Oracle® Fusion Middleware

Administrator's Guide for E-Business Suite Adapter for Oracle Enterprise Content Management

11*g* Release 1 (11.1.1)

E15865-01

January 2010

ORACLE®

Oracle Fusion Middleware Administrator's Guide for E-Business Suite Adapter for Oracle Enterprise Content Management, 11*g* Release 1 (11.1.1)

E15865-01

Primary Author: Sarah Howland

Contributor: Tom Albrecht, Kevin de Smidt, Sancho Pinto, Kevin Cocilo, Alex Barnett

Contents

3 AXF Tables

4 E-Business Suite Tables

Index

Preface

The *Administrator's Guide for E-Business Suite Adapter for Oracle Enterprise Content Management* describes E-Business Suite solution configurations for Enterprise Content Management systems.

Audience

This document is intended for administrators configuring integration solutions between E-Business Suite and Oracle content management systems.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/support/contact.html` or visit `http://www.oracle.com/accessibility/support.html` if you are hearing impaired.

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

Solution Overview

This guide describes the E-Business Suite Adapter for Enterprise Content Management. The **Imaging Solution** provides imaging, capture, and workflow capabilities via Oracle Imaging and Process Management (I/PM).

Oracle Content Management solutions use the **Application Extension Framework (AXF)** infrastructure to integrate a business application with a content management application. See "About Application Extension Framework (AXF)" on page 1-1. An **AXF solution** is a micro-application whose components are created using the AXF infrastructure.

Solutions are installed on top of a base configuration of core AXF and **E-Business Suite** files standard to solutions that use AXF functionality. AXF-related tables are configured in E-Business Suite to specify which screens are enabled to execute configured AXF commands and PLL modules are modified. See "About E-Business Suite Components" on page 1-10.

1.1 About This Guide

This guide contains the following chapters:

- Chapter 1, "Solution Overview", introduces AXF components, the Imaging Solution, and E-Business Suite components.
- Chapter 2, "Configuring E-Business Suite Components" describes how to install and configure E-Business Suite components.
- Chapter 3, "AXF Tables" describes the AXF configuration tables used for the Imaging Solution, including commands and web user interface tools, and provides example implementations.
- Chapter 4, "E-Business Suite Tables" describes the tables used by E-Business Suite in the Imaging Solution.

1.2 About Application Extension Framework (AXF)

Oracle's Application Extension Framework (AXF) is a command-driven, web services integration between a business application such as E-Business Suite and a content management application such as I/PM. 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.

The Application Extension Framework includes the following components:

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.

AXF Commands

AXF provides reusable commands for implementing functionality. The Imaging Solution uses multiple AXF commands, as described in "About AXF Commands" on page 1-5.

AXF Web Tools

AXF provides web interface components for display to users, such as a task list and task viewer. Configured through the AXF tables, these web tools are used in some imaging solutions, and described in "About AXF Web User Tools" on page 1-6.

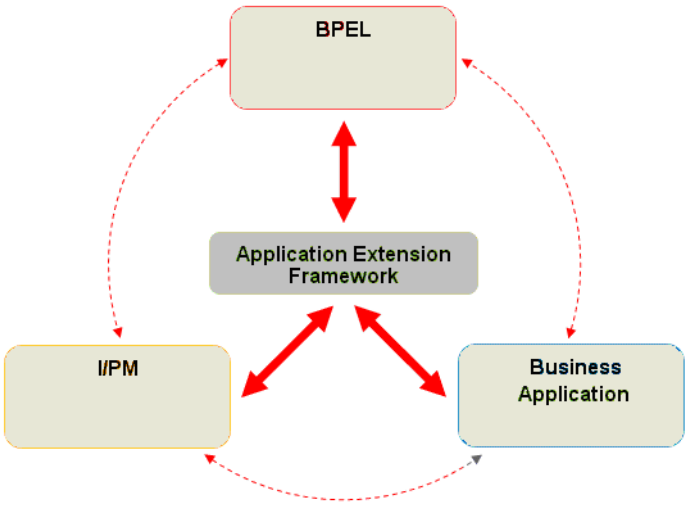
AXF Configuration Database Tables

You configure AXF solutions, commands, and web tools by configuring the AXF database tables. For information about each table and example implementations for the solution, see "AXF Tables" on page 3-1.

1.3 About the Imaging Solution

A workflow imaging solution is an integration between a business application such as E-Business Suite and BPEL (Oracle BPEL Process Manager), using Oracle I/PM as the imaging source, as illustrated in Figure 1–1. Through an AXF configuration, business users can process associated images and perform document-centric workflow tasks from their business application user interface.

Figure 1–1 Imaging Solution integrates Business Application, I/PM, and BPEL systems



This section covers the following topics:

- "Business User View for Imaging Solutions" on page 1-3

1.3.1 Business User View for Imaging Solutions

From a business user's perspective, the integration is virtually seamless. End-users use the Imaging Solution to:

- Launch I/PM from E-Business Suite, and select and perform workflow tasks. For example, users performing Invoice Processing tasks select a custom menu command integrated into their business application called Invoice Processing, 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 (Invoice Processing), 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.
- View attached images and metadata values. Use I/PM's tools for viewing, annotating, and redacting images, as permissions allow.
- Key entries in E-Business Suite while viewing images and related values in the I/PM viewer.
- Perform actions related to the workflow task, such as routing, canceling, updating, and completing tasks.
- Scan or upload supporting documents for a selected E-Business Suite record.
- View supporting images for an E-Business Suite record without leaving the E-Business Suite application.

1.3.1.1 Sample Scenario 1: Processing Invoices

An Imaging Solution configured for invoice processing might work as follows:

- A workflow process automatically generates user tasks.

  An invoice is uploaded, metadata values are assigned, and a task for processing the invoice is generated. Typically, tasks are pooled into profiles from which groups of users select. A user may have access to tasks in multiple profiles.
- From E-Business Suite, the user launches the Imaging Solution, by selecting a command called **Process Invoices** from the Zoom menu (or other special menu or key).
- The user selects a task from those listed for a selected profile. Once a task is selected (acquired), it is no longer available to other users.
- In the Task Viewer, users view the task's invoice image, key entries in E-Business Suite based on the image, and perform related commands.

  Additional action commands are typically provided in a side panel. Users might route the task to another user or user group for approval, add comments for others to view, skip the task, or re-scan or delete the task's document.
- Users complete the task and begin another, if desired.

Most often, changes users make in E-Business Suite are synchronized with I/PM, and vice versa.

Note: For details about Imaging Solution user tasks, see the *Oracle Fusion Middleware User's Guide for Enterprise Content Management Solutions for E-Business Suite*.

1.3.1.2 Sample Scenario 2: Capturing Supporting Employee Documents

An Imaging Solution configured for capturing supporting documents might work as follows:

- From E-Business Suite, a user retrieves a record such as an employee record.
- The user launches the document imaging solution, by selecting a command called **Scan Employee Document** from the Zoom menu (or other special menu or key).
- Oracle Distributed Document Capture launches and automatically initiates a scan (if a scanner is attached to the desktop) or allows the user to upload electronic images from desktop.
- The user enters index values (metadata) in Oracle Distributed Document Capture to store with the images.
- The user clicks Send, which transmits the captured document images and their metadata from Oracle Distributed Document Capture to Oracle I/PM.

1.3.1.3 Sample Scenario 3: Viewing Supporting Employee Documents

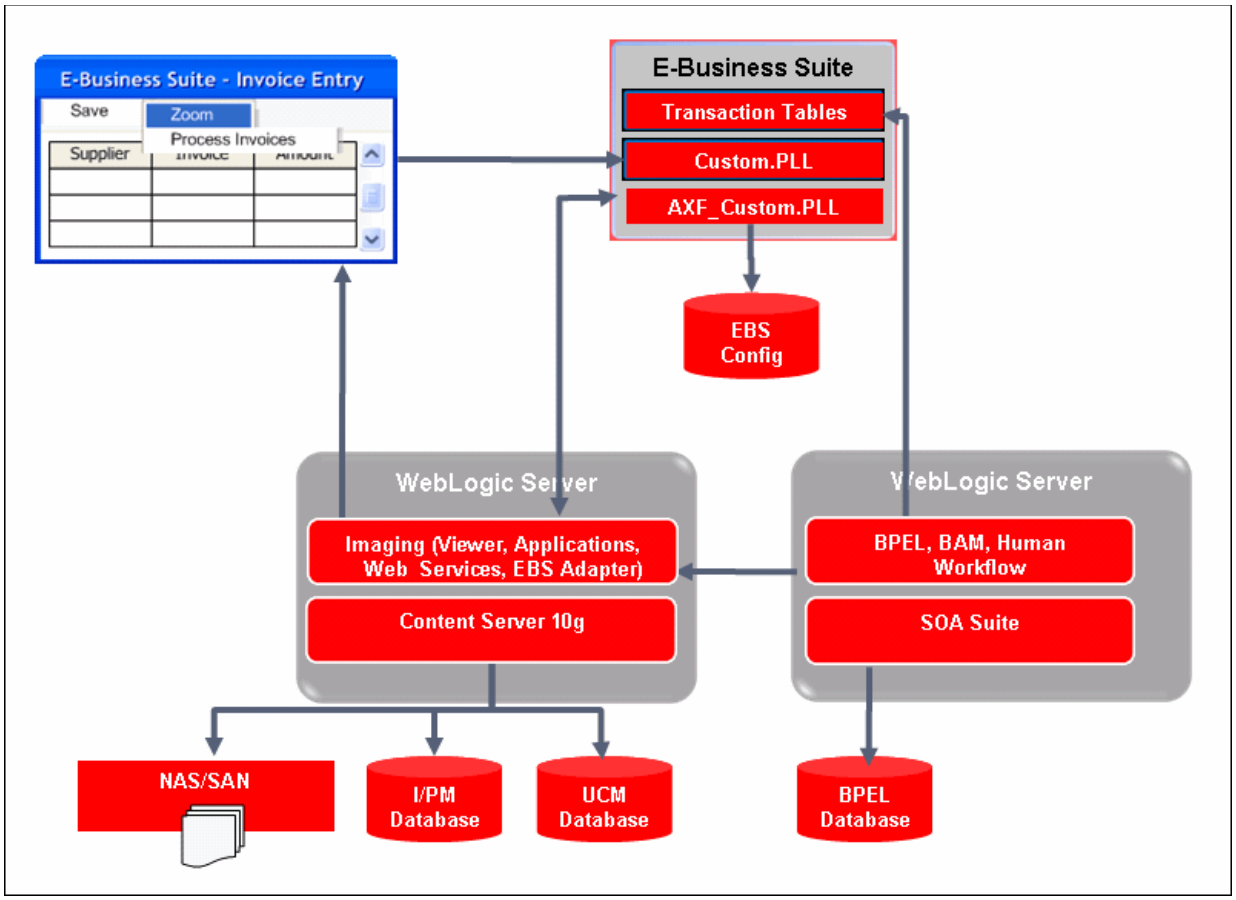
An Imaging Solution configured for viewing supporting documents might work as follows:

- From E-Business Suite, a user retrieves a record such as an employee record.
- A user launches the document imaging solution, by selecting a command called **View Employee Documents** from the Zoom menu (or other special menu or key).
- From the list of documents associated with the employee record and their metadata values, the user selects a document.
- The document is displayed in the Oracle I/PM viewer, where the user can view its images, and with appropriate permissions, apply annotations or redactions.

1.3.2 System Architecture

Figure 1–2 illustrates a workflow configuration for E-Business Suite.

Figure 1–2 System Architecture for AXF and E-Business Suite Configuration



1.3.3 About 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 3-25.

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 3-26.
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 3-26.
Release Task	Initializes the AXF Task List web tool for display (regardless of Autotask mode) and releases a human workflow task. See "Release Task Command" on page 3-27.
Complete Task	Completes a human workflow task and updates BPEL payload attribute values. If using the Autotask Command, claims the next task and displays it in the Task Viewer. See "Complete Task Command" on page 3-28.
Redirect	Redirects the current AXF web page to any URL specified in the configuration. See "Redirect Command" on page 3-29.

AXF Command	Description
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 3-29.
Update Task From Procedure	Calls a stored procedure using a specified data source and updates values in the BPEL payload using XPATH. See "Update Task From Procedure Command" on page 3-31.
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 3-33.

Note: You can also deploy custom commands to execute via AXF. See "Custom Commands" on page 3-34.

1.3.4 About AXF Web User Tools

The Imaging Solution provides the following user interface components. These are web interface components displayed to users and configured through the AXF tables.

- "About the Task List" on page 1-6
- "About the Task Viewer" on page 1-7
- "About the Enumeration Picker" on page 1-8
- "About the Identity Picker Web Tool" on page 1-8
- "About Comments" on page 1-9

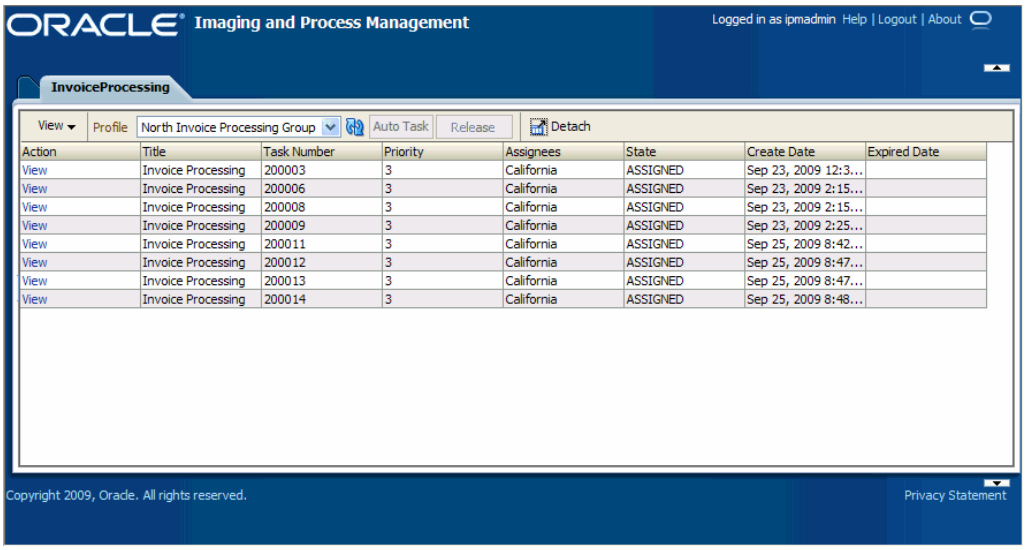
1.3.4.1 About the Task List

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.

Note: Use the BPM Worklist application to create views and share them with other users or groups.

For configuration information, see "Task List Web Tool" on page 3-14.

Figure 1–3 Task List Web Tool



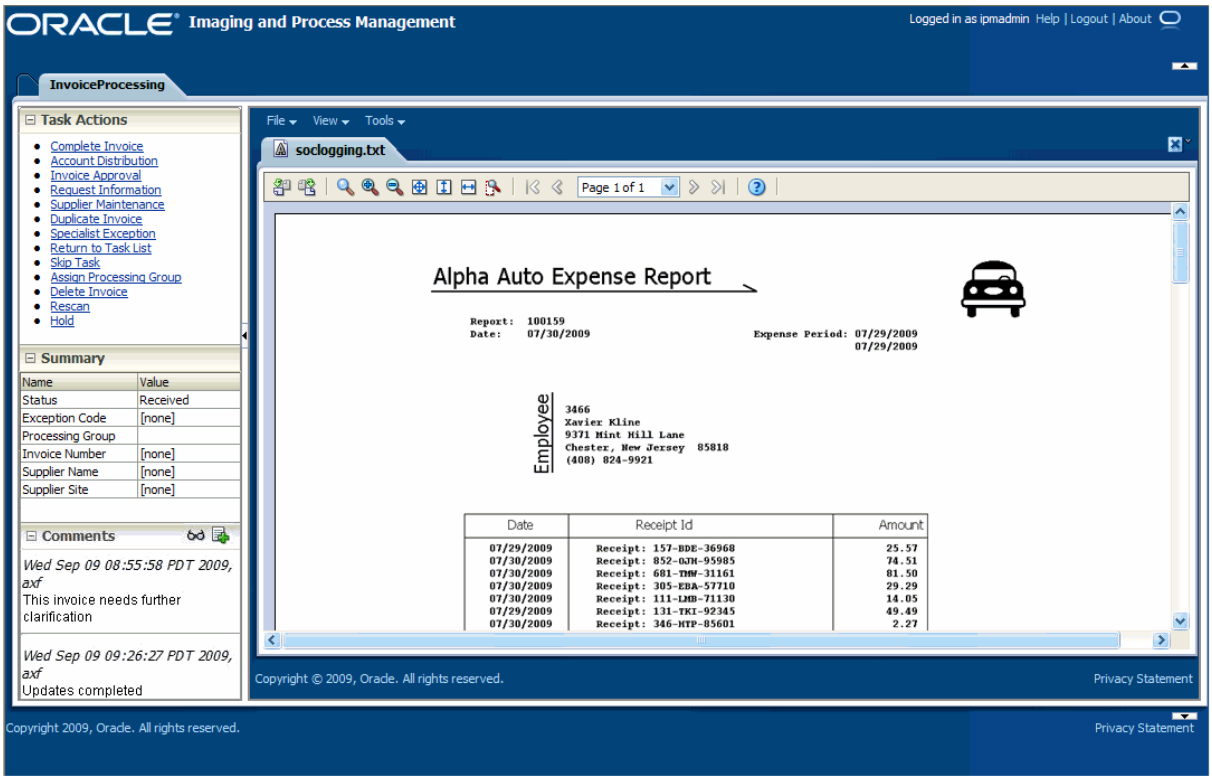
The Task List web tool can also display a list of AXF actions just like the Task Viewer, using AXF action commands. These action commands are menu components configured in the AXF_ACTIONS Table for display on a web page.

1.3.4.2 About the 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 typically displays a side menu containing AXF action commands configured in the AXF_ACTIONS Table. It may also include a Comments side pane; see "About Comments" on page 1-9.

For configuration information, see "Task Viewer Web Tool" on page 3-16.

Figure 1–4 Task Viewer Web Tool

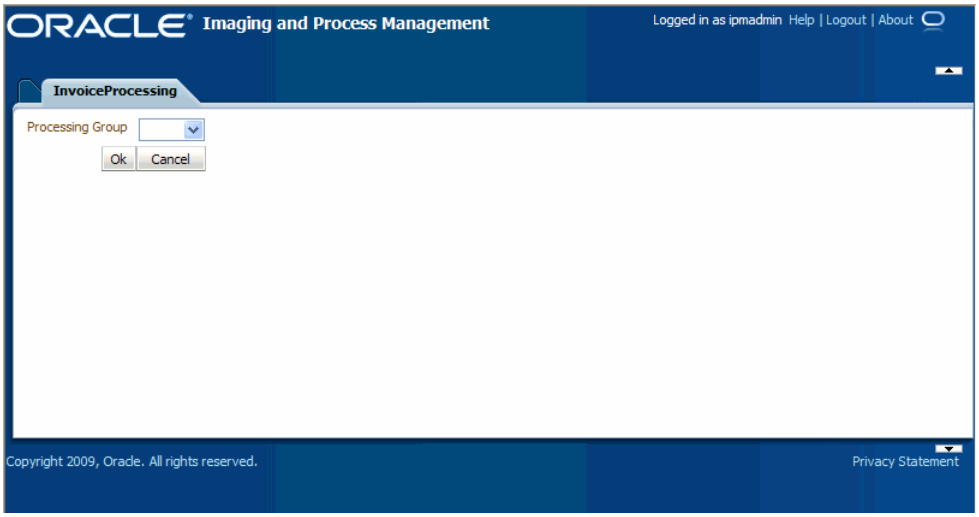


1.3.4.3 About the Enumeration Picker

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

For configuration information, see "Enumeration Picker Web Tool" on page 3-20.

Figure 1–5 Enumeration Picker Web Tool



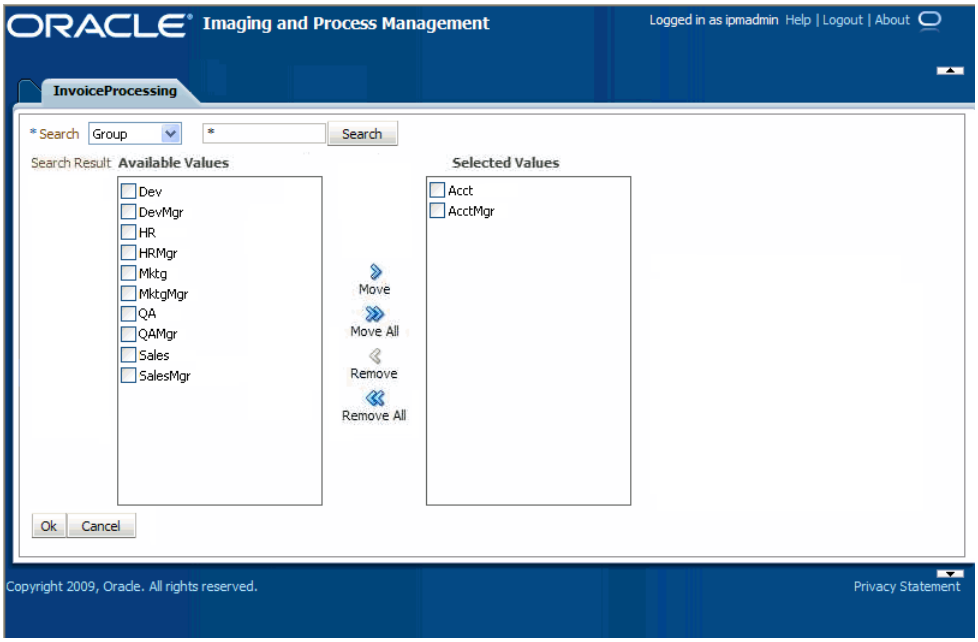
1.3.4.4 About the Identity Picker Web Tool

The Identity Picker web page allows users to select one or more users or groups from an identity store configured for BPEL. 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 3-24.

Figure 1–6 Identity Picker Web Tool

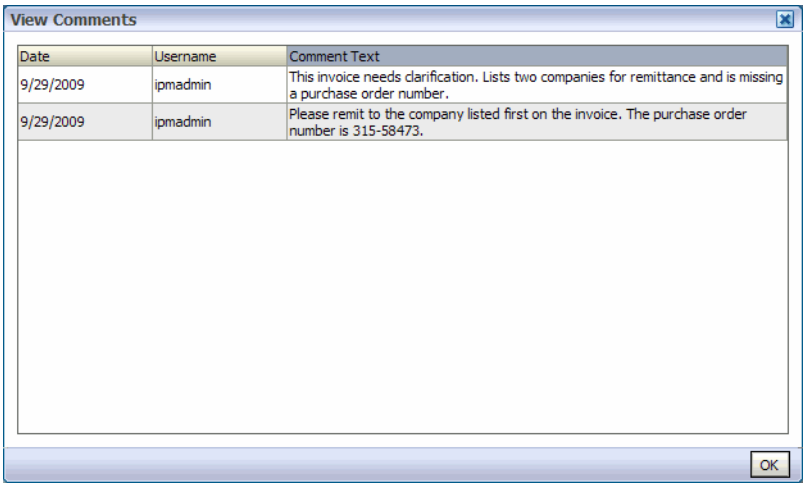


1.3.4.5 About Comments

The Comments web page allows users to enter comments related to the human task during the transaction's processing. Comments can be displayed in a side pane on the Task Viewer. Comments persist for the entire process, allowing users to view and add comments. Comments are saved using the native comments capabilities of BPEL's workflow task.

For configuration information, see "Comments" on page 3-20.

Figure 1–7 Comments Web Tool



1.4 About E-Business Suite Components

As part of AXF configuration, solution integrators configure the following E-Business Suite components.

- "PLL Modules" on page 1-10
- "PL/SQL Procedures" on page 1-10
- "AXF-Related Tables in E-Business Suite" on page 1-11

1.4.1 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 E-Business Suite event occurs, allowing AXF to determine if it relates to AXF functionality. See "Compiling E-Business Suite Forms" on page 2-4.

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 E-Business Suite configuration database.

1.4.2 PL/SQL Procedures

The following PL/SQL procedures are provided. (In some cases, separate E-Business Suite release 11 and 12 versions are provided.) See "Configuring the E-Business Suite Database" on page 2-2.

PL/SQL Procedure	Purpose
AXF_CREATE_TABLES_SYNONYM.sql	Creates the tables and synonyms used by AXF.

PL/SQL Procedure	Purpose
AXF_ADD_EBS_ATTACHMENT_PROC.sql	Creates a stored procedure that adds an attachment to a transaction in E-Business Suite.
AXF_SOAP_CALL_PROC.sql	Creates a stored procedure to make SOAP calls from PL/SQL.
AXF_APPS_INIT.sql	Required for EBS attachment.
AXF_EBS_PROPERTIES.sql	Populates the AFX_PROPERTIES table with security information for various E-Business Suite calls to AXF.

1.4.3 AXF-Related Tables in E-Business Suite

Configuring AXF for E-Business Suite requires configuring AXF-related tables in E-Business Suite. These tables are used to specify which screens are enabled to execute configured AXF commands. See "About the AXF Tables in E-Business Suite" on page 4-1.

AXF-related E-Business Suite tables include the following:

- AXF_CONFIGS Table
- AXF_COMMANDS Table
- AXF_COMMAND_PARAMETERS Table
- AXF_PROPERTIES Table

2

Configuring E-Business Suite Components

This chapter describes how to install and configure the E-Business Suite components used by ECM solutions.

This chapter covers the following topics:

2.1 System Requirements

The E-Business Suite Adapter is comprised of optional solutions installed over a base configuration of AXF and I/PM files standard to adapters that use AXF functionality. Requirements for the E-Business Suite Adapter for Enterprise Content Management are listed below.

> **Note:** Supported configuration and certification information is available at: http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html

E-Business Suite Release 11.5.10+, 12.0.4+ or 12.1.1+

A fully functioning Oracle E-Business Suite system.

- E-Business Suite Forms Builder is required for .PLL compilation.
  - For Oracle E-Business Suite 11i, Forms 6.0 Version 6.0.8.25.2+
  - For Oracle E-Business Suite 12, Forms Builder Version 10.1.2.0.2+
- To avoid duplicate logins, Oracle Single Sign On or Oracle Access Manager is required.

> **Note:** The E-Business Suite Adapter for ECM supports E-Business Suite Forms only. OAF web pages are not currently supported. In addition, Zoom Menu names are limited to a single language.

Oracle SOA Suite 11*g*

SOA 11*g*R1 (with patchset 1) is required.

You need a BPEL server instance. BPEL is part of SOA 11*g*R1.

2.2 Configuring E-Business Suite Solution Components

Installation of the E-Business Suite portion of AXF requires an active connection to the E-Business Suite database, general database experience, and knowledge of E-Business Suite Forms Builder. Consult your local DBA for assistance with these tasks. The instructions in this section assume the use of SQL*PLUS, but you can use any tool capable of querying the Oracle Database.

This section describes how to configure E-Business Suite components for the solution. It covers the following topics:

2.2.1 Creating the AXF E-Business Suite Configuration Schema User (AXF)

The adapter uses an E-Business Suite database to store PL/SQL procedures and E-Business Suite configuration information. Follow these steps to create a database user for use by AXF within the E-Business Suite database.

1. Create a user named `AXF`.

   A system account username and password is required to create the user. For assistance creating the user, contact your DBA.

2. Assign the configuration schema user the following access privileges:
   - Create table
   - Create sequences
   - Create public synonyms
   - Create session
   - Create procedure
   - Unlimited tablespace

2.2.2 Configuring the E-Business Suite Database

Follow these steps:

1. Using SQL*PLUS, log in to the E-Business Suite database as the AXF E-Business Suite configuration schema user.

   This user was previously created, as described in "Creating the AXF E-Business Suite Configuration Schema User (AXF)" on page 2-2.

2. As AXF, execute the **AXF_CREATE_TABLES_SYNONYM** script for your E-Business Suite version, from the applicable location listed below. This script creates the tables and synonyms used by AXF.

   E-Business Suite 12: AXF_INSTALL_DIR/ebs/R12/AXF_CREATE_TABLES_SYNONYM.sql

   E-Business Suite 11: AXF_INSTALL_DIR/ebs/R11/AXF_CREATE_TABLES_SYNONYM.sql

To execute the script, enter:

```
@AXF_CREATE_TABLES_SYNONYM.sql
```

Verify that the following tables were created: AXF_COMMAND_PARAMETERS, AXF_COMMANDS, AXF_CONFIGS, AXF_PROPERTIES, and AXF_FND_MAP.

3. As the APPS user, execute the **AXF_SOAP_CALL_PROC** script for your E-Business Suite version, from the applicable location listed below. This script creates a stored procedure to make SOAP calls from PL/SQL.

   E-Business Suite 12: AXF_INSTALL_DIR/ebs/R12/AXF_SOAP_CALL_PROC.sql

   E-Business Suite 11: AXF_INSTALL_DIR/ebs/R11/AXF_SOAP_CALL_PROC.sql

   Execute the script by entering:

   ```
   @AXF_SOAP_CALL_PROC.sql
   ```

4. As the APPS user, execute the following scripts for your E-Business Suite version, from the applicable location listed below.

   - AXF_INSTALL_DIR/ebs/AXF_APPS_INIT.sql
   - AXF_INSTALL_DIR/ebs/AXF_MANAGED_ATTACH_AVAIL.sql
   - AXF_INSTALL_DIR/ebs/AXF_MANAGED_ATTACH_VALUES.sql
   - AXF_INSTALL_DIR/ebs/AXF_MANAGED ATTACHMENT_DATA.sql

   Execute the script by entering:

   - `@AXF_APPS_INIT.sql`
   - `@AXF_MANAGED_ATTACH_AVAIL.sql`
   - `@AXF_MANAGED_ATTACH_VALUES.sql`
   - `@AXF_MANAGED_ATTACHMENT_DATA.sql`

5. As AXF, execute the **AXF_EBS_PROPERTIES_DATA.sql** script for your E-Business Suite version, from the location listed below.

   AXF_INSTALL_DIR/ebs/AXF_EBS_PROPERTIES_DATA.sql

   To execute the script, enter:

   ```
   @AXF_EBS_PROPERTIES_DATA.sql
   ```

6. As the APPS user, execute the **AXF_ADD_EBS_ATTACHMENT_PROC** sql script for your E-Business Suite version, from the applicable location listed below. This script creates a stored procedure for inserting attachments to the transaction record.

   E-Business Suite 12: AXF_INSTALL_DIR/ebs/R12/AXF_ADD_EBS_ATTACHMENT_PROC_R12.sql

   E-Business Suite 11: AXF_INSTALL_DIR/ebs/R11/AXF_ADD_EBS_ATTACHMENT_PROC_R11.sql

   Execute the script by entering the command appropriate for your version:

   @AXF_ADD_EBS_ATTACHMENT_PROC_R12.sql

   @AXF_ADD_EBS_ATTACHMENT_PROC_R11.sql

Note: This compilation may result in warnings, which can be ignored.

2.2.3 Compiling E-Business Suite Forms

AXF installation requires certain files to be uploaded to the E-Business Suite system, which enables a seamless integration of custom actions with existing E-Business Suite Forms.

Note: For information on using Oracle Forms Builder, see the following E-Business Suite documentation:

`http://www.oracle.com/technology/documentation/applications.html`

Follow these steps to copy the AXF_CUSTOM.pld file, convert it to an AXF_CUSTOM.pll file, make modifications, and then compile it to an AXF_CUSTOM.plx file.

1. For the applicable version listed below, copy the **AXF_CUSTOM.pld** file to the E-Business Server (to **FORMS_PATH** for E-Business Suite 12, or **FORMS60_PATH** for E-Business Suite 11).

   E-Business Suite 12: AXF_INSTALL_DIR/ebs/R12/AXF_CUSTOM.pld

   E-Business Suite 11: AXF_INSTALL_DIR/ebs/R11/AXF_CUSTOM.pld

   Note: If you are using a Linux/UNIX system and copied the .PLDs from a Windows system, issue the `dos2unix` command before converting it below.

2. Open Oracle Forms Builder and connect to the E-Business Suite database as the APPS user. Forms Builder is typically located in the `/bin/` subdirectory of your database's Oracle home.

   Note: If you fail to connect, verify the tnslistener.ora file.

3. In Forms Builder, open and convert AXF_CUSTOM.pld to AXF_CUSTOM.pll, by selecting **File** > **Administration** > **Convert**. Select **PL/SQL libraries** and **Text to binary** while converting the file.

   Note: If the following error is displayed during conversion of AXF_CUSTOM.PLD to AXF_CUSTOM.PLL, repeat this step until the file successfully converts.

   PDE-PL1038 - Can not open file as a PL/SQL Library

4. From the File menu, open AXF_CUSTOM.pll. Select **Program** > **Compile pl/sql** > **All** (E-Business Suite 12) or **Program > Compile > All** (E-Business Suite 11).

5. Compile AXF_CUSTOM into a module (.plx) by selecting **Program > Compile Module** (E-Business Suite 12) or **File** > **Administration** > **Compile File** (E-Business Suite 11).
6. Select **File** > **Connect** and ensure that you are connected to the database as an APPS user.
7. Open CUSTOM.pll by selecting **File** > **Open** and selecting **PL/SQL Libraries** (*.pll) in the Files of Type field. After opening the file and expanding Program Units, right-click the custom package body of CUSTOM.pll and select pl/sql editor.

   WARNING: Modifications to CUSTOM.pll are modifications to the E-Business Suite infrastructure. Ensure that this file is appropriately backed up before making changes.

8. Modify CUSTOM.pll by including the modifications formatted in bold italics in the following text. If the file contains other customizations, place these modifications after the existing code inside each function/procedure.

```
function zoom_available return boolean is
begin

-- Required for ALL integrations
return AXF_CUSTOM.zoom_available();
end zoom_available;

procedure event(event_name varchar2) is
begin

-- Required for AXF integrations
AXF_CUSTOM.event(event_name);
null;

end event;
```

   Note: If implementing the UCM Managed Attachments solution, replace this line:

   return AXF_CUSTOM.zoom_available();

   with the following line:

   return true;

9. With CUSTOM.pll open, verify that AXF_CUSTOM is listed as an Attached Library. If not, add it by highlighting Attached Libraries under CUSTOM and clicking the plus (+) symbol; browse to AXF_CUSTOM.pll and select it. Confirm the path warning if prompted.
10. Select **Program** > **Compile pl/sql < All** (E-Business Suite 12) or **Program > Compile > All** (E-Business Suite 11).
11. Compile CUSTOM into a module (.plx), by selecting **Program > Compile Module** (E-Business Suite 12) or **File** > **Administration** > **Compile File** (E-Business Suite 11).

Notes:

- AXF_CUSTOM must be compiled using the APPS schema user ID.
- If you encounter the following identifier or other errors referencing objects in APPCORE.pll while compiling, this indicates that the `APPCORE.pll` file must be attached to your form:

  `'APP_SPECIAL.ENABLE' must be declared` (a).

12. Save all before exiting Forms Builder. Verify that the Zoom menu command is displayed in the appropriate E-Business Suite forms.

2.2.4 Setting User Locales

To prevent issues with different locales when invoking AXF, E-Business Suite users should set the same values for their user locale preference and their browser locale.

2.2.5 Configuring and Viewing Log Files

When troubleshooting, you may want to examine the following AXF-related logs:

- AXF logs
- E-Business Suite logs

2.2.5.1 Configuring AXF Logging

Use the AXF logs to isolate issues in solution configuration. By default, some AXF logging automatically occurs as part of Application Server logging. Follow these steps to configure more detailed and separate AXF logging.

1. Add a log handler to the Application Server configuration. Add the handler inside the <log_handlers> tag in the logging.xml file, at the following location:

   $DOMAIN/config/fmwconfig/servers/$SERVER/logging.xml

```
<log_handler
name='axf-handler'class='oracle.core.ojdl.logging.ODLHandlerFactory'
level='ALL'>
   <property name='path'
value='${domain.home}/servers/${weblogic.Name}/logs/axf.log'/>
   <property name='maxFileSize' value='5485760'/>
   <property name='maxLogSize' value='54857600'/>
</log_handler>
```

2. Add a logger and set the level from the Log Levels (ODL Message Types) listed in Table 2–1. You can set the logging level in the XML file or via Enterprise Manager.

```
<logger name='oracle.imaging.axf' level='TRACE:32' useParentHandlers='false'>
     <handler name='axf-handler'/>
     <handler name='console-handler'/>
</logger>
```

Note: Remove the console-handler tag to omit logging on the console.

Table 2–1 Available Logging Levels

Log Type	Description	Log Level (ODL Message Type)
NULL	The logger inherits the log level set for its parent.	n/a
SEVERE	Log system errors requiring attention from the system administrator.	ERROR:1
WARNING	Log actions or conditions discovered that should be reviewed and may require action before an error occurs.	WARNING:1
INFO	Log normal actions or events. This could be a user operation, such as login completed, or an automatic operation, such as a log file rotation.	NOTIFICATION:1
CONFIG	Log configuration-related messages or problems.	NOTIFICATION:16
FINE	Log trace or debug messages used for debugging or performance monitoring. Typically contains detailed event data.	TRACE:1
FINER	Log fairly detailed trace or debug messages.	TRACE:16
FINEST	Log highly detailed trace or debug messages.	TRACE:32

3. Restart oc4j if it is running. The logger is displayed in Enterprise Manager. You can change the logging level at run time.

2.2.5.2 Configuring E-Business Suite Logging

You enable logging for specific forms in the AXF_CONFIGS Table. For details, see "Enabling E-Business Suite Logging" on page 4-3.

2.3 Uninstalling the Adapter

Follow the steps listed below to uninstall the adapter:

- "Uninstalling AXF from E-Business Suite" on page 2-7

2.3.1 Uninstalling AXF from E-Business Suite

Follow these steps to uninstall AXF from E-Business Suite.

1. Remove the AXF database schema and all associated data is removed.
   - Remove AXF_CUSTOM.* (AXF_CUSTOM.pll, AXF_CUSTOM.pld, and AXF_CUSTOM.plx) from `FORMS_PATH` (or `FORMS60_PATH` on E-Business Suite 11 systems).
   - Revert CUSTOM.PLD to a backed up version.
   - Restore and compile the original (backed up) AXF_CUSTOM.pll and AXF_CUSTOM.plx files.
2. Execute the AXF_DROP_TABLES_SYNONYM script for your E-Business Suite version, from the applicable location listed below. This script drops all tables, synonyms, and sequences created by the AXF_CREATE_TABLES_SYNONYM script run during installation.

   E-Business Suite 12: AXF_INSTALL_DIR/ebs/R12/AXF_DROP_TABLES_SYNONYM.sql

   E-Business Suite 11: AXF_INSTALL_DIR/ebs/R11/AXF_DROP_TABLES_SYNONYM.sql

Execute the script by entering:

@AXF_DROP_TABLES_SYNONYM.sql

3

AXF Tables

This chapter describes the AXF configuration tables used for the Imaging Solution, including commands, web user interface tools, and example implementations. This chapter covers the following topics:

3.1 Overview of AXF Configuration Tables

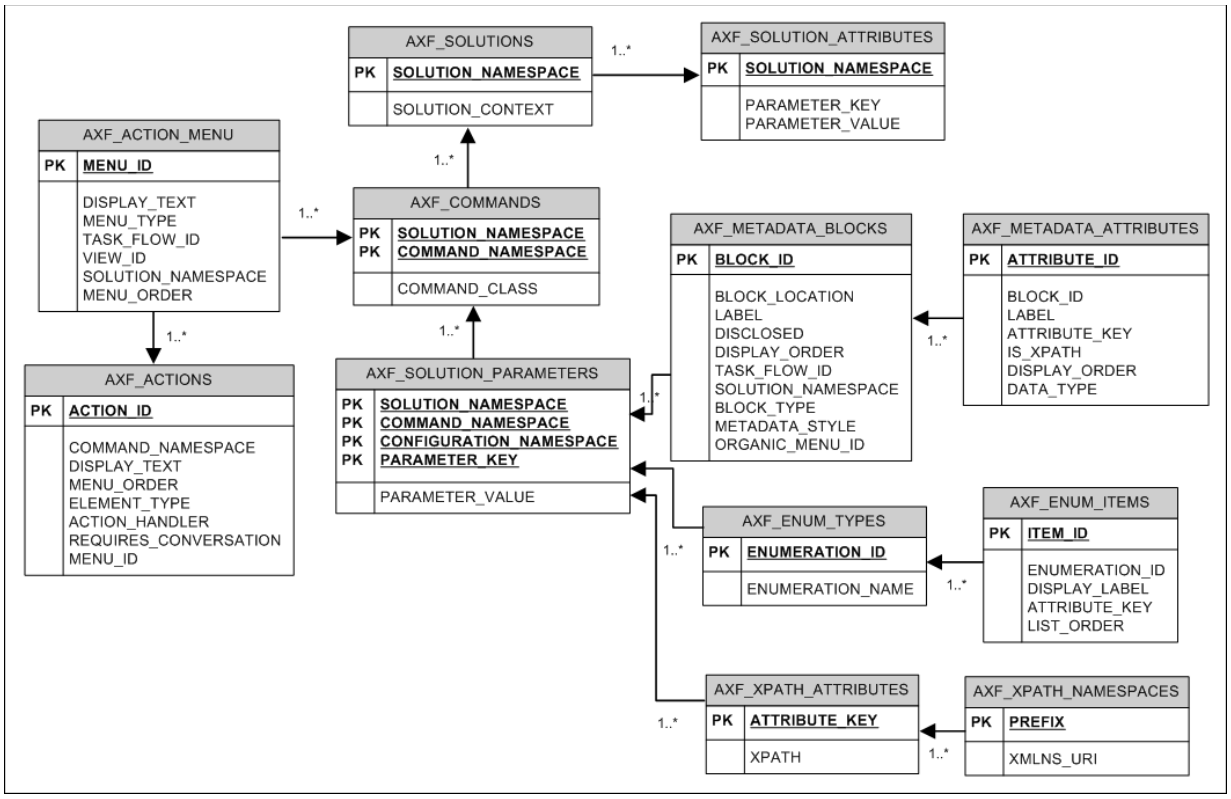
Note: Running an AXF and an Oracle BPM Worklist session at the same time can result in the session first opened ending. For example, launching an AXF session with an Oracle BPM Worklist session open ends the BPM Worklist session.

This conflict occurs because console session information is retained in browser cookies whose names are domain specific, but default to the same initial value. To prevent this conflict, set cookie names unique for each domain. To set cookie names, use the console on the advanced section of the Domain Configuration/General page.

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 3–1 AXF Configuration Tables



AXF Table	Description
AXF_SOLUTIONS Table AXF_SOLUTION_ATTRIBUTES Table	Define AXF solutions, and general parameters for infrastructure, services, and solutions.
AXF_COMMANDS Table	Define AXF commands within solutions.
AXF_SOLUTION_PARAMETERS Table	Define parameters for AXF commands and AXF web tools.
AXF_ACTION_MENU Table AXF_ACTIONS Table	Define task action pane itself, links in the pane, and their parameters.
AXF_METADATA_BLOCKS Table, AXF_METADATA_ATTRIBUTES Table	Define optional panes, such as Summary and Comments in the Task Viewer.
AXF_ENUM_TYPES Table, AXF_ENUM_ITEMS Table	Define enumeration pickers and their values.
AXF_XPATH_ATTRIBUTES Table, AXF_XPATH_NAMESPACES Table	Define XPATH attributes for payload elements.

3.2 AXF Tables

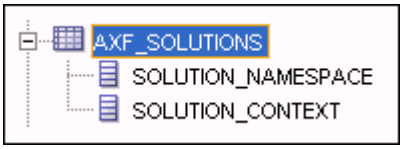
This section describes the following AXF tables. See "AXF Web User Tools" on page 3-14 for web tool-related tables.

- "AXF_SOLUTIONS Table" on page 3-4
- "AXF_SOLUTION_ATTRIBUTES Table" on page 3-5
- "AXF_COMMANDS Table" on page 3-7
- "AXF_SOLUTION_PARAMETERS Table" on page 3-9

- "AXF_ACTION_MENU Table" on page 3-10
- "AXF_ACTIONS Table" on page 3-11
- "AXF_METADATA_BLOCKS Table" on page 3-18
- "AXF_METADATA_ATTRIBUTES Table" on page 3-19
- "AXF_XPATH_ATTRIBUTES Table" on page 3-13
- "AXF_XPATH_NAMESPACES Table" on page 3-14
- "Configuring the BPEL Connection" on page 3-5

3.2.1 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.



3.2.1.1 Column Description

Table 3–1 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.

3.2.1.2 Example Implementation

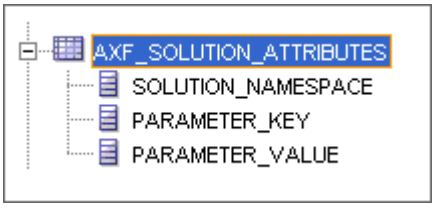
This example table shows the AXF solutions defined. Each of the solutions uses AxfCommandMediator as its solution implementation.

Table 3–2 Example AXF_SOLUTIONS Table

SOLUTION_NAMESPACE	SOLUTION_CONTEXT
InvoiceProcessing	ejb.AxfCommandMediator#oracle.imaging.axf.service.AxfCommandMediatorRemote
AccountDistributionApproval	ejb.AxfCommandMediator#oracle.imaging.axf.service.AxfCommandMediatorRemote
SupplierMaintenance	ejb.AxfCommandMediator#oracle.imaging.axf.service.AxfCommandMediatorRemote
RequestInvoiceInformation	ejb.AxfCommandMediator#oracle.imaging.axf.service.AxfCommandMediatorRemote
AccountDistribution	ejb.AxfCommandMediator#oracle.imaging.axf.service.AxfCommandMediatorRemote
InvoiceApproval	ejb.AxfCommandMediator#oracle.imaging.axf.service.AxfCommandMediatorRemote
Rescan	ejb.AxfCommandMediator#oracle.imaging.axf.service.AxfCommandMediatorRemote

3.2.2 AXF_SOLUTION_ATTRIBUTES Table

This table defines general attributes for use by infrastructure, services, or solutions. For example, use this table to define error message addresses, connections, and conversation timeout settings.



3.2.2.1 Column Description

Table 3–3 Column Description for AXF_SOLUTION_ATTRIBUTES Table

Column	Description
SOLUTION_NAMESPACE	Specifies the functional area that utilizes the parameter. Must correspond to a valid BPEL value. ■ *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_KEY	Name of the parameter. Used when retrieving the parameter value from the database. Parameters include: ■ BPEL_CONNECTION: Identifies the BPEL connection to be used. ■ 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. ■ IDENTITY_SERVICE_ENDPOINT: Specifies the URL point to BPEL identity web services to query the defined users in BPEL. ■ 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 3-27.
PARAMETER_VALUE	Value of the parameter.

3.2.2.2 Example Implementation

This example table sets solution attributes for the Invoice Processing solution.

Table 3–4 Example AXF_SOLUTION_ATTRIBUTES Table

SOLUTION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
InvoiceProcessing	BPEL_CONNECTION	axfconnection
InvoiceProcessing	CONNECTION_PROVIDER	oracle.imaging.axf.servicemodules.bpel.workflow.AxfWorkflowServiceModule
InvoiceProcessing	USE_AUTOTASK_LOCKING	TRUE

3.2.2.3 Configuring the BPEL Connection

Configuring the BPEL connection for use by an AXF solution involves the following tasks:

- "Creating a CSF Credential Alias" on page 3-6
- "Creating a Connection in I/PM Imaging Connections" on page 3-6
- "Referencing the Connection in the AXF_SOLUTION_ATTRIBUTES Table" on page 3-6
- "Configuring the URI to Display Images in the Task Viewer" on page 3-7

3.2.2.3.1 Creating a CSF Credential Alias The Credential Store Framework (CSF) enables you to create a username/password alias for use in an I/PM connection configuration. With a CSF alias, you supply a key instead of a username and password, and use this key in creating an I/PM connection. (You can use one CSF key for multiple imaging connections.)

For information about creating keys and aliases, see the *Oracle Fusion Middleware Administrator's Guide*.

3.2.2.3.2 Creating a Connection in I/PM Imaging Connections Follow these steps to create a connection and specify the CSF alias key, BPEL server name and port.

1. Log in to the I/PM imaging system as an administrator.
2. From Manage Connections in the side pane, click the + (plus) sign document icon for creating a BPEL connection.
3. Enter a name for the connection, and click **Next**.

   This name is referenced in the AXF_SOLUTION_ATTRIBUTES Table to establish the connection.
4. On the BPEL Settings step, enter the system name, port, and credential alias (previously created as described in "Creating a CSF Credential Alias" on page 3-6).

   For example:

   - System: <system>
   - Port: 8001
   - Credential Alias: axfconnection
5. Click **Next**, then **Submit**.

3.2.2.3.3 Referencing the Connection in the AXF_SOLUTION_ATTRIBUTES Table Follow these steps to identify the I/PM imaging connection to the AXF solution. Run the commands from SQL Developer (or other suitable tool that can connect to the imaging database schema).

1. Run the two configuration rows specified below, where:

   - *CONNECTION_NAME* identifies the connection name configured in I/PM Imaging Connections, as described in "Creating a Connection in I/PM Imaging Connections" on page 3-6.
   - *SOLUTION_NAMESPACE* identifies the solution. *InvoiceProcessing* is used in the example below. Modify this value if needed.

```
Insert into AXF_SOLUTION_ATTRIBUTES (SOLUTION_NAMESPACE,PARAMETER_KEY,PARAMETER_VALUE) values \
('InvoiceProcessing','BPEL_CONNECTION','CONNECTION_NAME');
Insert into AXF_SOLUTION_ATTRIBUTES (SOLUTION_NAMESPACE,PARAMETER_KEY,PARAMETER_VALUE) values \
('InvoiceProcessing','CONNECTION_PROVIDER','oracle.imaging.axf.servicemodules.bpel.workflow.
AxfWorkflowServiceModule');
```

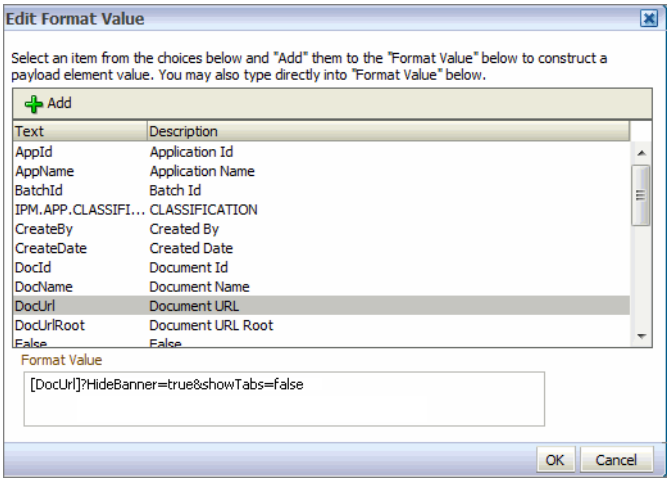
3.2.2.3.4 Configuring the URI to Display Images in the Task Viewer Follow the steps below to configure the URI for displaying images in the Task Viewer. You configure the DocURL payload element from the BPEL Injector as the URI and hide the banner.

1. Under Manage Applications, display the BPEL Payload Properties page for the I/PM application.

   For more information, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Imaging and Process Management*.

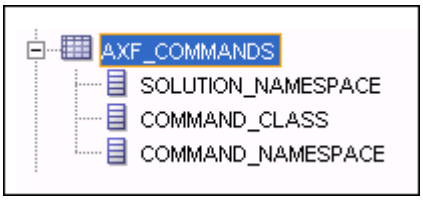
2. For the URI payload element, choose **Format Value** in the Mapped Value field.
3. Click the adjacent Edit Format Value icon.
4. In the Edit Format Value screen that is displayed (shown below), select DocUrl and click **Add**.
5. Construct the following value in the Format Value field and click **OK**.

   [DocUrl]?HideBanner=true&showTabs=false



3.2.3 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.



3.2.3.1 Column Description

Table 3–5 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.

Table 3–5 (Cont.) Column Description for AXF_COMMANDS Table

Column	Description
COMMAND_CLASS	The fully qualified class name in which the command is defined. This class is loaded and the execute() method representing the command is executed. For information about a specific task, see the specific task, listed under "AXF Tables" on page 3-1.

3.2.3.2 Example Implementation

This example shows commands defined for the Invoice Processing solution.

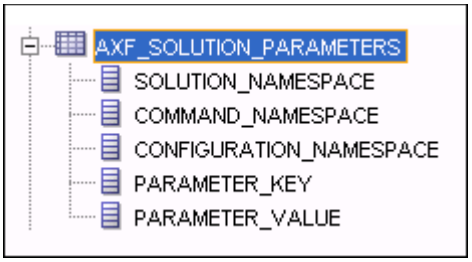
Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 3–6 Example AXF_COMMANDS Table

COMMAND_CLASS	COMMAND_NAMESPACE
oracle.imaging.axf.commands.bpel.AutotaskCommand	AutoOpenTask
oracle.imaging.axf.commands.bpel.ReleaseTaskCommand	ReleaseTask
oracle.imaging.axf.commands.bpel.ReleaseTaskCommand	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

3.2.4 AXF_SOLUTION_PARAMETERS Table

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



3.2.4.1 Column Description

Table 3–7 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 3-14 ■ "Task Viewer Web Tool" on page 3-16 ■ "Enumeration Picker Web Tool" on page 3-20 ■ "Identity Picker Web Tool" on page 3-24 ■ "Comments" on page 3-20 AXF Commands: ■ "Open Task Command" on page 3-26 ■ "Autotask Command" on page 3-26 ■ "Release Task Command" on page 3-27 ■ "Complete Task Command" on page 3-28 ■ "Redirect Command" on page 3-29 ■ "Update Task Command" on page 3-29 ■ "Update Task From Procedure Command" on page 3-31 ■ "Validate Task Command" on page 3-33
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 comes from the AXF_XPATH_ATTRIBUTES Table.

3.2.4.2 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.

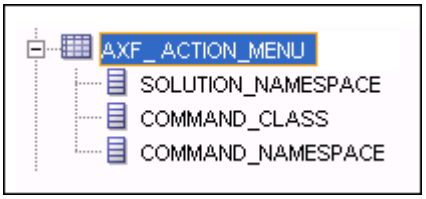
Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 3–8 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.axf.web.backing.TaskList	CMD_OPEN_ TASK_BUTTON	OpenTask
StartInvoiceProcessing	oracle.imaging.axf.web.backing.TaskList	CMD_AUTO_ TASK_BUTTON	AutoOpenTask
StartInvoiceProcessing	oracle.imaging.axf.web.backing.TaskList	DEFAULT_VIEW	(null)
StartInvoiceProcessing	oracle.imaging.axf.web.backing.TaskList	NO_OF_LINES	20
StartInvoiceProcessing	oracle.imaging.axf.web.backing.TaskList	SHOW_INBOX	FALSE
StartInvoiceProcessing	oracle.imaging.axf.web.backing.TaskList	CONNECTION_ NAME	default
StartInvoiceProcessing	oracle.imaging.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

3.2.5 AXF_ACTION_MENU Table

Use this table to insert and customize an action menu on the Task Viewer or Task List screen. A common use is to display a Task Actions pane in the Task Viewer for users to click action links related to the displayed task, as shown in Figure 3–2. Use the AXF_ACTIONS Table to define a specified menu's actions.



3.2.5.1 Column Description

Table 3–9 Column Description for AXF_ACTION_MENU Table

Column	Description
MENU_ID	Specifies a primary key to the AXF_ACTIONS Table, identifying the menu in which to place menu actions.
DISPLAY_TEXT	Specifies the pane's title (for example, *Task Actions*, *Shortcuts*, or *Re-Assignments*).
MENU_TYPE	Specifies where on the screen the menu is displayed and its type. (LEFT_SIDEBAR displays a side pane leftmost on the screen.)

Table 3–9 (Cont.) Column Description for AXF_ACTION_MENU Table

Column	Description
TASK_FLOW_ID	Specifies a task flow String that corresponds to a known task flow ID which loads a page or pages on the task flow.
VIEW_ID	(Reserved for future use.)
SOLUTION_NAMESPACE	Identifies the AXF solution, as defined in the AXF_SOLUTIONS Table.
MENU_ORDER	Defines the order in which the menu is displayed if multiple menus are set for display.

3.2.5.2 Example Implementation

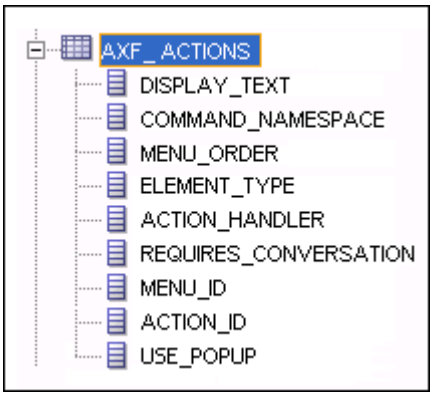
This example table implements a pane entitled *Task Actions* in the left side of the Task Viewer for the Invoice Processing solution.

Table 3–10 Example AXF_ACTION_MENU

MENU_ID	DISPLAY_TEXT	MENU_TYPE	TASK_FLOW_ID	VIEW_ID	SOLUTION_NAMESPACE	MENU_ORDER
0	Task Actions	LEFT_SIDEBAR	axf-taskviewer-tfd	null	InvoiceProcessing	0

3.2.6 AXF_ACTIONS Table

This table defines the task actions used in an AXF solution. You can display action menus in a Task Viewer page (see "Task Viewer Web Tool" on page 3-16) or a Task List (see "Task List Web Tool" on page 3-14). This table links to the AXF_COMMANDS Table.



3.2.6.1 Column Description

Table 3–11 Column Description for AXF_ACTIONS Table

Column	Description
DISPLAY_TEXT	Specifies the name of the action (link, for example) in the pane.
COMMAND_NAMESPACE	Specifies the command that is called as a result of the action, as defined in the AXF_COMMANDS Table.
MENU_ORDER	Specifies the display order of the action in the pane.
ELEMENT_TYPE	Specifies how to render the action on the page, where: ■ LINK: Displays an HTML link ■ BUTTON: Displays a button

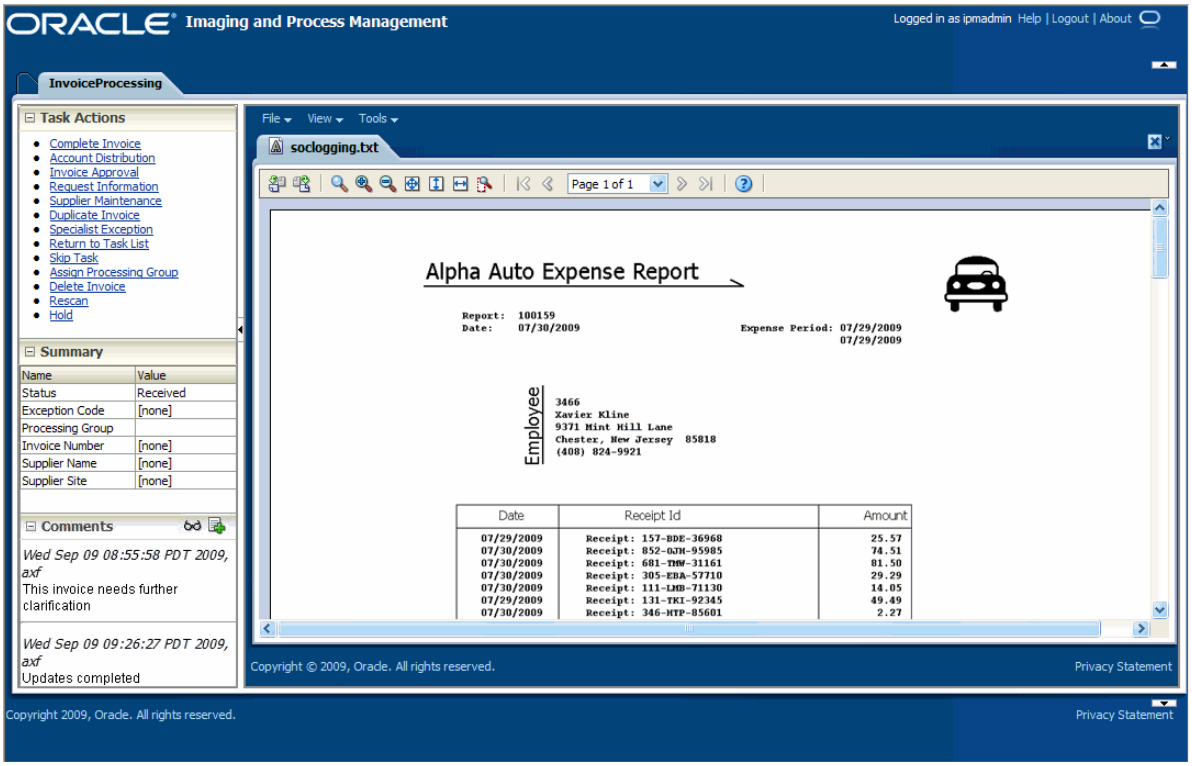
Table 3–11 (Cont.) Column Description for AXF_ACTIONS Table

Column	Description
ACTION_HANDLER	Determines how the command is handled, where COMMAND sends the command specified in the COMMAND_NAMESPACE column to the Solution Mediator. **Note:** If left (null), this value defaults to COMMAND.
REQUIRES_CONVERSATION	Specifies whether the action requires a conversation ID.
MENU_ID	Specifies the ID from the AXF_ACTION_MENU Table and defines the menu in which the action is displayed.
ACTION_ID	Defines the action's unique numeric identifier.
USE_POPUP	Reserved for future use.

3.2.6.2 Example Implementation

The tables that follow provide an example AXF_ACTIONS Table.

Figure 3–2 Task Viewer Page with Task Actions, Summary, and Comments Enabled



Fields not shown in Table 3–12:

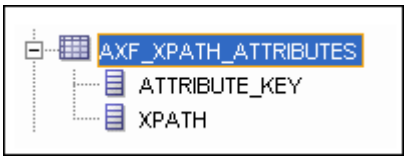
- ACTION_HANDLER=COMMAND
- REQUIRES_CONVERSATION=TRUE

Table 3–12 Example AXF_ACTIONS Table

DISPLAY_TEXT	COMMAND_ NAMESPACE	MENU_ ORDER	ELEMENT_ TYPE	MENU_ID	ACTION_ID
Invoice Approval	InvoiceApprovalEdit	1	LINK	0	AXF_ACTIONS_SEQ.NEXTVAL
Return to Task List	ReleaseTask	0	LINK	0	AXF_ACTIONS_SEQ.NEXTVAL
Skip Task	SkipTask	0	LINK	0	AXF_ACTIONS_SEQ.NEXTVAL
Complete Invoice	CompleteTask	0	LINK	0	AXF_ACTIONS_SEQ.NEXTVAL

3.2.7 AXF_XPATH_ATTRIBUTES Table

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



3.2.7.1 Column Description

Table 3–13 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.

3.2.7.2 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 comes from the AXF_XPATH_ATTRIBUTES table.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 3–14 Example AXF_SOLUTION_PARAMETERS Table

COMMAND_NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
AssignProcessingGroupEdit	oracle.imaging.axf.web.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 3–15 Example AXF_XPATH ATTRIBUTES Table

ATTRIBUTE_KEY	XPATH
InvoiceProcessing_ProcessingGroup	//task:processingGroup

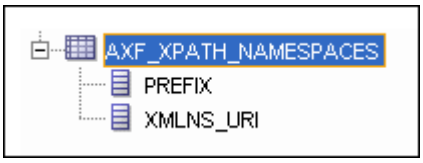
In the AXF_NAMESPACES table that follows, the XMLNS_URI column displays where within the XML file to locate the processingGroup task information.

Table 3–16 Example AXF_XPATH_NAMESPACES Table

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

3.2.8 AXF_XPATH_NAMESPACES Table

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



3.2.8.1 Column Description

Table 3–17 Column Description for AXF_XPATH_NAMESPACES Table

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

3.2.8.2 Example Implementation

Table 3–18 Example AXF_XPATH_NAMESPACES Table

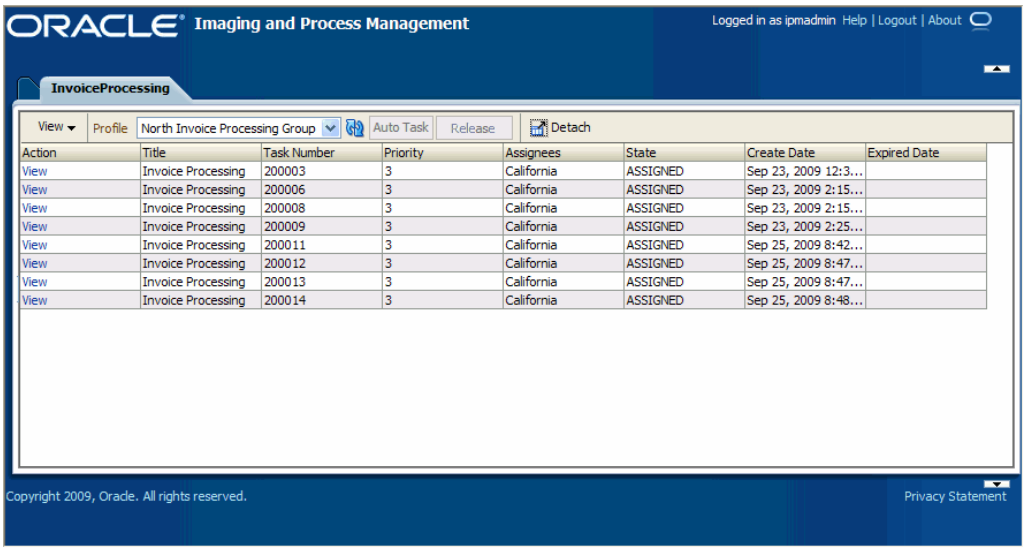
Prefix	XMLNS_URI
task	http://xmlns.oracle.com/bpel/workflow/task
documentContent	http://xmlns.oracle.com/imaging/axf/documentContentTypes
solution	http://xmlns.oracle.com/imaging/axf/solutionTypes
invoiceProcessing	http://xmlns.oracle.com/imaging/axf/InvoiceProcessing
xml	http://www.w3.org/XML/1998/namespace

3.3 AXF Web User Tools

This section covers the following topics:

3.3.1 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

- The **Profile** field uses standard BPEL views to restrict the task list view based on user/group, BPEL Process versions, and BPEL payload attribute values. (Use the BPM Worklist application to create views and share them with other users or groups.)
- Users can select a task from the table by clicking its **View Task** link, 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 Task**.
- Users can skip (release) an assigned task by clicking the **Release** button. The task is then released back into the pool of available tasks. If the user clicks Release but did not previously acquire the selected task, a message indicates that the task cannot be released.
- You can configure the Task List to include a side pane action list with links.

3.3.1.1 Task List Parameters

Table 3–19 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 link on the Task List web page.
CMD_AUTO_TASK_BUTTON	Specify a COMMAND_NAMESPACE to be executed when a user clicks the Auto Task button on the Task List web page.
CONNECTION_NAME	Specify the BPEL connection, as defined in the AXF_SOLUTION_ATTRIBUTES Table.
NO_OF_LINES	Specify the maximum number of tasks to be displayed before multiple pages are used.

Table 3–19 (Cont.) Task List Parameters in AXF_SOLUTION_PARAMETERS Table

Parameter	Description
BPEL_CONNECTION	Specify the BPEL connection that obtains the task list from BPEL, defined in the AXF_SOLUTION_ATTRIBUTES Table.
VIEW_LIST	Specify the list of views (defined in the Human workflow system) displayed to users in the Profile field.
DEFAULT_VIEW	Specify the default profile.
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 and the Inbox profile is selected.
REDIRECT_URL	Specify either: ▪ a task flow String corresponding to a task flow ID which loads one or more pages on the task flow. ▪ a standard URL string that redirects to the specified URL.

3.3.1.2 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. The remaining rows define the task list's behavior.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

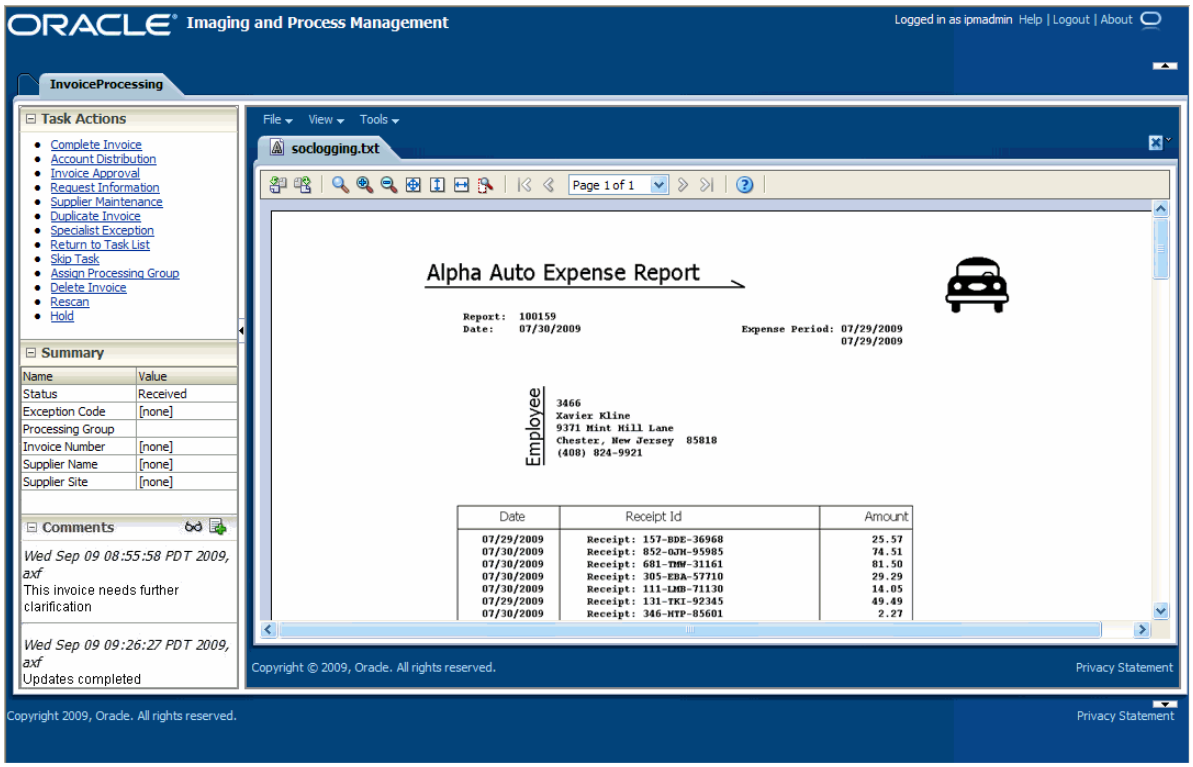
Table 3–20 Example Task List Parameters in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_ NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
StartInvoiceProcessing	oracle.imaging.axf.commands.bpel.RedirectCommand	REDIRECT_URL	taskflow://WEB-INF/taskflows/axf-tasklist-tfd.xml#axf-tasklist-tfd
StartInvoiceProcessing	oracle.imaging.axf.web.backing.TaskList	CMD_OPEN_TASK_BUTTON	OpenTask
StartInvoiceProcessing	oracle.imaging.axf.web.backing.TaskList	CMD_AUTO_TASK_BUTTON	AutoOpenTask
StartInvoiceProcessing	oracle.imaging.axf.web.backing.TaskList	DEFAULT_VIEW	(null)
StartInvoiceProcessing	oracle.imaging.axf.web.backing.TaskList	SHOW_INBOX	FALSE
StartInvoiceProcessing	oracle.imaging.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

3.3.2 Task Viewer Web Tool

The AXF Task Viewer web tool is a reusable web interface that displays 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 Java commands or AXF action commands.



Task Viewer Features

- Users view I/PM image documents in the **Image Viewer pane**, using either the basic or advanced I/PM viewer. Typically, the Task Viewer uses the I/PM 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 **Task Actions pane**, which invoke a solution's AXF commands. You enable the Task Actions pane in the AXF_ACTION_MENU Table, configure the action links to invoke AXF commands in the AXF_ACTIONS Table, and the commands themselves in the AXF_COMMANDS Table.
- Users can view a **Summary** section that displays metadata values about the task. You configure these items for display in the AXF_METADATA_BLOCKS Table and AXF_METADATA_ATTRIBUTES Table. You can also configure the section's title and the task payload values displayed.
- If configured, users can also view a **Comments** section that displays comment fragments and provides icons for displaying full comments or adding them for the task. You configure comments for display in the AXF_METADATA_BLOCKS Table.
- If autotask mode is selected, users disable it by returning to the Task List, typically by clicking a **Return to Task List** link in the Task Actions pane.

Configuring the Task Viewer

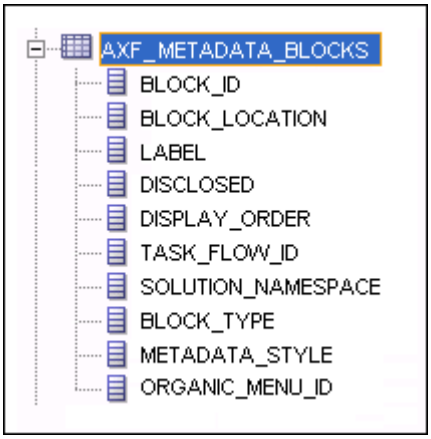
You configure the Task Viewer in the following tables:

- AXF_SOLUTION_PARAMETERS Table

- AXF_ACTION_MENU Table
- AXF_ACTIONS Table
- AXF_METADATA_BLOCKS Table
- AXF_METADATA_ATTRIBUTES Table

3.3.2.1 AXF_METADATA_BLOCKS Table

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



3.3.2.1.1 Column Description

Table 3–21 Column Description for AXF_METADATA_BLOCKS Table

Column	Description
BLOCK_ID	Identifies the row in the database. Links to the AXF_METADATA_ATTRIBUTES Table.
BLOCK_LOCATION	Specifies where the block is displayed on the Task Viewer page. *LEFT_SIDEBAR* displays a left sidebar pane. (Currently, this is the only value supported.)
LABEL	Defines the pane's label (for example, *Summary* or *Comments*).
DISCLOSED	TRUE if the block is displayed; FALSE if it is not displayed.
DISPLAY_ORDER	Specifies the order in which the block is displayed. The default value is 1.
TASK_FLOW_ID	Specifies the task flow on which to display the metadata block (for example, *axf-taskviewer-tfd* or *axf-tasklist-tfd*).
SOLUTION_NAMESPACE	Specifies the AXF solution name.
BLOCK_TYPE	Specifies the type of values contained in the block (for example, METADATA or COMMENT.
METADATA_STYLE	Reserved for future use.
ORGANIC_MENU_ID	Reserved for future use.

3.3.2.1.2 Example Implementation

This table displays the Summary and Comments section on the Task Viewer page.

Columns not shown: DISCLOSED=TRUE

METADATA_STYLE=null

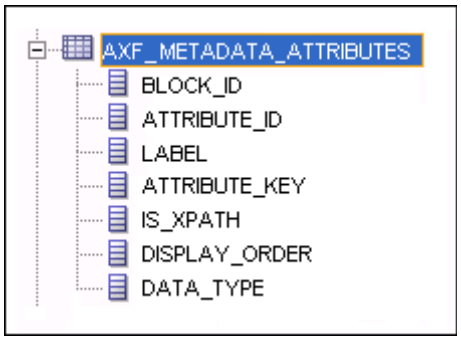
ORGANIC_MENU_ID=null

Table 3–22 Example AXF_METADATA_BLOCKS Table

BLOCK_ID	BLOCK_LOCATION	LABEL	DISPLAY _ORDER	TASK_FLOW_ID	SOLUTION_ NAMESPACE	BLOCK_ TYPE
1	LEFT_SIDEBAR	Summary	1	axf-taskviewer-tfd	InvoiceProcessing	METADATA
AXF_METADATA_ BLOCKS_ SEQ.NEXTVAL	LEFT_SIDEBAR	Comments	1	axf-taskviewer-tfd	InvoiceProcessing	COMMENT

3.3.2.2 AXF_METADATA_ATTRIBUTES Table

This table defines the labels and values to be shown in metadata sections specified in the "AXF_METADATA_BLOCKS Table" on page 3-18. It also defines how attribute values are retrieved for display via Xpath attributes.



3.3.2.2.1 Column Description

Table 3–23 Column Description for AXF_METADATA_ATTRIBUTES Table

Column	Description
BLOCK_ID	References the AXF_METADATA_BLOCKS Table in which to display metadata labels and values.
ATTRIBUTE_ID	Primary key for the metadata attribute.
LABEL	Specifies the metadata label displayed to users in the metadata section (for example, *Status* in a Summary section).
ATTRIBUTE_KEY	Specifies an attribute key that matches the Xpath attribute key in the AXF_XPATH_ATTRIBUTES Table, where it is used to look up and display the metadata value.
IS_XPATH	If TRUE, the attribute value comes from the xpath in the BPEL payload. If FALSE, the value comes from BPEL system attributes.
DISPLAY_ORDER	Specifies the order in which the metadata label/value are displayed in the metadata section.
DATA_TYPE	Specifies the metadata item's data type (for example, String).

3.3.2.2.2 Example Implementation

This table defines metadata labels and values displayed in a Task Viewer's Summary section, as shown in Figure 3–2.

Columns not shown:

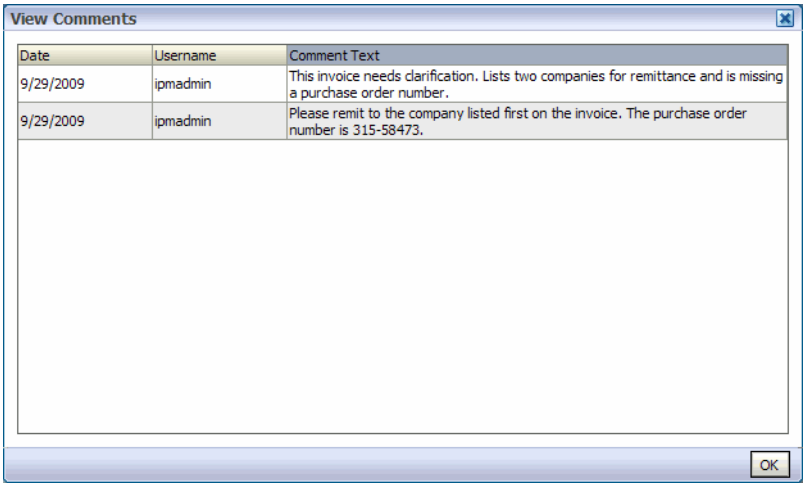
DATA_TYPE=String

Table 3–24 Example AXF_METADATA_ATTRIBUTES Table

BLOCK_ID	ATTRIBUTE_ID	LABEL	ATTRIBUTE_KEY	IS_ XPATH	DISPLAY_ ORDER
1	AXF_METADATA_ ATTRIBUTES_SEQ.NEXTVAL	Status	InvoiceProcessing_Status	TRUE	1
1	AXF_METADATA_ ATTRIBUTES_SEQ.NEXTVAL	Exception Code	InvoiceProcessing_ExceptionCode	TRUE	2
1	AXF_METADATA_ ATTRIBUTES_SEQ.NEXTVAL	Processing Group	InvoiceProcessing_ProcessingGroup	TRUE	3
1	AXF_METADATA_ ATTRIBUTES_SEQ.NEXTVAL	Invoice Number	InvoiceProcessing_InvoiceNumber	TRUE	4
1	AXF_METADATA_ ATTRIBUTES_SEQ.NEXTVAL	Supplier Name	InvoiceProcessing_SupplierName	TRUE	5
1	AXF_METADATA_ ATTRIBUTES_SEQ.NEXTVAL	Supplier Site	InvoiceProcessing_SupplierSiteName	TRUE	6

3.3.2.3 Comments

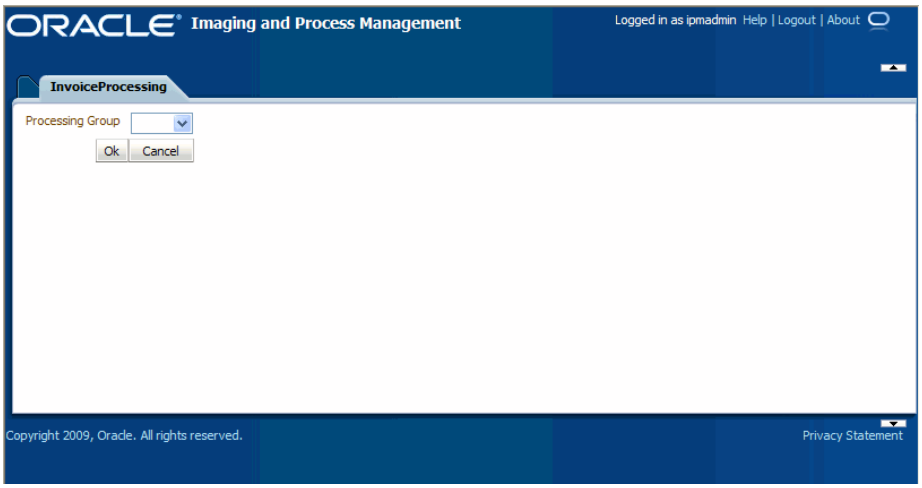
The Comments pane allows users to view and enter comments related to the human task during the transaction's processing. You configure comments in the AXF_METADATA_BLOCKS Table.



3.3.3 Enumeration Picker Web Tool

The Enumeration Picker web tool allows users to select a choice from a list of values configured in AXF tables. For example, the Enumeration Picker shown in the graphic that follows displays a Processing Group dropdown field containing North, South, East, and West values.

After the user selects a value, the value is updated into the BPEL payload 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 3–25).
- Define the picker in the AXF_ENUM_TYPES Table.
- Define the picker's values in the AXF_ENUM_ITEMS Table.

3.3.3.1 Enumeration Picker Parameters

Table 3–25 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 3-30. 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.

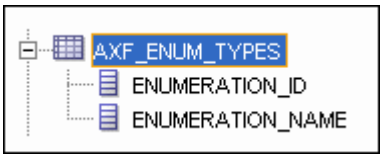
Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 3–26 Example Enumeration Picker Parameters in AXF_SOLUTIONS Table

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

3.3.3.2 AXF_ENUM_TYPES Table

This table defines Enumeration Pickers.



Column Description

Table 3–27 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

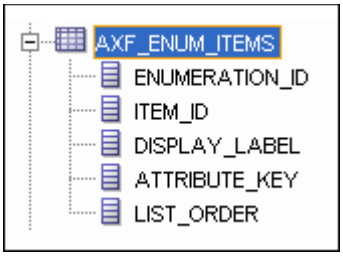
This example defines enumeration pickers for the Invoice Processing solution. Each picker's items are defined in the AXF_ENUM_ITEMS Table.

Table 3–28 Example AXF_ENUM_TYPES Table

ENUMERATION_ID	ENUMERATION_NAME
1	ProcessingGroups
2	SupplierMaintenanceCodes
3	SpecialistExceptionCodes
4	RescanCodes

3.3.3.3 AXF_ENUM_ITEMS Table

This table defines a specified Enumeration Picker's values.



Column Description

Table 3–29 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.
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 3–30 Example AXF_ENUM_ITEMS Table

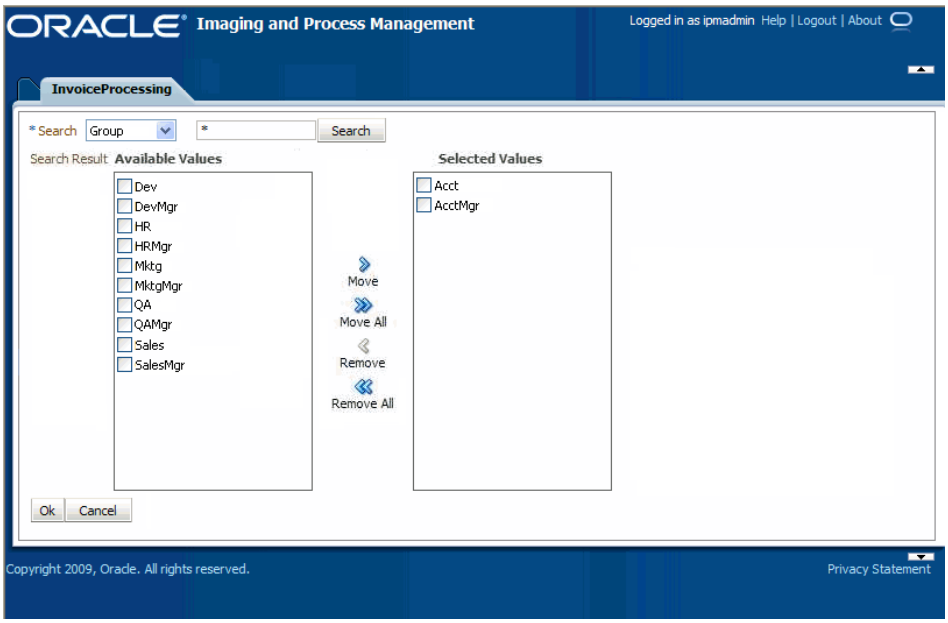
ENUMERATION_ID	ITEM_ID	DISPLAY_LABEL	ATTRIBUTE_KEY	LIST_ORDER
1	1	North	North	1
1	2	South	South	2
1	3	East	East	3
1	4	West	West	4
2	5	No Supplier	No Supplier	1
2	6	No Supplier Site	No Supplier Site	2
3	7	Duplicate Invoice	Duplicate Invoice	1
3	8	Invalid Invoice Number	Invalid Invoice Number	2
3	9	No 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

3.3.4 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. 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 OK. 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.



3.3.4.1 Identity Picker Parameters

Table 3–31 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.

Table 3–31 (Cont.) Identity Picker Parameters in AXF_SOLUTION_PARAMETERS Table

Parameter Key	Description
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. **Note:** Specify USER or GROUP for a command. To allow both search types, create an additional command that uses the other type to open the identity picker. For example, you might create AssignByGroup and AssignByUser commands.
IDENTITY_ATTRIBUTE	This attribute is updated in the BPEL task when a user clicks OK on the Identity Picker page. The Attribute value is a constant as defined under "System Attributes" on page 3-30. If the value has an XPATH: prefix, then the value comes from the AXF_XPATH_ATTRIBUTES Table and it is the XPATH to update the value in the task payload.

3.3.4.2 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.

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 3–32 Example Identity Picker Parameters in AXF_SOLUTION_PARAMETERS table

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

3.4 AXF Commands

AXF commands include:

- "Open Task Command" on page 3-26
- "Autotask Command" on page 3-26
- "Release Task Command" on page 3-27
- "Complete Task Command" on page 3-28
- "Redirect Command" on page 3-29
- "Update Task Command" on page 3-29
- "Update Task From Procedure Command" on page 3-31
- "Terminate Conversation Command" on page 3-33
- "Validate Task Command" on page 3-33

AXF command-related topics include:

- "Custom Commands" on page 3-34

- "Configuring Chained Commands and Web Tools" on page 3-34

3.4.1 Open Task Command

This command acquires a task from BPEL (human work flow) for a given task ID; the specific task is likely selected from the task list. If the task can be acquired by the user, the command obtains the details of the task and displays the specified web page.

3.4.1.1 Open Task Command Parameters

Table 3–33 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

Table 3–33 Parameters for OpenTask Command

Parameter Key	Description
TASK_VIEW_URL	This task flow is returned in the response command upon executing this command. The value for the TASK_VIEW_URL parameter uses one of the following strings to represent task flows. Each string can be thought of as a special URL where *taskflow:* is the protocol instead of *http*. ■ taskflow://WEB-INF/taskflows/axf-tasklist-tfd.xml#axf-tasklist-tfd (displays the Task List) ■ taskflow://WEB-INF/taskflows/axf-taskviewer-tfd.xml#axf-taskviewer-tfd (displays the Task Viewer) ■ taskflow://WEB-INF/taskflows/axf-identity-picker-tfd.xml#axf-identity-picker-tfd (displays the Identity Picker) ■ taskflow://WEB-INF/taskflows/axf-enumeration-picker-tfd.xml#axf-enumeration-picker-tfd (displays the Enumeration Picker) ■ taskflow://WEB-INF/taskflows/axf-comments-tfd.xml#axf-comments-tfd (displays Comments)

3.4.1.2 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 3–34 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	taskflow://WEB-INF/taskflows/axf-taskviewer-tfd.xml#axf-taskviewer-tfd

3.4.2 Autotask Command

This command initializes autotask mode, in which a new human workflow task is automatically claimed for the user.

3.4.2.1 Autotask Command Parameters

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

Table 3–35 Autotask Command Parameters in AXF_SOLUTION_PARAMETERS Table

Parameter Key	Description
TASK_VIEW_URL	Task flow 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.

3.4.2.2 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 3–36 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	taskflow://WEB-INF/taskflows/axf-taskviewer -tfd.xml#axf-taskviewer-tfd
AutoOpenTask	oracle.imaging.axf.commands.bpel. AutotaskCommand	CMD_ON_NO_ TASKS	StartInvoiceProcessing
AutoOpenTask	oracle.imaging.axf.commands.bpel. AutotaskCommand	BPEL_TRY_AUTO	3000

3.4.2.3 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 prevents 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_SOLUTION_ATTRIBUTES Table by inserting the following row:

NAMESPACE	PARAMETER_ TYPE	PARAMETER_NAME	PARAMETER_VALUE
BPEL.default	connection	USE_AUTOTASK_LOCKING	true

3.4.3 Release Task Command

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

3.4.3.1 Release Task Command Parameters

Table 3–37 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

Table 3–37 Release 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.

3.4.3.2 Example Implementation

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 3–38 Example Release Task Commands in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_ VALUE
SkipTask	oracle.imaging.axf.commands.bpel.ReleaseTaskCommand	CMD_AUTOTASK_OFF	StartInvoiceProcessing
SkipTask	oracle.imaging.axf.commands.bpel.ReleaseTaskCommand	CMD_AUTOTASK_ON	AutoTaskOpen
ReleaseTask	oracle.imaging.axf.commands.bpel.ReleaseTaskCommand	CMD_AUTOTASK_OFF	StartInvoiceProcessing
ReleaseTask	oracle.imaging.axf.commands.bpel.ReleaseTaskCommand	CMD_AUTOTASK_ON	StartInvoiceProcessing

3.4.4 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.

In addition, the Complete Task command can also update BPEL payload attribute values using request parameters to the command. If auto-task mode is active, the command claims the next available task and displays in the Task Viewer. If auto-task mode is not active, the command displays the Task List.

3.4.4.1 Complete Task Command Parameters

Table 3–39 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

Table 3–39 CompleteTask Command Parameters

Parameter Key	Description
OUTCOME	Specify the outcome defined for the human work flow system. Default bpel outcomes are singular, APPROVE, 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.

3.4.4.2 Example Implementation

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 3–40 Example Complete Task Command in AXF_SOLUTION_PARAMETERS Table

COMMAND_ NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_ VALUE
DeleteInvoice	oracle.imaging.axf.commands.bpel.CompleteTaskCommand	CMD_AUTOTASK_OFF	StartInvoiceProcessing
DeleteInvoice	oracle.imaging.axf.commands.bpel.CompleteTaskCommand	CMD_AUTOTASK_ON	AutoOpenTask
DeleteInvoice	oracle.imaging.axf.commands.bpel.CompleteTaskCommand	OUTCOME	DELETE_INVOICE

3.4.5 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 3–41 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

3.4.5.1 Redirect Command Parameters

Table 3–41 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.

3.4.5.2 Example Implementation

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 3–42 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
StartInvoiceProcessing	oracle.imaging.axf.commands.system. RedirectCommand	REDIRECT_URL	taskflow://WEB-INF/taskflows/axf-tasklist-tfd.xml#axf-tasklist-tfd

3.4.6 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 3-31.)

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 from those listed in "System Attributes" on page 3-30. 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.

3.4.6.1 Update Task Parameters

Table 3–43 Parameters for UpdateTaskCommand

Parameter Key	Description
outcome	Specify the outcome defined for the human work flow system. Default bpel outcomes are singular, APPROVE, or REJECT.

3.4.6.2 System Attributes

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

3.4.6.3 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 3–44 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

3.4.7 Update Task From Procedure Command

The Update Task From Procedure command calls a stored pl/sql procedure using a specified data source and updates the task payload using XPATH.

3.4.7.1 Update Task From Procedure Parameters

Table 3–45 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 returned data 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.

3.4.7.2 Example Implementation

Fields not shown: SOLUTION_NAMESPACE=InvoiceProcessing

Table 3–46 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

Table 3–46 (Cont.) Example UpdateTaskFromProcedureCommand Parameters in AXF_SOLUTION_PARAMETERS Table

COMMAND_NAMESPACE	CONFIGURATION_NAMESPACE	PARAMETER_KEY	PARAMETER_VALUE
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

3.4.7.3 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;
```

3.4.8 Terminate Conversation Command

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

3.4.9 Validate Task Command

The Validate Task command validates BPEL system attribute data or BPEL payload data, and based on validation results, executes a subsequent command.

Table 3–47 lists configuration parameters for this command. These parameters are used in the AXF_SOLUTION_PARAMETERS Table to configure commands.

3.4.9.1 Validate Task Command Parameters

Table 3–47 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.
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.
FAIL_MESSAGE	Specifies the message to display if the validation fails.

3.4.9.2 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 3–48 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 before completing the task.

3.4.9.3 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 business application 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 3–49 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 3–48 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 3–50 AXF_ACTIONS Table

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

3.4.10 Custom Commands

You can also deploy custom commands to work within the AXF infrastructure. Custom commands must implement the oracle.imaging.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.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.

3.4.11 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 3–51 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 3–51 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

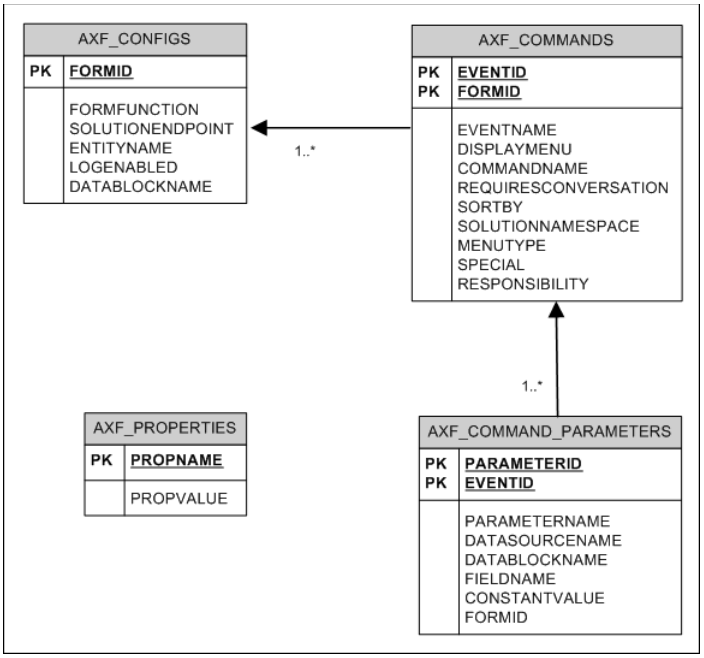
4

E-Business Suite Tables

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

4.1 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.



4.2 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 datablocks on a form rather than a specific datablock, 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 datablock. 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.)

4.2.1 Column Description

Table 4–1 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)

4.2.2 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.)

Table 4–2 Example AXF_CONFIGS Table

FORMID	FORMFUNCTION	SOLUTIONENDPOINT	ENTITYNAME	LOG ENABLED	DATABLOCKNAME
1	AP_APXINWKB	http://<ApplicationServerName>:<SOA Port>/axf-ws/AxfSolutionMediatorService	AP_INVOICES	YES	AXF_DEFAULT
2	AP_APXINWKB_ SUMMARY_VIEW	http://<ApplicationServerName>:<SOA Port>/axf-ws/AxfSolutionMediatorService	AP_INVOICES	YES	INV_SUM_FOLDER
6	AP_APXINWKB_ BATCHES	http://<ApplicationServerName>:<SOA Port>/axf-ws/AxfSolutionMediatorService	AP_INVOICES	YES	INV_SUM_FOLDER

4.2.3 Enabling E-Business Suite Logging

To enable logging for a particular Form function, set the LOGENABLED field to either `1`, `YES` or `TRUE` and the file is 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 continue to grow as items are appended.

4.3 AXF_COMMANDS Table

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

4.3.1 Column Description

Table 4–3 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
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

Table 4–3 (Cont.) Column Description for AXF_COMMANDS Table

Column Name	Description	Data Type	Nullable
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	Reserved for future use.	Varchar2(100 byte)	Yes

4.3.2 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 automatically invokes 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 4–4 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
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)

4.4 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.

4.4.1 Column Description

Table 4–5 Column Description for AXF_COMMAND_PARAMETERS Table

Column	Description
PARAMETERID	Defines a unique ID for the parameter.

Table 4–5 (Cont.) Column Description for AXF_COMMAND_PARAMETERS Table

Column	Description
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.

4.4.2 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 opens another instance of the browser. Users may then take additional steps in the newly created window.

Table 4–6 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)
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)

4.5 AXF_PROPERTIES Table

Use the AXF_PROPERTIES table to define properties for AXF integration for E-Business Suite.

4.5.1 Column Description

Table 4–7 Column Description for AXF_PROPERTIES Table

Column	Description
PROPNAME	Specifies properties to be used. Properties include: ■ SecureMode: To enable SSL, set this property to ON, and set values for AXFWalletPath and AXFWalletPwd properties. ■ AXFWalletPath: Certificate location (path). ■ AXFWalletPwd: Password for Wallet Manager. ■ AXF_VERSION: Specify `1` for AXF 10g, or `2` for AXF 11g. ■ AXF_SOAP_SECURITY: Specify TRUE to enable SOAP security, as described in "Setting SOAP Security" on page 4-6. ■ AXF_SOAP_USER: Specify the SOAP userid used in the SOAP header for authentication, as described in "Setting SOAP Security" on page 4-6. ■ AXF_SOAP_POLICY: Specify the name of the SOAP policy to be used. The currently supported policy is USER_NAME_TOKEN.
PROPVALUE	Specifies the property's value.

4.5.2 Example Implementation

This example table shows the default properties values.

Table 4–8 Example AXF_PROPERTIES Table

PROPNAME	PROPVALUE
SecureMode	OFF
AXFWalletPath	file:<walletpath>
AXFWalletPwd	walletpassword
AXF_VERSION	2
AXF_SOAP_POLICY	USER_NAME_TOKEN
AXF_SOAP_SECURITY	FALSE
AXF_SOAP_USER	AXF

4.5.2.1 Setting SOAP Security

Follow these steps to configure SOAP security, in which the application sends the SOAP user and password in the header for authentication.

1. Enable SOAP security by specifying `TRUE` for the AXF_SOAP_SECURITY property in the AXF_PROPERTIES Table.
2. Set the AXF_SOAP_POLICY property to `USER_NAME_TOKEN`.
3. Store the SOAP password in the database vault by executing the following command as APPS schema:

```
execute fnd_vault.put('AXF','<AXF_SOAP_USER>','<SOAP_
PASSWORD');
```

Where <AXF_SOAP_USER> is the SOAP user id used in the SOAP header for authentication, and <SOAP_PASSWORD> is the SOAP password.

4. Verify the previous command with this statement:

```
select fnd_vault.get ('AXF','<AXF_SOAP_USER>') from dual;
```

Index

A

B

C

E

G

I

L

O

P

R

S

T

U

V

X