Oracle® E-Business Suite Adapter for Imaging and Process Management

Application Extension Framework Configuration Guide

10.1.3.5

August 2010

ORACLE®

Oracle E-Business Suite Adapter for Imaging and Process Management Application Extension Framework Configuration Guide, 10.1.3.5

Primary Author: Sarah Howland

Contributor: Tom Albrecht, Sancho Pinto

Contents

Preface

The Application Extension Framework Configuration Guide describes the Application Extension Framework and leads you through the steps required to implement commands and configure them for use within the AXF infrastructure.

Audience

This document is intended for developers creating integration solutions between specific business applications, BPEL, and Oracle I/PM.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

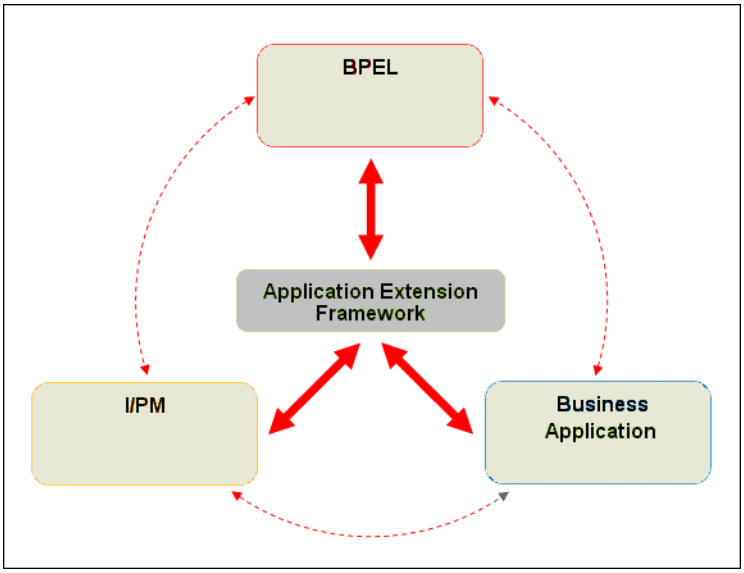
AXF Overview

The following topics are covered in this chapter:

1.1 About Application Extension Framework

Oracle's Application Extension Framework (AXF) is a command-driven, web services integration between a business application and BPEL (Oracle BPEL Process Manager), using Oracle I/PM as the imaging source. Through an AXF configuration, business users can process associated images and perform document-centric workflow tasks from their business application user interface. The open Java-based architecture of AXF allows integrators to configure and modify multiple business process solutions separate from the systems themselves, and to upgrade systems without affecting implemented AXF solutions.

Figure 1–1 Application Extension Framework integrates Business Application, I/PM, and BPEL systems



AXF Solutions

An AXF solution is a micro-application whose components are created using the AXF infrastructure.

AXF Solution Templates

Oracle provides templates for specific functions, such as automating invoice and receipt processing via BPEL-based workflows with associated approval rules, data entry forms, and reports.

> **Note:** To obtain a solution template, contact your systems integrator, Oracle Consulting, or Oracle Support.

Business User View

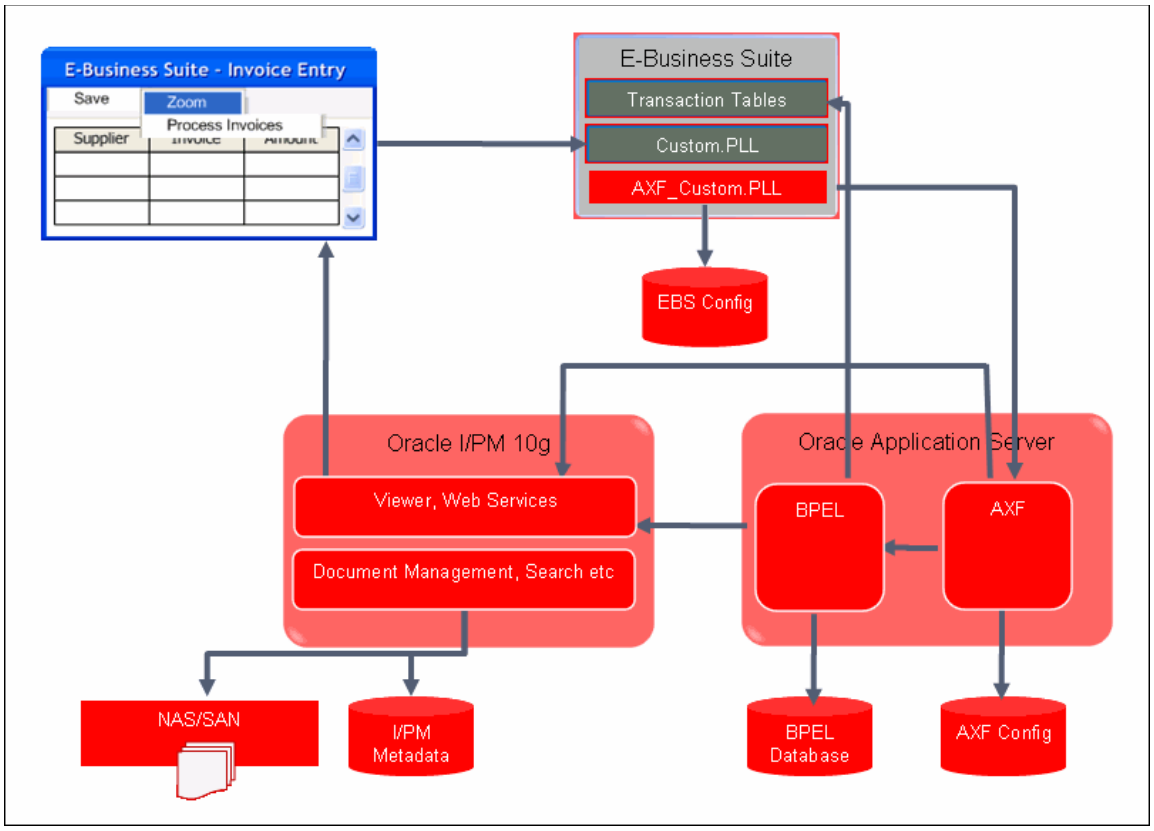
From a business user's perspective, the integration is virtually seamless. For example, users performing Accounts Payable tasks might select a custom menu command integrated into their business application called Start Processing Images, initiating the following processes:

- A SOAP request is generated and sent to AXF, passing the selected command along with additional parameters such as an AXF solution (Accounts Payable), an AXF command (Open_Tasklist), and a user name.
- AXF returns an Open_URL command with the URL to launch, such as an AXF Task List.
- The business application opens the Task List URL in a new browser window, enabling the user to start processing invoice images.

System Architecture

Figure 1–2 illustrates an AXF configuration for E-Business Suite.

Figure 1–2 System Architecture for AXF and EBS Configuration



1.2 AXF Components

The AXF framework includes these main components:

- "AXF Commands" on page 1-4
- "AXF Web User Tools" on page 1-5
- "AXF Configuration Database Tables" on page 1-9

1.2.1 AXF Commands

The reusable AXF commands allow you to implement the functionality described below. For information about these commands, including their parameters and example implementations, see "AXF Commands" on page 2-27.

AXF Command	Description
Open Task	Initializes and displays the AXF Task Viewer web page and claims a human workflow task. See "Open Task Command" on page 2-28.
Autotask	Initializes autotask mode, in which a new human workflow task is automatically claimed in the AXF Task Viewer without displaying the Task List. See "Autotask Command" on page 2-28.
Cancel Task	Initializes the AXF Task List web tool for display (regardless of Autotask mode) and releases a human workflow task. See "Cancel Task Command" on page 2-29.
Complete Task	Completes a human workflow task and updates BPEL payload attribute values. If using Autotask Command, claims the next task and displays it in the Task Viewer. See "Complete Task Command" on page 2-30.
Redirect	Redirects the current AXF web page to any URL specified in the configuration. See "Redirect Command" on page 2-30.
Terminate Conversation	Used by an external client to terminate a conversation with AXF. (This command does not include parameters.)
Update Task	Updates BPEL payload field values on a specified human task or values in the XML payload using XPATH. See "Update Task Command" on page 2-31.
Update Task From Procedure	Calls a stored procedure using a specified data source and returns results using XPATH. See "Update Task From Procedure Command" on page 2-32.
Validate Task	Used to validate BPEL system attribute data or BPEL payload data using the Regular Expression language, and based on validation results, execute a subsequent command. See "Validate Task Command" on page 2-34.

Note: You can also deploy custom commands to execute via AXF. See "Custom Commands" on page 2-36.

1.2.2 AXF Web User Tools

The Application Extension Framework provides the following user interface components. These are web interface components displayed to users, and configured through the AXF tables.

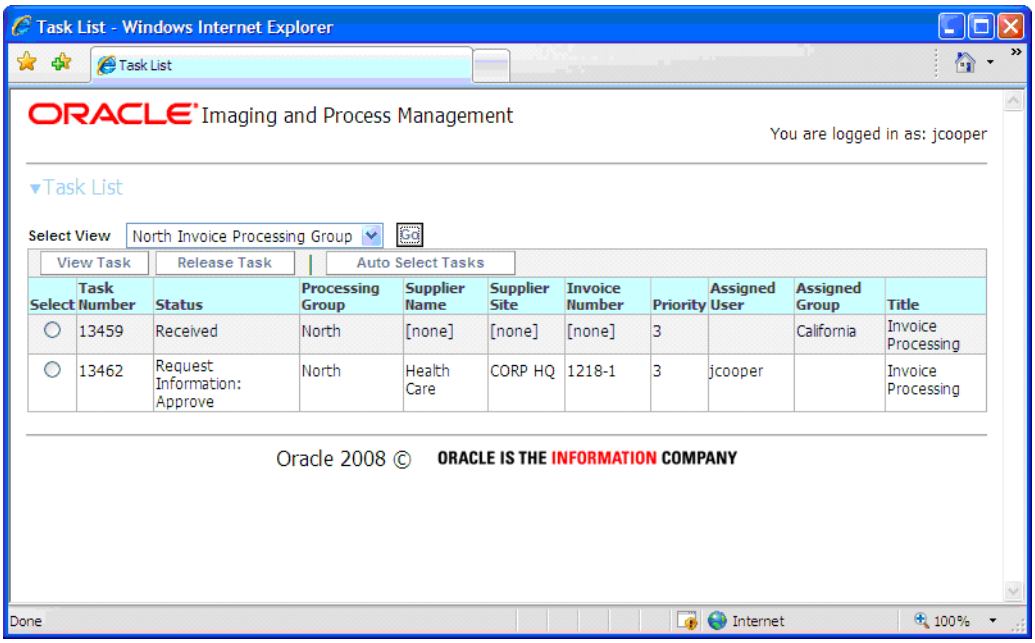
- "Task List" on page 1-5
- "Task Viewer" on page 1-6
- "Enumeration Picker" on page 1-7
- "User Picklist" on page 1-8
- "Comments" on page 1-9

1.2.2.1 Task List

The Task List web page displays a list of available tasks to users. It interacts with the AXF Infrastructure and BPEL to display the list using views configured in the BPEL Worklist application.

For configuration information, see "Task List Web Tool" on page 2-13.

Figure 1–3 Task List Web Tool



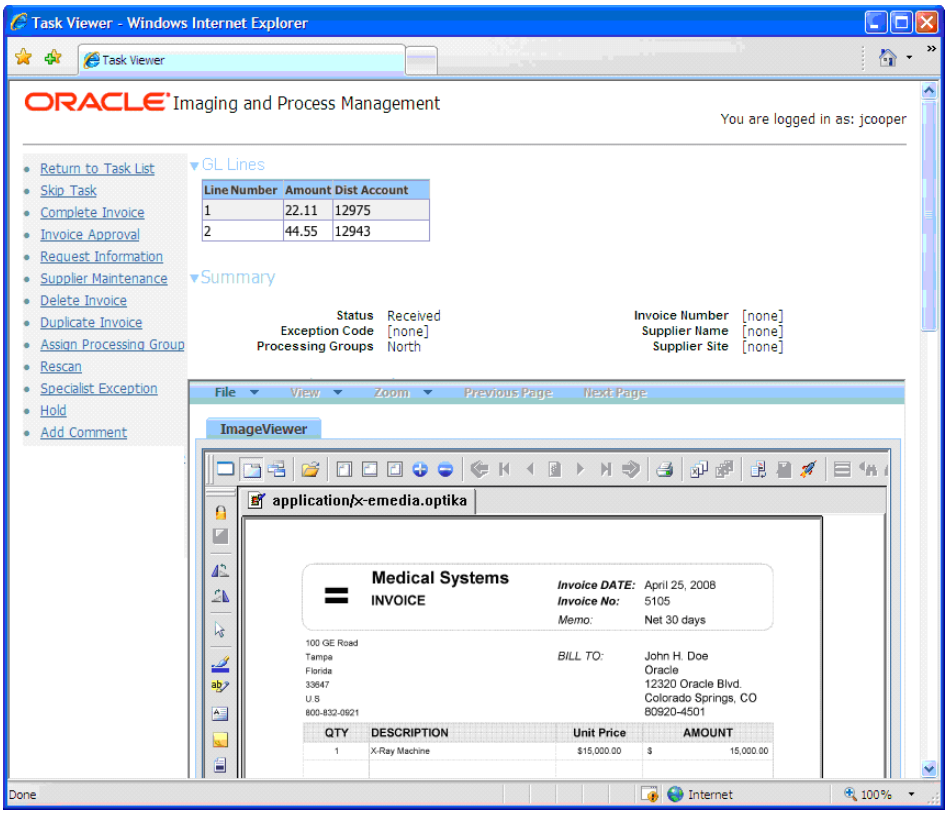
The Task List web tool can also display a list of AXF actions just like the Task Viewer, using AXF action commands. These action commands are menu components configured in the AXF_ACTIONS Table for display on a web page.

1.2.2.2 Task Viewer

The Task Viewer web page displays images and metadata values through interaction with the AXF Infrastructure, BPEL, I/PM, and the business application. It also displays a side menu containing AXF action commands configured in the AXF_ACTIONS Table.

For configuration information, see "Task Viewer Web Tool" on page 2-16.

Figure 1–4 Task Viewer Web Tool

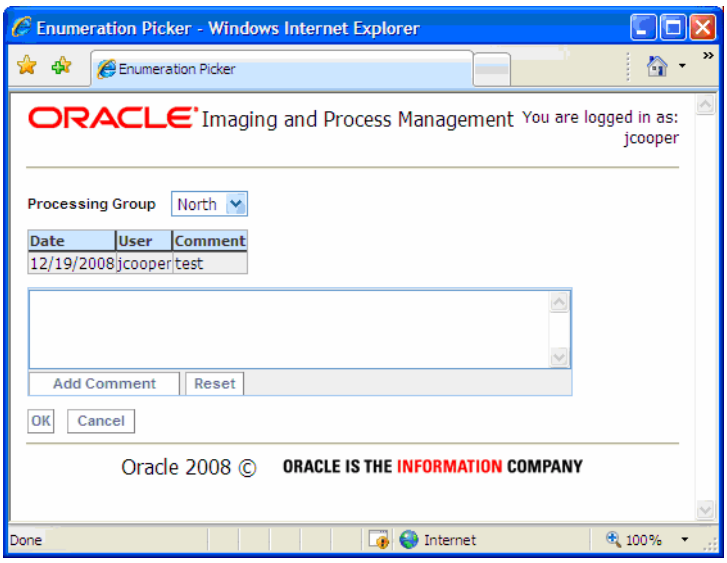


1.2.2.3 Enumeration Picker

The Enumeration Picker web page allows users to select from a list of enumerated values configured in the AXF database tables and to add comments.

For configuration information, see "Enumeration Picker Web Tool" on page 2-21.

Figure 1–5 Enumeration Picker Web Tool

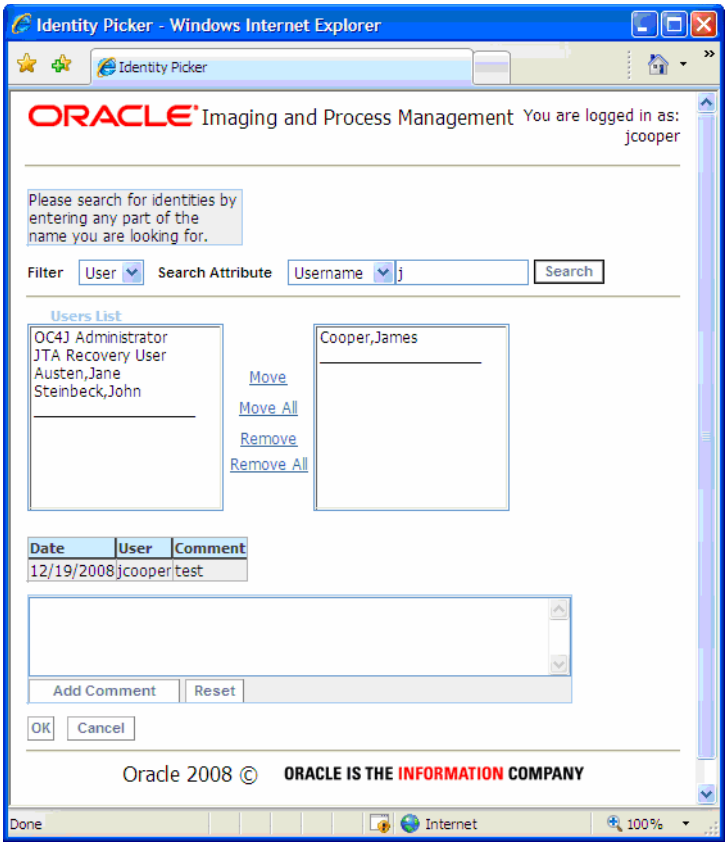


1.2.2.4 User Picklist

The Identity Picker web page allows users to select one or more users or groups from an identity store configured for BPEL, and to add comments. After choosing an identity, a related action is typically taken; most likely, a task is assigned or delegated to the selected user or group of users. For example, a business user who encounters a problem with a transaction might select an exception handler to send the transaction to, after entering a comment that describes the problem.

For configuration information, see "Identity Picker Web Tool" on page 2-24.

Figure 1–6 User Picklist Web Tool

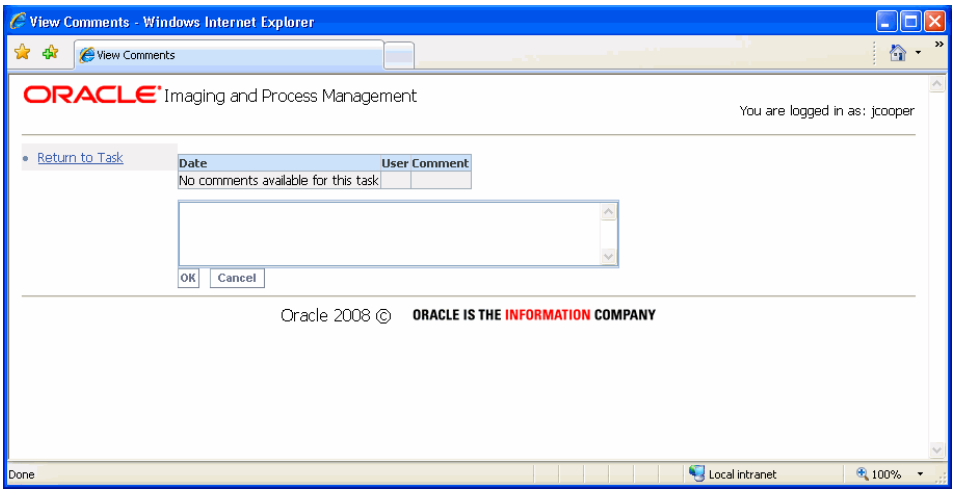


1.2.2.5 Comments

The Comments web page allows users to enter comments related to the human task during the transaction's processing. Comments persist for the entire process, allowing users to view, add, or edit comments. Users can add comments by clicking an action list link or on the Enumeration Picker or Identity Picker web pages. Comments are saved using the native comments capabilities of BPEL's workflow task.

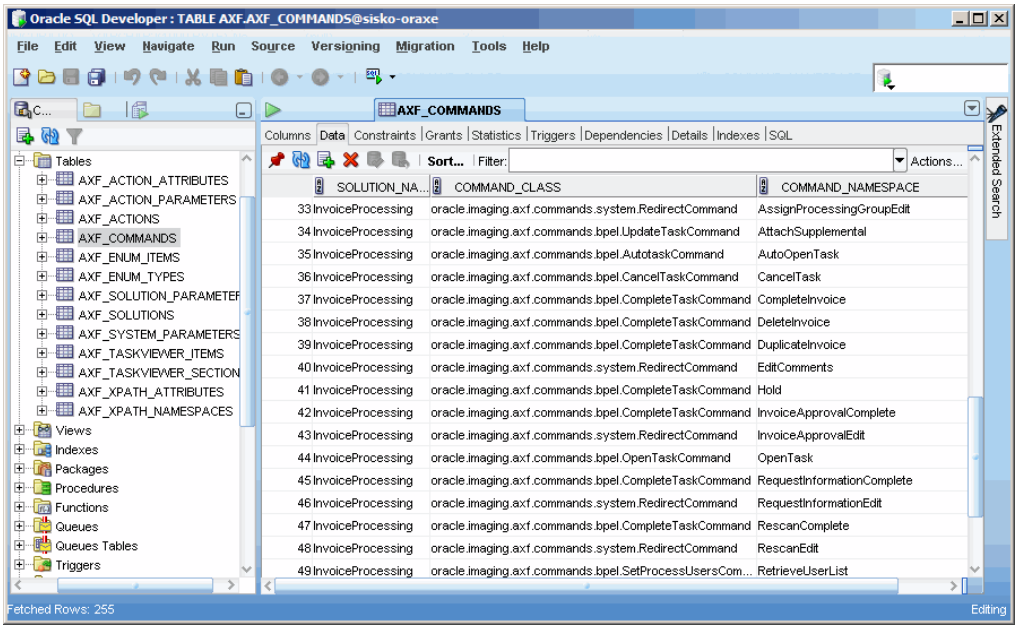
For configuration information, see "Comments Web Tool" on page 2-26.

Figure 1–7 Comments Web Tool



1.2.3 AXF Configuration Database Tables

You configure AXF solutions, commands, and web tools by configuring the AXF database tables. See "AXF Configuration Tables" on page 2-1 for information about each table and example implementations.



1.3 E-Business Suite Components

As part of AXF configuration, solution integrators configure the following E-Business Suite (EBS) components:

1.3.1 E-Business Suite Configuration Database

Configuring AXF for EBS requires configuring AXF-related tables in EBS. These tables are used to specify which screens are enabled to execute configured AXF commands.

AXF-related EBS tables include the following (covered in "AXF Tables in E-Business Suite" on page 3-1):

- AXF_CONFIGS Table
- AXF_COMMANDS Table
- AXF_COMMAND_PARAMETERS Table

1.3.2 PLL Modules

In AXF, user interface .PLL extension modules are used to access workflow tasks and documents associated with business records.

The *Custom.PLL* module is slightly modified during installation to call AXF functions. It notifies AXF each time an EBS event occurs, allowing AXF to determine if it relates to AXF functionality.

The *AXF_Custom.PLL* component performs the following functions:

- Calls out to a web service to execute an AXF Command
- Responds to the following AXF response commands:
  - Open Browser
  - Terminate Conversation
- Renders menus to expose AXF functionality based on the EBS configuration database.

1.3.3 PL/SQL Procedures

The following PL/SQL procedures are provided:

- *Add Attachment PL/SQL*: Use the axf_add_ebs_attachment store pl/sql procedure to add an attachment to a transaction in EBS.
- *AXF_SOAP_CALL_PROC.sql*: This script creates a stored procedure to make SOAP calls from PL/SQL. This script is configured in the *Application Extension Framework Installation Guide*.

2

AXF Configuration Tables

This chapter describes the AXF configuration tables used for AXF solutions, commands, and web user interface tools, and provides example implementations. For information about E-Business Suite tables, see "AXF Tables in E-Business Suite" on page 3-1. This chapter covers the following topics:

AXF Tables

AXF Web User Tools

AXF Commands

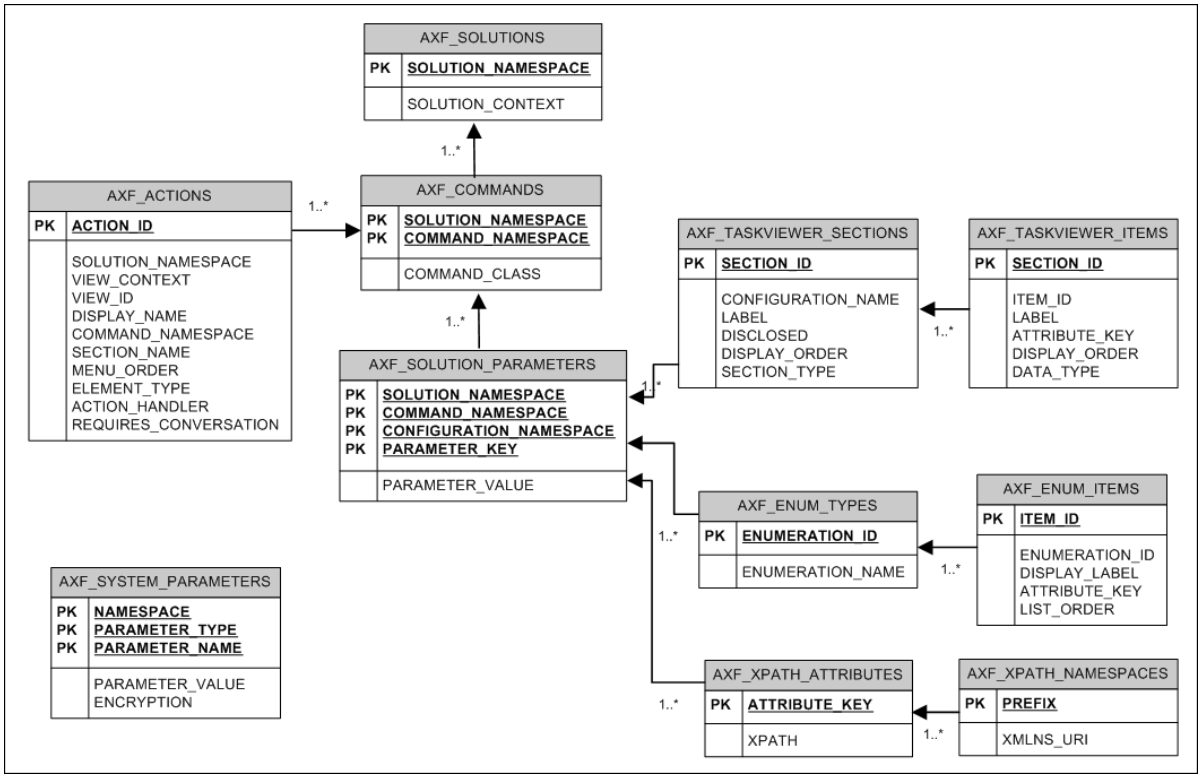
- "Validate Task Command" on page 2-34

2.1 Overview of AXF Configuration Tables

Note: If modifying AXF table values in a running system, either execute Clear DB Cache from the Driver page or restart the AXF application within the Application Server for the changes to take effect.

The diagram that follows displays the AXF configuration tables and their relationships.

Figure 2–1 AXF Configuration Tables



AXF Table	Description
AXF_SYSTEM_PARAMETERS Table	Define general parameters for infrastructure, services, and solutions.
AXF_SOLUTIONS Table	Define AXF solutions.
AXF_COMMANDS Table	Define AXF commands within solutions.
AXF_SOLUTION_PARAMETERS Table	Define parameters for AXF commands and AXF web tools.
AXF_ACTIONS Table	Define dynamic action menus for the Task Viewer web tool.
AXF_TASKVIEWER_SECTIONS Table, AXF_TASKVIEWER_ITEMS Table	Define the Task Viewer web tool.
AXF_ENUM_TYPES Table, AXF_ENUM_ITEMS Table	Define the Enumeration Picker web tool.

AXF Table	Description
AXF_XPATH_ATTRIBUTES Table, AXF_XPATH_NAMESPACES Table	Define XPATH attributes for payload elements.

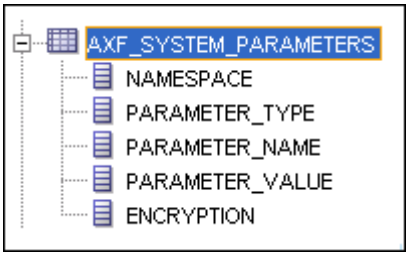
2.2 AXF Tables

This section describes the following AXF tables. See for "AXF Web User Tools" on page 2-13 for web tool-related tables.

- "AXF_SYSTEM_PARAMETERS Table" on page 2-3
- "AXF_SOLUTIONS Table" on page 2-5
- "AXF_COMMANDS Table" on page 2-6
- "AXF_SOLUTION_PARAMETERS Table" on page 2-7
- "AXF_ACTIONS Table" on page 2-9
- "AXF_XPATH_ATTRIBUTES Table" on page 2-12
- "AXF_XPATH_NAMESPACES Table" on page 2-13

2.2.1 AXF_SYSTEM_PARAMETERS Table

This table defines general system parameters for use by infrastructure, services, or solutions. For example, use this table to define administrator names and encrypted passwords, error message addresses, security settings, and conversation timeout settings.



2.2.1.1 Column Description

Table 2–1 Column Description for AXF_SYSTEM_PARAMETERS Table

Column	Description
NAMESPACE	Specifies the functional area that utilizes the parameter. ■ *AXF* namespace is used by AXF. ■ *AccountsPayable* is used by the AccountsPayable template. ■ *BPEL.default* specifies the name of the BPEL connection, where BPEL is a constant and default is the name of connection.
PARAMETER_TYPE	Name used to differentiate between parameters. Can be used to define custom types within a namespace. Additional parameters may be added as type *database*.

Table 2–1 (Cont.) Column Description for AXF_SYSTEM_PARAMETERS Table

Column	Description
PARAMETER_NAME	Name of the parameter. Used when retrieving the parameter value from the database. Parameters include: ■ ADMIN_PASSWORD: Specifies the BPEL Administrator's password (for the default connection). This parameter is stored as an encrypted string when surrounded with exclamation marks (!). For example, if your password is *password*, this column value should be *!password!*. The value is encrypted when first accessed by AxfConfigurationService. ■ ADMIN_USER: Specifies the BPEL Administrator's user id (for the default connection). ■ CONNECTION_PROVIDER: Defines the connection (BPEL or custom). If specifying a BPEL connection, this value is *AxfWorkflowServiceModule*. ■ ConversationTimeoutSeconds: Specifies the length of time for which a ConversationID (cid) is valid. The default is 43200 seconds of inactivity. ■ ERROR_URL: Generates the URLs for error messages generated by AXF. ■ IDENTITY_CONTEXT: Specifies the BPEL process' identity context (jazn.com). ■ IDENTITY_SERVICE_ENDPOINT: Specifies the URL point to BPEL identity Web services to query the defined users in BPEL. ■ PROVIDER_URL: Specifies the URL used to look up BPEL EJB services. ■ securityClass: Specifies the class name of a security provider for AXF. You can specify one of the following: *WebSecurityCheck* (default): Verifies that usernames provided for EBS and AXF match. *RequestorSecurityCheck*: Verifies that a valid username is provided for EBS. *NullSecurityCheck*: Bypasses any additional security check. To relax the restriction that the EBS and AXF usernames match, use the RequestorSecurityCheck, by specifying the following for the securityClass parameter value: `oracle.imaging.bai.axf.security.RequestorSecurityCheck` ■ WebSecurityCheckUrl: Specify the URL for the security check, if using WebSecurityCheck or RequestorSecurityCheck for the securityClass parameter. ■ USE_AUTOTASK_LOCKING: Specifies if autotask locking is enabled (TRUE) or disabled (FALSE). Enabling autotask locking can prevent collisions that may occur when multiple users are acquiring tasks in Autotask mode. See "Configuring Autotask Locking" on page 2-29.
PARAMETER_VALUE	Value of the parameter.
ENCRYPTION	Allows for the encryption of parameter values such as passwords. Possible values for this column include: ■ ENCRYPT: Causes the value in the parameter value column to be encrypted after the first time the configuration is loaded. After the encryption takes place, the column's value changes to ENCRYPTED. ■ ENCRYPTED: Denotes that the value in the PARAMETER_VALUE column is encrypted. ■ PLAIN_TEXT: Denotes that the value in the PARAMETER_VALUE column is in plain text.

2.2.1.2 Example Implementation

This example table displays system parameter setting for AXF and a BPEL connection.

Table 2–2 Example AXF_SYSTEM_PARAMETERS Table

NAMESPACE	PARAMETER_ TYPE	PARAMETER_NAME	PARAMETER_VALUE	ENCRYPTION
AXF	system	ConversationTimeoutSeconds	43200	PLAIN_TEXT
AXF	system	ERROR_URL	http://<ApplicationServerName>:<SOA-Port>/axf-web/faces/Error.jspx	PLAIN_TEXT
AXF	system	securityClass	oracle.imaging.bai.axf.security.WebSecurityCheck	PLAIN_TEXT
AXF	system	webSecurityCheckUrl	http://<ApplicationServerName>:<SOA-Port>/axf-web/faces/securityCheck.jspx	PLAIN_TEXT
default	bpel.connection	ADMIN_PASSWORD	Wi90KGqjZTM	ENCRYPTED
default	bpel.connection	ADMIN_USER	oc4jadmin	PLAIN_TEXT
default	bpel.connection	CONNECTION_PROVIDER	oracle.imaging.bai.axf.servicemodules.bpel.workflow.AxfWorkflowServiceModule	PLAIN_TEXT
default	bpel.connection	IDENTITY_CONTEXT	jazn.com	PLAIN_TEXT
default	bpel.connection	IDENTITY_SERVICE_ ENDPOINT	http://<ApplicationServerName>:<SOA-Port>/integration/services/IdentityService/identity	PLAIN_TEXT
default	bpel.connection	PROVIDER_URL	opmn:ormi://<ApplicationServerName>:<OPMNPort>:oc4j_soa/hw_services	PLAIN_TEXT
default	bpel.connection	USE_AUTOTASK_LOCKING	TRUE	PLAIN_TEXT

You can create another BPEL connection named *omicron*, as shown in Table 2–3.

Table 2–3 Example AXF_SYSTEM_PARAMETERS Table

NAMESPACE	PARAMETER _ TYPE	PARAMETER_NAME	PARAMETER_VALUE	ENCRYPTION
BPEL.omicron	connection	ADMIN_USER	oc4jadmin	PLAIN_TEXT
BPEL.omicron	connection	ADMIN_PASSWORD	Password	ENCRYPTED
BPEL.omicron	connection	IDENTITY_CONTEXT	jazn.com	PLAIN_TEXT
BPEL.omicron	connection	PROVIDER_URL	opmn:ormi://<ApplicationServerName>:<OPMNPort>:oc4j_soa/hw_Services	PLAIN_TEXT
BPEL.omicron	connection	IDENTITY_SERVICE_ ENDPOINT	http://<ApplicationServerName>:<SOA-Port>/integration/services/IdentityService/identity	PLAIN_TEXT

2.2.2 AXF_SOLUTIONS Table

The AXF_SOLUTIONS table defines the solutions used by AXF. It links to the AXF_COMMANDS Table via the SOLUTION_NAMESPACE column.

2.2.2.1 Column Description

Table 2–4 Column Description for AXF_SOLUTIONS Table

Column	Description
SOLUTION_CONTEXT	Defines the JNDI name of the AXF solution implementation. (Currently, AxfCommandMediator is the only solution implementation.)
SOLUTION_NAMESPACE	Defines the AXF solution name.

2.2.2.2 Example Implementation

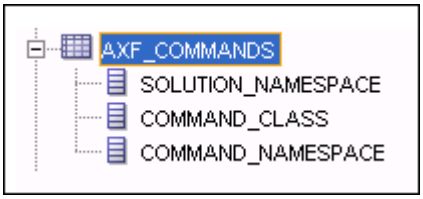
This example table shows the AXF solutions defined. Each of the solutions uses AxfCommandMediator as its solution implementation.

Table 2–5 Example AXF_SOLUTIONS Table

SOLUTION_NAMESPACE	SOLUTION_CONTEXT
InvoiceProcessing	AxfCommandMediator
AccountDistributionApproval	AxfCommandMediator
SupplierMaintenance	AxfCommandMediator
RequestInvoiceInformation	AxfCommandMediator
AccountDistribution	AxfCommandMediator
InvoiceApproval	AxfCommandMediator
Rescan	AxfCommandMediator

2.2.3 AXF_COMMANDS Table

Use this table to define AXF commands and their java classes for each solution. Note that you configure each command's parameters in the AXF_SOLUTION_PARAMETERS Table.



2.2.3.1 Column Description

Table 2–6 Column Description for AXF_COMMANDS Table

Column	Description
SOLUTION_NAMESPACE	The name of the solution, as defined in the AXF_SOLUTIONS Table.
COMMAND_NAMESPACE	Defines the unique name of the command within the solution.
COMMAND_CLASS	The fully qualified class name in which the command is defined. This class will be loaded and the execute() method representing the command will be executed. For information about a specific task, see the specific task, listed under "AXF Configuration Tables" on page 2-1.

2.2.3.2 Example Implementation

This example shows commands defined for the Invoice Processing solution.

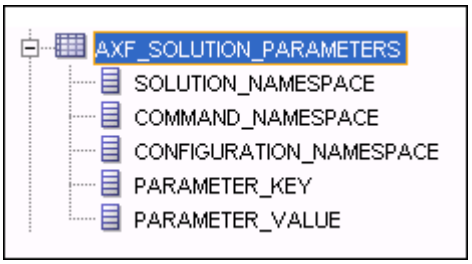
Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–7 Example AXF_COMMANDS Table

COMMAND_CLASS	COMMAND_NAMESPACE
oracle.imaging.axf.commands.bpel.AutotaskCommand	AutoOpenTask
oracle.imaging.axf.commands.bpel.CancelTaskCommand	CancelTask
oracle.imaging.axf.commands.bpel.CancelTaskCommand	SkipTask
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	AccountDistributionComplete
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	AssignProcessingGroupComplete
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	CompleteInvoice
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	DeleteInvoice
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	DuplicateInvoice
oracle.imaging.axf.commands.bepl.CompleteTaskCommand	Hold
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	InvoiceApprovalComplete
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	RequestInformationComplete
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	RescanComplete
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	SpecialistExceptionComplete
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	SupplierMaintenance
oracle.imaging.axf.commands.bpel.CompleteTaskCommand	SupplierMaintenanceComplete
oracle.imaging.axf.commands.bpel.OpenTaskCommand	OpenTask
oracle.imaging.axf.commands.bpel.UpdateTaskFromProcedureCommand	RetrieveUserList
oracle.imaging.axf.commands.bpel.UpdateTaskCommand	AttachSupplemental
oracle.imaging.axf.commands.bpel.UpdateTaskCommand	SaveInvoice
oracle.imaging.axf.commands.system.RedirectCommand	AccountDistributionEdit
oracle.imaging.axf.commands.system.RedirectCommand	AssignProcessingGroupEdit
oracle.imaging.axf.commands.system.RedirectCommand	EditComments
oracle.imaging.axf.commands.system.RedirectCommand	InvoiceApprovalEdit
oracle.imaging.axf.commands.system.RedirectCommand	RequestInformationEdit
oracle.imaging.axf.commands.system.RedirectCommand	RescanEdit
oracle.imaging.axf.commands.system.RedirectCommand	SearchIPM
oracle.imaging.axf.commands.system.RedirectCommand	SpecialistExceptionEdit
oracle.imaging.axf.commands.system.RedirectCommand	StartInvoiceProcessing
oracle.imaging.axf.commands.system.RedirectCommand	SupplierMaintenanceEdit
oracle.imaging.axf.commands.system.TerminateConversationCommand	TerminateConversation

2.2.4 AXF_SOLUTION_PARAMETERS Table

This table defines command parameters for the solution, AXF commands, and AXF web tools.



2.2.4.1 Column Description

Table 2–8 Column Description for AXF_SOLUTION_PARAMETERS Table

Column	Description
SOLUTION_NAMESPACE	Identifies the solution namespace, as defined in the AXF_SOLUTIONS Table.
COMMAND_NAMESPACE	Specifies the command name, as defined in the AXF_COMMANDS Table.
CONFIGURATION_NAMESPACE	Used to implement the command. Specify the complete package name of the implementation class. This namespace path provides the physical Java class to be instantiated. The namespace also differentiates commands within the same solution namespace.
PARAMETER_KEY	Specifies the parameter key to be used in the AXF command. For parameter details, see the specific command or web tool: Web Tools: ■ "Task List Web Tool" on page 2-13 ■ "Task Viewer Web Tool" on page 2-16 ■ "Enumeration Picker Web Tool" on page 2-21 ■ "Identity Picker Web Tool" on page 2-24 ■ "Comments Web Tool" on page 2-26 AXF Commands: ■ "Open Task Command" on page 2-28 ■ "Autotask Command" on page 2-28 ■ "Cancel Task Command" on page 2-29 ■ "Complete Task Command" on page 2-30 ■ "Redirect Command" on page 2-30 ■ "Update Task Command" on page 2-31 ■ "Update Task From Procedure Command" on page 2-32 ■ "Validate Task Command" on page 2-34
PARAMETER_VALUE	Specifies the value of the parameter key. (For parameter details, see the specific AXF command or web tool.) If the value has an XPATH: prefix, the attribute value will come from the AXF_XPATH_ATTRIBUTES Table.

2.2.4.2 Example Implementation

This example defines the StartInvoiceProcessing command for the Invoice Processing solution. The first row specifies that the task list be displayed, using the

RedirectCommand and corresponding URL. The remaining rows call the task list (in the CONFIGURATION_NAMESPACE column) and define its behavior.

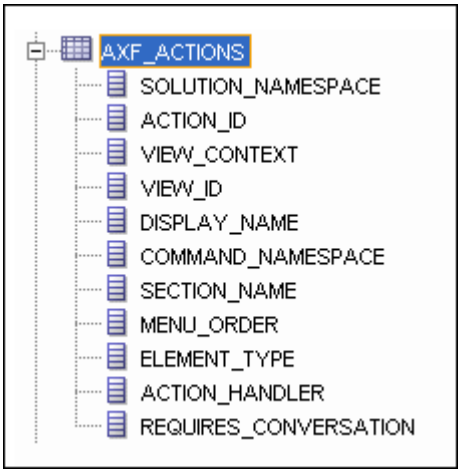
Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–9 Example AXF_SOLUTION_PARAMETERS Table for StartInvoiceProcessing Command

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
StartInvoiceProcessing	oracle.imaging.axf.commands.bpel.RedirectCommand	REDIRECT_URL	http://<ApplicationServerName>:<SOA-Port>/axf-web/faces/TaskList.jspx
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	CMD_OPEN_TASK_BUTTON	OpenTask
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	CMD_AUTO_TASK_BUTTON	AutoOpenTask
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	DEFAULT_VIEW	(null)
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	NO_OF_LINES	20
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	SHOW_INBOX	FALSE
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	CONNECTION_NAME	default
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	CMD_CANCEL_TASK_BUTTON	CancelTask
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	VIEW_LIST	North Invoice Processing Group, South Invoice Processing Group, East Invoice Processing Group, West Invoice Processing Group, My Holds,Exceptions

2.2.5 AXF_ACTIONS Table

This table defines the dynamic action menus used in an AXF solution. You can display action menus in a Task Viewer page (see "Task Viewer Web Tool" on page 2-16) or a Task List (see "Task List Web Tool" on page 2-13). This table links to the AXF_COMMANDS Table.



2.2.5.1 Column Description

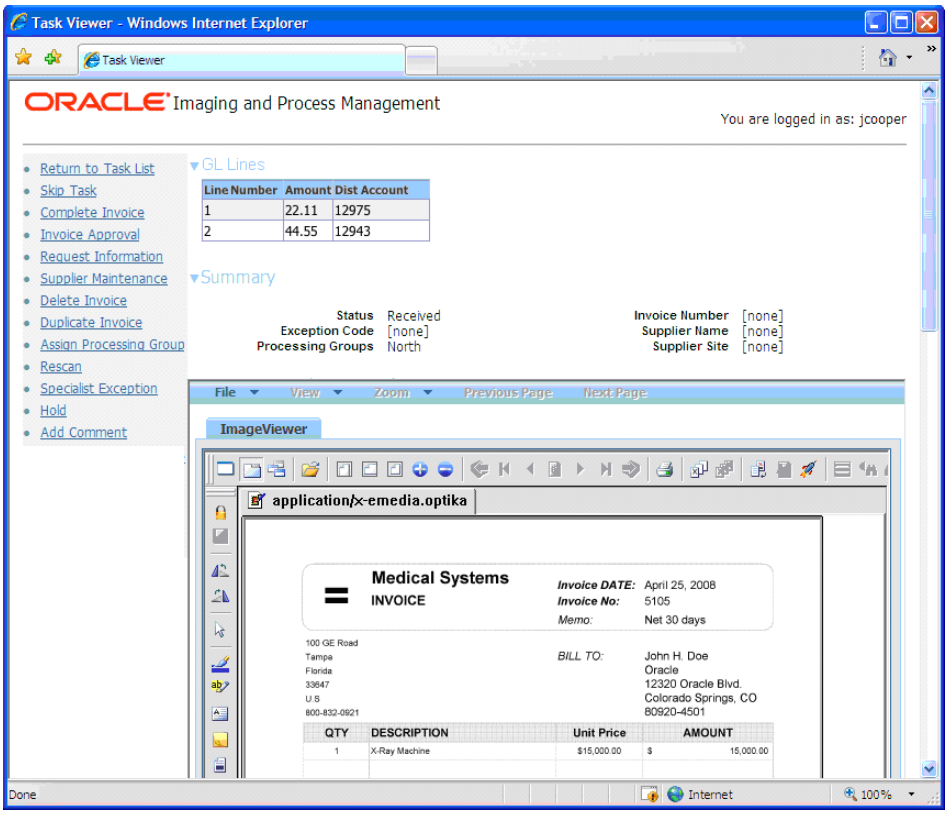
Table 2–10 Column Description for AXF_ACTIONS Table

Column	Description
SOLUTION_NAMESPACE	Identifies the solution namespace, as defined in the AXF_SOLUTIONS Table.
ACTION_ID	Specifies the name of the script action to take when the action link/button is selected on the JSP page.
VIEW_CONTEXT	Specifies the root web context of the web application in which the JSP page exists. In most cases, this value is /axf-web, but another context can be specified.
VIEW_ID	Specifies the name of the page on which to render the action. You can specify any AXF web tool, such as TaskViewer.jspx or CommentViewer.jspx.
DISPLAY_NAME	Specifies the name of the label on the button or link on the page.
COMMAND_NAMESPACE	Specifies the command that is called as a result of the action, as defined in the AXF_COMMANDS Table.
SECTION_NAME	(Reserved for future functionality.)
MENU_ORDER	Specifies the display order in the menu on the page.
ELEMENT_TYPE	Specifies how to render the action on the page, where: ■ LINK: Displays an HTML link ■ BUTTON: Displays a button
ACTION_HANDLER	Determines how the command is handled, where: ■ COMMAND: The command specified in the COMMAND_NAMESPACE column is sent to the Solution Mediator. ■ NONE: This type of command is not handled by AXF. These action types can be handled using Javascript or a custom action handler.
REQUIRES_CONVERSATION	Specifies whether the action requires a conversation ID.

2.2.5.2 Example Implementation

The tables that follow provide an example AXF_ACTIONS Table. The AXF action display names match the action list links shown in Table 2–2.

Figure 2–2 Action List Corresponding to AXF_ACTIONS Table



Fields not shown in Table 2–11:

- SOLUTION_NAMESPACE=InvoiceProcessing
- VIEW_CONTEXT=/axf-web
- SECTION_NAME=(null)
- ELEMENT_TYPE=LINK
- ACTION_HANDLER=COMMAND
- REQUIRES_CONVERSATION=TRUE

Table 2–11 Example AXF_ACTIONS Table

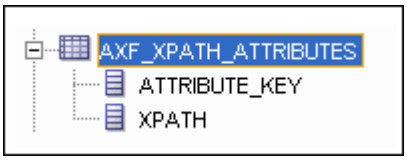
ACTION_ID	VIEW_ID	DISPLAY_NAME	COMMAND_NAMESPACE	MENU_ORDER
cancelTask	/TaskViewer.jspx	Return to Task List	skipTask	2
skipTask	/TaskViewer.jspx	Skip Task	cancelTask	1
CompleteInvoice	/TaskViewer.jspx	Complete Invoice	CompleteInvoice	3
InvoiceApproval	/TaskViewer.jspx	Invoice Approval	InvoiceApprovalEdit	4
RequestInformation	/TaskViewer.jspx	Request Information	RequestInformationEdit	5
SupplierMaintenance	/TaskViewer.jspx	Supplier Maintenance	SupplierMaintenance	6
DeleteInvoice	/TaskViewer.jspx	Delete Invoice	DeleteInvoice	7
DuplicateInvoice	/TaskViewer.jspx	Duplicate Invoice	DuplicateInvoice	8
AssignProcessingGroup	/TaskViewer.jspx	Assign Processing Group	AssignProcessingGroupEdit	9
Rescan	/TaskViewer.jspx	Rescan	RescanEdit	10

Table 2–11 (Cont.) Example AXF_ACTIONS Table

ACTION_ID	VIEW_ID	DISPLAY_NAME	COMMAND_NAMESPACE	MENU_ORDER
SpecialistException	/TaskViewer.jspx	Specialist Exception	SpecialistExceptionEdit	11
Hold	/TaskViewer.jspx	Hold	Hold	12
AddComment	/TaskViewer.jspx	Add Comment	EditComments	13

2.2.6 AXF_XPATH_ATTRIBUTES Table

This table defines the XPATH attributes used in the AXF framework. This XPATH is mainly defined for payload elements.



2.2.6.1 Column Description

Table 2–12 Column Description for AXF_XPATH_ATTRIBUTES Table

Column	Description
ATTRIBUTE_KEY	Attribute key referenced in the Parameter Value column in the AXF_SOLUTION_PARAMETERS Table.
XPATH	XPATH expression used to locate the value in the payload.

2.2.6.2 Example Implementation

This example follows an XPATH attribute specified for an AssignProcessingGroupEdit command in the AXF_SOLUTION_PARAMETERS table. The PARAMETER_VALUE column contains an XPATH: prefix, indicating that the attribute value will come from the AXF_XPATH_ATTRIBUTES table.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–13 Example AXF_SOLUTION_PARAMETERS Table

COMMAND_NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
AssignProcessingGroupEdit	oracle.imaging.bai.axf.web.backing.EnumerationPicker	ATTRIBUTE_NAME	XPATH:InvoiceProcessing_ProcessingGroup

In the AXF_XPATH_ATTRIBUTES table that follows, the corresponding XPATH column displays the XPATH expression used to locate the value in the payload.

Table 2–14 Example AXF_XPATH ATTRIBUTES Table

ATTRIBUTE_KEY	XPATH
InvoiceProcessing_ProcessingGroup	//task:processingGroup

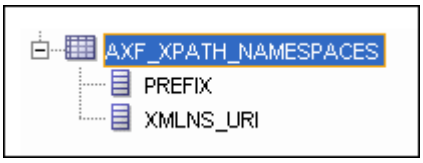
In the AXF_NAMESPACES table that follows, the XMLNS_URI column displays where within the XML file to locate the processingGroup task information.

Table 2–15 Example AXF_XPATH_NAMESPACES Table

Prefix	XMLNS_URI
task	http://xmlns.oracle.com/bpel/workflow/task

2.2.7 AXF_XPATH_NAMESPACES Table

The following table defines the namespaces used for the XPATH attributes. It links to the AXF_XPATH_ATTRIBUTES Table.



2.2.7.1 Column Description

Table 2–16 Column Description for AXF_XPATH_NAMESPACES Table

Column	Description
PREFIX	The namespace prefix used in the XPATH.
XMLNS_URI	Provides a unique identifier.

2.2.7.2 Example Implementation

Table 2–17 Example AXF_XPATH_NAMESPACES Table

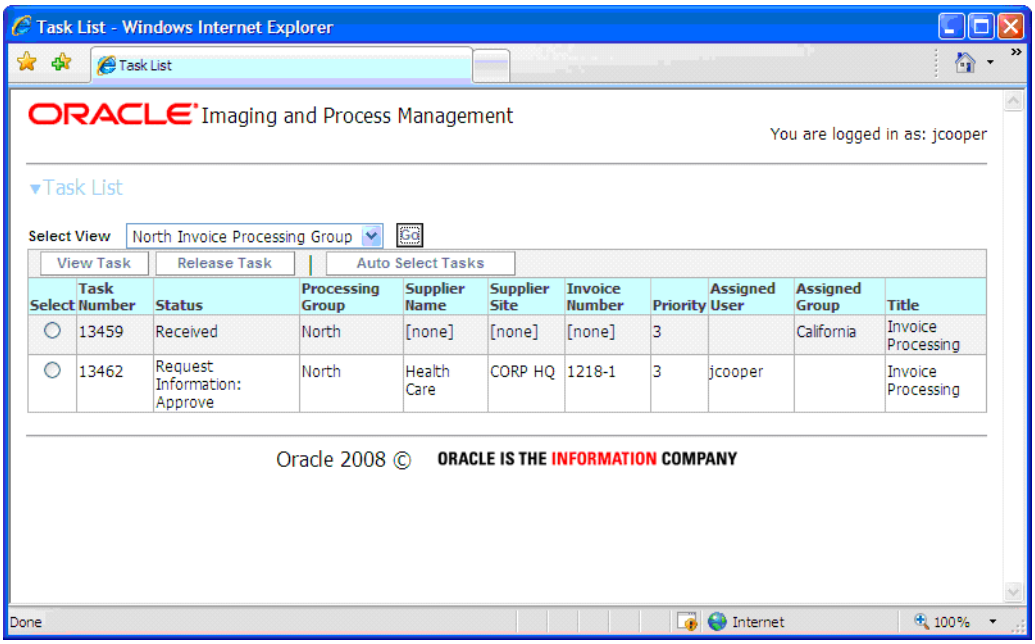
Prefix	XMLNS_URI
task	http://xmlns.oracle.com/bpel/workflow/task
documentContent	http://xmlns.oracle.com/imaging/axf/documentContentTypes
solution	http://xmlns.oracle.com/imaging/axf/solutionTypes
invoiceProcessing	http://xmlns.oracle.com/imaging/axf/InvoiceProcessing
xml	http://www.w3.org/XML/1998/namespace

2.3 AXF Web User Tools

This section covers the following topics:

2.3.1 Task List Web Tool

The AXF Task List web tool is a reusable web interface for displaying human workflow tasks controlled by an AXF solution.



Task List Features

- Users can select a task and open it by clicking **View Task**, which retrieves the workflow task from a specified BPEL server and process, claims it and displays it in the Task Viewer. After users complete the selected task, they return to the Task List.
- When autotask mode is selected, the AXF solution automatically claims and opens tasks as users complete them, until all of a user's tasks have been processed or the user chooses to stop processing tasks by returning to the Task List. Users activate autotask mode by clicking **Auto Select Tasks**.
- Users can skip (release) an assigned task, by clicking the **Release Task** button. The task is then released back into the pool of available tasks.
- The **Select View** option uses standard BPEL views to restrict the task list view based on user/group, BPEL Process versions, and BPEL payload attribute values. For information about uploading views, see "Configuring BPEL Task Views" on page A-1.
- You can configure the Task List to include a side pane action list with links that invoke a solution's AXF commands, as shown in the Task Viewer Web Tool.

2.3.1.1 Task List Parameters

Table 2–18 Task List Parameters in AXF_SOLUTION_PARAMETERS Table

Parameter	Description
CMD_OPEN_TASK_BUTTON	Specify a COMMAND_NAMESPACE to be executed when a user clicks the View Task button on the Task List web page.
CMD_AUTO_TASK_BUTTON	Specify a COMMAND_NAMESPACE to be executed when a user clicks the Auto Select Tasks button on the Task List web page.
CMD_CANCEL_TASK_BUTTON	Specify a COMMAND_NAMESPACE to be executed when a user clicks the Release Task button on the Task List web page?
CONNECTION_NAME	Specify the BPEL connection, as defined in the AXF_SYSTEM_PARAMETERS Table.
NO_OF_LINES	Specify the maximum number of tasks to be displayed before multiple pages are used.
BPEL_CONNECTION	Specify the BPEL connection that obtains the task list from BPEL, defined in the AXF_SYSTEM_PARAMETERS Table.
VIEW_LIST	Specify the list of views (defined in the Human workflow system) displayed to users in the Select View field.
DEFAULT_VIEW	Specify the default user view.
SHOW_INBOX	Specify whether the Inbox is listed in the view list. If TRUE, the Inbox is listed; if FALSE, the Inbox is not listed.
TASK_DEF	Specify the BPEL human workflow tasks to which the user has access. (For example, a value of *Rescan* means that Rescan tasks are displayed in the Inbox.) This parameter applies only when the SHOW_INBOX parameter is set to TRUE.
USE_CUSTOM_PAGINATION	If set to FALSE, allows for selecting specific rows in the task list (a dropdown with 1-10, 11-20, and 21-30, for example). If set to TRUE (default), allows for selecting normal previous/next controls.

2.3.1.2 Example Implementation

This example defines the StartInvoiceProcessing command for the Invoice Processing solution. The first row uses the Redirect Command to display the task list using the specified URL. The remaining rows define the task list's behavior.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–19 Example Task List Parameters in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_ NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
StartInvoiceProcessing	oracle.imaging.axf.commands.bpel.RedirectCommand	REDIRECT_URL	http://<ApplicationServerName>:<SOA-Port>/axf-web/faces/TaskList.jspx
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	CMD_OPEN_TASK_BUTTON	OpenTask
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	CMD_AUTO_TASK_BUTTON	AutoOpenTask
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	DEFAULT_VIEW	(null)

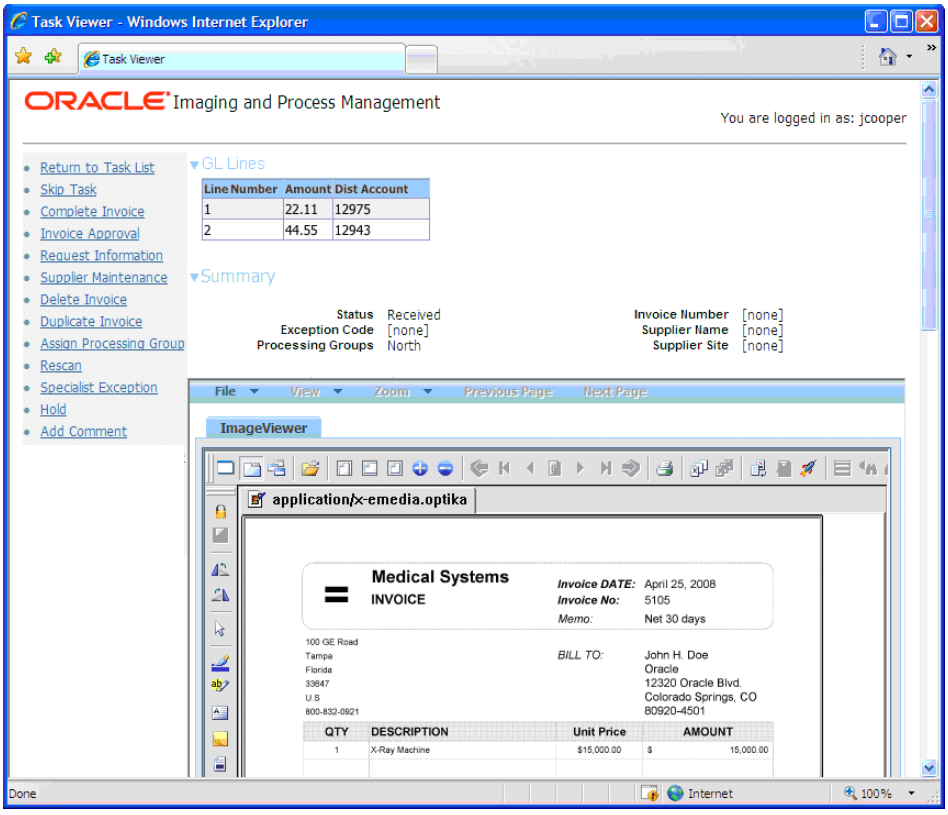
Table 2–19 (Cont.) Example Task List Parameters in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_ NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	NO_OF_LINES	20
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	SHOW_INBOX	TRUE
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	CONNECTION_NAME	default
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	CMD_CANCEL_TASK_BUTTON	CancelTask
StartInvoiceProcessing	oracle.imaging.bai.axf.web.backing.TaskList	VIEW_LIST	North Invoice Processing Group, South Invoice Processing Group, East Invoice Processing Group, West Invoice Processing Group, My Holds,Exceptions

2.3.2 Task Viewer Web Tool

The AXF Task Viewer web tool is a reusable web interface used to display the content associated with a Human Workflow Task. In a typical configuration, a business user displays the Task Viewer on one monitor, and keys values shown in the image into a business application on another monitor.

You can customize the web page via database configuration using AXF action commands.



Task Viewer Features

- Users view I/PM image documents in the Image Viewer section. Typically, the Task Viewer uses the I/PM public PAWSER viewer tool to render image

documents. However, another tool can be used; the Task Viewer uses whichever URL has been passed into the BPEL process by I/PM's BPEL Injector.

- Users can select actions in the side pane action list, which invoke a solution's AXF commands. You configure actions to invoke AXF commands in the AXF_ACTIONS Table, and the commands themselves in the AXF_COMMANDS Table. For example, clicking an **Add Comment link** displays the Comments web tool, allowing users to enter comments related to the task during the transaction's processing.
- Users can view a Summary section that displays metadata values about the task. You configure these items for display in the AXF_TASKVIEWER_ITEMS Table. You can also configure the section's title and the task payload values displayed.
- Users can also view a table of dynamic data from the BPEL payload XML (for example, General Ledger lines for an invoice processing solution). You configure the table in the AXF_TASKVIEWER_SECTIONS Table and its data lines in the AXF_TASKVIEWER_ITEMS Table. For information on formatting the XML file, see "Formatting XML Data For a Dynamic Data Table" on page 2-20.
- If autotask mode is selected, users disable it by returning to the Task List, typically by clicking a **Return to Task List** link.

Configuring the Task Viewer

You configure the Task Viewer in the following tables:

- AXF_SOLUTION_PARAMETERS Table
- AXF_TASKVIEWER_SECTIONS Table
- AXF_TASKVIEWER_ITEMS Table

2.3.2.1 Task Viewer Parameters

Table 2–20 Task Viewer Parameters in AXF_SOLUTION_PARAMETERS Table

Parameter	Description
SHOW_COMMENTS	If the value is TRUE, then the comments display in the Task Viewer page.
TASKVIEWER_CONFIG	Specify the name of the configuration to launch, as defined in the AXF_TASKVIEWER_SECTIONS Table.

Example Implementation

This example shows a Task Viewer configuration for the Invoice Processing solution. It includes settings for OpenTask and AutoOpenTask versions.

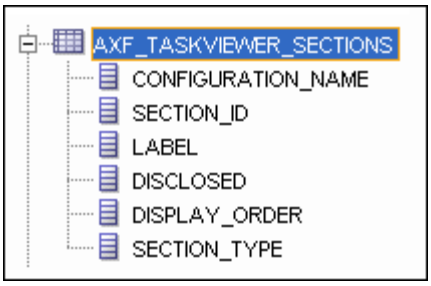
Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–21 Example Task Viewer Parameters in AXF_SOLUTION_PARAMETERS Table

COMMAND_NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
OpenTask	oracle.imaging.bai.axf.web.backing.TaskViewer	TASKVIEWER_CONFIG	InvoiceProcessingTaskViewer
AutoOpenTask	oracle.imaging.bai.axf.web.backing.TaskViewer	SHOW_COMMENTS	TRUE
AutoOpenTask	oracle.imaging.bai.axf.web.backing.TaskViewer	TASKVIEWER_CONFIG	InvoiceProcessingTaskViewer
OpenTask	oracle.imaging.bai.axf.web.backing.TaskViewer	SHOW_COMMENTS	TRUE

2.3.2.2 AXF_TASKVIEWER_SECTIONS Table

This table defines the task viewer itself and its sections to be displayed on the Task Viewer page.



Column Description

Table 2–22 Column Description for AXF_TASKVIEWER_SECTIONS Table

Column	Description
CONFIGURATION_NAME	Specify a name for the configuration. Configuration names are defined in the AXF_SOLUTION_PARAMETERS Table.
SECTION_ID	The section number in the Task Viewer page for a specific configuration name.
LABEL	Specify a name for the section to be displayed on the Task Viewer page.
DISCLOSED	(This column is reserved for future use.)
DISPLAY_ORDER	If the value is less then zero, the section is displayed above the Image Viewer section. If the value is greater then zero, then it is displayed below the Image Viewer section.
SECTION_TYPE	Specify the type of section to display. You can choose: ▪ **PanelHeader:** Includes a header for the section, named with the value contained in the Label column. ▪ **None:** Displays the section without a header. ▪ **BPELComments:** Displays a predefined section type that includes comments related to the task. Includes a header for the section, named with the value contained in the Label column. ▪ **table:** Displays a table of dynamic data from the BPEL payload XML. Includes a section header. See "Formatting XML Data For a Dynamic Data Table" on page 2-20. Note that this section is displayed only if there is data available for display.

Example Implementation

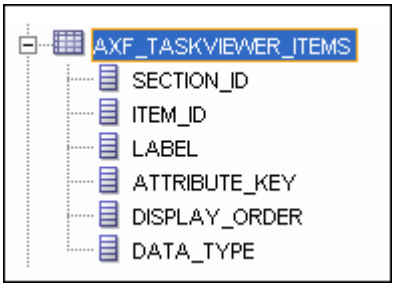
This example shows section settings for the InvoiceProcessingTaskViewer called in the AXF_SOLUTION_PARAMETERS table (see "Task Viewer Parameters" on page 2-17). It includes a Summary section and a Table section of dynamic data titled *GL Lines*. Their display order results in the table section appearing at the top, then the Summary section, then the Image Viewer.

Table 2–23 Example AXF_TASKVIEWER_SECTIONS Table

CONFIGURATION_NAME	SECTION_ID	LABEL	DISPLAY_ ORDER	SECTION_TYPE
InvoiceProcessingTaskViewer	1	Summary	-1	PanelHeader
InvoiceProcessingTaskViewer	2	GL Lines	-9	table

2.3.2.3 AXF_TASKVIEWER_ITEMS Table

This table defines section items for display on the Task Viewer page. It links to the AXF_TASKVIEWER_SECTIONS Table.



Column Description

Table 2–24 Column Description for AXF_TASKVIEWER_ITEMS Table

Column	Description
SECTION_ID	Specify an ID for the Task Viewer section.
ITEM_ID	Specify an ID for the item.
LABEL	Specify a name for the item for display on the Task Viewer page. When a dynamic data table is specified, this label becomes a column header.
ATTRIBUTE_KEY	This attribute is updated in the BPEL task when a user clicks the OK button on the Task Viewer page. The Attribute value is either a system attribute name (see "System Attributes" on page 2-31) or prefixed with XPATH, which uses the XPATH table to find the value. Only XPATH is supported for a dynamic data table. If the value has an XPATH: prefix, the value comes from the AXF_XPATH_ATTRIBUTES Table and is the XPATH to update the value in the task payload.
DISPLAY_ORDER	Specify the order in which to display the item on the Task Viewer page. For a dynamic data table, specify the column order.
DATA_TYPE	Specify the item's data type (for example, TEXT, TEXTMULTILINE, ACTION, and so on). This value must be TEXT for a dynamic data table.

Example Implementations

Table 2–25 shows items defined for the Summary section defined in the AXF_TASKVIEWER_SECTIONS Table (see example implementation). It specifies each item to include in the Summary section, its data type and display order, and where to locate the value via an XPATH variable.

Fields not shown include: DATA_TYPE=TEXT

Table 2–25 Example AXF_TASKVIEWER_ITEMS Table for Summary Section

SECTION_ID	ITEM_ID	LABEL	ATTRIBUTE_KEY	DISPLAY_ORDER
1	1	Status	XPATH:InvoiceProcessing_Status	0
1	2	Processing Groups	XPATH:InvoiceProcessing_ProcessingGroup	1
1	3	Exception Code	XPATH:InvoiceProcessing_ExceptionCode	2
1	4	Invoice Number	XPATH:InvoiceProcessing_InvoiceNumber	3
1	5	Supplier Name	XPATH:InvoiceProcessing_SupplierName	4
1	6	Supplier Site	XPATH:InvoiceProcessing_SupplierSiteName	5

Table 2–26 shows data lines configured for a dynamic table defined in the AXF_TASKVIEWER_SECTIONS Table (see example implementation). This example results in three data columns in the *GL Lines* table. It assumes that the XPATH attributes exist in the AXF_XPATH_ATTRIBUTES Table. For information about formatting the XML data in the BPEL payload, see "Formatting XML Data For a Dynamic Data Table" on page 2-20.

Fields not shown include: DATA_TYPE=TEXT

Table 2–26 Example AXF_TASKVIEWER_ITEMS Table for Dynamic Table Section

SECTION_ID	ITEM_ID	LABEL	ATTRIBUTE_KEY	DISPLAY_ORDER
2	8	Line Number	XPATH:DistributionLines_LineNumber	0
2	9	Dist Account	XPATH:DistributionLines_DistributionAccount	1
2	10	Amount	XPATH:DistributionLines_Amount	2

2.3.2.4 Formatting XML Data For a Dynamic Data Table

After adding the table in the AXF_TASKVIEWER_SECTIONS Table and configuring its data lines in the AXF_TASKVIEWER_ITEMS Table, follow the guidelines below to ensure that the XML data in the BPEL payload is correctly formatted for display in the table.

Below is an XML sample for display at any level within the XML payload. The *First column XPATH* retrieves the parent and its peer elements (*collectionItem*). Each of the configured XPATHs point to an *itemValue* element used to retrieve the cell values for the table from each collectionItem.

> **Note:** All columns must display within the same direct parent element.

```
<rootElement>
  <collectionContainerElement>
    <collectionItem>                                    <--First row for table
      <itemValue1>value1</itemValue1>                   <--First column XPATH
      <itemValue2>value2</itemValue2>
      <itemValue3>value3</itemValue3>
    </collectionItem>
    <collectionItem>                                    <--Second row
      <itemValue1>value1</itemValue1>
      <itemValue2>value2</itemValue2>
      <itemValue3>value3</itemValue3>
```

```
    </collectionItem>
    <collectionItem>
      <itemValue1>value1</itemValue1>
      <itemValue2>value2</itemValue2>
      <itemValue3>value3</itemValue3>
    </collectionItem>
  </collectionContainerElement>
</rootElement>
```

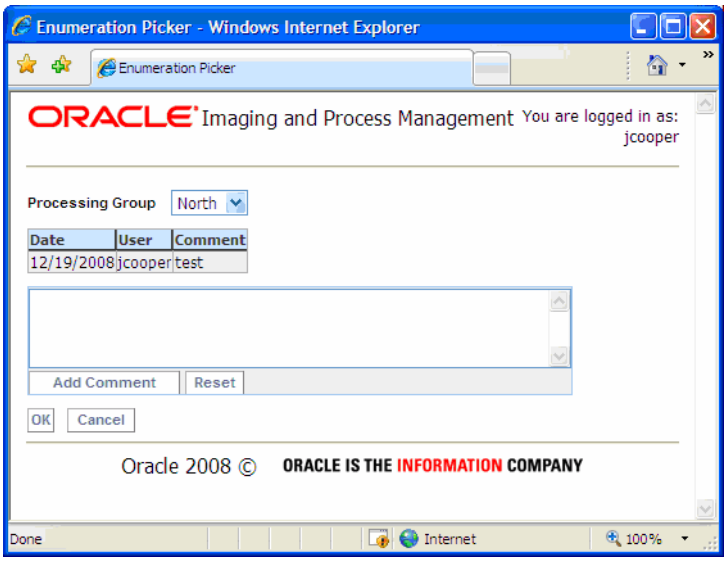
The first column XPATH for the above XML should be similar to the following:

```
/task:payload/task:rootElement/collectionContainerElement/collectionItem/itemValue1
```

2.3.3 Enumeration Picker Web Tool

The Enumeration Picker web tool allows users to select a choice from a list of values configured in AXF tables and to enter a comment related to the transaction's processing. For example, the Enumeration Picker shown in the graphic that follows displays a Processing Group dropdown field containing North, South, East, and West values, with North as the current value.

After the user selects a value, the value and comment (if provided) are updated before the configured command is executed, typically a command to return to the Task Viewer or to complete the command.



You configure the Enumeration Picker in the following tables:

- Define configuration parameters for the picker in the AXF_SOLUTION_PARAMETERS table (see Table 2–27).
- Define the picker in the AXF_ENUM_TYPES Table.
- Define the picker's values in the AXF_TASKVIEWER_ITEMS Table.

2.3.3.1 Enumeration Picker Parameters

Table 2–27 Enumeration Picker Parameters in AXF_SOLUTION_PARAMETERS Table

Parameter	Description
LOV_REFERENCE	This list of values reference links to the AXF_ENUM_TYPES Table, whose ID value links to the AXF_ENUM_ITEMS Table, where all picker values are stored.
ATTRIBUTE_NAME	This attribute is updated in the BPEL task when a user clicks the OK button on the Enumeration Picker web page. The attribute value is a constant; see "System Attributes" on page 2-31. If the value has an XPATH: prefix, the value comes from the AXF_XPATH_ATTRIBUTES Table and it is the XPATH to update the value in the task payload.
CMD_ON_CANCEL	Specify the command (COMMAND_NAMESPACE) to be executed when a user clicks the Cancel button on the Enumeration Picker page.
CMD_ON_OK	Specify the command (COMMAND_NAMESPACE) to be executed when a user clicks the OK button on the Enumeration Picker page.
ATTRIBUTE_LABEL	Specify the label name to display on the web page for attributes to be updated in the BPEL task.
DEFAULT_VALUE	Specify a default value for the picker. If no default is specified, a None value is displayed.
DEFAULT_ALWAYS	Specify TRUE to always show the value specified in the DEFAULT_VALUE parameter when displaying the Enumeration Picker, even if another value was previously selected. Otherwise, specify FALSE.

Example Implementation

This example shows an enumeration picker referenced for selecting the processing group.

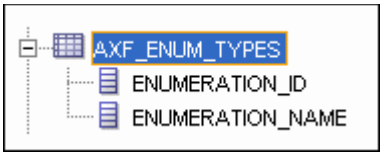
Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–28 Example Enumeration Picker Parameters in AXF_SOLUTIONS Table

COMMAND_NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
AssignProcessingGroupEdit	oracle.imaging.bai.axf.web.backing.EnumerationPicker	LOV_REFERENCE	ProcessingGroups
AssignProcessingGroupEdit	oracle.imaging.bai.axf.web.backing.EnumerationPicker	ATTRIBUTE_LABEL	Processing Group
AssignProcessingGroupEdit	oracle.imaging.bai.axf.web.backing.EnumerationPicker	CMD_ON_OK	AssignProcessingGroupComplete
AssignProcessingGroupEdit	oracle.imaging.bai.axf.web.backing.EnumerationPicker	ATTRIBUTE_NAME	XPATH:InvoiceProcessing_ProcessingGroup
AssignProcessingGroupEdit	oracle.imaging.bai.axf.web.backing.EnumerationPicker	CMD_ON_CANCEL	OpenTask
AssignProcessingGroupEdit	oracle.imaging.bai.axf.web.backing.EnumerationPicker	DEFAULT_VALUE	North
AssignProcessingGroupEdit	oracle.imaging.bai.axf.web.backing.EnumerationPicker	DEFAULT_ALWAYS	FALSE

2.3.3.2 AXF_ENUM_TYPES Table

This table defines Enumeration Pickers.



Column Description

Table 2–29 Column Description for AXF_ENUM_TYPES Table

Column	Description
ENUMERATION_ID	Specify an ID for the enumeration picker.
ENUMERATION_NAME	Specify a name for the enumeration picker configuration.

Example Implementation

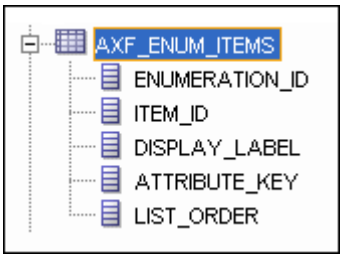
This example defines enumeration pickers for the Invoice Processing solution. Each picker's items are defined in the AXF_ENUM_ITEMS Table.

Table 2–30 Example AXF_ENUM_TYPES Table

ENUMERATION_ID	ENUMERATION_NAME
1	ProcessingGroups
2	SupplierMaintenanceCodes
3	SpecialistExceptionCodes
4	RescanCodes

2.3.3.3 AXF_ENUM_ITEMS Table

This table defines a specified Enumeration Picker's values.



Column Description

Table 2–31 Column Description for AXF_ENUM_ITEMS Table

Columns	Description
ENUMERATION_ID	Specify the picker's ID, as defined in the AXF_ENUM_TYPES Table.
ITEM_ID	Specify an ID for the picker item.
DISPLAY_LABEL	Specify the item name to be displayed in the picker field.

Table 2–31 (Cont.) Column Description for AXF_ENUM_ITEMS Table

Columns	Description
ATTRIBUTE_KEY	Specify the value to be stored in the payload. This value is often the same as the DISPLAY_LABEL's value, but can differ.
LIST_ORDER	Specify the order in which the value is to be listed in the picker field.

Example Implementation

This example defines the items for the ProcessingGroups, SupplierMaintenanceCodes, SpecialistExceptionCodes, and RescanCodes enumeration pickers defined in the AXF_ENUM_TYPES Table.

Table 2–32 Example AXF_ENUM_ITEMS Table

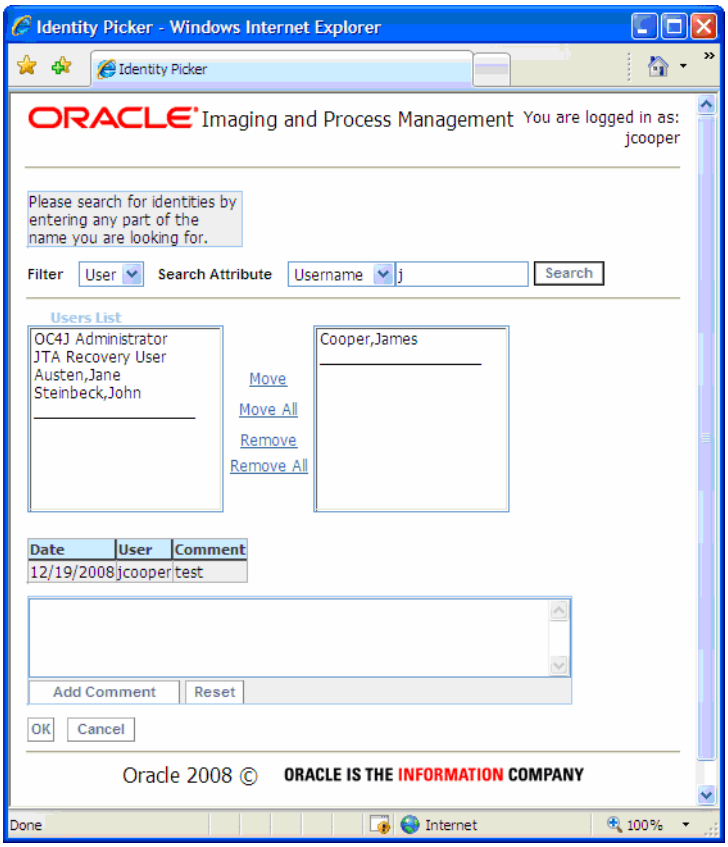
ENUMERATION_ID	ITEM_ID	DISPLAY_LABEL	ATTRIBUTE_KEY	LIST_ORDER
1	1	North	North	1
1	2	South	South	2
1	3	East	East	3
1	4	West	West	4
2	5	No Supplier	No Supplier	1
2	6	No Supplier Site	No Supplier Site	2
3	7	Duplicate Invoice	Duplicate Invoice	1
3	8	Invalid Invoice Number	Invalid Invoice Number	2
3	9	No PO	No PO	3
3	10	Invalid PO	Invalid PO	4
3	11	PO Overbill	PO Overbill	5
4	12	Poor Image Quality	Poor Image Quality	1
4	13	Pages Out of Order	Pages Out of Order	2
4	14	Pages Missing	Pages Missing	3
4	15	Other	Other	4

2.3.4 Identity Picker Web Tool

The Identity Picker web tool allows users to select one or more users or groups from an identity store configured for BPEL, and to add comments. Typically, a related action is taken after choosing an identity; for example, a task is assigned or delegated. The action to be taken after selecting an identity is configured in the AXF_SOLUTION_PARAMETERS Table.

Note: The command updates the task payload when the user clicks the OK button. The BPEL process is responsible for using this information to delegate the task.

Note: The Filter and Search Attribute settings utilize the BPEL Worklist views configuration. Changing these settings is done via the BPEL Workflow application.



2.3.4.1 Identity Picker Parameters

Table 2–33 Identity Picker Parameters in AXF_SOLUTION_PARAMETERS Table

Parameter Key	Description
CMD_ON_CANCEL	Specify the command (COMMAND_NAMESPACE) to be executed when a user clicks the Cancel button on the Identity Picker page.
CMD_ON_OK	Specify the command (COMMAND_NAMESPACE) to be executed when a user clicks the OK button on the Identity Picker page.
IDENTITY_FILTER	Define how the identity picker searches, where: ■ USER: The picker searches for user information defined in BPEL. ■ GROUP: The picker searches for group information defined in BPEL.
IDENTITY_ATTRIBUTE	This attribute is updated in the BPEL task when a user clicks the OK button on the Identity Picker page. The Attribute value is a constant as defined under "System Attributes" on page 2-31. If the value has an XPATH: prefix, then the value will come from the AXF_XPATH_ATTRIBUTES Table and it will be the XPATH to update the value in the task payload.

Use the following parameter to configure Comments in the Identity Picker.

Table 2–34 Identity Picker Comments in AXF_SOLUTION_PARAMETERS Table

Parameter Key	Description
COMMENT_READONLY	If TRUE, the comment is read only and users cannot add a comment. If FALSE, users can add a comment.

2.3.4.2 Example Implementation

This example shows an InvoiceApprovalEdit command that searches for user information stored in BPEL, and updates the BPEL task via an XPATH variable. The last row shows a comments configuration that allows users to add a comment to the transaction's processing.

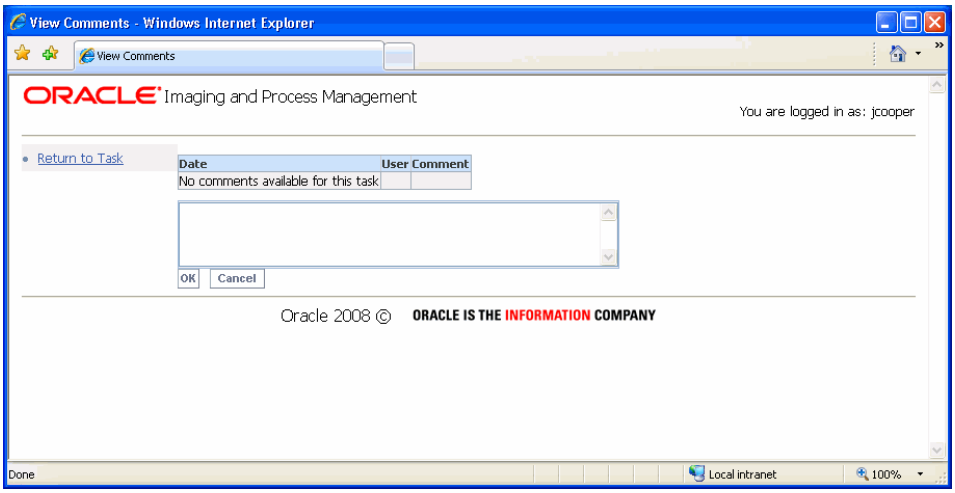
Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–35 Example Identity Picker Parameters in AXF_SOLUTION_PARAMETERS table

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
InvoiceApprovalEdit	oracle.imaging.bai.axf.web.backing.IdentityPicker	IDENTITY_FILTER	USER
InvoiceApprovalEdit	oracle.imaging.bai.axf.web.backing.IdentityPicker	CMD_ON_CANCEL	OpenTask
InvoiceApprovalEdit	oracle.imaging.bai.axf.web.backing.IdentityPicker	IDENTITY_ ATTRIBUTE	XPATH:InvoiceProcessing_ InvoiceApprovalAssignment
InvoiceApprovalEdit	oracle.imaging.bai.axf.web.backing.IdentityPicker	CMD_ON_OK	InvoiceApprovalComplete
InvoiceApprovalEdit	oracle.imaging.bai.axf.web.backing.Comments	COMMENT_ READONLY	FALSE

2.3.5 Comments Web Tool

The Comments web page allows users to enter comments related to the human task during the transaction's processing.



2.3.5.1 Comments Parameters

Table 2–36 Comments Parameters in AXF_SOLUTION_PARAMETERS Table

Parameter Key	Description
COMMENT_READONLY	If TRUE, the comment is read only and users cannot add a comment. If FALSE, users can add a comment.

2.3.5.2 Example Implementation

The following rows set comments functionality for the edit commands.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–37 Example Comments in AXF_SOLUTION_PARAMETERS Table

COMMAND_NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
RequestInformationEdit	oracle.imaging.bai.axf.web.backing.Comments	COMMENT_READONLY	FALSE
InvoiceApprovalEdit	oracle.imaging.bai.axf.web.backing.Comments	COMMENT_READONLY	FALSE
EditComments	oracle.imaging.bai.axf.web.backing.Comments	COMMENT_READONLY	FALSE
RescanEdit	oracle.imaging.bai.axf.web.backing.Comments	COMMENT_READONLY	FALSE

2.3.6 Modifying the Header and Footer of AXF Web Tools

Follow these steps to change the header and footer displayed in each AXF Web tool.

1. Make sure the AXF application is deployed to the Application Server.
2. Locate the *header.jspx* and *footer.jspx* files. These files are available at Application Server in the following directory:
$APP_SERVER/j2ee/home/applications/ImagingBaiaxf/imaging-bai-axf-web/fragments
3. Make a backup copy of the files.
4. Open the files in JDeveloper, modify their contents, and save the files in the same location.

> **Note:** You must use either the Oracle ADF component or the Java Faces component when modifying the footer and header file. If errors occur while compiling the file, you will not be able to view the Web page.

2.4 AXF Commands

AXF commands include:

- "Open Task Command" on page 2-28
- "Autotask Command" on page 2-28
- "Cancel Task Command" on page 2-29
- "Complete Task Command" on page 2-30
- "Redirect Command" on page 2-30
- "Update Task Command" on page 2-31
- "Update Task From Procedure Command" on page 2-32
- "Terminate Conversation Command" on page 2-34
- "Validate Task Command" on page 2-34

AXF command-related topics include:

- "Custom Commands" on page 2-36
- "Configuring Chained Commands and Web Tools" on page 2-36

2.4.1 Open Task Command

This command acquires a task from BPEL (human work flow) for a given task ID. If the task is already assigned to a user, the command obtains the details of the task and displays the Task List web page.

2.4.1.1 Open Task Command Parameters

Table 2–38 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

Table 2–38 Parameters for OpenTask Command

Parameter Key	Description
TASK_VIEW_URL	This URL is returned in the response command upon executing this command.

2.4.1.2 Example Implementation

This example uses the OpenTask command to display the Task Viewer for the Invoice Processing solution.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–39 Example Open Task Command in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
OpenTask	oracle.imaging.axf.commands.bpel. OpenTaskCommand	TASK_VIEW_URL	http://<ApplicationServerName>:<SOA-Port>/ axf-web/faces/TaskViewer.jspx

2.4.2 Autotask Command

This command initializes autotask mode, in which a new human workflow task is automatically claimed for the user in the AXF Task Viewer without displaying the Task List.

2.4.2.1 Autotask Command Parameters

These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure Autotask commands.

Table 2–40 Autotask Command Parameters in AXF_SOLUTION_PARAMETERS Table

Parameter Key	Description
TASK_VIEW_URL	URL returned in the response command upon executing this command.
CMD_ON_NO_TASKS	COMMAND_NAMESPACE executed when there are no tasks.
BPEL_TRY_AUTO	Time in milliseconds between attempts to get the next task from the Human workflow system.

2.4.2.2 Example Implementation

This example uses the Autotask command to automatically claim tasks and display them in the Task Viewer for the Invoice Processing solution.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–41 Autotask Command Parameters in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
AutoOpenTask	oracle.imaging.axf.commands.bpel. AutotaskCommand	TASK_VIEW_URL	http://<ApplicationServerName>:<SOA-Port>/ axf-web/faces/TaskViewer.jspx
AutoOpenTask	oracle.imaging.axf.commands.bpel. AutotaskCommand	CMD_ON_NO_TASKS	StartInvoiceProcessing
AutoOpenTask	oracle.imaging.axf.commands.bpel. AutotaskCommand	BPEL_TRY_AUTO	3000

2.4.2.3 Configuring Autotask Locking

In AXF configurations with multiple simultaneous users, collisions may occur when attempting to acquire tasks in Autotask mode. You can enable or disable autotask locking for each named BPEL connection in the AXF database. When locking is enabled, only one user may automatically acquire a task at a given time.

Enabling the lock functionality will prevent an error from appearing on the Task List if two users acquire a task simultaneously, and is the recommended setting. In situations where simultaneous acquisition is unlikely, disabling the lock functionality may increase performance.

The setting is configured in the AXF_SYSTEM_PARAMETERS Table by inserting the following row:

NAMESPACE	PARAMETER_ TYPE	PARAMETER_NAME	PARAMETER_VALUE
BPEL.default	connection	USE_AUTOTASK_LOCKING	true

2.4.3 Cancel Task Command

The Cancel command releases a human workflow task and displays the AXF Task List web tool, regardless of Autotask mode.

2.4.3.1 Cancel Task Command Parameters

Table 2–42 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

Table 2–42 Cancel Task Command Parameters

Parameter Key	Description
CMD_AUTOTASK_OFF	Specify the command (COMMAND_NAMESPACE) to be executed when AUTOTASK mode is off.
CMD_AUTOTASK_ON	Specify the command (COMMAND_NAMESPACE) to be executed when AUTOTASK mode is on.

2.4.3.2 Example Implementation

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–43 Cancel Task Commands in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
SkipTask	oracle.imaging.axf.commands.bpel.CancelTaskCommand	CMD_AUTOTASK_OFF	StartInvoiceProcessing
SkipTask	oracle.imaging.axf.commands.bpel.CancelTaskCommand	CMD_AUTOTASK_ON	AutoTaskOpen
CancelTask	oracle.imaging.axf.commands.bpel.CancelTaskCommand	CMD_AUTOTASK_OFF	StartInvoiceProcessing
CancelTask	oracle.imaging.axf.commands.bpel.CancelTaskCommand	CMD_AUTOTASK_ON	StartInvoiceProcessing

2.4.4 Complete Task Command

The Complete Task command updates the list of attributes and outcome for a specified task in the human task workflow. This command also takes the parameters defined for the Update Task Command.

2.4.4.1 Complete Task Command Parameters

Table 2–44 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

Table 2–44 CompleteTask Command Parameters

Parameter Key	Description
OUTCOME	Specify the outcome defined for the human work flow system (for example, APPROVED or REJECT).
CMD_AUTOTASK_ON	Specify the command (COMMAND_NAMESPACE) to be executed when AUTOTASK mode is on.
CMD_AUTOTASK_OFF	Specify the command (COMMAND_NAMESPACE) to be executed when AUTOTASK mode is off.

2.4.4.2 Example Implementation

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–45 Example Complete Task Command in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
DeleteInvoice	oracle.imaging.axf.command.bpel.CompleteTaskCommand	CMD_AUTOTASK_OFF	StartInvoiceProcessing
DeleteInvoice	oracle.imaging.axf.command.bpel.CompleteTaskCommand	CMD_AUTOTASK_ON	AutoOpenTask
DeleteInvoice	oracle.imaging.axf.command.bpel.CompleteTaskCommand	OUTCOME	DELETE_INVOICE

2.4.5 Redirect Command

The Redirect command redirects the browser to an AXF web tool or other URL. The request parameters included in this URL are:

- CID (Conversation ID)
- PID (ParameterSet ID)

Any user defined request parameters should be stored as part of the PID.

The base URL comes from the database configuration. This command returns the URL in the response command.

Table 2–46 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

2.4.5.1 Redirect Command Parameters

Table 2–46 RedirectCommand Parameters

Parameter Key	Description
REDIRECT_URL	This URL is returned in the response command upon executing this command.
EXTERNAL	If this has a value of TRUE, then the redirect page does not have a CID and PID appended to it. The re-direct URL is an external Web site and all request parameters are appended in the URL.

2.4.5.2 Example Implementation

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–47 Example Redirect Command in AXF_SOLUTION_PARAMETERS Table

COMMAND_NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
SearchIPM	oracle.imaging.axf.commands.system.RedirectCommand	EXTERNAL	TRUE

2.4.6 Update Task Command

The Update Task command updates the list of attributes in the BPEL task or updates values in the XML payload using XPATH. (For an XPATH example, see "Example Implementation" on page 2-32.)

You can create your own parameter keys for the Update Task command and use either a system attribute or an XPATH for the parameter value. AXF searches the request parameters and finds all the values that match the parameter keys (besides outcome), and pulls parameter keys for the list of attributes to use in that task payload.

To update a non-payload attribute in the BPEL task, use a system attribute listed in Table 2–49. For example, the UpdateTask command can take the value of `outcome` (defined as PARAMETER_KEY) from the request parameter and update the `OUTCOME` (defined as PARAMETER_VALUE) attribute value in the task.

2.4.6.1 Update Task Parameters

Table 2–48 Parameters for UpdateTaskCommand

Parameter Key	Description
outcome	Specify the outcome defined for the human work flow system (for example, APPROVED or REJECT).

2.4.6.2 System Attributes

Table 2–49 System Attributes

ACQUIREDBY
APPROVERS
ASSIGNEDDATE
ASSIGNEDGROUP //Cannot be updated
ASSIGNEDUSER //Cannot be updated
CREATEDATE

Table 2–49 (Cont.) System Attributes

CREATOR
DATEATTRIBUTE1-DATEATTRIBUTE5
EXPIREDDATE
ENDDATE
FORMATTRIBUTE1-FORMATTRIBUTE5
FROMUSER
NUMBERATTRIBUTE1-NUMBERATTRIBUTE5
OUTCOME
OWNERGROUP
OWNERUSER
PRIORITY
STATE
TASKID
TASKNUMBER //Cannot be updated
TITLE
TASKDEFINITIONNAME
TEXTATTRIBUTE1-TEXTATTRIBUTE10
UPDATEDBY
URLATTRIBUTE1 - URLATTRIBUTE5

2.4.6.3 Example Implementation

This XPATH example updates `transactionID` in the payload: the parameter key `InvoiceTransactionID` is the key defined in the request parameter. The value is `XPATH:TransactionID` where XPATH defines that the attribute `TransactionID` is defined in the AXF_XPATH_ATTRIBUTES Table.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–50 Example UpdateTaskCommand Parameters in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
AttachSupplemental	oracle.imaging.axf.commands.bpel. UpdateTaskCommand	InvoiceTransactionID	XPATH:InvoiceProcessing_TransactionID
SaveInvoice	oracle.imaging.axf.commands.bpel. UpdateTaskCommand	InvoiceTransactionID	XPATH:InvoiceProcessing_TransactionID

2.4.7 Update Task From Procedure Command

The Update Task From Procedure command calls a stored pl/sql procedure using a specified data source and returns results using XPATH.

2.4.7.1 Update Task From Procedure Parameters

Table 2–51 Parameters for UpdateTaskFromProcedure Command

Parameter Key	Description
XPATH_USERS	Specifies an XPATH variable contained in the AXF_XPATH_ATTRIBUTES Table that refers to the XPATH where the list of users is to be stored.
CMD_EMPTY_LIST	Specifies the command to be executed if no results are returned from the pl/sql function.
CMD_NON_EMPTY_LIST	Specifies the command to be executed if results are returned from the pl/sql function.
JNDI_DS	Specifies the name of the JNDI data source, configured on the Application Server, to use for execution of the pl/sql function.
PLSQL_PROC	Specifies the name of the pl/sql function to call.

2.4.7.2 Example Implementation

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–52 Example UpdateTaskFromProcedureCommand Parameters in AXF_SOLUTION_PARAMETERS Table

COMMAND_NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
RetrieveUserList	oracle.imaging.axf.commands.bpel.UpdateTaskFromProcedureCommand	XPATH_USERS	XPATH:InvoiceProcessing_InvoiceApprovalAssignment
RetrieveUserList	oracle.imaging.axf.commands.bpel.UpdateTaskFromProcedureCommand	CMD_NON_EMPTY_LIST	InvoiceApprovalEdit
RetrieveUserList	oracle.imaging.axf.commands.bpel.UpdateTaskFromProcedureCommand	CMD_EMPTY_LIST	CompleteInvoice
RetrieveUserList	oracle.imaging.axf.commands.bpel.UpdateTaskFromProcedureCommand	JNDI_DS	jdbc/AXFEBS11DataSource
RetrieveUserList	oracle.imaging.axf.commands.bpel.UpdateTaskFromProcedureCommand	PLSQL_PROC	AXFRETRIEVEUSERLIST

2.4.7.3 Example PL/SQL Procedure

The pl/sql procedure that follows loads the xml into the DOM, retrieves the invoice ID, queries for the invoice amount for that transaction, and based on that amount, returns a set of users.

To use this example, modify this procedure to retrieve the specific pieces of data from the payload you would like. The only requirement is that the pl/sql function you create must take a VARCHAR2 and return a VARCHAR2. The name of the function is in the AXF configuration.

```
create or replace FUNCTION axfretrieveuserlist(  xmlPayload IN VARCHAR2 ) RETURN VARCHAR2 IS

    v_node    xmldom.DOMNode;
    v_node2   xmldom.DOMNode;
    v_nl      xmldom.DOMNodeList;
    v_doc     xmldom.DOMDocument;
    v_elem    xmldom.DOMElement;
    v_parser  xmlparser.Parser;
    invoiceID     VARCHAR2(256);
    invoiceAmount NUMBER(8,2);
    userList      VARCHAR2(256);
```

```
BEGIN

    v_parser := xmlparser.newParser;
    xmlparser.parseBuffer(v_parser, xmlPayload);
    v_doc := xmlparser.getDocument(v_parser);
    xmlparser.freeParser(v_parser);

    -- Retrieve the invoice ID
    v_nl := xmldom.getElementsByTagName(v_doc, 'invoiceID');
    v_node := xmldom.item(v_nl, 0);
    v_node2 := xmldom.getFirstChild(v_node);
    invoiceID := xmldom.getNodeValue(v_node2);

    -- Retrieve Invoice Amount for given invoice id
    select INVOICE_AMOUNT into invoiceAmount from ap_invoices_all where INVOICE_ID = invoiceid;

    if invoiceamount > 10000 then
      userList := 'jlondon';
    else
      userList := 'jcooper,mtwain';
    end if;

    RETURN userList;

END;
```

2.4.8 Terminate Conversation Command

The Terminate Conversation Command is used by an external client to terminate a conversation with AXF.

2.4.9 Validate Task Command

The Validate Task command provides a means of validating BPEL system attribute data or BPEL payload data, and based on validation results, executing a subsequent command.

Table 2–53 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

2.4.9.1 Validate Task Command Parameters

Table 2–53 ValidateTaskCommand Parameters

Parameter Key	Description
ATTRIBUTE_TO_VALIDATE	Specifies the attribute in the BPEL task to validate. This can be either a system attribute or a payload attribute. If specifying a payload attribute, use a prefix value of *XPATH:* and reference a value from the AXF_XPATH_ATTRIBUTES Table.
REGULAR_EXPRESSION	Defines a standard Regular Expression for validating the specified attribute.

Table 2–53 (Cont.) ValidateTaskCommand Parameters

Parameter Key	Description
CMD_ON_PASS	Specifies the command to execute after this command, if the validation is successful.
CMD_ON_FAIL	Specifies the command to execute after this command if the validation fails. If this PARAMETER_KEY is provided, the FAIL_MESSAGE is not displayed.
FAIL_MESSAGE	Specifies the message to display if the validation fails.

2.4.9.2 Example Implementation

The following configuration validates that the invoice has been saved (Invoice Transaction ID is not 0). If it is 0, the command reports the error message specified in the FAIL_MESSAGE parameter.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 2–54 Example ValidateTask Command in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_ KEY	PARAMETER_VALUE
ValidateTransactionID	oracle.imaging.axf.commands.bpel. ValidateTaskCommand	ATTRIBUTE_TO_ VALIDATE	XPATH:InvoiceProcessing_TransactionID
ValidateTransactionID	oracle.imaging.axf.commands.bpel. ValidateTaskCommand	CMD_ON_PASS	CompleteInvoice
ValidateTransactionID	oracle.imaging.axf.commands.bpel. ValidateTaskCommand	REGULAR_ EXPRESSION	[^0]
ValidateTransactionID	oracle.imaging.axf.commands.bpel. ValidateTaskCommand	FAIL_MESSAGE	Please save the transaction in E-Business Suite before completing the task.

2.4.9.3 Example Implementation Instructions

Follow these steps to add a validation that verifies that a Transaction ID is present before allowing a task to be completed.

Note: This example assumes that you have installed the Invoice Processing template data.

Note: This configuration change should be applied only in use cases where users must create the EBS invoice before the task can be completed. This configuration would not apply in use cases where users may not create an invoice before completing the task (typically, for example, when the task is being completed with an outcome of SupplierMaintenance).

1. Add the following row to the AXF_COMMANDS table:

Table 2–55 Example AXF_COMMANDS Table

SOLUTION_ NAMESPACE	COMMAND_CLASS	COMMAND_ NAMESPACE
InvoiceProcessing	oracle.imaging.axf.commands.bpel.ValidateTaskCommand	ValidateTransactionID

2. Add the rows shown in Table 2–54 to the AXF_SOLUTION_PARAMETERS table.
3. In the AXF_ACTIONS Table, edit the row in which the Complete Task is configured, replacing the Complete action's COMMAND_NAMESPACE column with the ValidateTransactionID's command namespace.

Table 2–56 AXF_ACTIONS Table

ACTION_ID	VIEW_ID	DISPLAY_NAME	COMMAND_NAMESPACE	MENU_ORDER
CompleteInvoice	/TaskViewer.jspx	Complete Invoice	ValidateTransactionID	3

2.4.10 Custom Commands

You can also deploy custom commands to work within the AXF infrastructure. Custom commands must implement the oracle.imaging.bai.axf.commands.AxfCommand interface. The execute(AxfRequest) method is invoked by the infrastructure. Configure the implementation to execute in the AXF configuration database.

In addition, commands may implement the oracle.imaging.bai.axf.commands.ValidatableCommand interface, which provides a way for the AXF infrastructure to validate the configuration and operation of a command without executing it to provide a system command status.

2.4.11 Configuring Chained Commands and Web Tools

Some AXF commands have parameter keys that specify what occurs after the command is completed, allowing you to chain them. For example, Table 2–57 shows a portion of the AXF_SOLUTION_PARAMETERS table. After the CompleteTask command executes, additional AXF commands are executed (StartInvoiceProcessing and AutoOpenTask, based on program logic).

Table 2–57 Example AXF_SOLUTION_PARAMETERS Table for CompleteTask Command (InvoiceProcessing Solution)

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
DuplicateInvoice	oracle.imaging.axf.commands.bpel.CompleteTaskCommand	CMD_AUTOTASK_OFF	StartInvoiceProcessing
DuplicateInvoice	oracle.imaging.axf.commands.bpel.CompleteTaskCommand	CMD_AUTOTASK_ON	AutoOpenTask
DuplicateInvoice	oracle.imaging.axf.commands.bpel.CompleteTaskCommand	OUTCOME	DUPLICATE_INVOICE

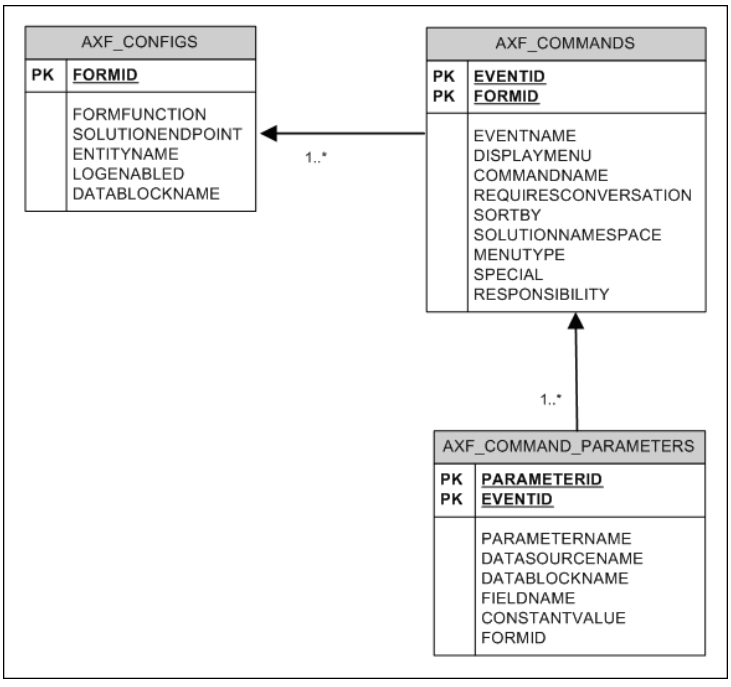
3

AXF Tables in E-Business Suite

Configuring AXF for EBS requires configuring AXF-related tables in EBS. This chapter covers the following topics:

3.1 About the AXF Tables in E-Business Suite

The following diagram shows how the tables used by the E-Business Suite system in AXF solutions are related.



3.2 AXF_CONFIGS Table

Use the AXF_CONFIGS table to enable the AXF solution on various E-Business Suite Forms. This table allows a fine level of granularity when selecting which Forms are AXF-enabled, up to the Data Block level.

Form events are invoked automatically when an action is performed on an E-Business Suite Form. The AXF_CUSTOM.pll makes all events available, such as POST-INSERT, for customization. You can decide which events to use, and how and when to use them.

When an action occurs, the customized code launches the specified solution and command configured for the event. In the case where the same form is being reused, such as Invoice Entry and Invoice Query, FORMFUNCTION and DATABLOCKNAME together uniquely identify each Form.

Note: You can enable all data blocks on a form rather than a specific data block, by specifying `AXF_DEFAULT` for the DATABLOCKNAME parameter. This allows AXF to be notified whenever a POST-INSERT event occurs for the form, regardless of its data block. Note, however, that setting the DATABLOCKNAME parameter to AXF_DEFAULT enables specified ZOOM or SPECIAL commands on all screens related to the form. (ZOOM and SPECIAL commands are set in the AXF_COMMANDS Table.)

3.2.1 Column Description

Table 3–1 Column Description for AXF_CONFIGS Table

Column Name	Description	Data Type
FORMID	Specifies the primary key of the table.	Number
FORMFUNCTION	Distinguishes each E-Business Suite Form based on the form's functionality.	Varchar2 (100 byte)
SOLUTIONENDPOINT	Specifies a URL to AXF.	Varchar2 (1000 byte)
ENTITYNAME	Used by the attachment functionality as a unique name, which links attachments to the correct Forms.	Varchar2 (100 byte)
LOGENABLED	Enables or disables the log for the specified form. Specify one of the following: ■ 1/TRUE/YES ■ 0/FALSE/NO	Varchar2 (10 byte)
DATABLOCKNAME	Specify the data block on the form to be enabled. Note that you can also specify `AXF_DEFAULT` to enable all data blocks on the form. A Form may be reused by E-Business Suite (for example, Invoice Entry and Invoice Query); the FORMFUNCTION and DATABLOCKNAME together uniquely identify each form.	Varchar2 (100 byte)

3.2.2 Example Implementation

This example defines that the entire Invoices Form is AXF-enabled. (Without the first row, the INV_SUM_FOLDER Data Block of the Invoices Form would be enabled.)

The last row in this example table shows the paperclip attachment feature enabled, by disabling the Managed Attachments solution. For more information, see the section on reenabling paperclip attachment in the *Oracle E-Business Suite Adapter for Oracle Imaging and Process Management Application Extension Framework Installation Guide*.

Table 3–2 Example AXF_CONFIGS Table

FORMID	FORMFUNCTION	SOLUTIONENDPOINT	ENTITYNAME	LOGENABLED	DATABLOCKNAME
1	AP_APXINWKB	http://<ApplicationServerName>:<SOA-Port>/imaging-bai-axf/AxfSolutionMediator	AP_INVOICES	YES	AXF_DEFAULT
2	AP_APXINWKB_SUMMARY_VIEW	http://<ApplicationServerName>:<SOA-Port>/imaging-bai-axf/AxfSolutionMediator	AP_INVOICES	YES	INV_SUM_FOLDER
6	AP_APXINWKB_BATCHES	http://<ApplicationServerName>:<SOA-Port>/imaging-bai-axf/AxfSolutionMediator	AP_INVOICES	YES	INV_SUM_FOLDER
7	AXF_MANAGED_ATTACHMENTS-DISABLED	http://<ApplicationServerName>:<SOA-Port>/imaging-bai-axf/AxfSolutionMediator	(null)	YES	(null)

3.2.3 Enabling EBS Logging

To enable logging for a particular Form function, set the LOGENABLED field to either `1`, `YES` or `TRUE` and the file will be created in the UTL_FILE_DIR folder. Consult with your DBA to verify that the UTL_FILE_DIR folder is available and accessible. Log files are named *Username_MASTER_LOG.txt*, and will continue to grow as items are appended.

3.3 AXF_COMMANDS Table

Use the AXF_COMMANDS table to describe the actions to be taken based on user activity. This table works in conjunction with the AXF_CONFIGS Table.

3.3.1 Column Description

Table 3–3 Column Description for AXF_COMMANDS Table

Column Name	Description	Data Type	Nullable
FORMID	Links to the AXF_CONFIGS Table.	Number	No
EVENTID	Primary key of the table.	Number	Yes
EVENTNAME	Name of the Event command to be invoked (for example, ZOOM, POST-INSERT).	Varchar2(100 byte)	Yes
DISPLAYMENU	Displays text of the menu for the command.	Varchar2(100 byte)	Yes
COMMANDNAMESPACE	Request command to be passed to the back-end when the menu is selected.	Varchar2(100 byte)	Yes
REQUIRESCONVERSATION	Indicates if the command requires a valid conversation or not.	Varchar2(10 byte)	Yes
SORTBY	Order in which the menu is displayed.	Number	Yes
SOLUTIONNAMESPACE	Name of the solution.	Varchar2(100 byte)	Yes

Table 3–3 (Cont.) Column Description for AXF_COMMANDS Table

Column Name	Description	Data Type	Nullable
MENUTYPE	Specify the menu type to display to users in E-Business Suite. You can choose: ■ ZOOM: Displays a Zoom menu in the toolbar. ■ ZOOMANDSPECIAL: Displays both a Zoom menu and a Special menu. (Enter a special key in the SPECIAL column.) ■ SPECIAL: Displays a Special menu on the toolbar. (Enter a special key in the Special column.)	Varchar2(25 byte)	Yes
SPECIAL	Create new menu entries by entering a unique number for the Special type menu, where: ■ SPECIAL1-15 creates entries in the Tools menu. ■ SPECIAL16-30 creates entries in the Reports menu. ■ SPECIAL31-45 creates entries in the Actions menu. (Consult the E-Business Suite Documentation for further information.)	Varchar2(10 byte)	Yes
RESPONSIBILITY	Use this column to filter the menu options based on user responsibility. Enter a value to display the menu only to end users with responsibilities associated with that value. ■ A null value indicates that the menu will be shown to all users. ■ If specifying multiple responsibilities, separate the values with a comma (as in 68766, 798767, 59968).	Varchar2(100 byte)	Yes

3.3.2 Example Implementation

This example shows two commands invoked from the Zoom menu (Attach Supplemental and Process Invoices). Each command is listed twice because the commands are enabling the same functionality, but on two different screens.

In addition, the solution has been configured to invoke the SaveInvoice command during the POST-INSERT event, which specifies that whenever an action inserts a new E-Business Suite transaction record, the integration will automatically invoke the SaveInvoice command on the back-end, performing the actions associated with the command. Note that POST-INSERT is not called by a subsequent save of the same transaction record in E-Business Suite.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing, SPECIAL=(null), RESPONSIBILITY=(null)

Table 3–4 Example AXF_COMMANDS Table

EVENT ID	FORM ID	EVENTNAME	DISPLAYMENU	COMMANDNAMESPACE	REQUIRESCONVERSATION	SORT BY	MENU TYPE
16	1	ZOOM	Attach Supplemental	AttachSupplemental	YES	2	ZOOM
14	6	ZOOM	Attach Supplemental	AttachSupplemental	YES	2	ZOOM
13	6	ZOOM	Process Batch Invoices	StartInvoiceProcessing	NO	1	ZOOM

Table 3–4 (Cont.) Example AXF_COMMANDS Table

EVENT ID	FORM ID	EVENTNAME	DISPLAYMENU	COMMANDNAMESPACE	REQUIRESCONVERSATION	SORT BY	MENU TYPE
10	1	ZOOM	Process Invoices	StartInvoiceProcessing	NO	1	ZOOM
11	1	POST-INSERT	(null)	SaveInvoice	YES	0	(null)
15	6	POST-INSERT	(null)	SaveInvoice	YES	0	(null)

3.4 AXF_COMMAND_PARAMETERS Table

Use the AXF_COMMAND_PARAMETERS table to define the information sent for each defined command. Each command may require or omit a different set of parameters.

3.4.1 Column Description

Table 3–5 Column Description for AXF_COMMAND_PARAMETERS Table

Column	Description
PARAMETERID	Defines a unique ID for the parameter.
EVENTID	Defines a unique ID for the event. Comes from the AXF_COMMANDS Table.
PARAMETERNAME	The name of the parameter to be passed.
DATASOURCENAME	Data Source for the parameter value. You can specify *Data* or *Constant*.
DATABLOCKNAME	Data Block of the Form from which the value is fetched
FIELDNAME	Field Name in the form from which the value is fetched.
CONSTANTVALUE	A constant value for the parameter.

3.4.2 Example Implementation

The example that follows contains two parameters sent for EventID 2: a constant value (InvoicesByVendor) and a data value (VENDOR_NAME) in the *INVOICES_QF* Data Block.

The COMPLETE command requires that a conversation is established between E-Business Suite and AXF. A Conversation is a session unique ID that allows communication between E-Business Suite and AXF Framework.

If a command requires a separate window to be opened, then E-Business Suite will open another instance of the browser. Users may then take additional steps in the newly created window.

Table 3–6 Example AXF_COMMAND_PARAMETERS Table

PARAMETERID	EVENTID	PARAMETERNAME	DATASOURCENAME	DATABLOCKNAME	FIELDNAME	CONSTANTVALUE
1	2	SearchName	CONSTANT	(null)	(null)	InvoicesByVendor
2	2	VendorName	DATA	INVOICES_QF	VENDOR_NAME	(null)
18	11	InvoiceTransactionID	DATA	INV_SUM_FOLDER	INVOICE_ID	(null)

Table 3–6 (Cont.) Example AXF_COMMAND_PARAMETERS Table

PARAMETERID	EVENTID	PARAMETERNAME	DATASOURCENAME	DATABLOCKNAME	FIELDNAME	CONSTANTVALUE
21	14	InvoiceTransactionID	DATA	INV_SUM_FOLDER	INVOICE_ID	(null)
20	15	InvoiceTransactionID	DATA	INV_SUM_FOLDER	INVOICE_ID	(null)
22	16	InvoiceTransactionID	DATA	INV_SUM_FOLDER	INVOICE_ID	(null)

A

Configuring BPEL Task Views

This appendix covers the following topics:

A.1 About Configuring BPEL Task Views

This appendix describes how to configure BPEL task views via a custom XML file. This XML file is provided as a convenience, as you typically configure views using the BPEL Worklist application. However, because this custom XML file augments capabilities provided by BPEL, configuring it instead saves you from individually creating BPEL views for users in the BPEL Worklist application.

This appendix highlights features added by AXF to the custom XML file. For information about creating BPEL views, see the *Oracle BPEL Process Manager Developer's Guide*.

The XML file:

- Maps BPEL flex fields for display in the Task List
- Defines views for display to specified sets of users

How the BPEL Task Views XML File is Structured

To configure the XML file, you can customize the sample XML file provided with the solution template (called *bpel\CreateSampleViews.xml*), or refer to the sample and create your own. See "Main Configuration Steps" on page A-2.

In the XML file portion that follows, <workflowSet> contains two main sections: <FFMSet> and <viewSet>:

- <*FFMSet*> defines flex field mapping using <FFMDefinition>. See "Flex Field Mapping Definitions (FFMSet)" on page A-2.
- <*viewSet*> defines views using <viewDefinition>. See "View Definitions (viewSet)" on page A-4.

```
<workflowSet>
  <FFMSet>
    <FFMDefinition>
      …
```

```
    </FFMDefinition>
  </FFMSet>

  <viewSet>
    <viewDefinition>
      …
    </viewDefinition>
  </viewSet>
</workflowSet>
```

A.2 Main Configuration Steps

Configuring the XML file involves the following main steps. After configuring, you must upload the file.

- "Flex Field Mapping Definitions (FFMSet)" on page A-2
- "View Definitions (viewSet)" on page A-4
- "Uploading the XML File" on page A-7

A.3 Flex Field Mapping Definitions (FFMSet)

Follow the steps in this section to define flex field mapping for the BPEL Views.

- "Define the Admin User" on page A-2
- "Define Flex Field Mapping" on page A-3
- "Define Flex Field Mapping Details" on page A-3

A.3.1 Define the Admin User

In <FFMDefinition> (Flex Field Mapping Definition), <userList> specifies the admin user with permissions to create flex field mapping.

> **Note:** The admin user you specify here must be a member of the oc4j-administrators group within OID.

```
<workflowSet>
  <FFMSet>
    <FFMDefinition>
        <userList>
          <user>oc4jadmin</user>
        </userList>
    </FFMDefinition>
  </FFMSet>
  …
</workflowSet
```

A.3.2 Define Flex Field Mapping

<FFMDetail> defines the details to create the flex field mapping. (See the parameters that follow.)

```
<workflowSet>
  <FFMSet>
    <FFMDefinition>
      …
        <FFMDetail>
          <name>InvoiceProcessing</name>
          <useVersionID>2.0</useVersionID>
        </FFMDetail>
    </FFMDefinition>
  </FFMSet>
  …
</workflowSet>
```

Parameters	Description
name	Specifies the task name (for example, InvoiceProcessing, AccountDistribution, or InvoiceApproval).
useVersionID	Specifies the version of the task used to create the flex field mapping. ■ If not specified, the latest version of workflow task is used. ■ If *All* is specified, flex field mappings are created for all versions of tasks.

A.3.3 Define Flex Field Mapping Details

This mapping links the flex field attributes listed in the Task List with flex field values in the database. (See the parameters that follow.)

```
<workflowSet>
  <FFMSet>
    <FFMDefinition>
      …
      <FFMDetail>
          …
        <mappings>
            <column>
              <name>invoiceNumber</name>
              <attribute>TextAttribute1</attribute>
              <type>TEXT</type>
              <label>Invoice Number</label>
            </column>
          </mappings>
        </FFMDetail>
    </FFMDefinition>
  </FFMSet>
  …
</workflowSet>
```

Parameters	Description
name	Payload attribute name
attribute	Task attribute
type	Data type
label	Payload label name

A.4 View Definitions (viewSet)

Follow the steps in this section to define BPEL views.

A.4.1 Define Views

You can include multiple view definitions, such *East Group*, *West Group*, etc. In <viewDefinition>, <userList> is the user for whom the view is being created. The user could be oc4jadmin or another user such as JCOOPER.

```
<workflowSet>
  …
  <viewSet>
    <viewDefinition>
    <userList>
      <user>oc4jadmin</user>
    </userList>
    </viewDefinition>
    …
  </viewSet>
</workflowSet>
```

A.4.2 Define View Details

In <userViewDetail>, define the details to create a view for this user.

```
<workflowSet>
  …
  <viewSet>
    <viewDefinition>
    …
      <userViewDetail viewType="VIEW">
        <name>North Invoice Processing Group</name>
        <hidden>false</hidden>
        <description>North Invoice Processing Group View with maximum amount of
options configured</description>
        …
      </userViewDetail>
    </viewDefinition>
    …
  </viewSet>
</workflowSet>
```

A.4.3 Define View Columns

You can define multiple columns in <viewColumns>.

```
<workflowSet>
  …
  <viewSet>
    <viewDefinition>
    …
      <userViewDetail viewType="VIEW">
        …
        <viewColumns>
          <column>taskNumber</column>
          …
        </viewColumns>
      </userViewDetail>
    </viewDefinition>
    …
  </viewSet>
</workflowSet>
```

A.4.4 Define Optional View Information

ViewOptionalInfo defines optional information, such as <taskOptionalInfo>.

```
<workflowSet>
  …
  <viewSet>
    <viewDefinition>
    …
      <userViewDetail viewType="VIEW">
        <viewOptionalInfo>
          <taskOptionalInfo>GroupActions</taskOptionalInfo>
        </viewOptionalInfo>
      </userViewDetail>
    </viewDefinition>
    …
  </viewSet>
</workflowSet>
```

A.4.5 Define View Predicates

If <column> is *taskDefinitionId* or *taskDefinitionName,* the workflow ID will be defined in <clause>. (See the second and third clauses in the example that follows.)

- If *createUsingTaskDefID="true"* is not given, the value of <column> should be *taskDefinitionId*, and the workflow ID should be explicitly given as <value>, as in *default_InvoiceProcessing_1.0_InvoiceProcessing*.
- If *createUsingTaskDefID="true"* is given, but *useTaskDefID* is not, the value of <column> should be *taskDefinitionName*; the workflow ID is pulled out from the latest version of task, and the task name is taken from <value>, as in *InvoiceProcessing*.
- If *createUsingTaskDefID="true"* and *useTaskDefID* are given, the value of <column> should be *taskDefinitionName*; the workflow ID is pulled out based on *useTaskDefID*, and the task name is taken from <value>, as in *InvoiceProcessing*.

```
<workflowSet>
  …
  <viewSet>
    <viewDefinition>
    …
      <userViewDetail viewType="VIEW">
        …
        <viewPredicate>
          <assignmentFilter>My+Group</assignmentFilter>
          <clause ignoreCase="false" joinOperator="AND">
            <column>State</column>
            <operator>EQ</operator>
            <value>ASSIGNED</value>
          </clause>

          <clause ignoreCase="false" joinOperator="AND">
            <column>taskDefinitionId</column>
            <operator>EQ</operator>
            <value>default_InvoiceProcessing_1.0_InvoiceProcessing</value>
          </clause>

          <clause ignoreCase="false" joinOperator="AND"
createUsingTaskDefID="true" useTaskDefID="1.0">
           <column>taskDefinitionName</column>
           <operator>EQ</operator>
           <value>InvoiceProcessing</value>
         </clause>
        </viewPredicate>
      </userViewDetail>
    </viewDefinition>
    …
  </viewSet>
</workflowSet>
```

A.4.6 Define Column Order and Sorting

Define the order by which attributes are listed and sorted in the Task List.

```
<workflowSet>
  …
  <viewSet>
    <viewDefinition>
    …
      <userViewDetail viewType="VIEW">
        …
        <viewOrdering>
          <clause>
            <column>textAttribute1</column>
            <sortOrder>ASCENDING</sortOrder>
            <nullFirst>false</nullFirst>
          </clause>
        </viewOrdering>
      </userViewDetail>
    </viewDefinition>
    …
  </viewSet>
</workflowSet>
```

A.5 Uploading the XML File

After configuring the XML file, follow these steps to upload it.

1. Log in as the BPEL administrative user.
2. Upload the XML file using the View Creation tool, which is accessible at:

   http://<SOAServerName>:<SOAPort>/axf-web/faces/ViewUploader.jspx?bpelconnection=default

> **Note:** The bpelconnection specified above needs to match the bpel.connection defined in the "AXF_SYSTEM_PARAMETERS Table" on page 2-3. For example, Table 2.2.1.2 shows a bpelconnection setting of *default*.

Index

A

U

V

X