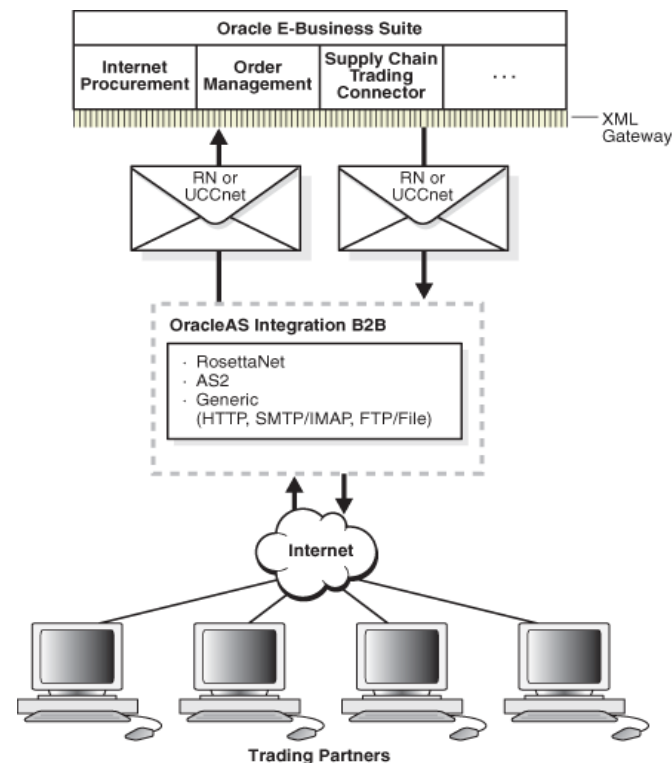


Oracle XML Gateway – A Key Message Broker for Oracle E-Business Suite

August 21, 2009 12:30 PM – 1:10 PM



Presented By

Shibu Kunneth

skunneth@astcorporation.com

Understanding XML, DTD

XML Gateway – What, Why and Architecture

XML Gateway – Components

XML Gateway – Message Designer

XML Gateway – Execution Engine

XML Gateway – Setups

Questions and Answers

References

■ XML

- ❖ XML stands for Extensible Markup Language . It was created to structure, store, and transport information. XML structures are not predefined. You must define it using DTD's.
- ❖ Sample XML would look like

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<PurchaseOrderMessage version="2.0">
  <Header>
    <MessageId>125691</MessageId>
    <Timestamp>2008-03-06 10:39:31</Timestamp>
    <Authentication>
      <Identity>XXX</Identity>
      <SharedSecret>XXXsecret</SharedSecret>
    </Authentication>
  </Header>
</PurchaseOrderMessage>
```

■ XML Standards

- ❖ Some of the leading standards in industry are RosettaNet, Open Applications Group (OAG), SOAP, iFX, cXML - commerce XML, OASIS / XML.Org- the Organization for the Advancement of Structured Information etc . Each has got their specific published Document Type Definitions (DTDs).
- ❖ Oracle has used OAG for mapping majority of the messages delivered with the Oracle E-Business Suite.

■ DTD

- ❖ A Document Type Definition (DTD) defines the legal building blocks of an XML document. It defines the document structure with a list of legal elements and attributes.
- ❖ A DTD can be declared inline inside an XML document, or as an external reference.
- ❖ Building blocks for DTD are Elements, Attributes, Entities, PCDATA ,CDATA
- ❖ Example of DTD

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PurchaseOrderMessage [
<!ELEMENT AccountCode (#PCDATA)>
<!ELEMENT AccountingDate (#PCDATA)>
<!ELEMENT Action (Code, Group, Description)>
<!ELEMENT Address (TemplateName, AddressCode?, Contact*, AddressLine+, City, State?,
    PostalCode, Country)>
<!ELEMENT AddressCode (#PCDATA)>
<!ELEMENT AddressLine (#PCDATA)>
<!ATTLIST AddressLine
    linenummer CDATA #REQUIRED
    label CDATA #IMPLIED
>
<!ELEMENT AreaCode (#PCDATA)>
]>
    
```

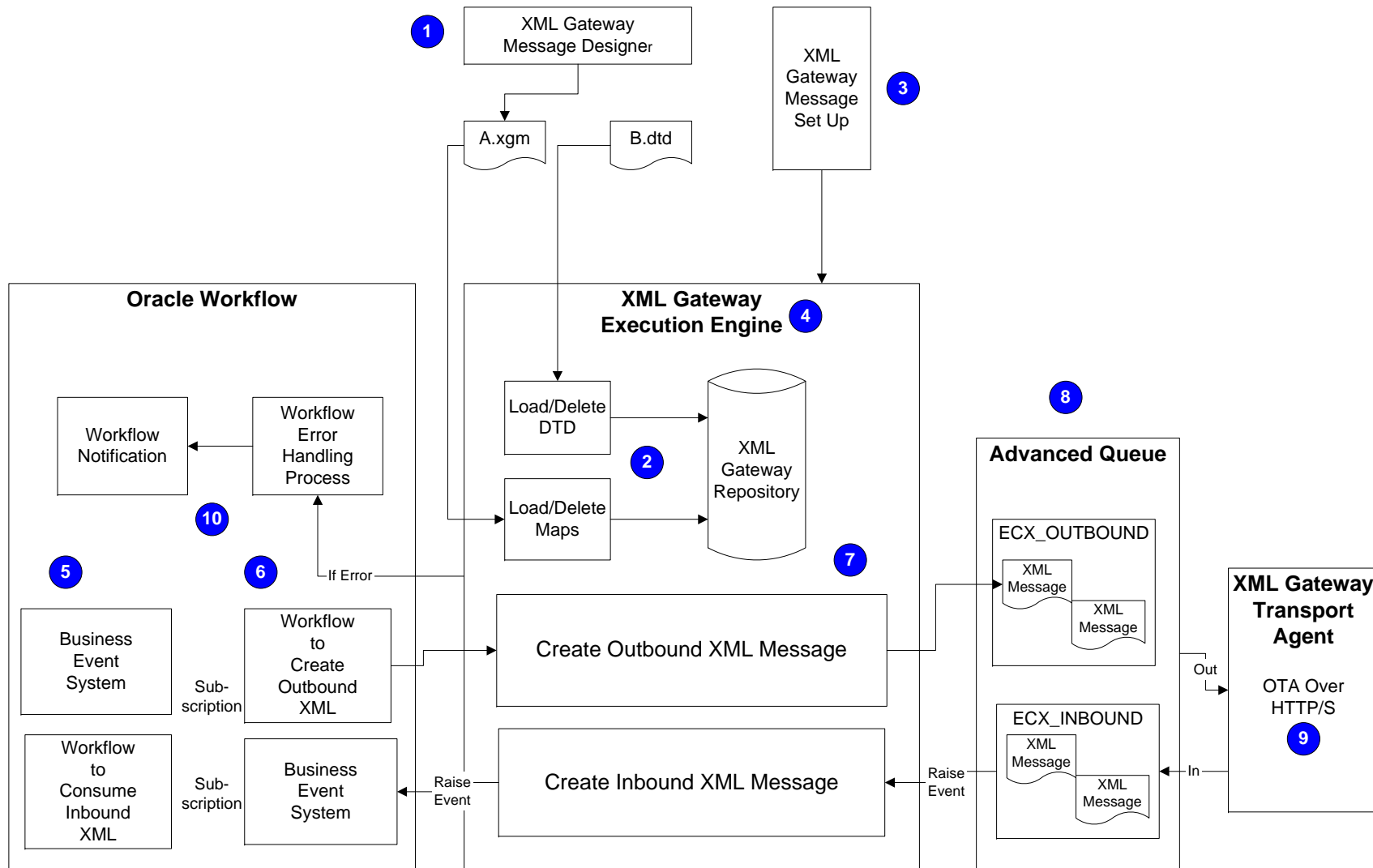
■ What is XML Gateway

- ❖ Oracle XML Gateway is a set of services that assures consistent bidirectional XML messaging implementation when integrating with the Oracle E-Business Suite, thereby enabling seamless collaboration, coordination, and communication of business-critical data. In short it's a key message broker for the Oracle E-Business Suite.
- ❖ It consume XML messages triggered by business events and acts as the agent to send information as an outbound transaction or to receive the information as an inbound transaction between Oracle E-Business Suite and external entities set up as Trading Partners in Oracle.
- ❖ Oracle E-Business Suite uses the capability of the Oracle Workflow Business Event System to publish and subscribe event-based XML message creation and consumption. Oracle XML Gateway consumes events raised by the Oracle E-Business Suite and integrates with Oracle Advanced Queuing to enqueue/dequeue a message and Oracle Transport Agent to deliver messages to business partners.

■ Why Use XML Gateway

- ❖ Oracle XML Gateway is real-time, event based and tend to be based on a single transaction thereby assuring a consistent XML message, lowering integration costs and expediting implementation.
- ❖ Oracle XML Gateway supports all DTD based XML standards.
- ❖ It provides a wizard-guided, GUI-oriented, repository-based Message Designer to define data source and targets, create hierarchy and element maps plus define actions for data transformation and process control.
- ❖ Has a very robust execution engine integrated with the Oracle e-Business Suite to create and consume XML messages based on a business event.
- ❖ Tightly integrated with Oracle Advanced Queuing to en-queue/de-queue outbound or inbound XML messages.
- ❖ Doesn't need a translator as the messages are created or consumed based on the message map (associated with the trading partner) stored in the XML Gateway repository.
- ❖ Capable of sending notification via Oracle Workflow to report errors detected by the XML Gateway Execution Engine, Oracle Advanced Queuing or Oracle Transport Agent.

Overview of the flow of an XML message through XML Gateway



■ Message Designer

- ❖ The XML Gateway Message Designer is a wizard-guided, repository-based tool used to define message maps. A message map represents the relationship between the source and target data elements. We use the XML Gateway Message Designer to perform the following functions.
 - Define Data Source and Data Target -- Each message map consists of a data source and data target representing where the data is coming from and where it is mapped to. The different scenarios could be Source RDBMS to Target XML-based on DTD , Source RDBMS to Target RDBMS, Source XML-based on DTD to Target XML-based on DTD , Source XML-based on DTD to Target RDBMS.
 - Map Source Data Structure to Target Data Structure – Here we relate the source data structure to the target data structure. This process is especially important if the data structure of the application data model is different from the data structure of the business document. XML Gateway Message Designer supports both expanding and collapsing hierarchies to ensure that data can be retrieved from or populated into the Oracle E-Business Suite data model . For example a business document like a PO based on CXML is represented in two levels, PO Headers and PO Lines. However, the application data model is represented in three levels - PO Headers, PO Lines and PO Distributions, so the data in the business document must be collapsed to accommodate the application data model.
 - Map Source Data Element to Target Data Element – Here we map the source data elements to the target data elements. The Message Designer user interface displays the data source and the data target in adjacent windows. A simple drag and drop between the source and target data elements creates a map relationship. The source data element name is noted next to the target data element name to identify the map relationship.

- Identify Data Transformation and Process Control Functions – Message Designer is used to identify data transformation and process control functions. These functions can be defined at the source or target . They can be set to be applied at the data element, document, root level or to be applied before, during, or after the message is created or consumed. The common process control functions involve calling procedures or database functions to extend the integration with the Oracle E-Business Suite. Other common process control functions allow you to inquire on the status of a transaction and manage the process flow based on the status. For serious errors, the process may be aborted with error notifications sent to the XML Gateway system administrator or Trading Partner contact.

■ XML Gateway Setup

- ❖ To implement a XML message with a trading partner/Supplier, we use the XML Gateway responsibility to perform the following
 - Define Trading Partner -- We set up each of the entity/entity site that we want to transmit the data via XML as a trading partner. Included in the Trading Partner definition are the following
 - Define Trading Partner name
 - Enable XML messages for the partner
 - Enable request for message confirmation
 - Define the message map to use for message creation
 - Define the e-mail address of the trading partner contact to notify for data errors
 - Define trading partner-specific code conversion values
 - Define transport protocol: SMTP, HTTP, HTTPS with credential and username and password as necessary

- Define Code Conversion -- We can cross-reference Oracle codes to codes that are meaningful to your recipient.
- Define Transactions – We can define a cross-reference between the Oracle transaction name and the external transaction name. The external transaction name will be based on what is meaningful per the XML standard used by the recipient. The external transaction name will appear on the message envelope to support message transport.

■ XML Gateway Execution Engine

- ❖ The XML Gateway Execution Engine is responsible for interacting with several Oracle technologies to process and transport XML messages to and from Trading Partners. The Oracle technologies involved include the following
 - Oracle E-Business Suite
 - Oracle Workflow Business Event System
 - Oracle Advanced Queuing
 - Oracle Transport Agent

■ Business Event System

- ❖ Oracle Workflow Business Event System publishes and subscribes to application business events to automatically trigger message creation or consumption. Business events and event subscriptions to send outbound messages that are pre-built within Oracle E-Business Suite application modules are delivered with the calls in the application module to raise a business event to indicate when something of interest has occurred. This includes event points indicating when a document was created, changed, confirmed or deleted. The XML Gateway Execution Engine interfaces with the Oracle E-Business Suite via business events and event subscriptions to retrieve data from Oracle e-Business Suite tables. For example when a PO document is approved an event will be raised to create XML message and then an event to send the XML document to the desired Trading Partner. The events that are enabled and have event subscriptions for the XML PO creation and transmittal are
 - oracle.apps.po.event.setxml -> This event is raised when a user goes to the PO Approvals form as the XML option is already selected.
 - oracle.apps.po.event.xmlpo -> This event is raised if a trading partner is enabled for XML PO transmittal.

■ Workflow to Create XML

- ❖ The delivered Oracle XML Gateway Standard workflow does the task of creating the XML based on Transaction Delivery Required Function used to determine if the Trading Partner is enabled for the transaction. If the Trading Partner is enabled for the transaction then the workflow's Create Trading Partner Message process kicks in which in turn calls the Generate Trading Partner XML Document function and Send Document function.
- ❖ The Generate Trading Partner XML Document function retrieves data from the Oracle E-Business Suite. The data for the XML message retrieved from the Oracle E-Business Suite is stored in the Event Message attribute. The Event Message is then processed according to the subsequent Workflow instruction which is to send to the WF_OUT agent. The Send Document function triggers outbound message creation. The XML message is created immediately as Send Mode attribute value is of "Immediate". This function also enqueue the XML message.

■ Advanced Queue

- ❖ Queues are tables on a database that are managed by Oracle Advanced Queuing.
- ❖ The XML Gateway uses queues specifically at two points in the process, as well as employing a general error queue. The first point is at the transport agent level between the transport agent module and the XML Gateway. The second point is at the transaction level between base Oracle E-Business Suite products and the XML Gateway.

■ Transport Agent

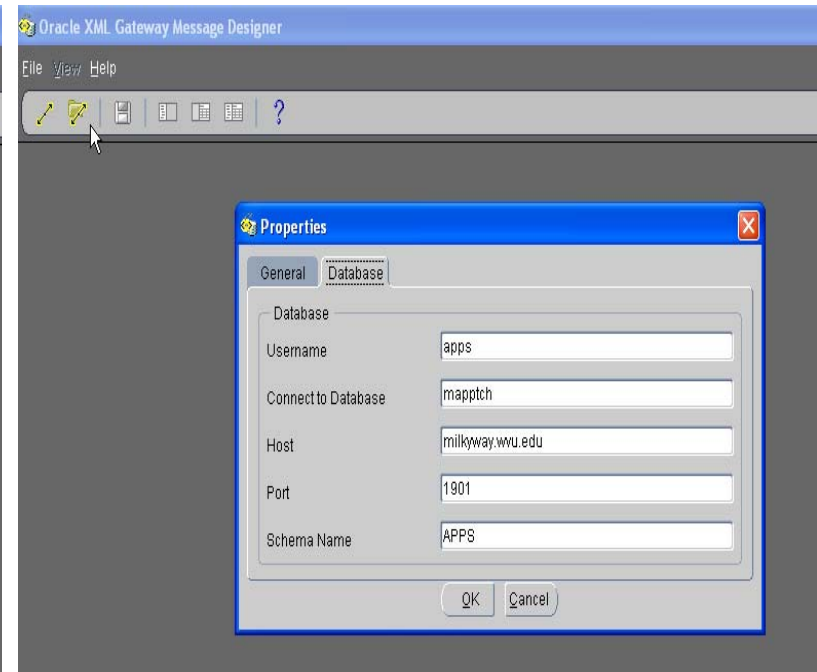
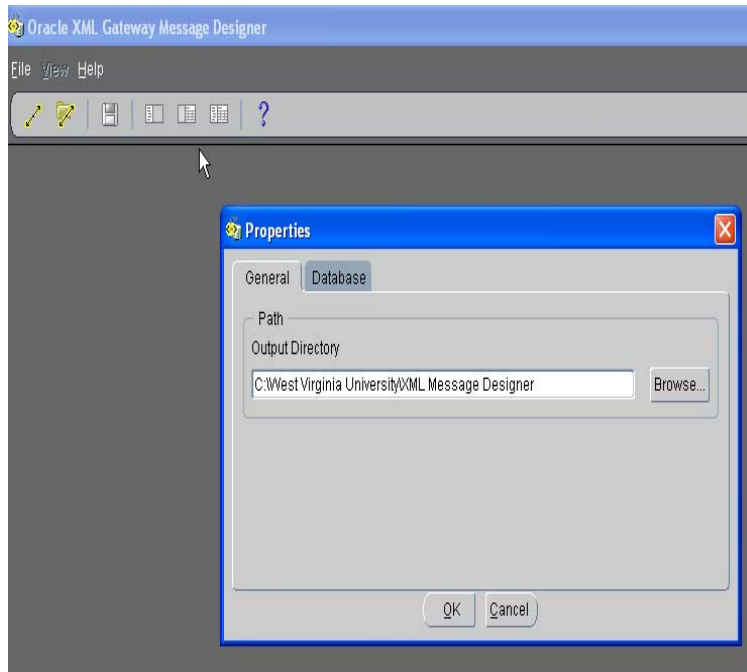
- ❖ Oracle Transport Agent (OTA) is a lightweight messaging platform for transmitting documents over HTTP and Secure HTTP (HTTPS).
- ❖ OTA implements a messaging protocol on top of the HTTP Application protocol. The OTA server is a Java-based servlet that uses the OTA messaging protocol to support the following requirements
 - Guaranteed, exactly-once delivery of a message over HTTP(S)
 - Complete audit and history tracking of messages sent and received
 - Outbound e-mail delivery of messages (SMTP)
 - Server certificate authentication (when using SSL mode)
 - Client certificate authentication
 - Built-in Application user authentication to Oracle e-Business Suite

■ Workflow Error

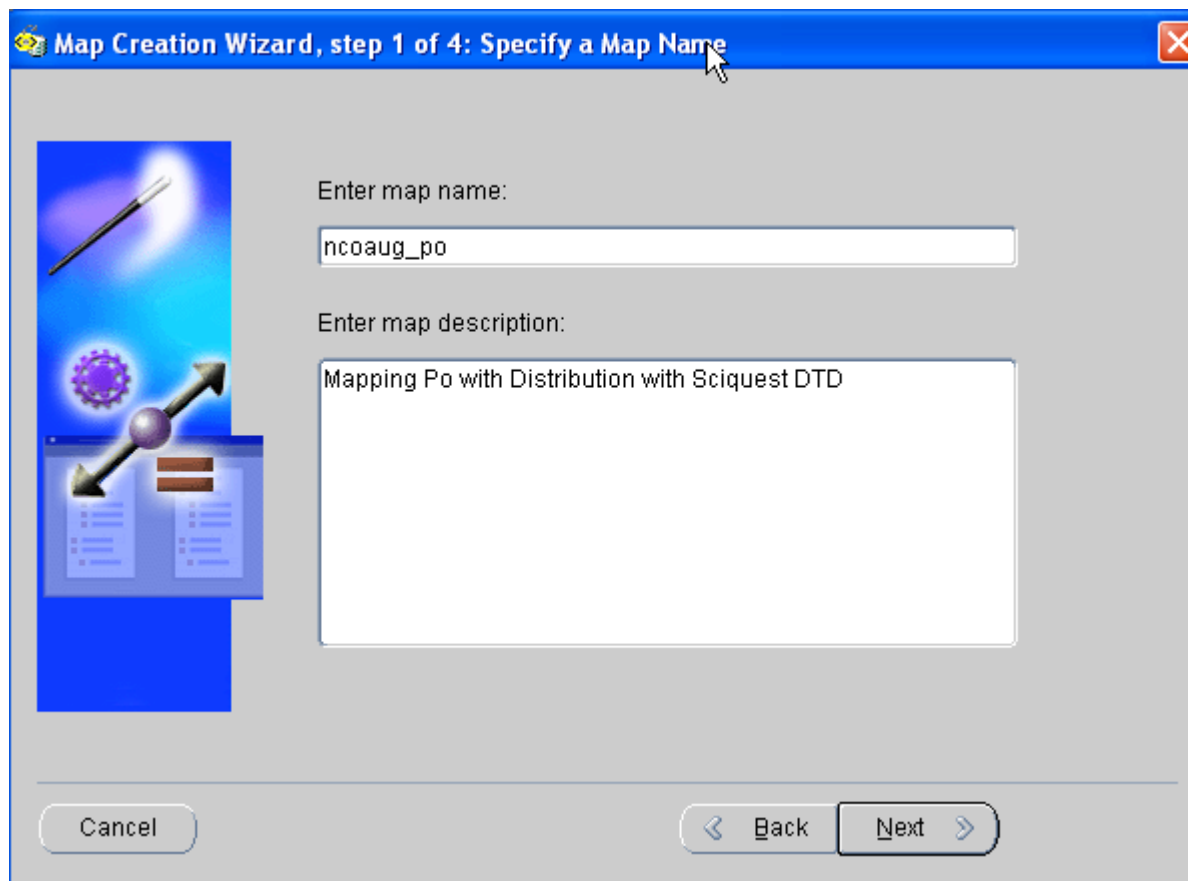
- ❖ The XML Gateway Error workflow contains error handling processes to manage errors detected by Oracle Workflow Business Event System or Oracle XML Gateway.
- ❖ Oracle Workflow sends a notification to the Trading Partner contact for data errors or to the XML Gateway system administrator using standard workflow error handling procedures for system or process errors. For errors that require collaboration between the Trading Partner contact and the XML Gateway system administrator, a notification is sent to both parties to encourage discussion and to expedite problem resolution.

■ Step 1 -- Set up the properties

- ❖ On the General tab, you specify the Output Directory. This is the default directory used to store the data definition and message map files created using the Message Designer. On the Database tab, provide the default database connection information. The default values will be provided to the Data Definition and Map Creation Wizards as well as to the Procedure Call action.



- Step 2 - Creating Transaction Map



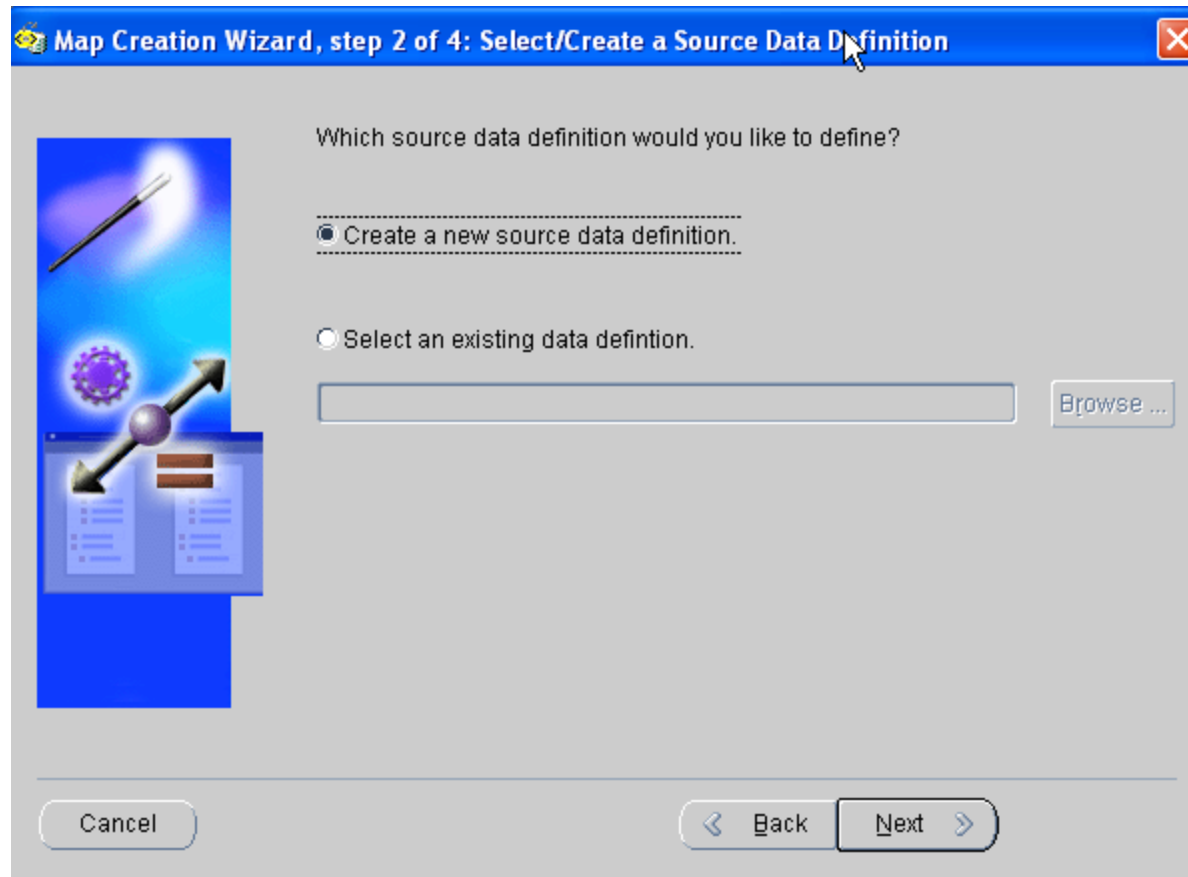
Map Creation Wizard, step 1 of 4: Specify a Map Name

Enter map name:

Enter map description:



Buttons: Cancel, < Back, Next >

■ Step 3 -- Creating Source Data Definition



- Step 4 -- Source Data Definition Name

Map Creation Wizard, step 3 of 5: Specify Source Data Definition Name and Type

Enter data definition name:

Enter data definition description:

Select data definition type:

Cancel < Back Next >

■ Step 5 -- Connection to Database



Map Creation Wizard, step 4 of 8: Specify Source Database Information

User Name:

Password:

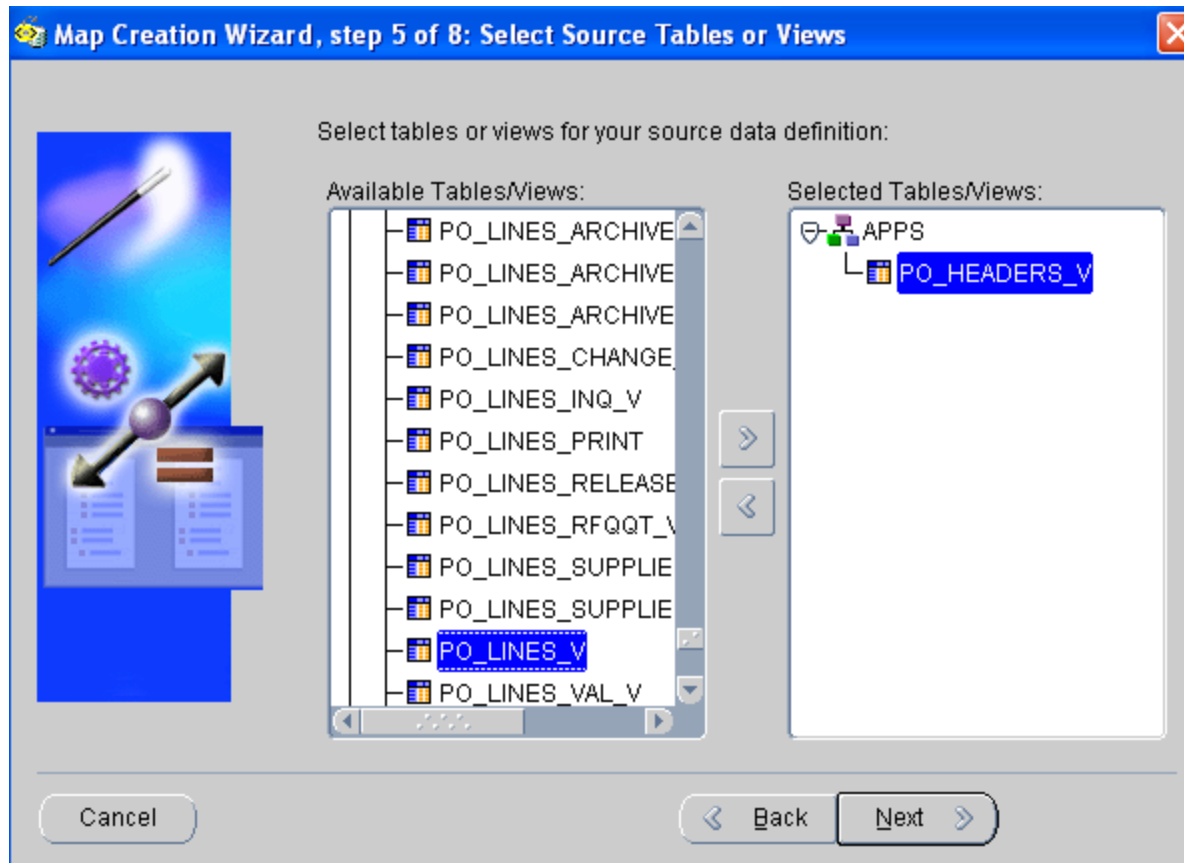
Connect String:

Host:

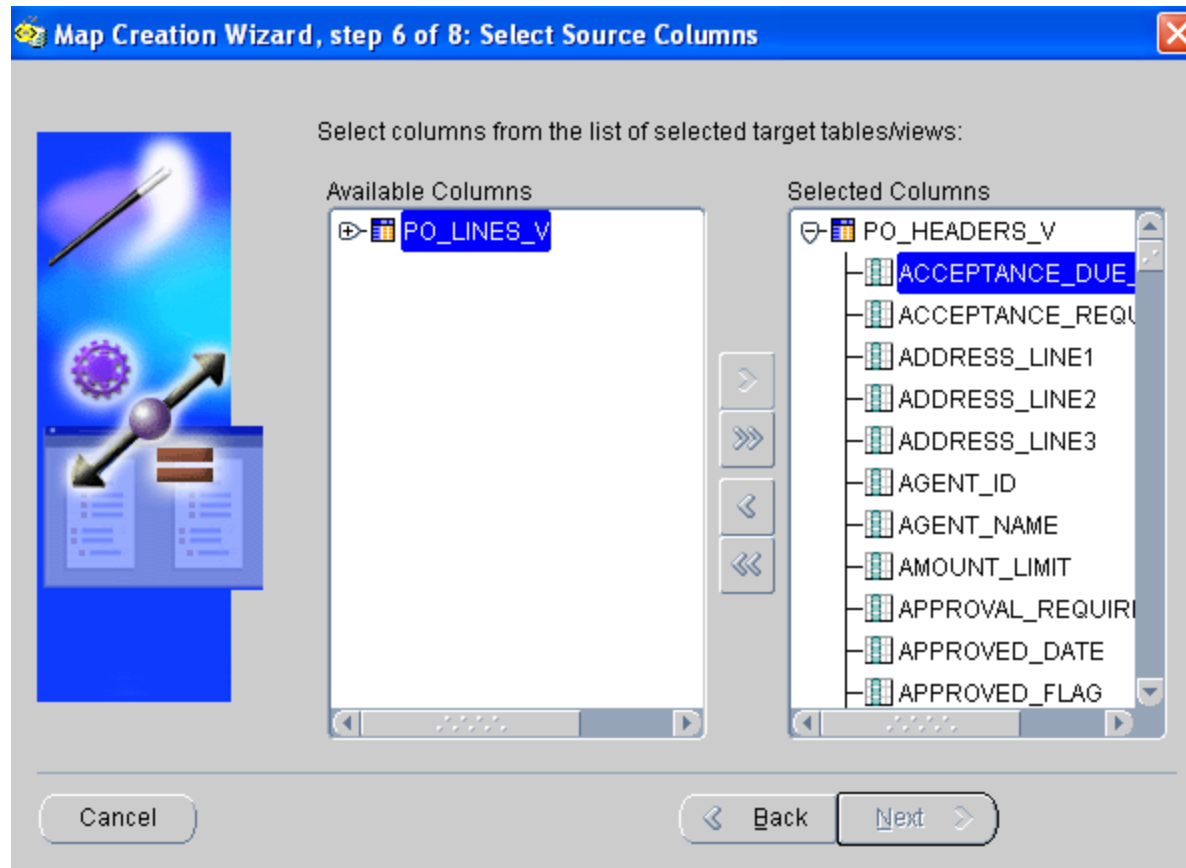
Port:

Cancel < Back Next >

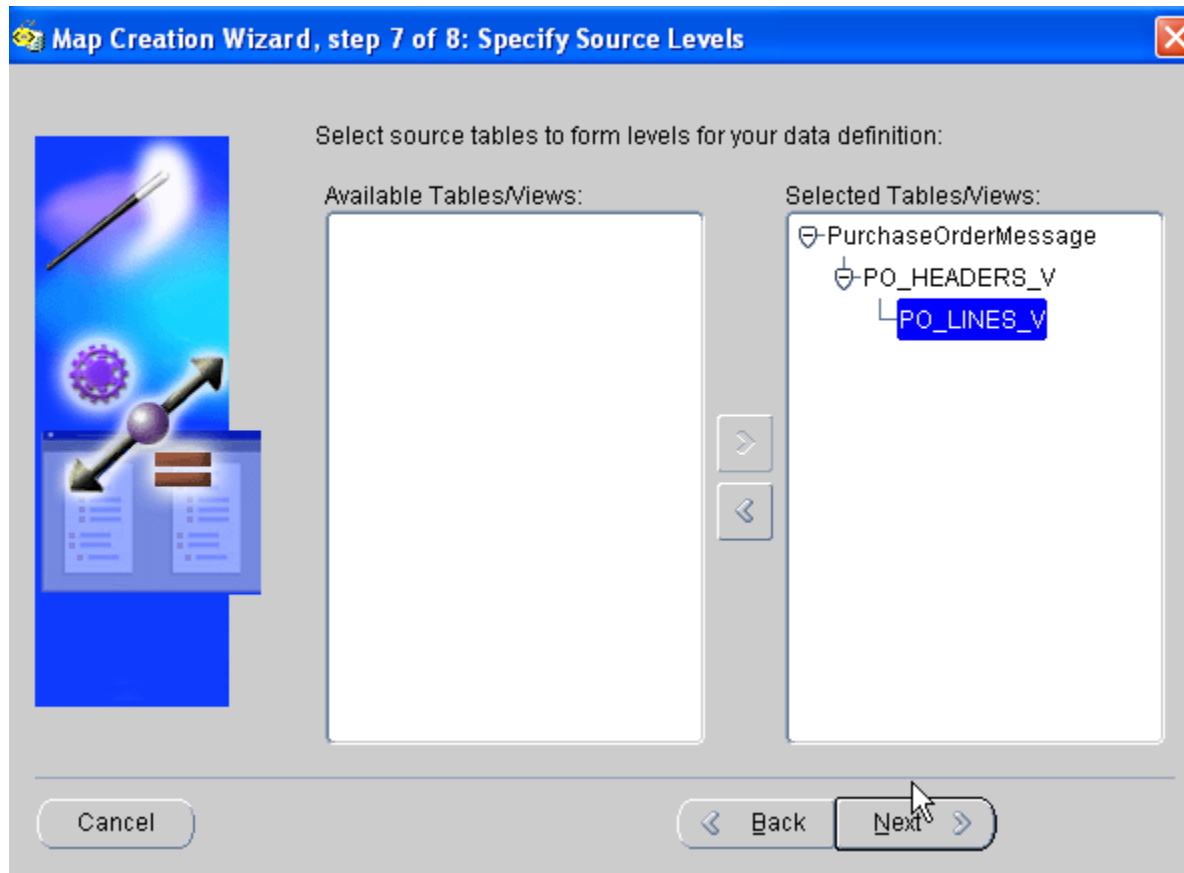
■ Step 6 -- Selecting Database Objects



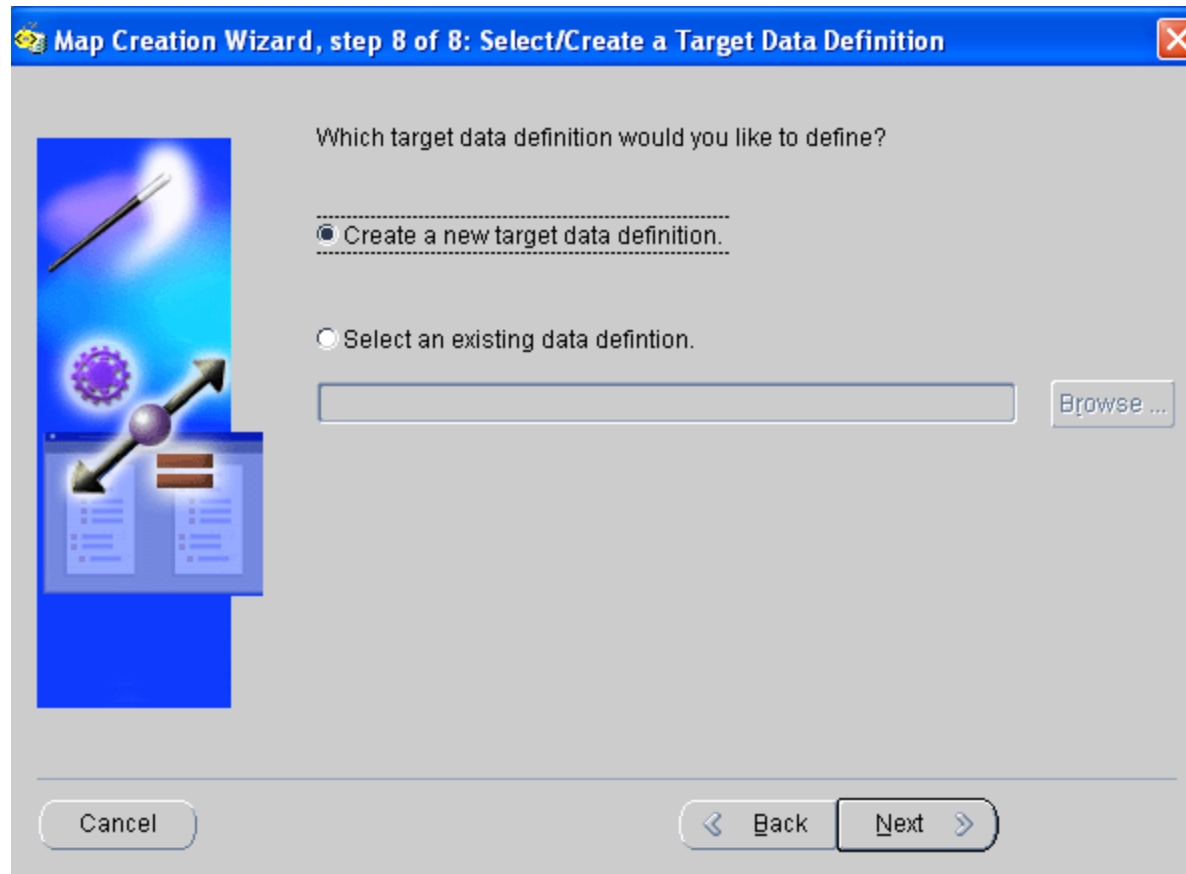
■ Step 7 -- Selecting Required Columns



- Step 8 -- Create hierarchy of the source data definition




■ Step 9 -- Creating Target Data Definition



- Step 10 -- Target Data Definition Name

Map Creation Wizard, step 9 of 9: Specify Target Data Definition Name and Type



Enter data definition name:


Enter data definition description:

Select data definition type:

Cancel < Back Next >

- Step 11 -- Target Data Definition source

Map Creation Wizard, step 10 of 10: Specify Target XML file and root element



Enter XML standard:

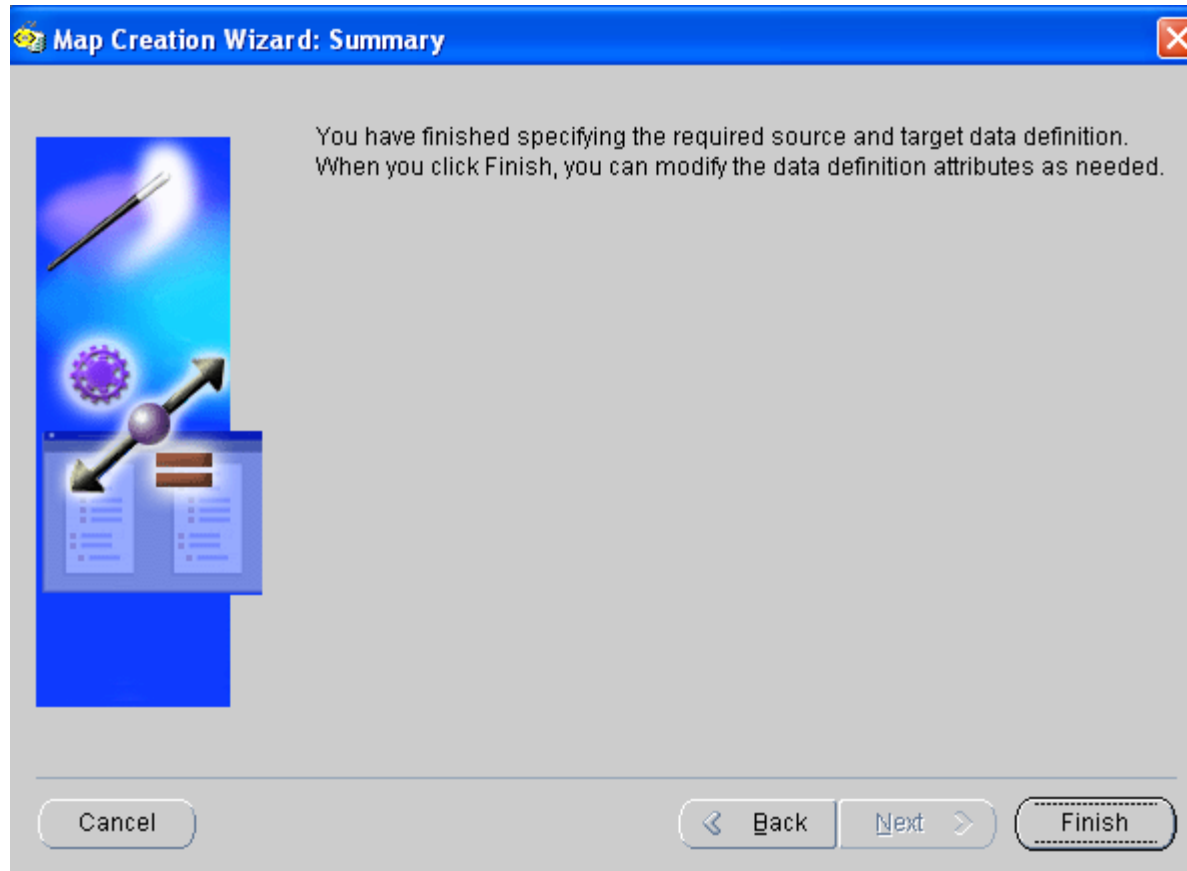
Select a DTD/XML file:

Identify the XML root element:

Enter DTD file name:

Identify the runtime location of the dtd:

- Step 12 -- Transaction Map created



Step 13 -- Source Definition

TransactionMap:WVUPurchaseOrderWithDist

SourceDefinition | Target Definition | Level Mapping | Element Mapping

Source

- PurchaseOrderMessage
 - WVUFIN_SQ_PO_HEADER_V

Source Elements

	Field	ItemType	Default	Category	Data Type	DB Column
1	PurchaseOrderMessage	Level				<input type="checkbox"/>
2	WVUFIN_SQ_PO_HEA...	Level				<input type="checkbox"/>
3	BILLTO_ADDRESS_C...	Element			VARCHAR2	<input checked="" type="checkbox"/>
4	BILLTO_ADDRESS_LI...	Element			VARCHAR2	<input checked="" type="checkbox"/>
5	BILLTO_ADDRESS_LI...	Element			VARCHAR2	<input checked="" type="checkbox"/>
6	BILLTO_ADDRESS_LI...	Element			VARCHAR2	<input checked="" type="checkbox"/>
7	BILLTO_CITY	Element			VARCHAR2	<input checked="" type="checkbox"/>
8	BILLTO_COUNTRY	Element			VARCHAR2	<input checked="" type="checkbox"/>
9	BILLTO_COUNTRY_N...	Element			VARCHAR2	<input checked="" type="checkbox"/>
10	BILLTO_LOCATION_ID	Element			NUMBER	<input checked="" type="checkbox"/>
11	BILLTO_PHONE1	Element			VARCHAR2	<input checked="" type="checkbox"/>
12	BILLTO_PHONE2	Element			VARCHAR2	<input checked="" type="checkbox"/>
13	BILLTO_PHONE3	Element			VARCHAR2	<input checked="" type="checkbox"/>
14	BILLTO_STATE	Element			VARCHAR2	<input checked="" type="checkbox"/>
15	BILLTO_ZIP	Element			VARCHAR2	<input checked="" type="checkbox"/>
16	BUYER_CODE	Element			VARCHAR2	<input checked="" type="checkbox"/>
17	BUYER_EMAIL	Element			VARCHAR2	<input checked="" type="checkbox"/>
18	BUYER_FULL_NAME	Element			VARCHAR2	<input checked="" type="checkbox"/>
19	BUYER_ID	Element			NUMBER	<input checked="" type="checkbox"/>
20	BUYER_TELEPHONE	Element			VARCHAR2	<input checked="" type="checkbox"/>
21	CARRIER_ORG_NAME	Element			VARCHAR2	<input checked="" type="checkbox"/>
22	COMMENTS	Element			VARCHAR2	<input checked="" type="checkbox"/>
23	CONFIRMING_ORDE...	Element			VARCHAR2	<input checked="" type="checkbox"/>

Step 14 -- Target Definition

TransactionMap:WVUPurchaseOrderWithDist

SourceDefinition | Target Definition | Level Mapping | Element Mapping

Target

- [-] PurchaseOrderMessage
 - version
 - [+] Header
 - [+] PurchaseOrder

Target Elements

	Field	ItemType	Default	Data Type	Node Type
1	PurchaseOrderMessage	Level			Element
2	version	Element		VARCHAR2	Attribute
3	Header	Level		VARCHAR2	Element
4	MessageId	Element		VARCHAR2	Element
5	Timestamp	Element		VARCHAR2	Element
6	Authentication	Element		VARCHAR2	Element
7	Identity	Element		VARCHAR2	Element
8	SharedSecret	Element		VARCHAR2	Element
9	PurchaseOrder	Level		VARCHAR2	Element
10	POHeader	Level		VARCHAR2	Element
11	id	Element		VARCHAR2	Attribute
12	RevisionNumber	Element		VARCHAR2	Element
13	RevisionDate	Element		VARCHAR2	Element
14	PONumber	Element		VARCHAR2	Element
15	Requestor	Element		VARCHAR2	Element
16	UserProfile	Element		VARCHAR2	Element
17	username	Element		VARCHAR2	Attribute
18	FirstName	Element		VARCHAR2	Element
19	LastName	Element		VARCHAR2	Element
20	Email	Element		VARCHAR2	Element
21	BuyerInfo	Element		VARCHAR2	Element
22	BuyerContactName	Element		VARCHAR2	Element
23	Email	Element		VARCHAR2	Element

Step 15 -- Level Mapping

TransactionMap:WVUPurchaseOrderWithDist

SourceDefinition Target Definition **Level Mapping** Element Mapping

Source

- [-] PurchaseOrderMessage
 - [+] WVUFIN_SQ_PO_HEADER_V

Target

- [-] PurchaseOrderMessage
 - version
 - [+] Header => WVUFIN_SQ_PO_HEADER_V
 - [-] PurchaseOrder => WVUFIN_SQ_PO_HEADER_V
 - [+] POHeader => WVUFIN_SQ_PO_HEADER_V
 - [+] POLine => WVUFIN_SQ_PO_LINES_V

Step 16 -- Element Mapping

TransactionMap:WVUPurchaseOrderWithDist

SourceDefinition Target Definition Level Mapping **Element Mapping**

Source

- WVUFIN_SQ_PO_HEADER_V
 - BILLTO_ADDRESS_CODE
 - BILLTO_ADDRESS_LINE1
 - BILLTO_ADDRESS_LINE2
 - BILLTO_ADDRESS_LINE3
 - BILLTO_CITY
 - BILLTO_COUNTRY
 - BILLTO_COUNTRY_NAME
 - BILLTO_LOCATION_ID
 - BILLTO_PHONE1
 - BILLTO_PHONE2
 - BILLTO_PHONE3
 - BILLTO_STATE
 - BILLTO_ZIP
 - BUYER_CODE
 - BUYER_EMAIL
 - BUYER_FULL_NAME
 - BUYER_ID
 - BUYER_TELEPHONE
 - CARRIER_ORG_NAME
 - COMMENTS
 - CONFIRMING_ORDER_FLAG
 - CUSTOMER_NUM
 - EDI_LOCATION_CODE
 - END_DATE_ACTIVE

Target

- PurchaseOrderMessage
 - version
 - Header => WVUFIN_SQ_PO_HEADER_V
 - MessageId => PO_HEADER_ID
 - Timestamp
 - Authentication
 - Identity
 - SharedSecret
 - PurchaseOrder => WVUFIN_SQ_PO_HEADER_V
 - POHeader => WVUFIN_SQ_PO_HEADER_V
 - id => PO_HEADER_ID
 - RevisionNumber => PO_REVISION_NUM
 - RevisionDate => PO_CREATION_DATE
 - PONumber => PO_NUMBER
 - Requestor
 - BuyerInfo
 - CreateDateTime
 - Supplier
 - BillTo
 - ShipTo
 - PaymentInfo
 - ExternalInfo
 - FreightOnBoard => FOB_NAME
 - CustomFieldValueSet
 - CustomFieldValueSet

■ Step 17

❖ Action Levels

An Action may be defined for any of the following entities:

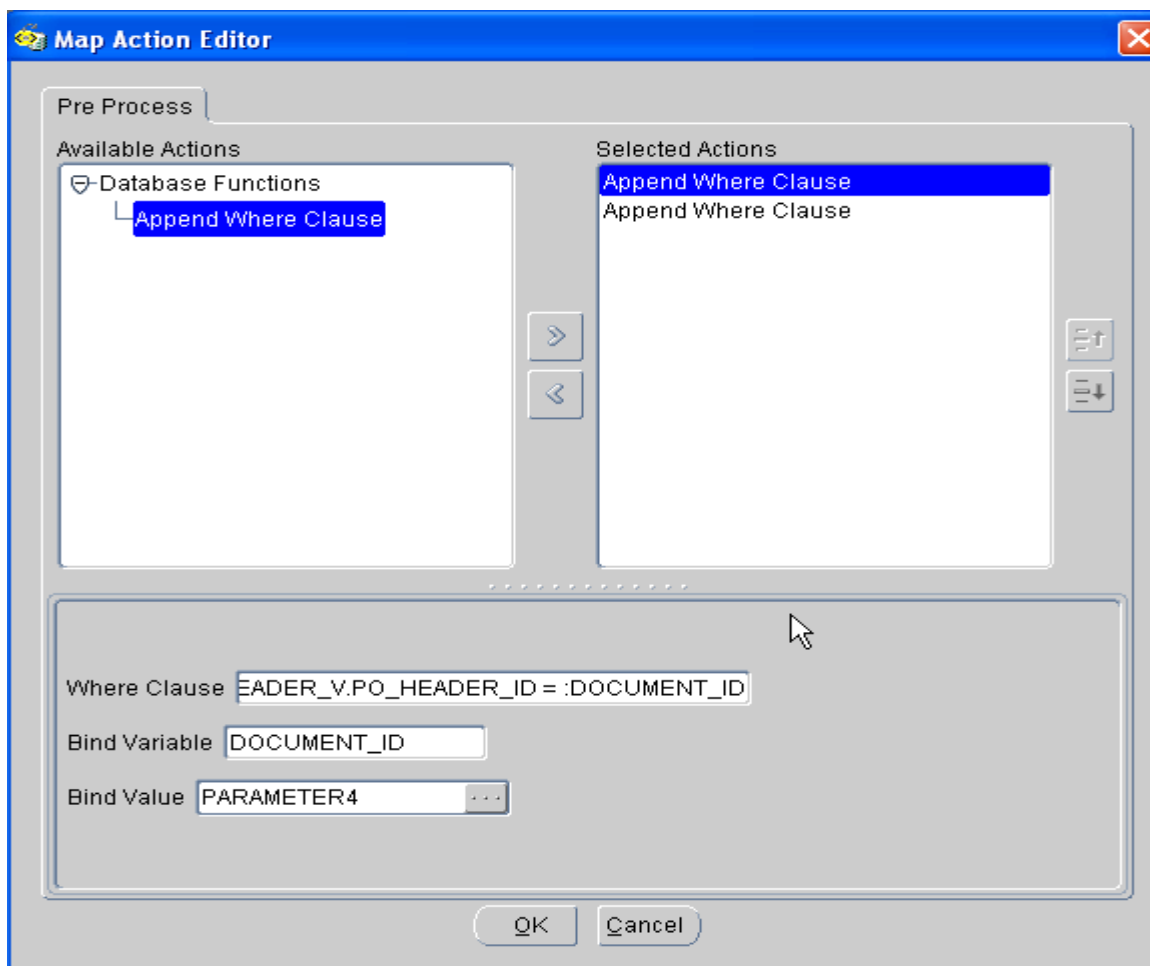
- Element - An element is the smallest unit of a message. An action defined at the element level is applied to that element only.
- Document - A document is a collection of elements representing a business document. An action defined at the document level is applied to the document.
- Root - A root represents a collection of documents. An action defined at the root level is applied to all documents contained by the root.

Some actions are designed to be applied to the element only, while others are intended for the document only. Most Actions are defined on the target side of the Element Map with the exception of the Append Where Clause action type, which is defined on the source side of the Element Map if the source is based on database views or tables.

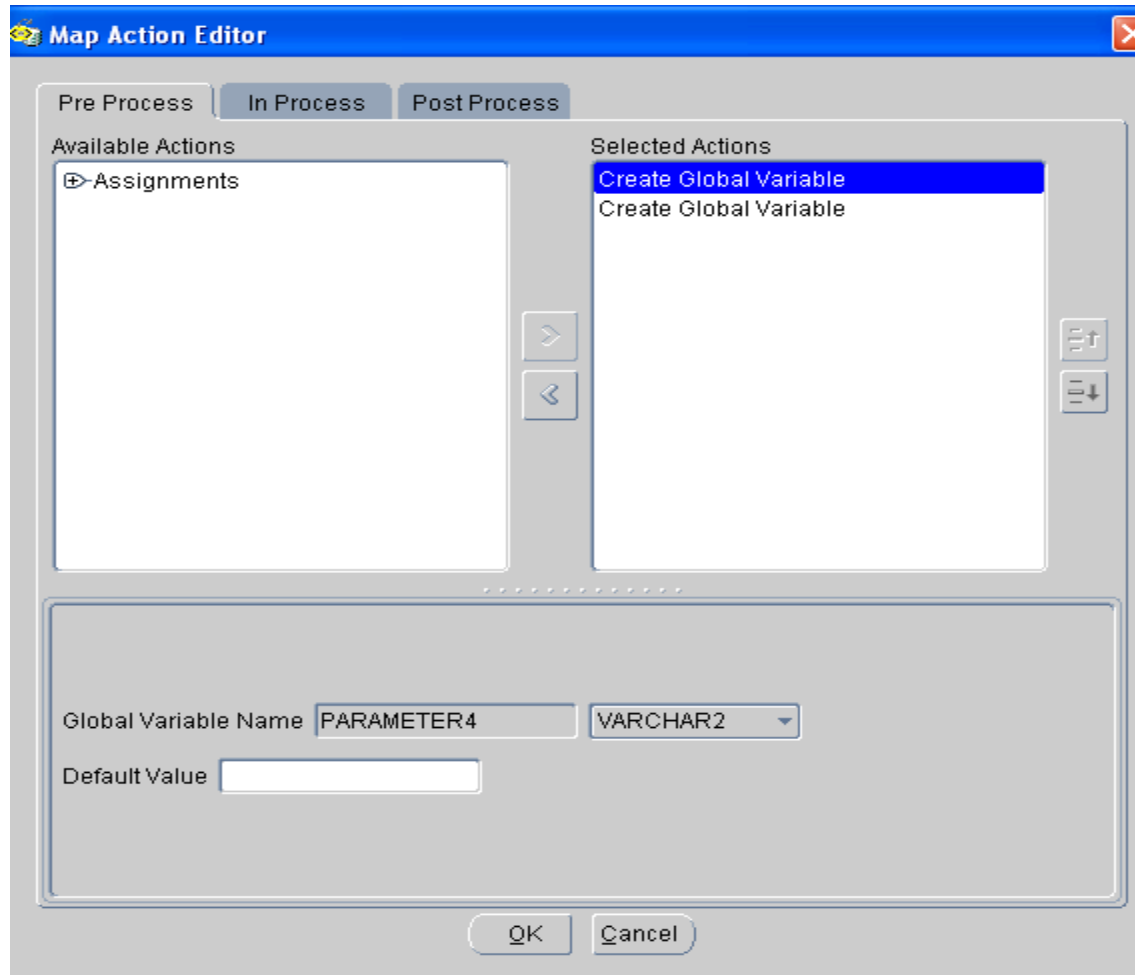
An Action can be applied at any of the following stages of message creation or consumption:

- Pre Process -- A preprocess action is executed before the message is created or consumed. The Create Global Variable action is an example of a preprocess action. The variable must be defined before you can use it.
- In Process -- An in-process action is executed during message creation or consumption. The Math and String Functions are examples of in-process actions used to perform a computation or to manipulate a value.
- Post Process -- A postprocess action is executed after the message is created or consumed. The Insert into Database Table action is an example of a postprocess action. The row cannot be inserted into the database until all the data for the row has been processed.

- Step 18 -- Source Action



■ Step 19 -- Target Action



The image shows a 'Map Action Editor' dialog box with a blue title bar and a close button. It has three tabs: 'Pre Process', 'In Process', and 'Post Process'. The 'In Process' tab is selected. The dialog is divided into two main sections: 'Available Actions' on the left and 'Selected Actions' on the right. In the 'Available Actions' list, 'Assignments' is selected. In the 'Selected Actions' list, 'Create Global Variable' is selected. Between the two lists are two arrow buttons: a right-pointing arrow and a left-pointing arrow. To the right of the 'Selected Actions' list are two buttons: one with a plus sign and a right arrow, and another with a minus sign and a right arrow. Below these sections is a large rectangular area containing a 'Global Variable Name' field with the text 'PARAMETER4', a data type dropdown menu set to 'VARCHAR2', and a 'Default Value' text input field. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

- **Step 20 -- Loading Map and DTD to Database**
 - ❖ Create the directory \$CUSTOM_TOP/xml/oag721 on the database server to keep XML message files and DTD files.
 - ❖ Upload the Message (ncoaug_po.xgm) and DTD (wvupo.dtd) files to \$CUSTOM_TOP/xml/oag721 directory via ftp.
 - ❖ Import the DTD file into database using following command from \$CUSTOM_TOP/xml/oag721 directory –

```
java oracle.apps.ecx.loader.LoadDTDToClob<DB username><DB password><Hostname>:<Port>:<SID><mydtd.dtd>  
<RootElementName><Location>
```
 - ❖ Import XGM (XML Message Designer) file into database using following command from \$CUSTOM_TOP /xml/oag721 directory.

```
java DownloadMap <DB username><DB Password><Hostname>:<Port>:<SID><MAP_CODE>
```

■ XML Gateway Execution Engine

- ❖ The execution engine is the driver that begins its operation once Oracle Workflow Business Event System detects an outbound transaction to be processed, or that an inbound message has arrived on the queue and does the following:
 - Validate whether Trading Partner or Hub is defined and the required document is defined and enabled.
 - Retrieve Message Map from Repository associated with the Trading Partner and required document.
 - Execute the Message Map by either gathering application data from Oracle EBS using the database views and columns identified in the message map for outbound message or maps the data in the XML message to its target data fields in Oracle EBS tables and columns identified in the message map for inbound messages.
 - Apply code conversions, data transformations, and process control functions defined on the message map.
 - Validate Message using XML Parser to check whether the inbound message or the newly created outbound message is well-formed and valid (based on a DTD stored in the DTD directory) before proceeding further.
 - Create XML Message if its outbound message or Consume XML Message if it's inbound.
 - Dequeue Message from Inbound Queue or Enqueue Message to Outbound Queue
 - Detect and Report Processing Errors via email notification to appropriate authority.
 - Create or Consume Confirmation Messages if it is enabled for the Trading Partner.

- ❖ All this is possible only if the following are fulfilled
 - Message maps are created and loaded into the repository along with their associated DTDs.
 - Trading Partners are defined.
 - Code Conversions are defined.
 - Transactions are defined.
 - Oracle Workflow Business Event System events are published by the Oracle EBS and subscriptions to those events are defined.
 - Engine and listeners are started.

■ Define System Profile Options

- ❖ ECX: Log File Path - Log File Path where the XML messages and runtime log are stored (Required)
- ❖ ECX: Maximum XML Size - Specifies the maximum size of an outbound XML document (in characters) beyond which parsing is performed based on the value of ECX: XML Validate Flag. If the size is not set, the document will be parsed by default. (Not Required , Default value – 2MB)
- ❖ ECX: Server Time Zone – The time zone in which the database server is running (Required)
- ❖ ECX: System Administrator Email Address – XML Gateway System Administrator e-mail address (Required)
- ❖ ECX: XML Suppress Empty Tags – Specifies whether an outbound document should have empty tags in it or not. (Not Required)
- ❖ ECX: XML Validate Flag - Specifies whether an outbound document should continue to be parsed by the engine after the ECX: Maximum XML Size has been met. (Not Required, Default Value - Y)
- ❖ ECX: XSLT File Path – Path where XSLT style sheets are stored. (Not Required)

■ Assign XML Gateway Responsibility

- ❖ Use the System Administrator responsibility to assign the Oracle XML Gateway responsibility to a user to access the Oracle XML Gateway database and forms. Use standard procedures to assign the responsibility

■ Define the UTL_FILE_DIR parameters

- ❖ To use Oracle XML Gateway, you must first create directories where the XML message process log and XSLT style sheets will be stored. Oracle XML Gateway uses the UTL_FILE package to read and write to the server. UTL_FILE can only write to accessible directories. The directories are defined by the utl_file_dir parameter in the init<SID>.ora file. Within this file, each accessible directory is indicated by a line such as utl_file_dir=<directory_name>. The value for *directory_name* must be a physical directory. It cannot be a variable, a logical, or an alias. In addition, the value for *directory_name* must match the value defined in the Oracle XML Gateway profile for ECX_UTL_LOG_DIR File Path (ECX: Log File Path) and ECX_UTL_XSLT_DIR File Path (ECX: XSLT File Path).

■ Define Hubs

- ❖ A hub is an integration point within your network (either your intranet or the internet). Hubs are typically used to route documents to and from trading partners. Oracle Exchange is an example of a hub.

■ Define XML Standards

- ❖ Defines standards bodies for XML messages.

■ Define Transactions

- ❖ We define the transactions that will be used by the XML Gateway Execution Engine. We will then associate these transactions with a trading partner in the Trading Partner Setup form. Most of the seeded transactions setup within Oracle XML Gateway suffices and requires no additional setup. For example for an outbound PO we would choose the following seeded transaction

- Party Type : Supplier
- Transaction Type: PO
- Transaction Subtype: PRO
- Transaction Description: Purchase Order outbound when a new PO is created
- Standard Code: <whatever standard you are following>
- Direction: OUT
- External Transaction Type: ORDER
- External Transaction Subtype: REQUEST

Define Transactions

Party Type: **Supplier**

Transaction Type: **PO**

Transaction Subtype: **PRO**

Transaction Description: **Purchase Order outbound when a new PO is created.**

External Transactions

Standard Code	Direction	External Transaction Type	External Transaction Subtype	Queue
CXML	OUT	ORDER	REQUEST	
OAG	OUT	PO	PROCESS	
OAG	OUT	POPI	POPI	
OAG	OUT	SOP	SOP	

■ Define Trading Partners

- ❖ This is one of the most important steps in setups. This is the component that will enable a message to be processed through the XML Gateway engine.
- ❖ In the XML Gateway, the term "Trading Partner" refers to an entity such as a customer, supplier, bank branch, or internal locations at a particular address with which you exchange messages. Since a given entity may have several locations, you must define one Trading Partner for each customer address, supplier site, or bank branch as required for processing transactions by the Oracle XML Gateway.
- ❖ The Trading Partner Setup form is used to:
 - Enable messages for the trading partner by identifying the internal and external transaction type and transaction subtype codes, and the XML standard associated with the message.
 - Access the Trading Partner User Setup form.
 - Access the Trading Partner Code Conversion form.
 - Select a message map for the trading partner.
 - Identify the communications protocol and address for a message.
 - Trading Partner setup in XML Gateway is organization-dependent.

- ❖ During message processing, Trading Partner data is used to
 - Link a particular address location in Oracle E-Business Suite to the Trading Partner definition in the Gateway.
 - Provide a means of telling the Execution Engine which Trading Partner message map to use.
 - Enable specific transactions for Trading Partners.
 - Determine how to deliver the message.
- ❖ The Trading Partner Setup form requires an entry for each Transaction Type and Transaction Subtype associated with this trading partner. For example for an outbound PO the following data needs to be filled in:
 - Trading Partner Type: Supplier
 - Trading Partner Name: Name of the Supplier/Vendor
 - Trading Partner Site: Site of the Supplier/Vendor for which we need to set up the record
 - Company Admin Email: Email Address of a Contact from Supplier/Vendor
 - Transaction Type: PO
 - Transaction Subtype: PRO
 - Standard Code: XML standard being used
 - External Transaction Type: ORDER

- External Transaction Subtype: REQUEST
- Direction: OUT
- Map: The mapping which would be used
- Connection/Hub: DIRECT
- Protocol Type: Whatever Protocol needs to be used
- Username: Username given by Trading Partner
- Password: Password given by Trading Partner
- Protocol Address: Complete URL (including service/servlet) where the Transport Agent will attempt to post the XML Document.
- Source Trading Partner Location Code: A unique code that identifies your organization
- Destination Trading Partner Location Code: A unique code that identifies your final destination organization in case its being routed through a hub.
- Document Confirmation: Indicator for the confirmation level that this Trading Partner would like to send or receive a confirmation. Possible values are
 - 0 (Default value) means Never send a confirmation
 - 1 means Send a confirmation only if there are errors
 - 2 means Always send a confirmation
- Routing: Whether Static or Dynamic

Functions Documents

Define Trading Partners

Define Trading Partners

Define Hubs

Define XML Standards

Define Code Conversion

Define Transactions

Define Trading Partners

Define Lookup Values



Trading Partner Setup

Trading Partner Type **Supplier**

Trading Partner Name **Office Supplies, Inc.**

Trading Partner Site **3605 Warrensville Center Road Shaker Heights OH 44122**

Company Admin Email **customer@company.com**

User Setup Code Conversion

Trading Partner Details

Username	Password	Protocol Address	Source Trading Partner Location Code	Destination Trading Partner Location Code	Document Confirm	Routing
ast		http://www.astcorp	AST		0	

■ Define Standard Code Conversion

- ❖ The Oracle XML Gateway code conversion function provides a method to cross-reference the codes defined in Oracle E-Business Suite to codes used by trading partners, the XML standard, or other standard codes in the transactions. The external code may be the XML standard codes such as OAG's code, the universal code such as ISO codes, or trading partner-specific. The trading partner-specific code may override an XML standard or the other universally used code conversion values

Trading Partner Code Conversion: Office Supplies, Inc., 3605 Warrensville Center Road Shaker Heights OH 44122

Category Code	Description
ACTION_CODE	Action Code (such as Add/Change/De
AC_HANDLING	Allowance and Charge Handling Co
AC_SPEC_CHARGES	Allowance and Charge Special Char
AC_SPEC_SERVICES	Allowance and Charge Special Serv
ADDSERCODE	Additional Service

Category Values

Standard Code	Oracle Value	Description	From Trading Partner Value	To Trading Partner Value	Standard	Data Seeded
CUSTOM					<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>

Revert All Revert

Questions and Open Discussions



Contact Information

Shibu Kunneth

Email: skunneth@astcorporation.com

Phone: 630-631-7080

Presentation Copy

www.astcorporation.com/papers/ncoaug

- **Oracle XML Gateway User's Guide, Release 11i Part No. B10665-04**
- **Metalink Note 204162.1**
- **Oracle Purchasing 11i XML XML Transaction Delivery Setup Guide Part No. A96668-02**
- **Metalink Note 213169.1**
- **Metalink Note 337428.1**