRP, Inventory, or Work in Process systems. Another scenario of which the Change RFQ XML refers to is the integration of a buyer's business software components to supplier's business software components.

A Change RFQ XML requests that another business application component make changes to an existing Request for Quotation. The environment for this XML can be within the enterprise or outside the enterprise. A request is modified to include answers to questions or clarification of specifications using a Change RFQ XML.

Cancel RFQ

Request for Quotation (RFC) is a document used to solicit supplier quotations for goods or services needed. Possible sources of RFQs would be Purchasing, Master Scheduling or MRP, Inventory, or Work in Process systems. Another scenario of which the Cancel RFQ XML refers to is the integration of a buyer's business software components to supplier's business software components.

The purpose of the Cancel RFQ XML is to publish to a business application or system the need to cancel an entire Request for Quotation or one or more of its line items. The environment for this XML can be within the enterprise or outside the enterprise. A Buyer organization's Purchasing system may send a Cancel RFQ XML to a Supplier organization's Order Management system to cancel an existing RFQ or lines of a RFQ.

Respond RFQ

Request for Quotation (RFC) is a document used to solicit supplier quotations for goods or services needed. Possible sources of RFQs would be Purchasing, Master Scheduling or MRP, Inventory, or Work in Process systems. Another scenario of which the Respond RFQ XML refers to is the integration of a buyer's business software components to supplier's business software components.

A Respond RFQ XML communicates from one business application to one or more other business applications that additional data related to a Request for Quotation is available. The environment for this XML can be within the enterprise or outside the enterprise. The Respond RFQ XML is typically involved in a Custom Build to Order workflow. The workflow sequence for custom, built to order involves more activity than that of finished goods. The buyer's focus involves finding the best price for an item or quantity of items that are not currently available. A long-term business relationship is often involved leading to expectations that must be addressed within the quote. Customized products require a higher level of specification. Individual supplier inquiries may be answered formally for all participants to insure competitive parity.

Conversely, buyers may have questions related to an individual quotation that lead to a change. Suppliers respond with questions and/or clarification issues using a Respond RFQ XML.

Getlist RFQ and List RFQ

Request for Quotation (RFC) is a document used to solicit supplier quotations for goods or services needed. Possible sources of RFQs would be Purchasing, Master Scheduling or MRP, Inventory, or Work in Process systems. Another scenario of which the Getlist RFQ XML refers to is the integration of a buyer's business software components to supplier's business software components.

Getlist RFQ is an XML that enables a business application module to request information containing summary information for one or more Request for Quotations. The response to this request is LIST RFQ. The environment for this XML can be within the enterprise or outside the enterprise. The Getlist RFQ also enables the retrieval of information across several documents by using selection fields. An example of this could be requesting all QUALFICATN Data Type occurrences for a specific RFQLINE. This type of functionality is limited to the capabilities of the responding application and needs to be determined during the implementation project. The typical scenario for use of this XML is that a Supplier or Buyer requests a list of RFQs.

The List is sent to the requestor by using a List RFQ XML. This XML publishes one or more summary listings of Request for Quotation information to other business application component. This may be in response to a Getlist RFQ request or to proactively publish a listing of summary Request for Quotation information for a business event. When a receiving application receives this XML, the information can be used as is or it may be used to initiate a selection of a specific Request for Quotation through the Get RFQ XML. The processing is designed to provide multiple occurrences of summary data.

Get RFQ and Show RFQ

Request for Quotation (RFC) is a document used to solicit supplier quotations for goods or services needed. Possible sources of RFQs would be Purchasing, Master Scheduling or MRP, Inventory or Work in Process systems. Another scenario of which the Get RFQ and Show RFQ XML refer to is the integration of a buyer's business software components to supplier's business software components.

Get RFQ XML document enables a business applications module to request Request for Quotation (RFQ) information from another business application. The response to this XML is a Show RFQ. The environment for this XML can be within the enterprise or outside the enterprise. A Supplier's Order Management system sends a Get RFQ XML to a Buyer's Purchasing system to request RFQ information. It may also be that intermediaries are used to manage RFQ exchange activities between a Buyer and a Seller. The workflow sequence for scenarios using intermediaries involves additional activity. An intermediary may be independent or an extension of a large organization. Much of the additional activity follows from the aggregation provided by an intermediary. Multiple RFQs could reside with the intermediary. The Get RFQ XML may be used to select one of the items returned by a List RFQ XML for review. The full RFQ is sent by using a Show RFQ XML.

The Show RFQ XML communicates to a business application module or system the sending systems representation of Request for Quotation information. This request may be used as a response to a Get RFQ XML or as a push notification of an event.

Sync Quote

A Quotation is a statement of the price, terms, and conditions of sale a supplier offers for an item or items. A quotation usually includes a detailed description (specifications) of goods or services the supplier offers. Suppliers consider quotations as an offer to sell when given in response to an inquiry. The integration scenario for the Sync Quote XML is the integration of a Buyer's business software components to Supplier's business software components.

A Sync Quote XML ensures that all business software components in a specific integration instance have the current Quotation information. This XML is commonly used to publish the need to create or update a Quotation in a publish and subscribe

integration environment. The scenario of which the Sync Quote XML refers to is the integration of a Buyer's business software components to Supplier's business software components.

Add Quote

A Quotation is a statement of the price, terms, and conditions of sale a supplier offers for an item or items. A quotation usually includes a detailed description (specifications) of goods or services the supplier offers. Suppliers consider quotations as an offer to sell when given in response to an inquiry. The integration scenario for the Add Quote XML is the integration of a Buyer's business software components to Supplier's business software components.

An Add Quote XML indicates a response to a supplier provided quotation or buyer question on a quotation. The purpose of the Add Quote XML is to communicate from one business application to one or more other business applications that additional data related to a quote has been added or needs to be added, depending on the business case. This XML commonly causes updates to occur. This XML may be used individually, or as part of a larger integration scenario. An example is the sending of an Add Quote XML from a Supplier's Order Management system to a Buyer's Purchasing system, or through an intermediary.

Change Quote

A Quotation is a statement of the price, terms, and conditions of sale a supplier offers for an item or items. A quotation usually includes a detailed description (specifications) of goods or services the supplier offers. Suppliers consider quotations as an offer to sell when given in response to an inquiry. A quotation often represents a binding document that may change based on negotiations between the buyer or RFQ "owner" and the seller responding to the RFQ. The integration scenario for the Change Quote XML is the integration of a Buyer's business software components to Supplier's business software components.

The purpose of the Change Quote XML is to request that another business application component make changes to an existing Quotation. The environment for this XML can be within the enterprise or outside the enterprise. The Change Quote XML is sent by a Supplier's Order Management system to a Buyer's Purchasing system, or through an intermediary.

The Change Quote XML is used in pre-purchase activities for Quantities of Finished Items and Custom Build to Order Items. Using a Change Quote XML a Supplier may alter a quotation during negotiations in an effort to win the order.

Cancel Quote

A Quotation is a statement of the price, terms, and conditions of sale a supplier offers for an item or items. A quotation usually includes a detailed description (specifications) of goods or services the supplier offers. Suppliers consider quotations as an offer to sell when given in response to an inquiry. A quotation often represents a binding document that may change based on negotiations between the buyer or RFQ "owner" and the seller responding to the RFQ. The integration scenario for the Cancel Quote XML is the integration of a Buyer's business software components to Supplier's business software components.

Cancel Quote XML is used to publish to a business application or system the need to cancel an entire quotation or one or more of its line items. The environment for this XML can be within the enterprise or outside the enterprise. This XML commonly causes updates to occur.

Respond Quote

A Quotation is a statement of the price, terms, and conditions of sale a supplier offers for an item or items. A quotation usually includes a detailed description (specifications) of goods or services the supplier offers. Suppliers consider quotations as an offer to sell when given in response to an inquiry. The integration scenario for the Respond Quote XML is the integration of a Buyer's business software components to Supplier's business software components.

Respond Quote XML indicates a response to a supplier-provided quotation or buyer question on a quotation. The purpose of the Respond Quote XML is to communicate from one business application to one or more other business applications that an additional data related to a quotation has been added or needs to be added, depending on the business case. This XML commonly causes updates to occur.

Getlist Quote and List Quote

A Quotation is a statement of the price, terms, and conditions of sale a supplier offers for an item or items. A quotation usually includes a detailed description (specifications) of goods or services the supplier offers. Suppliers consider quotations as an offer to sell when given in response to an inquiry. The integration scenario for the Getlist and List Quote XML is the integration of a Buyer's business software components to Supplier's business software components.

Getlist Quote XML enables a business application module to request information containing summary information for one or more quotations. The response to this request is LIST QUOTE. The environment for this XML can be within the enterprise or outside the enterprise. The Getlist Quote XML also enables the retrieval of information across several documents by using selection fields. An example of this could be requesting all SALESINFO Data Type occurrences for a specific QTELINE. This type of functionality is limited to the capabilities of the responding application and needs to be determined during an implementation project. This XML does not usually cause updates to occur. This XML may be used individually, or as part of a larger integration scenario. There are many possible business applications in several environments that may use this capability. An example is a Buyer's Purchasing system sending a Getlist Quote XML to a Supplier's Order Management system, or through an intermediary.

The purpose of the List Quote XML is to publish one or more summary listings of quotation information to other business application component. This may be in response to a Getlist Quote request or to proactively publish a listing of summary quotation information for a business event. The environment for this XML can be within the enterprise or outside the enterprise. When a receiving application receives this XML, the information can be used as is or it may be used to initiate a selection of a specific Quote through the GET QUOTE request. The processing is designed to provide multiple occurrences of summary data.

Get Quote and Show Quote

A Quotation is a statement of the price, terms, and conditions of sale a supplier offers for an item or items. A quotation usually includes a detailed description (specifications) of goods or services the supplier offers. Suppliers consider quotations as an offer to sell when given in response to an inquiry. The integration scenario for the Get and Show Quote XML is the integration of a Buyer's business software components to Supplier's business software components.

Get Quote XML enables a business applications module to request this quotation information from another business application. The response to this XML is the SHOW QUOTE. The environment for this XML: can be within the enterprise or outside the enterprise. This XML does not usually cause updates to occur.

This XML may be used individually, or as part of a larger integration scenario. There are many possible business applications in several environments that may use this capability. An example is a Buyer's Purchasing system sending a Get Quote XML to a Supplier's Order Management system, or through an intermediary.

The purpose of the Show Quote XML is to communicate to a business application module or system the sending systems representation of quotation information. This request may be used as a response to a GET QUOTE request or as a push notification of an event. The environment for this XML can be within the enterprise or outside the enterprise.

Show Shipment

Shipment is an XML business document that details the intent to transport a specific quantity of material goods from a supplier to a single customer business partner destination. The Shipment has been modeled after similar proprietary documents on popular business software packages (Oracle Applications' Delivery document, SAP's Delivery Note, and so on). The environment for this XML can be within the enterprise or outside the enterprise.

A Shipment is typically derived from the shipping schedule associated with a customer's purchase or sales order, once overall demand and various other business factors which prioritize the availability of the supplier's goods inventory have been evaluated. The Shipment document is designed to have a dynamic structure & content. Initial shipment planning information can be updated and significant detail (actual

picked inventory attributes, ship unit packaging, and so on) may be added during the execution phase of the supplier's order fulfillment and shipping business processes. The final form of the Shipment document provides detail about the carrier and level of service used to transport the material, the exact quantity and attributes of the material shipped, and how that material is physically packaged and identified for transport.

To aid the customer's planning and receiving business processes, the supplier may transmit the final Shipment document to customer in advance so that they can prepare for carrier arrival and then efficiently accept and utilize the ordered material. In this use case, the Shipment document may function as a traditional Advance Ship Notice (ASN).

This XML may be used individually, or as part of a larger interface scenario. For example, a Shipping Management System uses a Show Shipment XML to show shipment information to a Buyer’s Purchasing System where the receipt of goods are recorded.

Sync Planschd

PLANSCHD indicates a demand forecast sent from a customer to a supplier, or a supply schedule sent from a supplier to a customer. Sync Planschd XML communicates requirement information (part number, quantity, and so on) between a customer and their supplier on a regular basis, for example daily, weekly and so on, or for a user-defined time bucket type definition that is sent as part of this PLANSCHD.

Since collaboration between a customer and a supplier can potentially go through several rounds of negotiations, both parties can use the same XML to communicate their current demand or supply schedule in response to what they received from the other party. Either party can indicate detailed exception descriptions along with reason code and action code to the other party why the original demand or supply requirements need to be adjusted.

The Sync Planschd XML may be used individually, or as part of a larger interface scenario. An example would be the synchronization of plan schedules between a Buyer’s Scheduling application and the Supplier’s Release Management application. The Supplier Scheduling Application maintains schedules regarding planned demand, shipping and sequence of delivery. The Release management Application drives the MRP, the shipment and the sequence schedules in the Suppliers ERP application.

Get Planschd and Show Planschd

PLANSCHD indicates a demand forecast sent from a customer to a supplier, or a supply schedule sent from a supplier to a customer. The Get Planschd XML enables business applications module to request plan schedule information from another business application. The response to this XML is Show PLANSCHD.

Since collaboration between a customer and a supplier can potentially go through several rounds of negotiations, both parties can use the same XML to communicate their current demand or supply schedule in response to what they received from the other party. Either party can indicate detailed exception descriptions along with reason code and action code to the other party why the original demand or supply requirements need to be adjusted.

This XML may be used individually, or as part of a larger interface scenario. An example would be the Supplier's Release Management application sending a Get Planschd XML to a Buyer's Scheduling application. It could be that the supplier Management application quotes a planning schedule that does not exist in the Demand Management application causing the Demand Management Application to request the planning schedule. This interface scenario assumes that the Supplier Scheduling Application maintains schedules regarding planned demand, shipping and sequence of delivery. The Release management Application drives the MRP, the shipment and the sequence schedules in the Suppliers ERP application.

Show Planschd XML communicates to a business application module or system the sending systems representation of plan schedule information. This request may be used as a response to a Get Planschd XML or as a push notification of an event. This XML is used to publish planning schedule from a Supplier Scheduling application.

Sync Seqschd (Sequence Schedule)

Sync Seqschd XML enables the exchange of sequence schedule information authorizing a sequenced shipment of parts for specific trading partners and addresses. The environment for this XML can be within the enterprise or outside the enterprise. Commonly, the sequence schedule is generated by a work in process application and transmitted to an order or material planning application. This XML commonly causes updates to occur.

The Sync Seqschd XML may be used individually, or as part of a larger interface scenario. This event is the publication from the supplier scheduling application of a Shipment schedule to the buyer release management application. This interface scenario assumes that the Supplier Scheduling Application maintains schedules regarding planned demand, shipping and sequence of delivery. The Release management Application drives the MRP, the shipment and the sequence schedules in the Suppliers ERP application.

Get Seqschd (Sequence Schedule) and Show Seqschd (Sequence Schedule)

A Get Seqschd XML enables a Release Management application to request sequence schedule information from another business application such as a Supplier Scheduling application. The response to this XML is Show SEQSCHD. This XML does not usually cause updates to occur.

Commonly, the sequence schedule is generated by a work in process application and transmitted to an order or material planning application. The environment for this XML can be within the enterprise or outside the enterprise. The SCHEDULEID Field Identifier is used as a selection field. This is the only Field Identifier value that is used to send a Get Seqschd XML.

A Show Seqschd XML communicates to a business application module or system the sending systems representation of sequence schedule information. This request may be used as a response to a Get Seqschd request or as a push notification of an event. Information detailing general description of partners, including addresses and contacts, sequence schedule date and time information, list of items ordered along with the quantity, delivery date and other schedule information is data that may be returned with a Show Seqschd XML. The information obtained may be used by a Demand Management application to do demand forecasting.

Sync Shipschd (Shipment Schedule)

Sync Shipschd XML enables the exchange of shipment schedule information, authorizing a shipment quantity and date for specific trading partners and addresses. Commonly, the ship schedule is generated by a material planning application and

transmitted to an order management application. Shipment schedules associated with a customer's purchase or sales order is used to typically derive a shipment. This is completed once overall demand and various other business factors which prioritize the availability of the supplier's goods inventory have been evaluated.

The Sync Shipschd XML is used in an interface scenario between a Supplier Scheduling Application and a Release Management application. This interface scenario assumes that the Supplier Scheduling Application maintains schedules regarding planned demand, shipping and sequence of delivery. The Release management Application drives the MRP, the shipment and the sequence schedules in the Suppliers ERP application.

Get Shipschd (Shipment Schedule) and Show Shipschd (Shipment Schedule)

The Get ShipSchd XML enables a Seller's Release Management application to request shipment schedule information from a Buyer's Supplier Scheduling application. This interface scenario assumes that the Supplier Scheduling Application maintains schedules regarding planned demand, shipping and sequence of delivery. The Release management Application drives the MRP, the shipment and the sequence schedules in the Suppliers ERP application. The response to this XML is the SHOW SHIPSCHD.

The Show Shipschd XML communicates to a business application module or system the sending systems representation of SHIPSCHD information. This request may be used as a response to a GET SHIPSCHD request or as a push notification of an event. Information on the partner including the location and contacts, the items ordered along with the quantity, delivery date and other schedule information and line item exceptions based on predefined business rules or contract agreements may be returned using a Show Shipschd XML.

Process Invoice

Process Invoice is an XML that transmits an invoice from a supplier to a customer, thus involving an interface between a Supplier's Accounts Receivable application with a Customer's Accounts Payable application. Invoice in this interface scenario refers to either an item based customer invoice or as a general purpose customer credit or debit

memo. A customer invoice is a document that is created in Receivables that lists amounts owed for the purchases of goods or services. This document also lists any tax, freight charges and payment terms. The Process Invoice XML transmits invoice information such as line items, charges, allowances and tax, partner information including location and contacts, invoice payment terms and information about a business partner document reference (that is, customer's purchase order, supplier's sales order) for the invoice as a whole or for a particular invoice line.

Process WIPSPLIT

Process WIPSLIT (Work in Process Split) is an XML business service request that notifies a Manufacturing Application of the creation of multiple production lots from a single production lot of a product being made on a production order. The business environment most likely to require this capability is a lot-based discrete manufacturing scenario. The environment for this XML is normally within the enterprise. Process WIPSPLIT XML may be used in an integration that involves a manufacturing execution system (MES) and a central ERP system. The MES is the source of production events and activities and the ERP (with Order Management, Bill of Material, Cost Management and Inventory) is the source of related setup events which occur prior to production.

The Process WIPSPLIT XML communicates the originating lot, the resulting lots, their quantities, and the processing step at which this event occurred, along with the time at which this event occurred. This XML commonly causes updates to occur. Process WIPSPLIT XML may be used individually, or as part of a larger interface scenario. The workflow could start with the placement and synchronization of orders in the ERP system and the issuance of materials and charging of resources to the production order. The production order moves through the operations on the shop floor and throughout the production process, the order may experience a number of moves and splits or merges, and the creation of one or more bonus lots of material may possibly occur. The result of a WIP split may cause changes in the cost of the finished goods.

Process WIPMERGE

Process WIPMERGE (Work in Process Merge) is an XML business service request that notifies a Manufacturing Application of the creation of a single production lot from multiple production lots of a product being made on a production order. The business environment most likely to require this capability is a lot-based discrete manufacturing scenario. The environment for this XML is normally within the enterprise. Process WIPMERGE XML may be used in an integration that involves a manufacturing execution system (MES) and a central ERP system. The MES is the source of production events and activities and the ERP (with Order Management, Bill of Material, Cost Management and Inventory) is the source of related setup events which occur prior to production.

The Process WIPMERGE XML communicates the originating lots, the resulting lot, lot quantities, and the processing step at which this event occurred, along with the time at which this event occurred. This XML commonly causes updates to occur. This XML may be used individually, or as part of a larger interface scenario. The workflow could start with the placement and synchronization of orders in the ERP system and the issuance of materials and charging of resources to the production order. The production order moves through the operations on the shop floor and throughout the production process, the order may experience a number of moves and splits or merges, and the creation of one or more bonus lots of material may possibly occur. The result of a WIP merge may cause changes in the cost of the finished goods.

Process WIPRECOVER

Process WIPRECOVER is an XML business service request that notifies a Manufacturing Application of the creation of usable production materials from material previously considered unsuitable for production use. This is most often likely to represent a return to production of scrap material. The business environments most likely to require this capability are any type of manufacturing scenario (that is, discrete, process).

This XML communicates what is being recovered, the quantity being recovered, and the processing step at which the recovered material is to re-enter the production process, along with the time at which this event occurred. This XML commonly causes updates to occur. This XML may be used individually, or as part of a larger interface scenario. Process WIPRECOVER XML may be used in an integration that involves a

manufacturing execution system (MES) and a central ERP system. The MES is the source of production events and activities and the ERP (with Order Management, Bill of Material, Cost Management and Inventory) is the source of related setup events which occur prior to production.

Get WIPSTATUS and Show WIPSTATUS

Get WIPSTATUS is an XML business service request that enables a business application module to request information concerning the progress of a production order at a point in time from another business application. The business environments most likely to require this capability are any type of manufacturing scenario (that is, process, discrete) where business service requests for individual manufacturing transactions and events are not utilized. In a discrete manufacturing scenario, discrete job statuses describe the various stages in the life cycle of a job and control the activities that one can perform on it. Examples of Work in Process statuses are Complete, Released, Unreleased, On Hold, Cancelled or Close. The environment for this XML can be within the enterprise or outside the enterprise.

The Get WIPSTATUS XML communicates what quantities of end product reside at which processing steps along with the time this snapshot view was taken. The response to this XML is Show WIPSTATUS. This XML does not usually cause updates to occur. This XML may be used individually, or as part of a larger interface scenario. For example, a Work in Process system sends a Get WIPSTATUS XML to a Manufacturing Execution System to request the status or progress of a production order.

To respond to a Get WIPSTATUS XML, the Show WIPSTATUS XML is used. This Show WIPSTATUS XML notifies a Manufacturing Application of the progress of a production order at a point in time. The business environments most likely to require this capability are any type of manufacturing scenario (discrete, process) where business service requests for individual manufacturing transactions and events are not utilized. The environment for this XML can be within the enterprise or outside the enterprise. This XML communicates what quantities of end product reside at which processing steps along with the time this snapshot view was taken. This XML commonly causes updates to occur.

Process WIPMOVE

Process WIPMOVE (Work in Process Move) is an XML business service request that notifies a Manufacturing Application of the progression through the production processing steps or operations of a product being made on a production order. The business environments most likely to require this capability are any type of manufacturing scenario (that is, discrete, process). The environment for this XML is normally within the enterprise. Process WIPMOVE XML may be used in an integration that involves a manufacturing execution system (MES) and a central ERP system. The MES is the source of production events and activities and the ERP (with Order Management, Bill of Material, Cost Management and Inventory) is the source of related setup events which occur prior to production.

The Process WIPMOVE XML communicates which processing step the product is coming from and which step it is being moved to, along with the quantity moving and the time this event occurred. This XML assumes that the applications involved in this business scenario have already synchronized the production item and its BOM or Routing information. This XML commonly causes updates to occur.

Process WIPMOVE XML may be used individually, or as part of a larger interface scenario. The workflow could start with the placement and synchronization of orders in the ERP system and the issuance of materials and charging of resources to the production order. The production order moves through the operations on the shop floor and throughout the production process, the order may experience a number of moves and splits or merges, and the creation of one or more bonus lots of material may possibly occur. In an environment such as Oracle Manufacturing, move transactions moves assemblies within an operation, such as from Queue to Run, or from operation to the next. Move transactions can charge resources and overheads.

Allocate Resource

Allocate Resource is an XML that notifies a Manufacturing Application of the use of required labor or machine time on a production order making a product. The business environments most likely to require this capability are any type of manufacturing scenario (that is, discrete, process).

The Allocate Resource XML communicates what machine was utilized or which person performed the work and their labor skill class, along with the amount of time worked and at what time this event occurred. This XML commonly causes updates to

occur. This XML may be used individually, or as part of a larger interface scenario. For example, interface between a Manufacturing Execution System and a Cost Management system to make updates to the cost of a product with the allocated resources. Information as to the resource transaction, work order the resource is to be charged against, resources to be charged, the production order against which the resources are to be charged, the lot and/or serial number information on the production items, the operation at which resources are to be charged, and the resources (worker, machine and tools) required to perform the operation assists in allocating the right resources for the order.

Get Customer and Show Customer

Get Customer is an Way XML used to request customer information. This is likely between an Order Management system getting customer information from a customer management system. In Oracle Applications, customer records are maintained and managed in the Oracle Accounts Receivable system. Customer records include information on the customer (that is, customer name, payment method, partner type), customer's addresses and contacts. This XML does not cause updates to occur. An order in an Order Entry system can only be processed when customer records are in place and can be validated.

As a response to a GET Customer XML or a trigger by a business event, the Show Customer XML is used. The purpose of the Show Customer XML is to provide Customer information to a requesting business application, such as an Order Management system. This XML does not usually cause updates to occur.

Get Supplier and Show Supplier

Get Supplier is an Way XML used to request supplier information. This is likely between an Order Management system getting supplier information from a supplier management system. In Oracle Applications, supplier records are maintained and managed in the Oracle Purchasing system. Supplier records include information on the supplier (that is, supplier ID, currency, payment method, partner type), supplier's addresses and contacts. This XML does not cause updates to occur.

The purpose of the Show Supplier XML is to provide Supplier information to a requesting business application, such as Order Management system. This XML may be in response to a Get Supplier XML, or it may be triggered by a business event. This XML does not usually cause updates to occur.

Sync Ecatalog

Sync ECATALOG is an XML that synchronizes catalog information between two systems. The environment for this XML can be within the enterprise or outside the enterprise, including trading partners.

There are many possible business applications in several environments that may use this capability. Some examples of usage scenarios are a Manufacturer synchronizing catalogs with distributors, suppliers, or e-marketplaces and distributors, suppliers, or e-marketplaces synchronizing catalogs with buyers or other trading partners. It may also be necessary to support Component Supplier Management (CSM) scenarios. In this scenario, a company provides a service of sourcing and codifying the products of many companies and publishing a consolidated catalog.

The existing definition does not attempt to model configuration rules, and is therefore not sufficient to model configure to order environments that require cross option validation. Assemble to order environments that can be modeled using a marketing bill of material is addressed. The Catalog exchange scenario can be implemented either as a simple scenario using a single XML, or in the case of large catalogs involving complex pricing scenarios or partner specific details, as multiple XMLs.

Get ECATALOG and Show ECATALOG

Get ECATALOG, is an XML that enables a business application module or system to request catalog information. The environment for this XML can be within the enterprise or outside the enterprise, including trading partners. The Catalog information that is requested by the Get CATALOG XML may include item identifiers, specifications, pricing information agreed on, price lists and availability and delivery information.

There are many possible business applications in several environments that may use this capability. Some examples of usage scenarios are Manufacturer exchanging catalogs with distributors, suppliers, or e-marketplaces and distributors, suppliers or

e-marketplaces exchanging catalogs with buyers or other trading partners. It may also be necessary to support Component Supplier Management (CSM) scenarios. In this scenario, a company provides a service of sourcing and codifying the products of many companies and publishing a consolidated catalog. The Catalog exchange scenario can be implemented either as a simple scenario using a single XML, or in the case of large catalogs involving complex pricing scenarios or partner specific details, as multiple XMLs.

The purpose of the Show ECATALOG XML is to supply a business application module such as a Catalog Management System with requested catalog information. The environment for this XML can be within the enterprise or outside the enterprise, including trading partners. A Catalog Publisher should be able to publish catalog information to a customer or agent on information such as description of the product or service, party specific part identifiers, pricing information agreed on and other information depending on the needs of the subscriber. A Catalog Searcher should be able to request catalog information for a part or service identifier including description of product or service, part specific part numbers and other relevant information to the catalog.

Sync Invoice

Sync Invoice is an XML that synchronizes or transmits an invoice from a supplier to a customer, thus involving an interface between a Supplier's Billing system with a Customer's Accounts Payable application. Invoice in this interface scenario refers to either an item-based customer invoice or as a general purpose customer credit or debit memo. A customer invoice is a document that is created in Receivables that lists amounts owed for the purchases of goods or services. This document also lists any tax, freight charges and payment terms. The Sync Invoice XML synchronizes invoice information such as line items, charges, allowances and tax, partner information including location and contacts, invoice payment terms and information about a business partner document reference (that is, customer's purchase order, supplier's sales order) for the invoice as a whole or for a particular invoice line.

Get Invoice and Show Invoice

A Get Invoice XML enables a request of a customer invoice. This XML may be used as a request by a Customer to its Supplier. The interface would involve a Customer's Accounts Payable system requesting for invoice information from a Supplier's billing application. This allows the customer to get details of the invoice for processing and payment. A billing invoice may include information such as line items, charges, allowances and tax, partner information including location and contacts, invoice payment terms and information about a business partner document reference.

A response from the Supplier's billing system is to be carried through a Show Invoice XML. A Show Invoice XML transmits an invoice from a supplier to a customer. This XML may be used as a response to a Get Invoice request or as a push notification of an event.

Get Consumption and Show Consumption

The most common use of the Get Consumptn XML is to request a buyer's usage information about an item or product for the supplier of such item or product. This XML does not create or update either buyer's or supplier's inventory records. The receiver of the request is responsible to make effective use of this information. The Get Consumption XML may be used for a supplier of goods to request from the buyer the consumption status of goods. This XML may also be used for a vendor to request from the retailer if retail sales of goods have been made and for inventory systems to request consumption status from plant data collection and warehouse management systems.

The most common use of the Show Consumtn XML is to share a buyer's usage information about an item or product with the supplier of such item or product. This XML does not create or update either buyer's or supplier's inventory records. The receiver of the request is responsible to make effective use of this information.

This is an outline of the business flow that these XML supports:

1. Overall purchase, replenishment or vendor managed inventory agreement is in place and/or a Get Consumptn message is sent by the supplier.

2. Show Consumptn Message is returned to the supplier, distributor or third party logistics provider, that material has been consumed. This is done in response to events such as these (and/or the Get message), depending on implementation context:
3. Material is replenished to line side at manufacturing facility.
4. Material is assembled into final product.
5. Material is purchased and removed from facility by customer.
6. Supplier, distributor, third party logistics provider replenishes material, using information provided in the Show Consumptn message, the demand and shipment forecasts, and the terms of the overall purchase or vendor managed inventory agreement.

Create Requisitn

Create Requisitn is an XML request that notifies another business application of the need to order parts in a specific quantity, for a specific need by date. This XML usually causes updates to occur and may be used as part of a larger integration scenario or as a single tool for communicating demand. A Material Resource Planning (MRP) or an Inventory system may send a Create Requistn XML to a Purchasing System that handles purchase requisitions and orders maybe because reorder points indicate a need to request materials for production. Requisitions signify demand for goods and services to another business application for consideration of buying or in some way obtaining requested items. In Oracle, this facilitates the sending of a purchase requisition to trigger the issuance of a purchase order for items requested.

Getlist LDGRACTUAL and List LDGRACTUAL

A **Getlist LDGRACTUAL** is an XML that requests information containing summary information for one or more ledgers. Many applications in the enterprise environment create data that causes changes in the account balances of a general ledger application (that is, Benefits, Costing, Human Resources and Payroll, Inventory, Manufacturing, Production, Treasury). The integration scenario for this XML is for enterprise applications business software components to obtain and receive accounting data from the general ledger business software component. The request or response style of

messaging which the enterprise application sends a request message and expects to receive a response message back. The general ledger application receives the request message and produces the response message that is sent back to the enterprise application. The general ledger application uses information in the request message to know how to send the response message back to the enterprise application. This scenario assumes that the general ledger component "owns" the chart of accounts definition and the instances of data within it.

A List LDGRACTUAL XML is used to publish one or more summary listings of ledger information. This may be in response to a Getlist LDGRACTUAL request or to published proactively as a listing of summary ledger information for a business event. This ensures that subsidiary ledgers which keep details of the financial transactions are reflected correctly in summary in the general ledger application. Ledger actuals may also be used in the analysis of accounts

Get LDGRACTUAL and Show LDGRACTUAL

Get LDGRACTUAL XML enables an enterprise application to request actual detailed accounting data from a general ledger system. General Ledger is a central repository of accounting data that is summarized into account balances for analysis and reporting in the books. Each subsidiary ledger usually sends journal data (usually in summary) to the general ledger. The general ledger system then posts this details to the appropriate accounts and books. The Get LDGRACTUAL XML is used to request specific ledger information.

The purpose of the Show LDGRACTUAL XML is to communicate to an enterprise application (subsidiary ledgers) the sending systems representation of ledger information specifically requested. This may be used as a to a Get LDGRACTUALS XML or as a push notification of an event (updated general ledger data). This XML may be used individually or as part of a larger interface scenario. This XML uses a Data Type called LDGRACTUAL, which represents any of the general ledger elements corresponding to the Account Structure hierarchy that can be queried.

Acknowledge Delivery

A DELIVERY is an XML business document that details the receipt of goods by a customer or consignee destination from a supplier business partner using the services of a transportation provider (carrier). The DELIVERY document has been modeled for use in conjunction with the SHIPMENT XML when issued by the shipping business partner as an Advance Shipment Notification. The environment for this XML can be within the enterprise or outside the enterprise. DELIVERY was designed to support the physical process of receiving of goods from a carrier and is not biased toward receiving against a particular purchase, transfer or sales order reference document. This feature enables the receiving of goods simultaneously across multiple orders that were consolidated for shipping efficiency.

The Acknowledge Delivery XML may be used individually, or as part of a larger integration scenario. The ACKNOWLEDGE DELIVERY document may be used to notify the shipping business partner that the shipment has been received by the customer or consignee destination, and alert them to any discovered discrepancies. The acknowledgement may contain the full detail of the receipt as created by the receiving party or just the discrepancies and other exception conditions. This XML supports receipt acknowledgements at either the line item level and/or the ship unit level. Intermediate transportation or logistics providers or freight forwarding partners can use this document to acknowledge the receipt of entire shipping units without detailing the corresponding contents.

Receive Delivery

A DELIVERY is an XML business document that details the receipt of goods by a customer or consignee destination from a supplier business partner using the services of a transportation provider (carrier). The DELIVERY document has been modeled for use in conjunction with the SHIPMENT XML when issued by the shipping business partner as an Advance Shipment Notification. The environment for this XML can be within the enterprise or outside the enterprise. DELIVERY was designed to support the physical process of receiving of goods from a carrier and is not biased toward receiving against a particular purchase, transfer or sales order reference document. This feature enables the receiving of goods simultaneously across multiple orders that were consolidated for shipping efficiency.

The Receive Delivery XML may be used individually, or as part of a larger integration scenario. The RECEIVE DELIVERY document may be used to update the receiver's internal receiving and order management business applications to indicate that the requested material has arrived, including any unexpected quantity, condition or other exception discrepancies. This XML supports receiving at either the line item level and/or the ship unit level. Intermediate transportation or logistics providers or freight forwarding partners can use this document to acknowledge the receipt of entire shipping units without detailing the corresponding contents.

Get Delivery and Show Delivery

A DELIVERY is an XML business document that details the receipt of goods by a customer or consignee destination from a supplier business partner using the services of a transportation provider (carrier). The DELIVERY document has been modeled for use in conjunction with the SHIPMENT XML when issued by the shipping business partner as an Advance Shipment Notification. The environment for this XML can be within the enterprise or outside the enterprise. DELIVERY was designed to support the physical process of receiving of goods from a carrier and is not biased toward receiving against a particular purchase, transfer or sales order reference document. This feature enables the receiving of goods simultaneously across multiple orders that were consolidated for shipping efficiency.

The Get Delivery XML may be used individually, or as part of a larger integration scenario. For example, a Warehouse Management system may send a Get Delivery XML to an Order Management system. The GET DELIVERY XML may be used to request information about a specific expected (unreceived) or previously received goods delivery.

The Show Delivery is an XML document that may be used to obtain information about a specific expected (unreceived) or previously received goods delivery. The SHOW DELIVERY XML supports describing shipment content at either the line item level and/or the ship unit level. Intermediate transportation or logistics providers or freight forwarding partners can use this document to acknowledge the receipt of entire shipping units without detailing the corresponding contents. The Show Delivery XML may be issued in response to a Get Delivery request, or emitted asynchronously for notification upon some business event.

For expected deliveries, the SHOW DELIVERY document content may act as a receiving template or checklist to identify the quantity and shipping configuration of the expected goods. In this manner the SHOW DELIVERY XML may be considered

logically equivalent to the Advance Ship Notice information in a SHOW SHIPMENT document. This similarity is by design, as the receiver's SHOW DELIVERY may be directly derived from shipper's SHOW SHIPMENT content after the receiver's business logic has appropriately validated the Advance Ship Notification information.

Getlist Delivery and List Delivery

A DELIVERY is an XML business document that details the receipt of goods by a customer or consignee destination from a supplier business partner using the services of a transportation provider (carrier). The DELIVERY document has been modeled for use in conjunction with the SHIPMENT XML when issued by the shipping business partner as an Advance Shipment Notification. The environment for this XML can be within the enterprise or outside the enterprise. DELIVERY was designed to support the physical process of receiving of goods from a carrier and is not biased toward receiving against a particular purchase, transfer or sales order reference document. This feature enables the receiving of goods simultaneously across multiple orders that were consolidated for shipping efficiency.

The GETLIST DELIVERY is an XML that is used to request information about a set of expected (unreceived) or previously received goods deliveries meeting certain selection criteria. The response to the GETLIST DELIVERY request is LIST DELIVERY XML. The GetList Delivery XML may be used individually, or as part of a larger integration scenario. For example, a Warehouse Management system may send a Getlist Delivery XML to an Order Management system to request information on previously received goods deliveries.

The LIST DELIVERY is an XML document used to obtain limited information listing about an expected (unreceived) or previously received goods deliveries that match certain selection criteria in a Getlist Delivery request. Additional information about a specific DELIVERY may be obtained through a Show Delivery XML by using the listing information to populate a GET DELIVERY request. The XML provides general information about the delivery receipt document, the partner including the address and contacts, information on the item inventory being delivered, information on the particular shipping unit being received, and specific quantity of goods contained within a shipping unit. The List Delivery XML returns specific data that matches the selection criteria provided in a Getlist Delivery XML.