Oracle® BPEL Process Manager

Administrator's Guide

10*g* (10.1.3.1.0)

Part No. B28982-03

January 2007

Oracle BPEL Process Manager Administrator’s Guide, 10*g* (10.1.3.1.0)

Part No. B28982-03

Primary Author: Mark Kennedy

Contributor: Oracle BPEL Process Manager development, product management, and quality assurance teams

Contents

2 Service Configuration

4 Configuring and Viewing BPEL Process Logs

A Demo User Community

Index

Preface

This manual describes how to administer Oracle BPEL Process Manager.

This preface contains the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This manual is intended for anyone who is interested in administering Oracle BPEL Process Manager.

> **Note:** The chapter on Oracle BPEL Process Manager performance tuning that appeared in this guide for 10.1.2.0.2 has been moved to the *Oracle Application Server Performance Guide* for release 10.1.3.1.0.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see the following Oracle resources:

- *Oracle Application Server Performance Guide* for Oracle BPEL Process Manager tuning and performance information
- *Oracle BPEL Process Manager Quick Start Guide*
- *Oracle BPEL Process Manager Order Booking Tutorial*
- *Oracle BPEL Process Manager Developer's Guide*
- *Oracle Adapters for Files, FTP, Databases, and Enterprise Messaging User's Guide*
- *Oracle Application Server Adapter Concepts*
- *Oracle Application Server Adapter for Oracle Applications User's Guide*

Printed documentation is available for sale in the Oracle Store at

http://oraclestore.oracle.com/

To download free release notes, installation documentation, white papers, or other collateral, visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

http://www.oracle.com/technology/membership/

To download Oracle BPEL Process Manager documentation, technical notes, or other collateral, visit the Oracle BPEL Process Manager site at Oracle Technology Network (OTN):

http://www.oracle.com/technology/bpel/

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

http://www.oracle.com/technology/documentation/

See the *Business Process Execution Language for Web Services Specification*, available at the following URL:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbizspec/html/bpel1-1.asp

See the *XML Path Language (XPath) Specification*, available at the following URL:

http://www.w3.org/TR/1999/REC-xpath-19991116

See the *Web Services Description Language (WSDL) 1.1 Specification*, available at the following URL:

http://www.w3.org/TR/wsdl

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

Oracle BPEL Process Manager Security

It is critical to control access to BPEL processes and to the Web services they use. Preventing unauthorized users from breaking into your system is required to protect both the integrity of your processes and the personal information of your customers. This chapter describes the methods available for securing BPEL processes and invoking secured Web services with Oracle BPEL Process Manager.

This chapter contains the following topics:

Security Overview

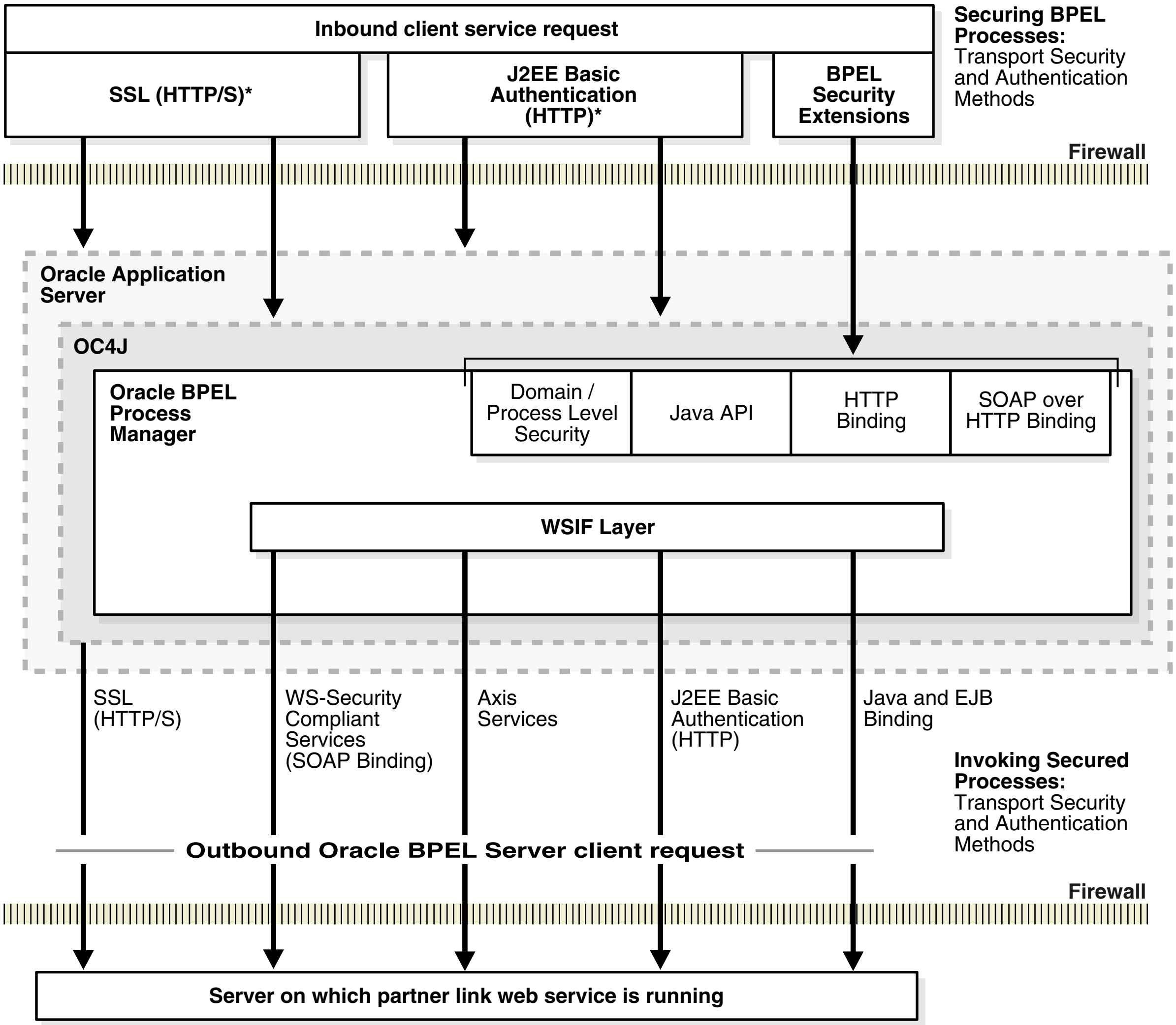
Security in Oracle BPEL Process Manager is implemented as follows:

- Securing a BPEL process in which interaction is initiated by an inbound client service request sent to Oracle BPEL Server. The following transport security and authentication methods are available:
  - SSL (HTTP/S)
  - J2EE basic authentication (HTTP)
  - BPEL security extensions
- Invoking secured services in which interaction is initiated by an outbound client request sent from Oracle BPEL Server to the server on which the partner link Web service is running. The following transport security and authentication methods are available:
  - SSL (HTTP/S)
  - WS-Security-compliant services
  - Axis services

- J2EE basic authentication (HTTP)
- Java and Enterprise Java Bean (EJB) binding

Figure 1–1 provides an overview of these transport security and authentication methods available for securing BPEL processes (inbound) and invoking secured services (outbound):

Figure 1–1 Inbound and Outbound Transport Security and Authentication Methods


Inbound client service request
SSL (HTTP/S)*
J2EE Basic Authentication (HTTP)*
BPEL Security Extensions
Securing BPEL Processes: Transport Security and Authentication Methods
Firewall
Oracle Application Server
OC4J
Oracle BPEL Process Manager
Domain / Process Level Security
Java API
HTTP Binding
SOAP over HTTP Binding
WSIF Layer
SSL (HTTP/S)
WS-Security Compliant Services (SOAP Binding)
Axis Services
J2EE Basic Authentication (HTTP)
Java and EJB Binding
Invoking Secured Processes: Transport Security and Authentication Methods
Outbound Oracle BPEL Server client request
Firewall
Server on which partner link web service is running

* With the Oracle BPEL Process Manager for OracleAS Middle Tier installation type, inbound client service requests that use SSL transport security and J2EE basic authentication are verified by Oracle Application Server. With the Oracle BPEL Process Manager for Developers installation type, inbound client service requests that use SSL transport security and J2EE basic authentication are verified by OC4J.

This section provides an overview of the following security features in the context of Oracle BPEL Process Manager. References are also provided to sections that describe these features in more detail:

- WS-Security
- Authentication
- Authorization
- Encryption and Decryption
- Secure Socket Layer
- Digital Signatures for Integrity and Nonrepudiation
- BPEL Security Extensions

WS-Security

WS-Security provides a mechanism for adding three levels of security to simple object access protocol (SOAP) messages. These security levels are as follows:

- Authentication tokens – Used for passing user name and password information, as well as X.509 certificates, within the SOAP message headers.
- XML encryption – Used for message confidentiality.
- XML digital signatures – Used for message integrity, source and origin validation, and nonrepudiation.

See Also:

- "WS-Security-Compliant Services" on page 1-18
- Web Services Security (WS-Security) Specifications available at the following URL:

  `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss`

Authentication

Authentication is the process of proving the identity of a user. Oracle BPEL Process Manager supports basic authentication (HTTP), certificate-based authentication (HTTP/S), and native BPEL security extension authentication.

See Also: The following sections:

- "Securing BPEL Processes (Inbound)" on page 1-5 for instructions on securing a BPEL process
- "Invoking Secured Services (Outbound)" on page 1-14 for instructions on invoking secured services from BPEL

Authorization

Authorization is the evaluation of security constraints to send a message or make a request. Authorization uses specific criteria to determine whether to permit the request. The criteria are authentication and restriction. Oracle BPEL Process Manager has no current native support for inbound authorization. Oracle Web Services Manager can instead be used to provide this capability.

See Also: "Authorization" on page 1-29

Encryption and Decryption

Encryption is the practice of encoding (encrypting) data in such a way that only an intended recipient can decode (decrypt) and read the data. Oracle BPEL Process Manager has no current native support for XML encryption. Oracle Web Services Manager can instead be used to provide this capability.

See Also: "Message Encryption and Decryption" on page 1-29

Secure Socket Layer

Secure Socket Layer (SSL) is a standard for the secure transmission of documents over the Internet using HTTP/S (secure HTTP). SSL uses digital signatures to prevent data from being compromised.

See Also:

- "Using SSL for Certificate-Based Authentication" on page 1-6 for details about using SSL to secure BPEL processes
- "Using SSL for Certificate-Based Authentication" on page 1-15 for details about using SSL to invoke secured services

Digital Signatures for Integrity and Nonrepudiation

A digital signature is a code attached to an electronic document that reliably identifies the author or sender, and verifies that the document has not been compromised. Oracle BPEL Process Manager has no current native support for digital signatures. Oracle Web Services Manager can instead be used to provide digital signatures and signature verification capabilities.

See Also: "Digital Signatures" on page 1-29

BPEL Security Extensions

BPEL security extensions are fully integrated into Oracle BPEL Process Manager.

Regardless of which channel you use to invoke a process (such as HTTP, SOAP, Java API, and so on), the same security constraints apply. However, the way credentials are passed differs amongst channels.

BPEL security extensions are intended for BPEL developers who want to enhance the security of Oracle BPEL Process Manager. These extensions are technical and require a good understanding of Oracle BPEL Server, including the various technologies used for invoking processes (for example, SOAP and HTTP). There are also many references to the Oracle BPEL Java API, so a good knowledge of Java is required.

Oracle BPEL Process Manager’s API includes BPEL security extensions that enable developers to create custom security. This is necessary in secure environments where users must be authenticated and authorized to use certain BPEL processes.

See Also:

- "Using the Native BPEL Security Extensions" on page 1-10
- "Domain and Process Level Security" on page 1-11
- "Java API" on page 1-13
- "HTTP Binding" on page 1-13
- "SOAP over HTTP Binding" on page 1-13
- "Java and EJB Binding (10.1.3)" on page 1-21
- "Default and Custom Validators" on page 1-25
- *Oracle BPEL Process Manager Workflow Services API Reference*, which is also located in

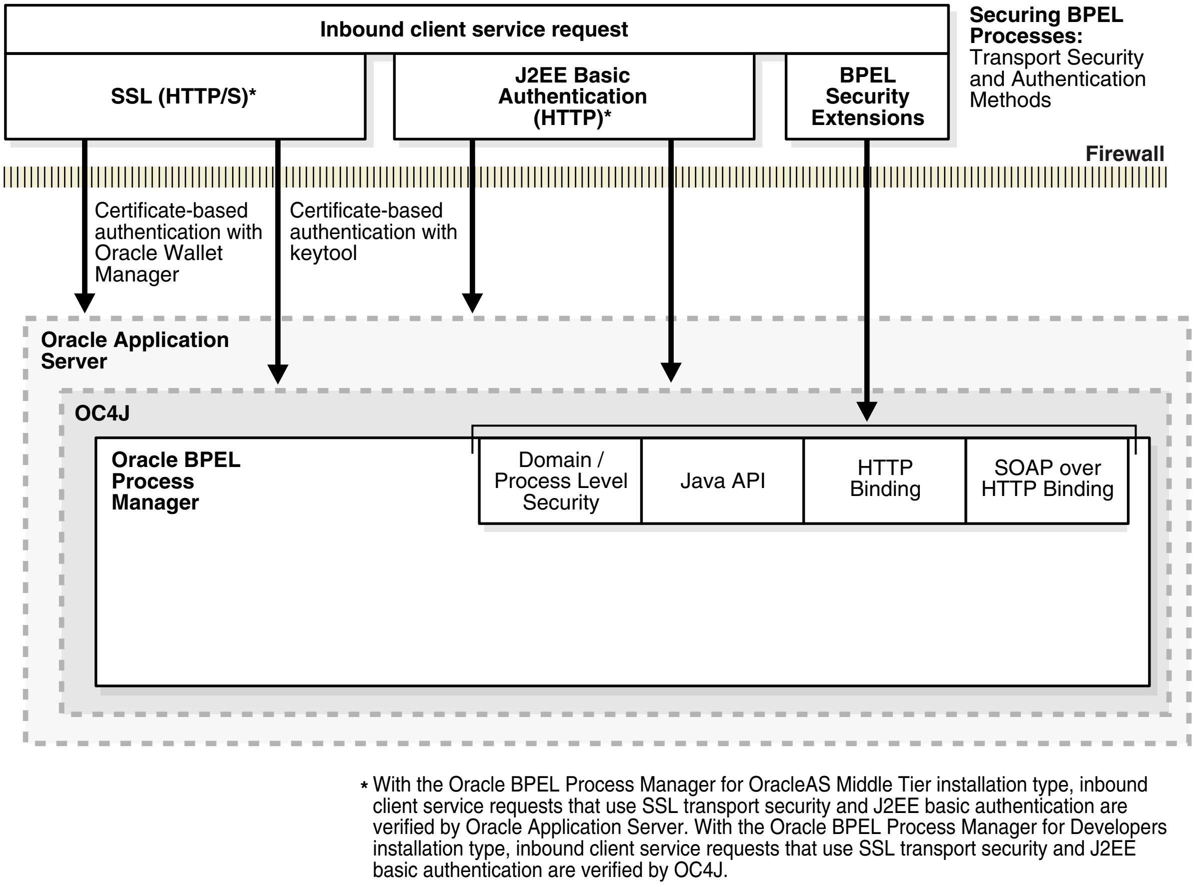
  SOA_Oracle_Home\bpel\docs\apidocs

Securing BPEL Processes (Inbound)

You can secure a BPEL process in which interaction is initiated by an inbound client service request sent to Oracle BPEL Server.

Figure 1–2 provides a high-level overview of the transport security and authentication methods available for securing BPEL processes (inbound).

Figure 1–2 Securing BPEL Processes (Inbound)


Inbound client service request
SSL (HTTP/S)*
J2EE Basic Authentication (HTTP)*
BPEL Security Extensions
Securing BPEL Processes: Transport Security and Authentication Methods
Firewall
Certificate-based authentication with Oracle Wallet Manager
Certificate-based authentication with keytool
Oracle Application Server
OC4J
Oracle BPEL Process Manager
Domain / Process Level Security
Java API
HTTP Binding
SOAP over HTTP Binding
* With the Oracle BPEL Process Manager for OracleAS Middle Tier installation type, inbound client service requests that use SSL transport security and J2EE basic authentication are verified by Oracle Application Server. With the Oracle BPEL Process Manager for Developers installation type, inbound client service requests that use SSL transport security and J2EE basic authentication are verified by OC4J.

This section describes how to provide BPEL process security through the following methods:

- Using SSL for Certificate-Based Authentication
- Using J2EE Basic Authentication
- Using the Native BPEL Security Extensions

> **Note:** Oracle recommends that you create an environment in which one or more instances of a server are dedicated to secure business processes and other instances are set up to host nonsecure processes.

Using SSL for Certificate-Based Authentication

BPEL processes are usually invoked using SOAP over HTTP. While basic authentication ensures that only authenticated users access BPEL processes, user names and password are prone to identification by network packet sniffers. Therefore, the need exists for securing the network connection through use of HTTP/S instead of HTTP. If you use HTTP/S as the authentication schema, both the client and server can be configured to exchange certificates. A successful SSL handshake confirms authentication.

The following types of certification authentication can be used:

- Server certificate authentication

  In this scenario, the client asks the server for the certificate and authenticates the trustworthiness of the server. The client does not present its certificate unless it is requested by the server to do so. The type of server presenting the certificate is based upon the Oracle BPEL Process Manager installation type you are using:

  - For Oracle BPEL Process Manager for OracleAS Middle Tier, the server is Oracle Application Server (and its version of OC4J)
  - For Oracle BPEL Process Manager for Developers, the server is the standalone OC4J in which Oracle BPEL Process Manager is deployed.

- Server and client certificate authentication

  In this scenario, both the client and server exchange certificates and a successful SSL handshake confirms authentication. This is called client authentication mode. The server (either the standalone OC4J in which Oracle BPEL Process Manager is deployed or Oracle Application Server (and its version of OC4J)) must be configured to request the client's certificate during the SSL handshake and authenticate the trustworthiness of the client. In the context of securing BPEL processes, this means that a client invoking a service presents a valid certificate issued by a mutually-trusted certificate authority. Server and client certificate authentication is not as frequently used.

The following sections describe the SSL configuration method to use based on the Oracle BPEL Process Manager installation type you are using:

- Oracle BPEL Process Manager for OracleAS Middle Tier
- Oracle BPEL Process Manager for Developers

Oracle BPEL Process Manager for OracleAS Middle Tier

SSL configuration for Oracle BPEL Process Manager for OracleAS Middle Tier is a two-step process:

- Step 1: Configuring OC4J
- Step 2: Configuring Oracle BPEL Server

Step 1: Configuring OC4J Use Oracle Wallet Manager to enable certificate-based authentication with the Oracle BPEL Process Manager for OracleAS Middle Tier installation type. (See Figure 1–2 on page 1-5.) Oracle Wallet Manager is an application for managing and editing security credentials in Oracle wallets. A wallet is a password-protected container that stores authentication and signing credentials, including private keys, certificates, and trusted certificates, all of which are used by SSL for strong authentication.

> **Note:** Do *not* use the default certificate included with Oracle Wallet (named `test`). The default certificate does not use the proper server host name. Instead, obtain a certificate from a certificate authority. This certificate must contain the proper server host name in the CN entry.

> **See Also:** *Oracle Application Server Administrator's Guide* for the following SSL configuration details:
>
> - Setting up a wallet and using Oracle Wallet Manager
> - Obtaining a certificate from a certificate authority

Step 2: Configuring Oracle BPEL Server Oracle BPEL Server must be configured with the SOAP server URL and SOAP callback URL.

1. Log in to Oracle BPEL Admin Console.

   ```
   http://localhost:port/BPELAdmin
   ```

2. Enter the `oc4jadmin` username and password.
3. Set the following two parameters under the **Configuration** tab:

Parameter	Description	Example
soapServerUrl	The BPEL SOAP server endpoint URL of a process	`http://`*`hostname`*`:`*`port`*
soapCallbackUrl	The BPEL SOAP callback URL of a process	`http://`*`hostname`*`:`*`port`*

4. Delete the default `.bpel_TaskManager_1.0.jar` and `.bpel_TaskActionHandler_1.0.jar` directories under *`SOA_Oracle_Home`*`\bpel\domains\`*`domain_name`*`\tmp`.

   where *`domain_name`* is the name of the domain in which the BPEL process to secure is located.

5. Restart Oracle BPEL Server.

   This recreates the correct service bindings and WSDL files for the TaskManager and TaskActionHandler processes and makes them available from HTTP/S-based endpoints. Processes deployed into Oracle BPEL Process Manager are now securely hosted at the new HTTP/S endpoint.

Oracle BPEL Process Manager for Developers

SSL configuration for Oracle BPEL Process Manager for Developers is a two-step process:

- Step 1: Configuring OC4J
- Step 2: Configuring Oracle BPEL Server

Step 1: Configuring OC4J

1. See the *Oracle Containers for J2EE Security Guide* for the following SSL configuration instructions:

   - Using `keytool` to enable certificate-based authentication. This tool generates a keystore and a self-signed certificate. A keystore is a protected database that holds keys and certificates for an enterprise. Access to a keystore is guarded by a password. The password is defined at the time the keystore is created by the user who creates the keystore, and is changeable only when providing the current password.
   - Configuring OC4J to use SSL. When complete, OC4J listens for SSL requests on one port and non-SSL requests on another.

   Notes:

   - Ensure that you shut down and restart OC4J after configuring SSL. This is accomplished by shutting down and restarting Oracle BPEL Server.
   - Instead of a self–signed certificate for production environments, use a certificate from a trusted certificate authority like Verisign/Thawte by submitting a certificate request generated by `keytool`.

Step 2: Configuring Oracle BPEL Server The steps to configure Oracle BPEL Server for the Oracle BPEL Process Manager for Developers installation type are the same as with the Oracle BPEL Process Manager for OracleAS Middle Tier installation type.

See "Step 2: Configuring Oracle BPEL Server" on page 1-7 for instructions on configuring Oracle BPEL Server.

Using J2EE Basic Authentication

J2EE basic authentication involves authentication through unsigned tokens, namely a user name and password.

Table 1–1 describes the supported features of this method.

Table 1–1 J2EE Basic Authentication Supported Features

Authentication Schemas	Service Access Protocols	User Repository	Customization Permitted	Granularity
Basic authentication (user name and password)	HTTP only	Oracle Application Server JAZN repository types: ■ OID ■ JAZN XML ■ JAAS custom plug-in	JAAS custom authorization plug-in	Individual process level security

The following sections describe the configuration method to use based on the Oracle BPEL Process Manager installation type:

- Oracle BPEL Process Manager for OracleAS Middle Tier
- Oracle BPEL Process Manager for Developers

Oracle BPEL Process Manager for OracleAS Middle Tier

J2EE basic authentication with the Oracle BPEL Process Manager for OracleAS Middle Tier installation type involves delegating authentication to Oracle Application Server. (See Figure 1–2 on page 1-5.) The following steps describe this process.

1. Oracle HTTP Server receives a service request.
2. Oracle HTTP Server forwards the request to OC4J.
3. OC4J validates the user name and password received in the HTTP headers against the configured identity service user repository:
    - Oracle Internet Directory (OID)
    - Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider (JAZN) XML
    - Custom JAAS plug-in
4. If the user name and password are authenticated, the request is sent to Oracle BPEL Server for servicing.

> **See Also:** *Oracle Containers for J2EE Security Guide* for configuration instructions

Oracle BPEL Process Manager for Developers

J2EE basic authentication with the Oracle BPEL Process Manager for Developers installation type involves delegating authentication to the OC4J in which Oracle BPEL Process Manager is deployed. (See Figure 1–2 on page 1-5.) The following steps describe this process.

1. Set up new users and roles in the JAZN repository:

    For example, for users in JAZN XML, configure a new user and role in `SOA_Oracle_Home\bpel\system\appserver\oc4j\j2ee\home\config\system-jazn-data.xml` as follows:

```
<user>
   <name>jsmith</name>
   <credentials>{903}9XRK6pyPRTVYN7bW5dkG1Z06C2pkBRW6</credentials>
</user>
. . .
. . .
<role>
   <name>jsmithrole</name>
   <members>
      <member>
        <type>user</type>
        <name>jsmith</name>
      </member>
   </members>
</role>
```

2. Configure the *SOA_Oracle_Home*\bpel\system\appserver\oc4j\j2ee\home\applications\orabpel_ear\META-INF\orion-application.xml file.

   Map the physical security roles maintained in OC4J (for example, JAAS principals and realms) to logical J2EE groups and users by adding the following sections:

   ```
   <security-role-mapping name="jsmithrole">
      <group name=" jsmithrole" />
   </security-role-mapping>
   ```

3. Configure the *SOA_Oracle_Home*\bpel\system\appserver\oc4j\j2ee\home\applications\orabpel_ear\startup_war\WEB-INF\web.xml file to protect the BPEL service endpoint URLs.

   A code segment from web.xml protecting an endpoint URL http://localhost/orabpel/default/HelloWorld/1.0) is as follows:

   ```
   <security-constraint>
      <web-resource-collection>
         <web-resource-name>Default Domain Pages</web-resource-name>
         <description>These pages are only accessible by authenticated
            users.</description>
         <url-pattern>*/orabpel/default/HelloWorld/1.0</url-pattern>
         <url-pattern>*/orabpel/default/HelloSecureWorld/1.0</url-pattern>
      </web-resource-collection>
      <auth-constraint>
         <role-name>jsmithrole</role-name>
      </auth-constraint>
   </security-constraint>
   <login-config>
      <auth-method>BASIC</auth-method>
      <realm-name>jazn.com</realm-name>
   </login-config>
   <security-role>
      <description>BPEL PM User</description>
      <role-name>jsmithrole</role-name>
   </security-role>
   ```

Using the Native BPEL Security Extensions

Native BPEL security extensions code can also handle authentication. (See Figure 1–2 on page 1-5.) The following steps describe this process.

1. Oracle HTTP Server receives a service request.
2. Oracle HTTP Server forwards the request, part of which is intercepted by Oracle BPEL Process Manager.
3. The BPEL security extension code of Oracle BPEL Process Manager validates the message received against the configured identity service user repository:
   - OID
   - JAZN XML
   - Custom JAAS plug-in
4. If the user name and password are authenticated, Oracle BPEL Server services the request.

Table 1–2 describes the supported features of this method.

Table 1–2 Native BPEL Security Extensions Supported Features

Authentication Schemas	Service Access Protocols	User Repository	Customization Permitted	Granularity
Basic authentication (user name and password)	HTTP	Oracle Application Server JAZN repository types: ■ OID ■ JAZN XML ■ Database-based repository ■ Custom	Custom user repository using the custom validator class **See Also:** "Creating a Custom Validator" on page 1-26	Fine grained: ■ Individual process level security (for example, `service1` with `username1/password1` and `service2` with `username2/password2`) ■ Supports domain level protection and all services in that domain with one user name and password
Normalized message properties	■ Java API ■ Remote method invocation (RMI)			
WS-Security (in accordance with the *WS-Security Web Services Security Specification*)	SOAP			

This section contains the following topics:

- Domain and Process Level Security
- Java API
- HTTP Binding
- SOAP over HTTP Binding

Domain and Process Level Security

Within Oracle BPEL Server, a message handler framework is used to control and modify inbound (calls to Oracle BPEL Server) and outbound (calls from Oracle BPEL Server) message flows. One of these plug-in handlers is the security interceptor. The security interceptor provides two levels of security:

- Domain level security

  If only this level is set, enables you to secure *all* processes running in a specific domain.

- Process level security

  If this level is also set, enables you to specify which processes to secure, and which not to secure, in a specific domain.

> **Note:** The following section only explains the configuration of the security interceptor, and not the framework itself.

By default, domain and process security is not enabled. However, both security levels can be easily enabled by modifying the *SOA_Oracle_Home*`\bpel\domains\`*domain_name*`\config\message-handlers.xml` file.

1. If you want to enable domain level security, remove the comment markers shown in bold from around the `security` attribute (for this example, the domain is named `default`):

   ```
   <inbound-flow>
      <message-handler id="default" />

   <!-- uncomment for inbound security
       <message-handler id="security" />
   -->
   </inbound-flow>
   ```

   This enables the security chain:

   ```
   <message-handler id="security">
           <classname>com.collaxa.cube.security.Authenticator</classname>
           <comment>
              <![CDATA[This is the handler for security interception]]>
              </comment>
           <property id="ACLManager">
               <value>com.oracle.bpel.security.validator.bpmid.
                 BPMIdentityValidator</value>
           <comment>BPMIdentityValidator uses the server configured security
                    such as JAAS to validate the user against</comment>
           </property>
   <!--
      <property id="SecuredProcesses">
         <value>CreditRatingService</value>
            <comment>Processes can be secured explicitely without having effect
                on the whole domain, put their names in here and comma
                separate them
            </comment>
      </property>
   -->
   </message-handler>
   ```

2. If you also want to enable security at the process level, remove the comment markers shown in bold from around the `SecuredProcesses` attribute in the same file. The section contains a `value` element that consists of a comma-separated list of process names:

   ```
   <!--
      <property id="SecuredProcesses">
         <value>CreditRatingService</value>
            <comment>Processes can be secured explicitely without having
               effect on the whole domain, put their names in here and comma
               separate them</comment>
      </property>
   -->
   </message-handler>
   ```

3. Specify the processes to secure in the `value` element of the `SecuredProcesses` section. For example:

   ```
   <value>CreditRatingService, HelloWorldService</value>
   ```

Any other processes in this domain that are *not* specified in the `value` element are *not* secured.

4. Restart Oracle BPEL Server.

   This enables the default validator bridge to be used for authentication and authorization.

> **See Also:** "Using the Default Validator" on page 1-25 for information about the validator bridge

Java API

For invocation of a process, use the `DeliveryService`. However, the normalized message (`com.oracle.bpel.client.NormalizedMessage`) needs the following properties (through `NormalizedMessage:setProperty(key, value)`) added:

```
secured = username
username = password
```

where *`username`* equals the user name that is sent, and the second pair consists of the *`username`* and the desired credential. For example:

```
secured = Clemens
Clemens = pwForClemens
```

> **Note:** You can also send an empty password; in this case, add only the first pair:
>
> ```
> secured = Clemens
> ```

HTTP Binding

When you use direct HTTP binding to invoke a process, there are multiple ways of specifying the credentials:

- As HTTP request parameters:

  ```
  <input type="hidden" name="bpelUser" value="clemens">
  <input type="hidden" name="bpelCredential" value="clemens">
  ```

- As basic authentication HTTP headers (base64-encoded):

  ```
  Authentication=BASIC <BASE64-HASH>
  ```

- As normal name-value HTTP header pairs, where the key for the user is `bpelUser` and the key for the password is `bpelCredential`

SOAP over HTTP Binding

When using SOAP binding, the only currently supported method for passing a user name credential is as a WS-Security compliant SOAP header. For example:

```
<wsse:Security soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
               soapenv:mustUnderstand="1"
               xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
                  wss-wssecurity-secext-1.0.xsd"
               xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
                  <wsse:UsernameToken
               xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
```

```
                  wss-wssecurity-utility-1.0.xsd"><wsse:Username>Clemens
                     </wsse:Username><wsse:Password Type=
                        "http://docs.oasis-open.org/wss/2004/01/oasis-200401-
                         wss-username-token-profile-1.0#PasswordText">pwForClemens
                        </wsse:Password>
                    </wsse:UsernameToken>
</wsse:Security>
```

When using Java to call a service endpoint through SOAP, the class `com.oracle.bpel.client.util.WSSecurityUtils` can generate a header element of the namespace. For example:

```
  /**
   * Create a WSSecurity compliant token from username and password -
 UsernameToken!!
   * @throws javax.xml.soap.SOAPException in case the element cannot be
 constructed
   * @return the headerElement needed for the header of the call
   * @param pCredential the credential
   * @param pUsername the username
   */
  public static SOAPHeaderElement createWSSecurityHeader (String pUsername,
                                                          String pCredential)
```

Note that `createWSSecurityHeader` represents the older Java standard. Since the change to the WS-Security standard in 2004, you must apply the new namespace or else it defaults to the `http://schemas.xmlsoap.org/ws/2002/07/secext` namespace. To create a `WSSE` header element with the new namespace, use this method located in the `WSSecurityUtils` class:

```
public static SOAPHeaderElement createOASISWSSecurityHeader
       (String pUsername,
        String pCredential,
        boolean pIsWSPolicyCompliant) throws SOAPException
  {
```

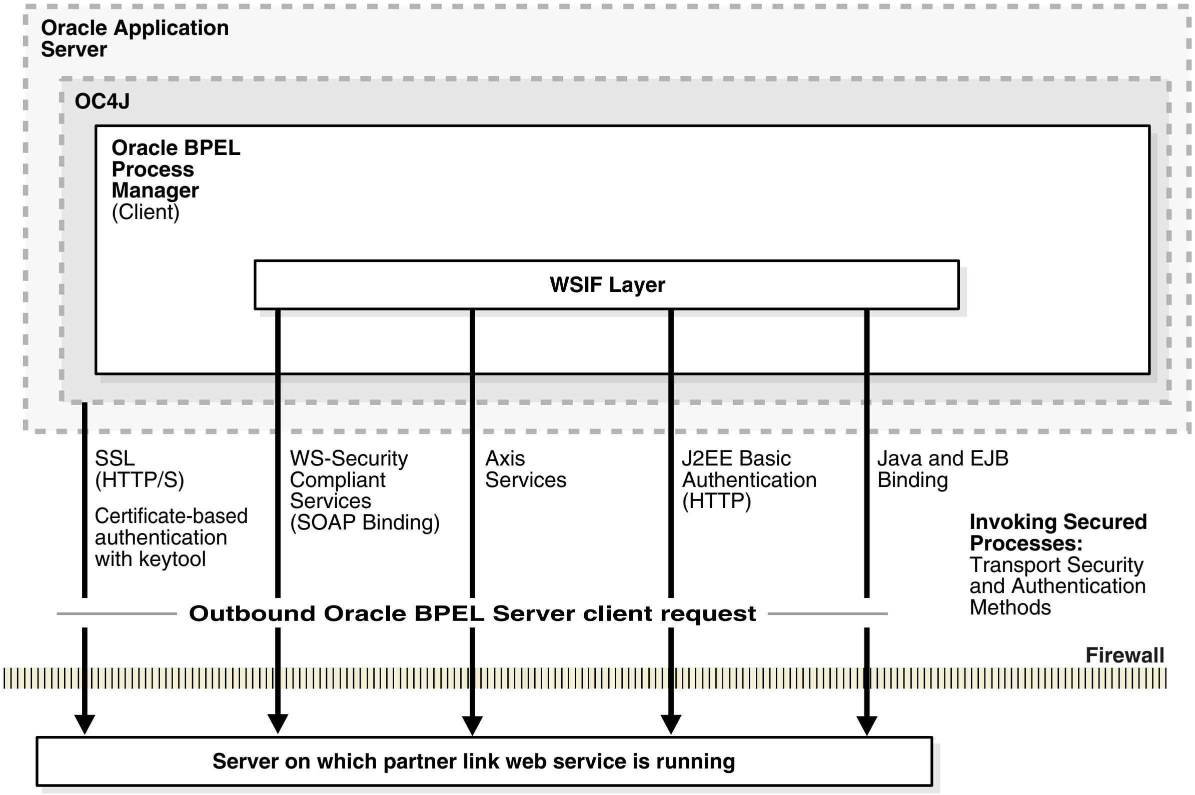
Invoking Secured Services (Outbound)

You can invoke secured services in which interaction is initiated by an outbound client request sent from Oracle BPEL Server to the server on which the partner link Web service is running. The configuration procedures for invoking secured services are the same for either Oracle BPEL Process Manager installation type:

- Oracle BPEL Process Manager for Developers
- Oracle BPEL Process Manager for OracleAS Middle Tier

Figure 1–3 provides an overview of the transport security and authentication methods available for invoking secured services (outbound):

Figure 1–3 Invoking Secured Services (Outbound)


Oracle Application Server
OC4J
Oracle BPEL Process Manager (Client)
WSIF Layer
SSL (HTTP/S)
Certificate-based authentication with keytool
WS-Security Compliant Services (SOAP Binding)
Axis Services
J2EE Basic Authentication (HTTP)
Java and EJB Binding
Invoking Secured Processes: Transport Security and Authentication Methods
Outbound Oracle BPEL Server client request
Firewall
Server on which partner link web service is running

This section contains the following topics:

- Using SSL for Certificate-Based Authentication
- WS-Security-Compliant Services
- Axis Services with Custom Authentication Handlers
- J2EE Basic Authentication Protected Services (HTTP)
- Java and EJB Binding (10.1.3)

Using SSL for Certificate-Based Authentication

If a partner exposes an HTTP/S-based service, the WSDL of that service contains the information in the service binding. You can invoke services from Oracle BPEL Process Manager that have a SOAP or HTTP binding.

Oracle BPEL Process Manager support for SSL is Java Secure Socket Extension (JSSE)-standards based and relies on the default SunJSSE provider for cryptographic services. For configuring the keystore and truststore, Oracle BPEL Process Manager relies on standard JSSE `keytool` and JSSE mechanisms. (See Figure 1–3 on page 1-15.)

The following types of certification authentication can be used:

- Server certificate authentication

  During the SSL handshake process, Oracle BPEL Process Manager, which acts as a client to the secured service of the partner link server, is required to verify the trustworthiness of the partner link (server authentication). Verifying the certificate presented by the partner link server satisfies this requirement. To do this, the default SunJSEE functionality is used by Oracle BPEL Process Manager and the truststore used in the process must contain the appropriate certificate entries. If the partner link server uses a self-signed certificate, this certificate must be placed as a trusted entry in the truststore. If the partner link server presents a certificate chain, then the root certificate of that chain must be part of the truststore.

- Server and client certificate authentication

  During the handshaking process, a partner link server can sometimes require that the client (in this scenario, Oracle BPEL Process Manager) present its certificate for verification. This is called client authentication mode. For these situations, you must also set up a certificate for Oracle BPEL Process Manager. The certificate can be self-signed or provided by a certificate authority. The `keytool` can be used to save that certificate and keys in the keystore and truststore.

  Note that it is not possible to know from the WSDL of the service if the partner link service requires this. This requirement is not in wide practice.

It is beneficial to set up a truststore in which trusted certificate entries are placed. This is different from the keystore, in which private and public key entries are present.

The default keystore and truststore files located in the `jre\lib\security` directory for your JDK installation are used:

- The `cacerts` file is the default keystore
- The `jssecacerts` file (if present) is the truststore file. If `jssecacerts` is not present, `cacerts` also serves as the truststore.

Keystore and truststore files are created and managed with JDK's `keytool`. This tool is useful for performing operations such as the following:

- Creating new keystores and truststores
- Reading and listing information present in the stores
- Updating and deleting existing entries in keystores and truststore

Notes:

- Do *not* use Oracle Wallet Manager to create a security certificate for communication between the client Oracle BPEL Server and the server on which the partner link Web service is running.
- No Oracle BPEL Server configuration is required when invoking secured services. This is because Oracle BPEL Server is the client in this type of interaction.

See Also:

- `http://java.sun.com/j2se/1.4.2/docs/guide/security/jsse/JSSERefGuide.html` for details about SSL, such as understanding how SSL works, creating keystores and truststores to use with JSSE, and debugging and troubleshooting issues
- `http://java.sun.com/products/jsse/` for JSSE details
- `http://java.sun.com/j2se/1.4.2/docs/tooldocs/tools.html#security` for details about using `keytool`
- *Oracle Containers for J2EE Security Guide* for details about using `keytool`

Oracle JDeveloper Design Time

To access secured WSDLs, Oracle JDeveloper must be configured at design time just like Oracle BPEL Server.

Oracle BPEL Server Runtime

This section describes how to configure HTTP/S with the partner link Web service server and the Oracle BPEL Server client side certificates.

HTTP/S with Partner Link Server Certificate Authentication Only Follow these steps to configure Oracle BPEL Process Manager for this environment:

1. Ensure that the keystore is configured appropriately to invoke the mutually-trusted certificate or the server certificate of the partner link.

   a. Connect through the Web browser to the endpoint URL of the service to invoke. After connecting to the server, a pop-up window displays the following message (if you have not already updated your browser's store with this certificate):

   ```
   Security Alert: Do you trust this certificate or not?
   ```

   b. Enter `yes` because you trust this server certificate.

   After the page is loaded on Internet Explorer, a lock displays in the status bar in the bottom right corner of your browser window.

   c. Click the lock to display a window that provides details about the certificate.

   d. Click the **Details** tab and copy the certificate to a file (for example, named `TestServiceServerCert.cer`). You can use the base64-encoded format.

   e. Use that file to import the server certificate to your default truststore. You can use `keytool` to help with this process.

   f. If the default truststore and keystore are the same, the command to import this certificate into the default `cacerts` keystore is as follows:

   ```
   SOA_Oracle_Home\jdk\bin\keytool -import -v -file TestServiceServerCert.cer
    -keypass keystore_password -keystore cacerts
   ```

   g. If you do not want to store the server certificate of the partner link server, you can ensure that a mutually-trusted root and certificate authority certificate is in your truststore or keystore.

2. Ensure that the correct keystore is used by OC4J and Oracle BPEL Process Manager:

   If your keystore is the default `cacerts` file keystore located in *`SOA_Oracle_Home`*`\jdk\jre\lib\security` directory, no changes are required. If not, then edit `obsetenv.bat` (or `obsetenv.sh` for UNIX installations) to include the following lines:

```
-Djavax.net.ssl.keyStore=path_to_your_certificate_store
 -Djavax.net.ssl.keyStorePassword=your_keystore_password
```

> **Note:** While you can also edit the `startorabpel.bat` file (or `startorabpel.sh` file for UNIX installations) to include these lines, Oracle recommends that you instead edit the `obsetenv.*` file for your operating system.

3. If you are using a different truststore from the default, you should enter the following:

```
-Djavax.net.ssl.trustStore=path_to_truststore
-Djavax.net.ssl.trustStorePassword=your_truststore_password
```

> **See Also:**
>
> http://java.sun.com/j2se/1.4.2/docs/tooldocs/tools.html#security for details about using keytool

HTTP/S with Partner Link Server and Oracle BPEL Server Client Certificate Authentication This section describes how to configure the Oracle BPEL Server client. The steps to configure the client to invoke partner links that require client authentication are similar to the steps to invoke partner links with only server side authentication enabled. The difference is the keystore that BPEL uses for this environment has the following certificates in the following locations:

- Its own (that is, the host OC4J server certificate in the keystore)
- The client certificate or a mutually-trusted CA certificate in the keystore and truststore

The high level steps involved are as follows:

1. Set up OC4J to use SSL, as described in "Step 1: Configuring OC4J" on page 1-8.
2. Ensure that a mutually-trusted certification authority certificate is in the truststore and keystore.
3. Ensure that the correct keystore and truststore are used by OC4J and BPEL, as described in Step 2 of "HTTP/S with Partner Link Server Certificate Authentication Only" on page 1-17.

WS-Security-Compliant Services

If a partner link expects WS-Security-compliant authentication tokens, BPEL can be configured to invoke the partner link with these. (See Figure 1–3 on page 1-15.) Table 1–3 shows the relevant properties. These properties are configurable at the individual partner link level.

Table 1–3 Properties

Property Name	Description	On Change
`wsseHeaders`	Creates a WS-Security `UsernameToken` with the following values: ■ `propagate` If the process has been invoked securely, these credentials are also used for the outbound direction. ■ `credentials` Passes credentials from the descriptor	Takes effect immediately
`wsseUsername`	The user name for the token (required)	Takes effect immediately
`wssePassword`	The password for the token (optional)	Takes effect immediately

See Also:

- *Oracle BPEL Process Manager Developer's Guide* for additional details about these deployment descriptor properties
- Web Services Security (WS-Security) Specifications available at the following URL:

  http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

SOAP Binding

When using SOAP binding, there are four possible cases:

- Case 1

  Propagation of the credentials over a partner link (if the process is invoked securely over any API) for WS-Security headers
- Case 2

  Propagation of the credentials over a partner link (if the process is invoked securely over any API) for basic authentication
- Case 3

  Static definition of a user name and password put into a WS-Security compliant user name token, and sent out
- Case 4

  Static definition of a user name and password that is used for `http-basic-authentication`, and sent out

Configuration By default, Oracle BPEL Server does not propagate any credentials, even if the process is invoked securely. All partner links that are used within a BPEL process are defined in `bpel.xml` (found in the BPEL suitcase).

```
<partnerLinkBindings>
  <partnerLinkBinding name="client">
    <property name="wsdlLocation">CreditRatingService.wsdl</property>
  </partnerLinkBinding>
</partnerLinkBindings>
```

For case 1, add the following property (which causes BPEL to add the `process-credentials` to the outgoing call):

```
<property name="wsseHeaders">propagate</property>
```

For case 2, add the following (attached to the SOAP call `(setUsername, setPassword)`):

```
<property name="basicHeaders">propagate</property>
```

For case 3, add the following (which builds a WS-Security Header):

```
<property name="wsseHeaders">credentials</property>
<property name="wsseUsername">your_user</property>
<property name="wssePassword">your_password</property>
```

For case 4, add the following (attached to the SOAP call `(setUsername, setPassword)`):

```
<property name="basicHeaders">credentials</property>
```

```
<property name="basicUsername">your_user</property>
<property name="basicPassword">your_password</property>
```

> **Note:** All these properties are on a per partner link basis, so they do not affect any other partner links as long as they are not specified on this specific binding.

Since the change to the WS-Security standard in 2004, you need to apply the new namespace or else it defaults to the `http://schemas.xmlsoap.org/ws/2002/07/secext` namespace. To apply the new namespace, add the following property:

```
<property name="wsseOASIS2004Compliant">true</property>
```

Axis Services with Custom Authentication Handlers

Table 1–4 shows the configurable properties at the partner link level for Axis services.

Table 1–4 Properties

Property Name	Description	On Change
`basicHeaders`	Creates a WS-Security `UsernameToken` with the following values: ■ `propagate` If the process has been invoked securely, these credentials are also used for the outbound direction. ■ `credentials` Passes credentials from the descriptor	Takes effect immediately
`basicUsername`	The user name for the token (required)	Takes effect immediately
`basicPassword`	The password for the token (optional)	Takes effect immediately

J2EE Basic Authentication Protected Services (HTTP)

This section describes HTTP basic authentication.

The section contains the following topics:

- HTTP Basic Authentication (10.1.2.0.2)
- HTTP Binding (10.1.3)

HTTP Basic Authentication (10.1.2.0.2)

Table 1–5 shows the deployment descriptor properties configurable at the partner link level. These properties can be set to authenticate services that use HTTP headers for authentication in 10.1.2.0.2.

Table 1–5 Properties

Property Name	Description	On Change
`httpUsername`	This is used for HTTP user name/password authentication	Takes effect immediately

Table 1–5 (Cont.) Properties

Property Name	Description	On Change
`httpPassword`	This is used for HTTP user name/password authentication	Takes effect immediately

HTTP Binding (10.1.3)

Starting with Oracle BPEL Process Manager release 10.1.3, all partner link properties are automatically propagated into the HTTP header. However, when outbound HTTP binding is used, credentials can be used for basic authentication, if configured:

```
<property name="httpBasicHeaders">credentials</property>

<property name="httpBasicUsername">your_username</property>
<property name="httpBasicPassword">your_password</property>
```

Or they can simply be propagated from the process instance:

```
<property name="httpBasicHeaders">propagate</property>
```

Java and EJB Binding (10.1.3)

Starting with Oracle BPEL Process Manager release 10.1.3, partner link properties can be propagated into the implementing class or EJB by implementing this interface:

```
com.oracle.bpel.client.wsif.IjavaEjbPlnkBindingInfo
```

It contains the following method:

```
/**
 * This method will be called immediately after the new instance
 * of the class/bean has been created
 *
 * @param pProperties the map containing name/value pairs
 */
public void setPlnkProperties(HashMap pProperties);
```

This method is called directly after the class or bean has been created, and the map contains all partner link properties.

Oracle BPEL Control and Oracle BPEL Admin Console Users and Roles

The Oracle Application Server oc4jadmin administrator account enables you to log into Oracle BPEL Control and Oracle BPEL Admin Console and manage BPEL processes. Beginning with this release, both consoles are fully integrated with Oracle Application Server J2EE and JAAS security features.

In addition, Oracle BPEL Process Manager automatically includes a set of users, roles, and domains for performing BPEL process management from Oracle BPEL Control and Oracle BPEL Admin Console. Table 1–6 describes these features:

Table 1–6 Oracle BPEL Process Manager Roles, Users, and Domains

Users	Roles	Domains
■ bpeladmin User account with the BPMSystemAdmin role and a default password of welcome1.	■ BPMSystemAdmin Provides access to all domains accessible through Oracle BPEL Control and Oracle BPEL Admin Console.	default — Enables you to partition and manage instances of your processes. You can create additional domains, as necessary.
■ default User account with the BPMDefaultDomainAdmin role and a default password of welcome1.	■ BPMDefaultDomainAdmin Provides access to *only* the default domain accessible through Oracle BPEL Control. This role does *not* provide access to Oracle BPEL Admin Console.	

The Oracle Application Server oc4jadmin administrator account also includes the BPMSystemAdmin role.

Passwords for the oc4jadmin, bpeladmin, and default users can be changed through Oracle Enterprise Manager 10*g* Application Server Control Console. Oracle recommends that you change the passwords for the bpeladmin and default users after installation.

Use the oc4jadmin user account when creating application server connections in the **Connection Navigator** of Oracle JDeveloper. Other user accounts, such as bpeladmin, default, or any new user accounts you created, do not have the RMI permissions and cannot be used when creating application server connections in Oracle JDeveloper.

You can create new users and groups and assign them to new domains or the default domain automatically included with Oracle BPEL Process Manager.

This section provides the following examples:

- Example 1: Creating New Users and Groups to Access New BPEL Domains
- Example 2: Creating a New User to Access the Default BPEL Domain
- Example 3: Creating a New User to Access All BPEL Domains

Notes: These examples use a:/home/oc4j/bpel/lib as the directory location for orabpel-boot.jar. Substitute this path with the one appropriate to your environment.

See Also:

- Appendix A, "Demo User Community" for additional details about the BPMSystemAdmin and BPMDefaultDomainAdmin roles
- *Oracle BPEL Process Manager Developer's Guide* for additional details about the BPMSystemAdmin and BPMDefaultDomainAdmin roles and domain management
- *Oracle Application Server Administrator's Guide* for instructions on changing the oc4jadmin, bpeladmin, and default passwords
- *Oracle Containers for J2EE Security Guide* for information about Java SSO (JSSO), OracleAS JAAS Provider Admintool examples, and additional security management tools available for file-based providers and Oracle identity management providers

Example 1: Creating New Users and Groups to Access New BPEL Domains

This section describes how to create a new user and group to access a new BPEL domain. In this example, you use Oracle Internet Directory to create the user and group and the OracleAS JAAS Provider Admintool of the XML-based JAZN provider to grant the necessary domain permissions to the new user and group. The management tool to use to create the user and group is based on the type of identity service provider you are using.

1. Configure the 10.1.3.1.0 identity service with 10.1.2 Oracle Internet Directory as described in "Configuring Identity Service 10.1.3.1.0 with 10.1.2 Oracle Internet Directory" on page 2-7.
2. Create a new domain in the Oracle BPEL Admin Console named soaAdmin.
3. Create a new user named soaAdmin and group named BPMsoaAdminDomainAdmin in Oracle Internet Directory.
4. Use the JAZN Admin tool to grant domain permissions to user soaAdmin or group BPMsoaAdminDomainAdmin:

```
java -Xbootclasspath/a:/home/oc4j/bpel/lib/orabpel-boot.jar -jar jazn.jar
-shell -grantperm jazn.com -user soaAdmin com.collaxa.security.DomainPermission
soaAdmin all
```

or

```
java -Xbootclasspath/a:/home/oc4j/bpel/lib/orabpel-boot.jar -jar jazn.jar
-shell -grantperm jazn.com -role BPMsoaAdminDomainAdmin
com.collaxa.security.DomainPermission soaAdmin all
```

where:

- com.collaxa.security.DomainPermission — Is the name of the permission class. This permission class does *not* provide access to Oracle BPEL Admin Console.
- soaAdmin — Is a parameter to the permission class. This parameter indicates the name of the domain to which the user has access.
- all — Is a parameter to the permission class. This parameter indicates the level of actions the user or group can perform.

> **Note:** The user `soaAdmin` or group `BPMsoaAdminDomainAdmin` receives either *all* or *no* privileges in the `soaAdmin` domain. You *cannot* assign specific actions to a user or group, such as specifying read-only permissions, update permissions, and so on.

5. Log into Oracle BPEL Control.

Example 2: Creating a New User to Access the Default BPEL Domain

This section describes how to create a new user to access the default BPEL domain automatically included with Oracle BPEL Process Manager. In this example, you use the OracleAS JAAS Provider Admintool to create a user and grant the `BPMDefaultDomainAdmin` role to the user. The management tool to use to create the user is based on the type of identity service provider you are using.

1. Create a user named `mike` in realm `jazn.com` with a password of `welcome`.

```
% java -Xbootclasspath/a:/home/oc4j/bpel/lib/orabpel-boot.jar -jar jazn.jar
 -shell -adduser jazn.com mike welcome
```

2. Grant the role `BPMDefaultDomainAdmin` to user `mike` in realm `jazn.com`.

```
% java -Xbootclasspath/a:/home/oc4j/bpel/lib/orabpel-boot.jar -jar jazn.jar
 -shell -grantrole BPMDefaultDomainAdmin jazn.com mike
```

3. Log into Oracle BPEL Control.

Example 3: Creating a New User to Access All BPEL Domains

This section describes how to create a new user to access all BPEL domains. In this example, you use the OracleAS JAAS Provider Admintool to create a user and grant the `BPMSystemAdmin` role to the user. The management tool to use to create the user is based on the type of identity service provider you are using. This user also receives the `com.collaxa.security.ServerPermission` permission. This permission provides access to all domains and to Oracle BPEL Admin Console. This permission also automatically includes the default (`com.collaxa.security.DomainPermission`) permission.

1. Create a user named `mike` in realm `jazn.com` with a password of `welcome`.

```
% java -Xbootclasspath/a:/home/oc4j/bpel/lib/orabpel-boot.jar -jar jazn.jar
 -shell -adduser jazn.com mike welcome
```

2. Grant the role `BPMSystemAdmin` to user `mike` in realm `jazn.com`.

```
% java -Xbootclasspath/a:/home/oc4j/bpel/lib/orabpel-boot.jar -jar jazn.jar
 -shell -grantrole BPMSystemAdmin jazn.com mike
```

3. Log into Oracle BPEL Control.

Default and Custom Validators

Two types of identity store validators are available for authenticating users:

- Using the Default Validator
- Creating a Custom Validator

Using the Default Validator

Oracle BPEL Process Manager provides a bridge to your identity store through the BPEL Identity Service. For example, in Oracle Application Server you can use JAZN, Oracle Internet Directory (OID), or a custom repository plug-in as your identity store.

If you want to invoke a BPEL process, your user name must be in the configured store, or, in the case of JAZN, created in `SOA_Oracle_Home\j2ee\home\config\system-jazn-data.xml` (for the Oracle BPEL Process Manager for OracleAS Middle Tier installation type). For example:

```
<user>
   <name>Clemens</name>
   <credentials>!yourpassword</credentials>
</user>
```

BPEL security validation is evaluated in the following order:

- If the BPEL suitcase contains the credentials within the `configurations` tag in `bpel.xml`. For example:

  ```
  <property name="user">Clemens</property>
  <property name="pw">your_password</property>
  ```

- If a role is specified in the BPEL suitcase, the user specified in the request must exist in the identity management store and must belong to that group.

  ```
  <property name="role">administrators</property>
  ```

  This method is useful when many processes are used and identity management cannot be reconfigured with a new role for each process.

- If neither of the security validators described above are found, BPEL concatenates the process name and `ExecutionRole` and expects the supplied user to belong to a role of that name. For example, if user `Clemens` invokes the `CreditRatingService` process, he must belong to a group named `CreditRatingServiceExecutionRole` as defined in your identity store (for example, `system-jazn-data.xml` if you are using JAZN):

  ```
  <role>
     <name>CreditRatingServiceExecutionRole</name>
        <members>
           <member>
              <type>user</type>
              <name>Clemens</name>
           </member>
         </members>
  </role>
  ```

See Also:

- "Configuring the Identity Service" on page 2-1
- *Oracle BPEL Process Manager Developer's Guide* for additional details about BPEL identity services

Creating a Custom Validator

It is sometimes necessary to implement a custom validator when the default does not meet your requirements. To accomplish this, the following interface must be implemented and the message handler reconfigured.

```
/**
 * This source is proprietary to ORACLE CORPORATION
 * 2005, All rights reserved
 */
package com.oracle.bpel.security;

import com.oracle.bpel.client.ServerException;

import com.oracle.bpel.client.NormalizedMessage;
import com.oracle.bpel.client.BPELProcessId;

/**
 * Public abstract class that has to be implemented
 * for having a valid ACLManager that is used by the BPEL server
 * for authentication & authorization
 *
 * @version 1.1
 */
public abstract class ACLManager extends BaseACLManager {

  /**
   * Public constructor that should use a cache for connections
   * and care about other stuff.
   * @throws com.oracle.bpel.client.ServerException
   * @since 1.0
   */
  public ACLManager() throws ServerException
  {
  }

  /**
   * Checks if a user is valid in the context of a secured Process
   *
   * @return valid or not
   * @param pMessage the message will hold all information, including
   * the domain information and headers
   * @throws ServerException in case something breaks
   */
  public abstract boolean validateUser
     (BPELProcessId pProcessID, NormalizedMessage pMessage) throws
        ServerException;

  /**
   * Checks if a user is allowed to execute (=invoke) a certain revision
   * (if given) of a process.
   *
   * @return true if he is otherwise false
   * @param pProcessId the name, domain and revision of the process
   * @param pMessage the message will hold all information, including
   * the domain information and headers
   * @throws ServerException in case something breaks
   */
  public abstract boolean isAllowedToExecuteProcess
   (BPELProcessId pProcessID, NormalizedMessage pMessage)
      throws ServerException;
  /**
   * Checks if a user is allowed to execute (=invoke) a certain activity
   * of a process.
   *
   * @return true if he is otherwise false
   * @param pProcessId the name, domain and revision of the process
   * @param pActivityName the name of the Activity
   * @param pMessage the message will hold all information, including
   * the domain information and headers
   * @throws ServerException in case something breaks
   */
  public abstract boolean isAllowedToExecuteActivity
     (BPELProcessId pProcessID, NormalizedMessage pMessage, String  pActivityName)
         throws ServerException;

  /**
   * Checks if a user is allowed to lookup  a certain revision
   * (if given) of a process.
   *
   * @return true if he is otherwise false
   * @param pMessage the message will hold all information, including
   * the domain information and headers
   * @param pProcessId the name, domain and revision of the process
   * @throws ServerException in case something breaks
   */
  public abstract boolean isAllowedToLookupProcess
     (BPELProcessId pProcessID, NormalizedMessage pMessage)
         throws ServerException;

  /**
   * Checks if a user is allowed to lookup a certain activity of a process.
   *
   * @return true if he is otherwise false
   * @param pActivityName the name of the Activity
   * @param pProcessId the name, domain and revision of the process
   * @throws ServerException in case something breaks
   */
  public abstract boolean isAllowedToLookupActivity
     (BPELProcessId pProcessID, NormalizedMessage pMessage, String  pActivityName)
         throws ServerException;

}
```

After implementation, the class must reside in `SOA_Oracle_Home\bpel\system\classes` to be reached by the class loader. The second step is to reconfigure the following property in `message-handlers.xml`:

```
<property id="ACLManager">
<value>com.oracle.bpel.security.validator.bpmid.BPMIdentityValidator</value>
   <comment>BPMIdentityValidator uses the server configured security such
      as JAAS to validate the user against
   </comment>
</property>
```

where `value` must point to the classname (including the package) of the implemented validator class.

Invoking a Partner Web Service through a Proxy Server

You can configure Oracle BPEL Process Manager to invoke a partner Web service through a proxy server.

For example, assume Oracle BPEL Process Manager is installed on a host named `internal123.company.com` and one of your deployed BPEL processes must invoke a synchronous Web service hosted outside the fire wall at `http://services.myPartner.com`. All the outbound HTTP traffic must be routed through an HTTP proxy server located at `myproxy004.company.com` on port `8090`.

Perform the following steps to configure `ant` tasks and Oracle BPEL Process Manager to invoke the partner Web service through an HTTP proxy server.

1. Open the *`SOA_Oracle_Home`*`\bpel\bin\obsetenv.bat` file on Windows or SOA_Oracle_Home`/bpel/bin/obsetenv.sh` file on Unix or Linux.
2. Modify the line `set OB_JAVA_PROPERTIES=` as follows:

   ```
   set OB_JAVA_PROPERTIES="-Dhttp.proxySet=true"
   "-Dhttp.proxyHost=myproxy004.company.com"
   "-Dhttp.proxyPort=8090" "-Dhttp.nonProxyHosts=internal123.company.com"
   ```

   By setting `http.proxySet` to `true`, you activate the client proxy and redirect all the outbound traffic through `http.proxyHost` and `http.proxyPort`. By setting the `http.nonProxyHosts` to the server that hosts Oracle BPEL Server, you prevent the local request from going through the proxy. You may want to expand the `nonProxyHosts` list to include other servers inside your corporate network or other logical names for the `internal123` host by using | as a delimiter.

Using Oracle Web Services Manager for Authorization, Message Encryption, and Digital Signatures

There are several security features for which Oracle BPEL Process Manager does not currently provide native support. For those features, Oracle Web Services Manager can be used. Oracle Web Services Manager provides sophisticated authentication capabilities. Oracle Web Services Manager supports authentication using HTTP basic authentication, COREid, Netegrity, LDAP, and X.509 Certificates, and WS-Security.

This section contains the following topics:

- Authorization
- Message Encryption and Decryption
- Digital Signatures

See Also:

- *Oracle Web Services Manager Installation Guide*
- *Oracle Web Services Manager Deployment Guide*
- *Oracle Web Services Manager User and Administrator Guide*
- *Oracle Web Services Manager Upgrade Guide*
- *Oracle Web Services Manager Extensibility Guide*

Authorization

Outbound authorization in the context of BPEL invoking a service is within the responsibility of the service provider and its implementation of authorization. While Oracle BPEL Process Manager has no current native support for inbound authorization, Oracle Web Services Manager provides the following capabilities to let authorized users access BPEL processes:

- Supports authorization based on the information contained in any part of the XML message or body
- Provides the following fine-grained access control:
  - Access control at the service level
  - Access control at the SOAP method level
- Supports WS-Security

Message Encryption and Decryption

This section describes the actual message encryption. XML encryption is covered by the WS-Security profile. While Oracle BPEL Process Manager has no current native support for XML encryption, Oracle Web Services Manager provides the following encryption and decryption features:

- WS-security compliant message and content encryption and decryption
- Full or partial message encryption, enabling you to specify an XPath expression to the part of the message that requires encryption.

Digital Signatures

While Oracle BPEL Process Manager has no current native support for digital signatures, Oracle Web Services Manager provides digital signatures and signature verification capabilities. When a client invokes a service, Oracle Web Services Manager performs the following tasks:

- Intercepts this request
- Checks if the service has a digital signature verification policy to be honored
- Verifies the signature
- Passes this request to BPEL to be serviced

Similarly, when BPEL invokes a partner link, Oracle Web Services Manager attaches a digital signature to the SOAP header of the message.

Summary

This chapter describes how to perform the following procedures:

- Secure a BPEL process in which interaction is initiated by an inbound client service request sent to Oracle BPEL Server. The following security methods are described: SSL authentication, J2EE basic authentication, and native BPEL security extension authentication.
- Invoke secured services in which interaction is initiated by an outbound client request sent from Oracle BPEL Server to the server on which the partner link Web service is running. The following security methods are described: SSL authentication, WS-Security-compliant services, HTTP basic authentication

protected services, Axis services with custom authentication handlers, and native BPEL security extensions.

This chapter also provides details about the default and custom identity store providers available with Oracle BPEL Process Manager. An overview of Oracle Web Service Manager is also provided. Oracle Web Service Manager can be used to provide authorization, message encryption and decryption, and digital signature support with Oracle BPEL Process Manager.

2 Service Configuration

This chapter describes configuration procedures for Oracle BPEL Process Manager services.

This chapter contains the following topics:

- Configuring the Identity Service
- Configuring the Notification Services
- Configuring the Workflow Service
- Integrating Oracle BPEL Process Manager with the Oracle Application Server Service Registry
- Summary

Configuring the Identity Service

The identity service is a thin Web service layer on top of the Oracle Application Server 10*g* security infrastructure, namely OracleAS JAAS Provider (JAZN), or any custom user repository. The identity service enables authentication and authorization of users and the lookup of user properties, roles, group memberships, and privileges.

The following sections describe how to configure the identity service.

- Structure of the Identity Service Configuration File
- Configuration for the XML-Based JAZN Provider
- Configuring Identity Service 10.1.3.1.0 with 10.1.2 Oracle Internet Directory
- Configuration for a Third-Party LDAP Server
- Configuration for Custom Identity Repository Plug-ins
- Setting Up Group Ownership

> **See Also:** *Oracle BPEL Process Manager Developer's Guide* for details about creating realms, users, and groups; supported identity service providers; and user and group properties

Structure of the Identity Service Configuration File

The identity service configuration is defined in the `is_config.xml` file. The file must be located in a directory that is included in the `CLASSPATH` of Oracle BPEL Process Manager. By default, it is stored in

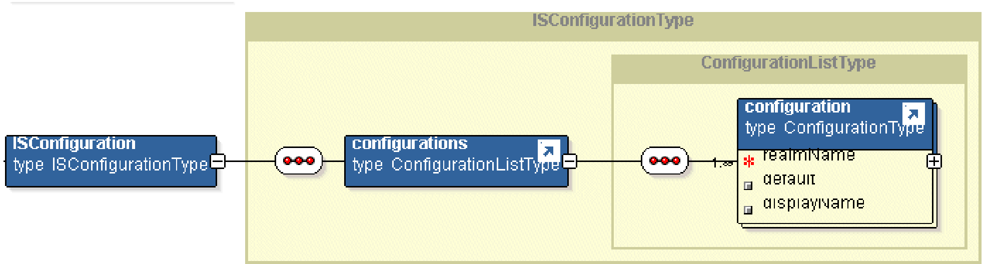
```
SOA_Oracle_Home\bpel\system\services\config
```

The XML schema for the `is_config.xml` file is stored in:

```
SOA_Oracle_Home\bpel\system\services\schema\is_config.xsd
```

Figure 2–1 shows the root element configuration, which can contain many configurations.

Figure 2–1 ISConfiguration Root Configuration



The identity service configuration file (as defined by `is_config.xsd`) consists of a root element `ISConfiguration`, which can have many configurations. Each configuration must be named in the `realmName` attribute. If several configurations are defined in `is_config.xml`, one configuration must be marked as default.

The identity service supports the following plug-in types: JAZN provider, third-party LDAP directories, or custom repository plug-ins. Each type defines providers used by the identity service.

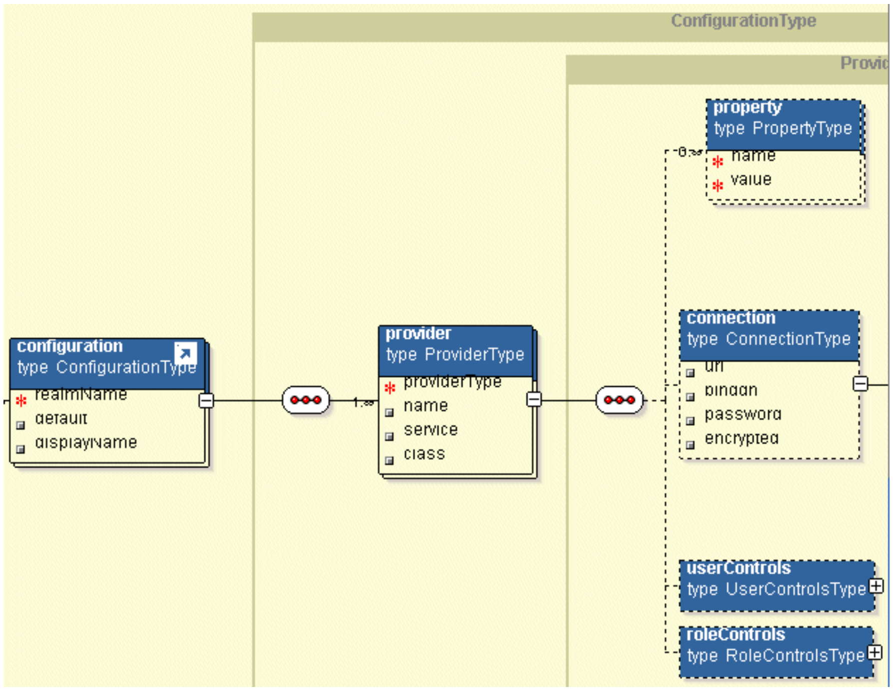
You can specify more then one provider for each configuration. A provider is always associated with one of the possible services (identity, authorization, or authentication). By default, the provider uses the identity service, unless otherwise specified in the service attribute. At least one provider associated with the identity service must be defined in the configuration.

provider Element

The provider element specifies the `providerType`, which can be `JAZN`, `LDAP`, or `CUSTOM`, provider name (optional), and any provider-specific properties.

Figure 2–2 shows the provider element configuration.

Figure 2–2 provider Element



Note: If the `providerType` is JAZN, the value for configuration attribute `realmName` must match the existing JAZN realm defined in `jazn.xml`.

For example, in the case of the JAZN XML provider, you must set the `providerType` attribute to `JAZN` and specify the value of the `userPropertiesFile` attribute. See "Configuration for the XML-Based JAZN Provider" on page 2-6 for more information about `userPropertiesFile`.

Similarly, if you use a custom plug-in to the identity service, you must set the `providerType` attribute to `CUSTOM`. You then specify the class name for custom identity service plug-in implementation, as follows:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<ISConfiguration xmlns="http://www.oracle.com/pcbpel/identityservice/isconfig">
   <configurations>
      <configuration realmName="jazn.com">
         <provider providerType="JAZN" name="xml" service="Identity">
            <property name="userPropertiesFile" value="users-properties.xml"/>
         </provider>
         <provider providerType="CUSTOM"
                   name="CustomPlugIn" service="Authentication"
                   class="package.name.CustomAuthenticationService" />
      </configuration>
   </configurations>
</ISConfiguration>
```

Multiple Service Providers The identity service supports multiple service parameters. In the code example above, one configuration is defined with two service providers. The configuration has one `JAZN` provider associated with the default identity service and another `CUSTOM` provider used for the authentication service. Therefore, the custom provider is used for user authentication calls while the JAZN XML provider is used for all authorization and identity service inquires.

> **See Also:** The "Oracle BPEL Process Manager Workflow Services" chapter of *Oracle BPEL Process Manager Developer's Guide* for additional details about the authentication, authorization, and identity service providers

Optional Parameters The provider can also define the following optional parameters in the configuration file. Most of these parameters apply to JAZN-based or LDAP-based providers, but can also be used by custom providers.

The `provider` element enables you to specify additional `property` elements, which can be used by custom plug-ins. An example follows:

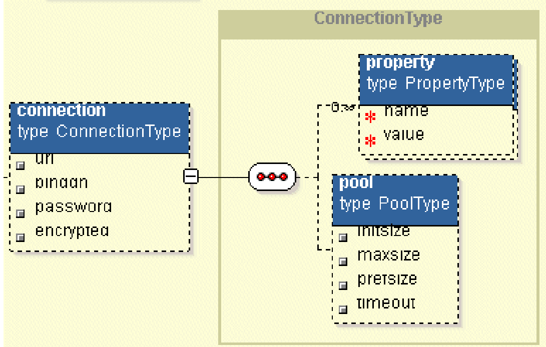
```
<?xml version = '1.0' encoding = 'UTF-8'?>
<ISConfiguration xmlns="http://www.oracle.com/pcbpel/identityservice/isconfig">
   <configurations>
      <configuration realmName="jazn.com">
          <provider providerType="CUSTOM"
                   name="CustomPlugIn" service="Identity"
                   class="package.name.CustomIdentityService">
             <property name="customProperty" value="customValue" />
      </configuration>
   </configurations>
</ISConfiguration>
```

In addition, the property element can be defined as part of any other elements (`userControls`, `searchControls`, `search`, and so on) in the configuration file.

connection Element

The `connection` element is used to specify the URL, the admin username (`binddn-` *`bind as this Distinguished name`*), the credential (`password)` for the LDAP or RDBMS connection used by the identity service, and a Boolean flag (`encrypted`) to specify that the password is either in plain text or is encrypted. The identity service overwrites the `is_config.xml` file after reading the configuration and encrypts the user password if it finds it in plain text. Figure 2–3 shows the structure of the connection configuration.

Figure 2–3 connection Configuration



The connection can specify connection pool properties by setting the following attributes on the pool element:

- `initsize`—initial size of the connection pool

- maxsize—maximum size of the connection pool
- prefsize— preferred size of the pool
- timeout—time after which the connection is released if there is no activity (in seconds)

The LDAP plug-in for the identity service uses the following default values:

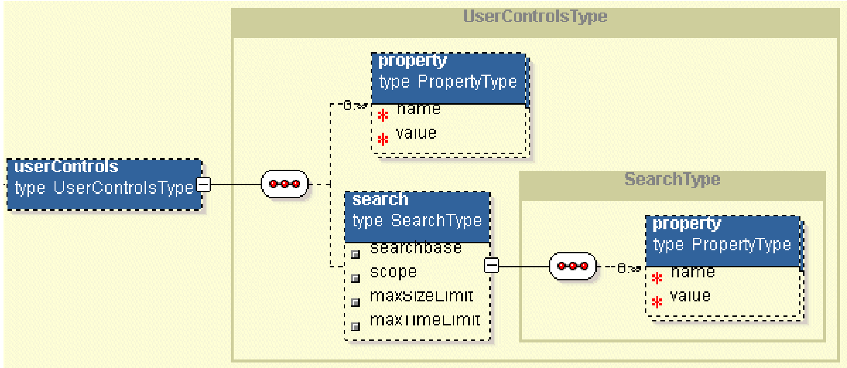
- initsize="2"
- maxsize="25"
- prefsize="10"
- timeout="60"

If you are using a custom identity service plug-in, you can also specify any additional connection-specific properties as name-value pairs.

userControls and roleControls Elements

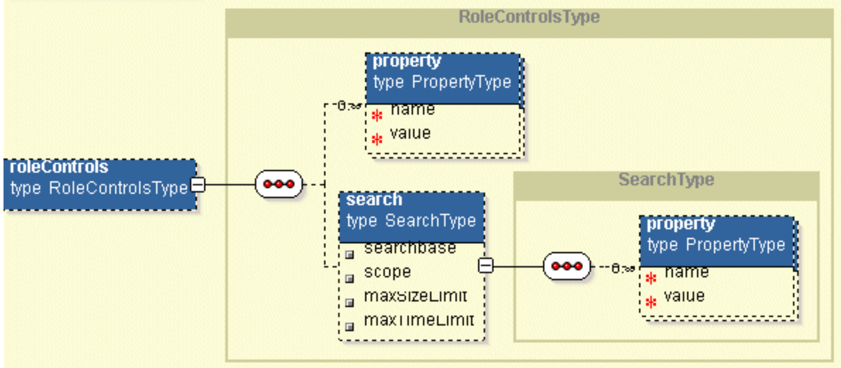
The userControls element is used to define user controls and to restrict the LDAP user search. Figure 2–4 shows the structure of the userControls element.

Figure 2–4 userControls Element



The roleControls element is used to define role controls and restrict the LDAP role search. Figure 2–5 shows the structure of the roleControls element.

Figure 2–5 roleControls Element



Both userControls and roleControls can have a search element that has the following optional attributes:

- searchbase—a list of LDAP entries, the distinguished names (DNs) of user or group containers.
- scope—determines the search level. The value can be onelevel, in which the search descends one level from the supplied DN, or subtree, in which the search descends the hierarchy from the DN to the lowest level in the tree.
- maxSizeLimit—the maximum number of elements that are fetched from LDAP during a search operation
- maxTimeLimit—the maximum time to wait to retrieve elements from an LDAP search

By default, the LDAP provider for the identity service uses the following values: maxSizeLimit ="1000",maxTimeLimit ="120" (sec), and scope="subtree".

Configuration for the XML-Based JAZN Provider

The default provider included with Oracle BPEL Process Manager is XML-based JAZN. The JAZN element is defined as follows:

```
jazn provider="XML" location="./system-jazn-data.xml"/
```

This element is in:

```
SOA_Oracle_Home\j2ee\home\config\jazn.xml
```

The identity service configuration file must specify the userPropertiesFile property and provide the value of the file name where all user properties are stored:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<ISConfiguration xmlns="http://www.oracle.com/pcbpel/identityservice/isconfig">
   <configurations>
      <configuration realmName="jazn.com">
         <provider providerType="JAZN" name="xml" service="Identity">
            <property name="userPropertiesFile" value="users-properties.xml"/>
         </provider>
   </configurations>
</ISConfiguration>
```

Note that the users-properties.xml file stores all extended user's properties. This is not required for JAZN authorization or authentication. However, the BPEL identity service requires this file to get contact details and the organizational hierarchy for users. If this file is not created, the following can occur:

- Certain workflow functionality such as notifications, manager views, or task escalation may not work.
- Workflow rule creation is disabled for all users who do not have the BPMWorkflowAdmin role.
- Workflow rule definitions for groups may not work.

By default, the identity service looks for users-properties.xml in the Oracle BPEL Process Manager classpath. Oracle Universal Installer stores the default users-properties.xml in

```
SOA_Oracle_Home\bpel\system\services\config
```

The XML schema for users-properties.xml is stored in

```
SOA_Oracle_Home\bpel\system\services\schema\IdentityService.xsd
```

Note: JAZN realms, users, and roles and groups can be created with the JAZN Admintool. To add a user to a specified realm, issue the following command:

```
java -jar jazn.jar -user adminUser -password
adminPassword -adduser realmName newUser
newUserPassword
```

See Also:

- JAZN documentation for information on how to configure the middle tier to use the XML-based JAZN provider and how to use the JAZN Admintool
- The "Oracle BPEL Process Manager Workflow Services" chapter of the *Oracle BPEL Process Manager Developer's Guide* for additional information on creating identity service users and roles

Configuring Identity Service 10.1.3.1.0 with 10.1.2 Oracle Internet Directory

This section describes how to configure the identity service of Oracle BPEL Process Manager 10.1.3.1.0 with Oracle Internet Directory 10.1.2.

- Task 1: Perform Preconfiguration Procedures
- Task 2: Perform Configuration Procedures
- Step 3: Test the Oracle Internet Directory Configuration
- Task 4: Configure the Middle Tier to use the LDAP-based JAZN provider with Secure Socket Layer (SSL)
- Troubleshooting
- Reverting from Oracle Internet Directory to the XML-Based JAZN Provider

Task 1: Perform Preconfiguration Procedures

Note: If you are using Oracle Internet Directory in an environment with multiple OC4J instances (for example, `*SOA_Oracle_Home*/j2ee/home` and `*SOA_Oracle_Home*/j2ee/oc4j_soa`), you must manually copy the relevant `<jazn>` element configuration section from the `jazn.xml` file of the `home` instance to the `jazn.xml` file of all other instances. See the section "Considering Multiple OC4J Instances when Associating Oracle Internet Directory" in the Oracle Identity Management chapter of the *Oracle Containers for J2EE Security Guide* for specific instructions.

Perform the following procedures to ensure that the Oracle Application Server instance is associated with Oracle Internet Directory.

1. Log in to the Oracle Enterprise Manager 10*g* Application Server Control Console:

   ```
   http://hostname:port/em
   ```

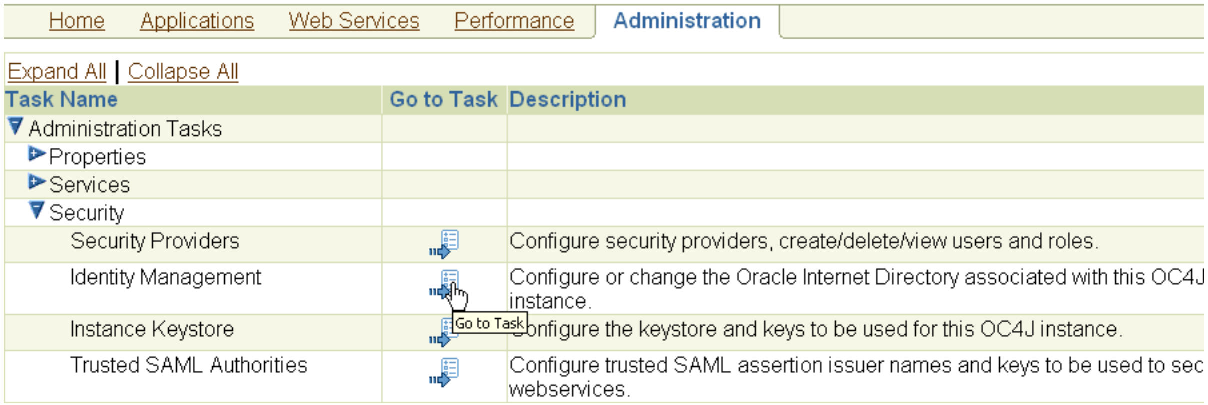
   where *`hostname`* is name of the host on which Oracle BPEL Process Manager is installed and *`port`* is the Oracle HTTP Server port.

The Cluster Topology page appears.

2. Click the OC4J instance name in the **Members** section.

   The OC4J: *oc4j_name* page appears.

3. Click the **Administration** tab.
4. Go to the **Security** section in the **Task Name** column.
5. Click the icon in the **Go to Task** column for **Identity Management**.



Task Name	Go to Task	Description
Administration Tasks		
Properties		
Services		
Security		
Security Providers		Configure security providers, create/delete/view users and roles.
Identity Management		Configure or change the Oracle Internet Directory associated with this OC4J instance.
Instance Keystore		Configure the keystore and keys to be used for this OC4J instance.
Trusted SAML Authorities		Configure trusted SAML assertion issuer names and keys to be used to sec webservices.

If Oracle Internet Directory is configured with this Oracle Application Server instance, details appear on this page.

6. See the following section based on whether Oracle Internet Directory is associated with an instance.

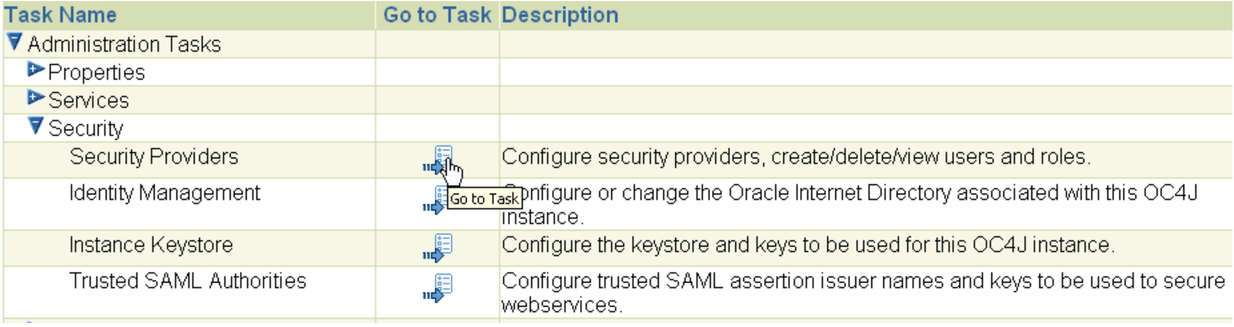
Is Oracle Internet Directory Associated with an Instance?	See...
Yes	"Oracle Internet Directory is Associated with an Oracle Application Server Instance" on page 2-8
No	"Oracle Internet Directory is Not Associated with an Oracle Application Server Instance" on page 2-9

> **See Also:** *Oracle Application Server Administrator's Guide* for details on using Oracle Enterprise Manager 10*g* Application Server Control Console

Oracle Internet Directory is Associated with an Oracle Application Server Instance

1. Review the details that appear on this page to ensure that this is the Oracle Internet Directory instance you want to use.
2. Return to the **Administration** tab of the OC4J: *oc4j_name* page you accessed in Step 3.
3. Go to the **Security** section.

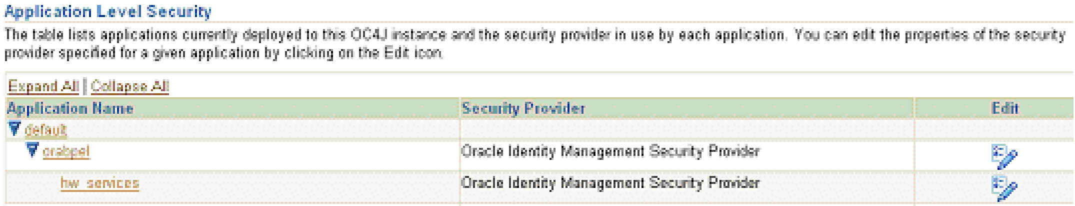
4. Click the icon in the **Go to Task** column for **Security Providers**.



Task Name	Go to Task	Description
▼ Administration Tasks		
▶ Properties		
▶ Services		
▼ Security		
Security Providers		Configure security providers, create/delete/view users and roles.
Identity Management	Go to Task	Configure or change the Oracle Internet Directory associated with this OC4J instance.
Instance Keystore		Configure the keystore and keys to be used for this OC4J instance.
Trusted SAML Authorities		Configure trusted SAML assertion issuer names and keys to be used to secure webservices.

5. Go to the **Application Name** section.

   The **orabpel** (for Oracle BPEL Process Manager) and **hw_services** (for human workflow) applications appear.



Application Level Security

The table lists applications currently deployed to this OC4J instance and the security provider in use by each application. You can edit the properties of the security provider specified for a given application by clicking on the Edit icon.

Expand All | Collapse All

Application Name	Security Provider	Edit
▼ default		
▼ orabpel	Oracle Identity Management Security Provider	✎
hw_services	Oracle Identity Management Security Provider	✎

6. Click the **Edit** column for **orabpel**.

   The Security Provider page appears.

7. Click **Change Security Provider**.

   The Change Security Provider page appears.

8. Select **Oracle Identity Management Security Provider** from the **Security Provider Type** list.

9. Click **OK**.

10. Repeat Steps 6 through 9 for **hw_services** (human workflow).

Oracle Internet Directory is Not Associated with an Oracle Application Server Instance

1. Click **Configure** to create a new association or **Change** to associate a different Oracle Internet Directory with the Oracle Application Server instance.

2. Provide appropriate responses to the questions that appear. If you want to use SSL, provide the specific SSL port number of your Oracle Internet Directory instance. If not, specify the non-SSL port of your Oracle Internet Directory instance.

3. Click **Next**.

4. If you want to associate Oracle Enterprise Manager 10*g* Application Server Control Console with Oracle Internet Directory, provide appropriate details on this page.

5. Click **Next**.

6. Click **orabpel** and **hw_services** to use Oracle Internet Directory as the security provider.

7. Click **Configure**.

Task 2: Perform Configuration Procedures

This section describes how to seed users into Oracle Internet Directory, configure the identity service, and grant privileges to BPM roles.

> **Note:** The path name delimiter used in this example, `/`, is for UNIX. If you are using Windows, assume that the path name delimiter is `\`.

1. Ensure that the `ORACLE_HOME` environment variable is set to the root directory of the Oracle Application Server instance being configured.
2. Open an operating system command prompt and go to the following directory, which includes the configuration scripts:

   ```
   SOA_Oracle_Home/bpel/system/services/install/ant-tasks
   ```

3. Execute either `configure_oid.bat` (for Windows operating systems) or `configure_oid.sh` (for Unix operating system) with the required parameters. Oracle recommends you use the bash shell to execute the script on Linux. For example, to run this script on Linux:

   ```
   sh ./configure_oid.sh oid_admin_user oid_admin_passwd
   oid_nonssl_port ssl_enabled oid_realm_name seedAllUsers | seedRequiredUsers
   oc4j_admin_user oc4j_admin_passwd oc4j_container_name
   ```

   For example:

   ```
   sh ./configure_oid.sh orcladmin welcome 389 false us seedAllUsers oc4jadmin
   welcome1 oc4j_soa
   ```

   The execution of this command internally modifies the *SOA_Oracle_Home*`/bpel/system/services/config/is_config.xml` file. The file contents look as follows:

   ```
   <?xml version = '1.0' encoding = 'UTF-8'?>
   <ISConfiguration
   xmlns="http://www.oracle.com/pcbpel/identityservice/isconfig">
           <configurations>
               <configuration realmName="us" displayName="us Realm">
                 <provider providerType="JAZN" name="OID">
                      <connection url="ldap://my.oid.com:389"
                       binddn="cn=orcladmin" password="passwd" encrypted="false"/>
                 </provider>
               </configuration>
           </configurations>
   </ISConfiguration>
   ```

   The command also modifies the *J2EE_Home*`/application-deployments/hw_services/orion-application.xml` and *J2EE_Home*`/application-deployments/orabpel/orion-application.xml` files and adds the Oracle Internet Directory details to the descriptor.

   where *J2EE_Home* is:

   - *SOA_Oracle_Home*`/j2ee/`*OC4J_Instance_Name* for Oracle Application Server SOA installations.
   - *SOA_Oracle_Home*`/bpel/system/appserver/oc4j/j2ee/`*OC4J_Instance_Name* for Oracle BPEL Process Manager for OracleAS Middle Tier installations.

The file contents look as follows:

```
<?xml version = '1.0'?>
<orion-application
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation=
        "http://xmlns.oracle.com/oracleas/schema/orion-application-10_0.xsd"
      deployment-version="10.1.3.1.0"
      default-data-source="jdbc/OracleDS"
      component-classification="internal-BPEL"
      schema-major-version="10"
      schema-minor-version="0">
   . . .
   . . .
   . . .
   <jazn provider="LDAP" jaas-mode="doAsPrivileged"/>
   . . .
   . . .
   . . .
</orion-application>
```

The `configure_oid` script also grants required privileges to the `BPMSystemAdmin` role and the `BPMDefaultDomainAdmin` role.

4. If you encounter operating system-specific problems, proceed as follows:

   a. Ensure that `OC4J_Home` points to the correct directory. This is typically *`SOA_Oracle_Home`*`/j2ee/home` for an Oracle BPEL Process Manager for OracleAS Middle Tier install and *`SOA_Oracle_Home`*`/bpel/system/appserver/oc4j/j2ee/home` for an Oracle BPEL Process Manager for Developers install.

   b. Ensure that `Ant_Home` points to the correct directory. This is typically *`SOA_Oracle_Home`*`/ant` for an Oracle BPEL Process Manager for OracleAS Middle Tier installation and *`SOA_Oracle_Home`*`/bpel/system/appserver/oc4j/ant` for an Oracle BPEL Process Manager for Developer's installation.

   c. Ensure that `Java_Home` points to the correct directory. This is typically *`SOA_Oracle_Home`*`/jdk`.

   d. Ensure that your `PATH` variable points to *`Ant_Home`*`/bin` and *`Java_Home`*`/bin`.

   e. Verify that you correctly specified all parameter values in Step 3.

   f. Execute the following command (entered on a single, wrapping line):

```
ant -f oid-config.xml -Doid.admin.user=oid_admin_user_name
-Doid.admin.pwd=oid_admin_password
-Doid.nonssl.port=non_ssl_port
-Dssl.enabled=is_ssl_enabled
-Doid.realm=oid_realm_name
-Doid.seed=seedAllUsers | seedRequiredUsers
-Doc4j.admin.user=oc4j_admin_user_name
-Doc4j.admin.pwd=oc4j_admin_password
-Doc4j.container=oc4j_container_name
```

   For example:

```
ant -f oid-config.xml -Doid.admin.user=orcladmmin
-Doid.admin.pwd=welcome1
-Doid.nonssl.port=389 -Dssl.enabled=false
```

```
-Doid.realm=us -Doid.seed=seedAllUsers
-Doc4j.admin.user=oc4jadmin -Doc4j.admin.pwd=welcome1
-Doc4j.container=oc4j_soa
```

Note: The username and realm are *not* specified as `cn=orcladmin` or `dc=us,dc=oracle,dc=com`, respectively. Instead use `orcladmin` and `us`.

The parameters you specify are defined as follows:

Parameter	Value
`oid.admin.user`	A user with administrative privileges over your Oracle Internet Directory instance. This is typically `orcladmin`.
`oid.admin.pwd`	The password of the user specified for the administrative user of Oracle Internet Directory.
`oid.nonssl.port`	The non-SSL port of the Oracle Internet Directory instance. The Oracle Internet Directory instance must be running on both the SSL and non-SSL ports.
`ssl.enabled`	Set to either of the following values: ■ `true` — Runs the identity service with SSL enabled. Note that JAZN must also be configured with SSL enabled. ■ `false` — Does not run the identity service with SSL enabled.
`oid.realm`	The realm under which you want to operate in Oracle Internet Directory.
`oid.seed`	Set either of the following values: ■ `seedAllUsers` — Seeds the demo users into Oracle Internet Directory ■ `seedRequiredUsers` — Seeds only those users required for proper functioning
`oc4j.admin.user`	A user with administrative privileges over your Oracle Application Server OC4J instance. This is typically `oc4jadmin`.
`oc4j.admin.pwd`	The password of the user specified for the administrative user of OC4J.
`oc4j.container`	The container name where the `hw_services` (for human workflow) and `orabpel` (for Oracle BPEL Process Manager) applications are deployed. To locate these names, go to `*SOA_Oracle_Home*/j2ee` and look for the following directories: ■ `*SOA_Oracle_Home*/j2ee/oc4j_home/application-deployments/orabpel` ■ `*SOA_Oracle_Home*/j2ee/oc4j_home/application-deployments/hw_services`

See Also:

- "Reverting from Oracle Internet Directory to the XML-Based JAZN Provider" on page 2-14
- Appendix A, "Demo User Community"

Step 3: Test the Oracle Internet Directory Configuration

There are multiple ways to test the Oracle Internet Directory configuration:

- Use the IdentityService servlet to look up users and roles by going to http://localhost:9700/integration/services/IdentityService/identity?operation=lookupUser.

  You can test with the bpeladmin user name to see if the user is seeded correctly.

- Go to the Oracle BPEL Worklist Application at http://localhost:9700/integration/worklistapp/Login and enter bpeladmin as the user name and welcome1 as the password to see if you can connect.

Task 4: Configure the Middle Tier to use the LDAP-based JAZN provider with Secure Socket Layer (SSL)

You must use NULL authentication when communicating with Oracle Internet Directory. NULL authentication means that data is encrypted with the Anonymous Diffie-Hellman cipher suite, but no certificates are used for authentication.

To use NULL authentication, add a <property> element to the <jazn> element in jazn.xml and to the <connection> element in is_config.xml to specify a protocol. You do not need to specify a wallet location or password, because NULL authentication does not use certificates.

Add the following property element to jazn.xml (shown in bold):

```
<jazn provider="LDAP" location="ldap://example.com:636" default-realm="us">
    <property name="ldap.user" value="cn=orcladmin"/>
    <property name="ldap.password" value="!welcome1"/>
    <property name="ldap.protocol" value="ssl"/>
</jazn>
```

See Also: *Oracle Containers for J2EE Security Guide* for additional details about JAZN configuration

Troubleshooting

If you cannot log in to Oracle BPEL Worklist Application or Oracle BPEL Control, ensure that the steps described above have been executed correctly. If the problem persists, follow these steps:

1. If you think the scripts did not complete successfully, perform the configuration steps again.
2. Go to *SOA_Oracle_Home*/j2ee/home/application-deployments/hw_services.
3. Open orion-application.xml in a text editor.
4. Verify that the jaas-mode attribute for the JAZN provider configuration is set to doAsPrivileged. For example:

   ```
   <jazn provider="LDAP" jaas-mode="doAsPrivileged"/>
   ```

5. Repeat these steps for *SOA_Oracle_Home*/j2ee/home/application-deployments/orabpel/orion-application.xml.

Reverting from Oracle Internet Directory to the XML-Based JAZN Provider

Follow these procedures if you need to revert from Oracle Internet Directory back to XML-based JAZN as the identity service provider.

1. Log in to the Oracle Enterprise Manager 10*g* Application Server Control Console:

   ```
   http://hostname:port/em
   ```

   where `hostname` is the name of the host on which Oracle BPEL Process Manager is installed and `port` is the Oracle HTTP Server port.

   The Cluster Topology page appears.

2. Click the OC4J instance name in the **Members** section.

   The OC4J: *oc4j_name* page appears.

3. Click the **Administration** tab.
4. Go to the **Security** section in the **Task Name** column.
5. Click the icon in the **Go to Task** column for **Security Providers**.
6. Go to the **Application Name** section.

   The **orabpel** (for Oracle BPEL Process Manager) and **hw_services** (for human workflow) applications appear.

7. Click the **Edit** column for **orabpel**.

   The Security Provider page appears.

8. Click **Change Security Provider**.

   The Change Security Provider page appears.

9. Select **File Based Security Provider** from the **Security Provider Type** list
10. Click **OK**.
11. Repeat Steps 7 through 10 for **hw_services**.
12. Log out of Oracle Enterprise Manager 10*g* Application Server Control Console.
13. Go to the `SOA_Oracle_home/bpel/system/services/config` directory.
14. Delete `is_config.xml`.
15. Rename `is_config.xml.BPM` to `is_config.xml`.
16. If you configured Oracle Internet Directory to use SSL in "Task 4: Configure the Middle Tier to use the LDAP-based JAZN provider with Secure Socket Layer (SSL)" on page 2-13, change the following line in the `jazn.xml` file.

   ```
   <property name="ldap.protocol" value="ssl"/>
   ```

   to

   ```
   <property name="ldap.protocol" value="no-ssl"/>
   ```

17. Restart the Oracle Application Server instance for the changes to take affect.

   ```
   SOA_Oracle_Home/opmn/bin> opmnctl stopall
   SOA_Oracle_Home/opmn/bin> opmnctl startall
   ```

Configuration for a Third-Party LDAP Server

Note the following considerations when using a third-party LDAP server. The configuration for Active Directory is slightly different. These differences are also described.

> **Note:** This section only describes Active Directory configuration on Windows 2003. This is because Windows 2000 does not permit nested security groups.

1. The third-party LDAP servers must be configured to use the following standard `objectClasses`:

For Active Directory	For Other Third-Party LDAP Servers
`top`	`top`
`person`	`person`
`organizationalPerson`	`organizationalPerson`
`user`	`inetOrgPerson`
`group`	`groupOfUniqueNames`

LDAP servers usually predefine the list of searchable attributes based on the `cn`, `firstname`, `lastname`, and `email` attributes. You can customize the attributes that can be searchable. The user manager attribute from `inetOrgPerson objectClass` should be searchable to allow workflow escalation. See the documentation for the third-party LDAP server you are using for how to set up the searchable attribute.

The recommended searchable attribute list is `cn`, `sn`, `givenName`, `uid`, `manager`, `title`, `mail`, and `telephoneNumber`.

2. When you seed Oracle BPEL Process Manager users and roles into the LDAP server, the process assumes that the users' and groups' container is created in LDAP.   To create system and optionally demo ldif files, open the following template files in:

```
SOA_Oracle_Home\bpel\system\services\config\ldap
```

For Active Directory	For Other Third-Party LDAP Servers
`system-winServer2003-ActDir.sbs`	`system-ldap.sbs`
`demo-winServer2003-ActDir.sbs`	`demo-ldap.sbs`
`demo-roleGrants-winServer2003-ActDir.sbs`	

Replace the substitution variables with the appropriate values, as shown in the following examples. The actual values to enter depend upon your domain:

LDAP Server	Substitution Variable	Replace With Value
Active Directory	`%s_UserContainerDN%`	`cn=Users,dc=us,dc=oracle,dc=com`
	`%s_GroupContainerDN%`	`cn=Users,dc=us,dc=oracle,dc=com`

LDAP Server	Substitution Variable	Replace With Value
Other Third-Party LDAP Servers	`%s_UserCommonNamingAttribute%`	`cn`
	`%s_UserContainerDN%`	`ou=People,dc=ldapus,dc=acmeoracle,dc=com`
	`%s_GroupContainerDN%`	`ou=Groups,dc=usldap,dc=acmeoracle,dc=com`

where:

- `%s_UserContainerDN%` with a DN, value of the entry under which all users are supposed to be added. The users container with:
  - dn: `cn=Users,dc=us,dc=oracle,dc=com` is used in this example for Active Directory
  - dn: `ou=People,dc=usldap,dc=acmeoracle,dc=com` is used in this example for other third-party LDAP servers
- `%s_GroupContainerDN%` with a DN value of the entry under which all public groups are supposed to be added. The groups' container with:
  - dn: `cn=Users,dc=us,dc=oracle,dc=com` is used in this example for Active Directory
  - dn: `ou=Groups,dc=ldapus,dc=acmeoracle,dc=com` is used in this example for other third-party LDAP servers
- `%s_UserCommonNamingAttribute%` with the value used to construct the user's DN. In this example for other third-party LDAP servers, the `cn` value is used. `%s_UserCommonNamingAttribute%` and value are not applicable to Active Directory.

Perform the following steps based on your type of third party LDAP server:

- For Active Directory:

  Run the following commands at the DOS command prompt on Windows 2003:

  ```
  ldifde.exe -i -k -f system-winServer2003-ActDir.ldif
  ldifde.exe -i -k -f demo-winServer2003-ActDir.ldif
  ldifde.exe -i -k -f demo-roleGrants-winServer2003-ActDir.ldif
  ```

  See the following Microsoft Active Directory Documentation for details about all bulk import options for Active Directory's `ldifde.exe`:

  ```
  http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies
  /directory/activedirectory/stepbystep/adbulk.mspx#ECAA
  ```

  The Windows system administrator must set passwords for all seeded users; otherwise, worklist application authentication does not work for those users.

- For other third-party LDAP servers:

  Store changes in the `system-ldap.ldif` and `demo-ldap.ldif` files. Then load the `system-ldap.ldif` file to the LDAP server by using the `ldapadd` utility. Optionally, load `demo-ldap.ldif` with the `ldapmodify` utility.

  For example:

  ```
  $ldapadd -c -h ldap.acme.com -p 389 -D "cn=admin" -w welcome -f
  ```

```
system-oid.ldif
$ldapmodify -c -h ldap.acme.com -p 389 -D "cn=admin" -w welcome -f
demo-oid.ldif
```

See the documentation for the third-party LDAP server you are using for information about the `ldapadd` and `ldapmodify` commands.

3. The identity service third-party LDAP provider must specify `connection`, `userControls`, and `roleControls` elements in the identity service configuration file.

   Identity service third-party LDAP provider implementation defines a set of user search properties that must be configured:

   - `nameattribute`—the name of the LDAP attribute that uniquely identifies the name of the user. In Sun Directory Server, it is `uid`; in Active Directory, it is `user`.
   - `objectClass`—the LDAP schema object class used to represent a user. In Sun Directory Server, it is `inetOrgPerson`.

   And a set of role search properties:

   - `nameattribute`—the name of the LDAP attribute that uniquely identifies the name of the role. In Sun Directory Server, it is `uniqueMember`; in Active Directory, it is `member`.
   - `objectclass`—the LDAP schema object class that is used to represent a group. In Sun Directory Server, it is `groupOfUniqueNames`. In Active Directory, it is `group`.
   - `membershipsearchscope`—specifies how deep in the LDAP directory tree to search for role membership. Supported values: `onelevel` or `subtree`.
   - `memberattribute`—The attribute of a static LDAP group object specifying the distinguished names (DNs) of the members of the group. In Sun Directory Server, it is `uniqueMember`; in Active Directory, it is `member`.

Both `userControls` and `roleControl` must define a search element with the `searchbase` attribute.

The `searchbase` attribute of the `userControls` search element is a space-separated list of DNs in the LDAP directory that contains users; for example, `cn=users,dc=us,dc=oracle.com,dc=com`.

The `searchbase` attribute of the `roleControls` search element is a space-separated list of DNs in the LDAP directory that contains roles; for example, `cn=Groups,dc=us,dc=oracle,dc=com`.

Examples of two realm LDAP server configurations are shown below:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<ISConfiguration xmlns="http://www.oracle.com/pcbpel/identityservice/isconfig">
   <configurations>
    <configuration realmName="us" default="true">
         <provider providerType="LDAP" name="iPlanet" service="Identity">
            <connection url="ldap://host:port"
         binddn="uid=admin,ou=administrators,ou=topologymanagement,o=netscaperoot"
         password="welcome1" encrypted="false">
               <pool initsize="2" maxsize="25" prefsize="10" timeout="60"/>
            </connection>
            <userControls>
               <property name="nameattribute" value="uid"/>
               <property name="objectclass" value="inetOrgPerson"/>
               <search searchbase="ou=People, dc=us,dc=oracle,dc=com "
                       scope="obelevel" maxSizeLimit="1000" maxTimeLimit="120"/>
            </userControls>
            <roleControls>
               <property name="nameattribute" value="cn"/>
               <property name="objectclass" value="groupOfUniqueNames"/>
               <property name="membershipsearchscope" value="onelevel"/>
               <property name="memberattribute" value="uniquemember"/>
               <search searchbase="ou=Groups,dc=us,dc=oracle,dc=com"
                       scope="onelevel" maxSizeLimit="1000" maxTimeLimit="120"/>
            </roleControls>
         </provider>
      </configuration>
      <configuration realmName="idc">
           <provider providerType="LDAP" name="openLDAP">
              <connection url="ldap://host:port"
                 binddn="cn=Manager,dc=oracle,dc=com"
                 password="welcome1" encrypted="true">
                <pool initsize="2" maxsize="25" prefsize="10" timeout="300000"/>
              </connection>
              <userControls>
                 <property name="nameattribute" value="uid"/>
                 <property name="objectclass" value="inetOrgPerson"/>
                 <search searchbase="ou=People,dc=idc,dc=oracle,dc=com"
                         scope="onelevel" maxSizeLimit="1000"
                         maxTimeLimit="120000"/>
              </userControls>
              <roleControls>
                 <property name="nameattribute" value="cn"/>
                 <property name="objectclass" value="groupOfUniqueNames"/>
                 <property name="membershipsearchscope" value="onelevel"/>
                 <property name="memberattribute" value="uniquemember"/>
                 <search searchbase="ou=Groups,dc=idc,dc=oracle,dc=com"
                         scope="onelevel" maxSizeLimit="1000"
                         maxTimeLimit="120000"/>
              </roleControls>
           </provider>
        </configuration>
     </configurations>
</ISConfiguration>
```

An example for Microsoft Active Directory follows:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<ISConfiguration xmlns="http://www.oracle.com/pcbpel/identityservice/isconfig">
  <configurations>
     <configuration realmName="AD" >
       <provider providerType="LDAP" name="Active Directory" service="Identity">
          <connection url="ldap://host:port"
             binddn="cn=administrator,cn=Users,dc=us,dc=oracle,dc=com"
             password="welcome1" encrypted="true">
             <pool initsize="2" maxsize="25" prefsize="10" timeout="300000"/>
          </connection>
          <userControls>
             <property name="nameattribute" value="sAMAccountName"/>
             <property name="objectclass" value="user"/>
             <search searchbase="cn=Users,dc=us,dc=oracle,dc=com"
                     scope="subtree" maxSizeLimit="1000" maxTimeLimit="120000"/>
          </userControls>
          <roleControls>
             <property name="nameattribute" value="cn"/>
             <property name="objectclass" value="group"/>
             <property name="membershipsearchscope" value="onelevel"/>
             <property name="memberattribute" value="member"/>
             <search searchbase="cn=Users,dc=us,dc=oracle,dc=com"
                     scope="subtree" maxSizeLimit="1000" maxTimeLimit="120000"/>
          </roleControls>
       </provider>
     </configuration>
  </configurations>
</ISConfiguration>
```

Configuration for Custom Identity Repository Plug-ins

The following example shows how to use configuration properties to configure custom plug-ins. In this example, the `CustomIdentityService` class is used to demonstrate custom repository plug-ins. This class implements the `BPMIdentityService` interface.

```
<?xml version =3D '1.0' encoding =3D 'UTF-8'?>
<ISConfiguration=xmlns=3D"http://www.oracle.com/pcbpel/identityservice/isconfig">
   <configurations>
      <configuration realmName=3D"default">
        <provider providerType=3D"CUSTOM" name=3D"DBProvider" class=3D =
         "is.custom.plugin.CustomIdentityService">
        <connection url=3D"jdbc:polite4@host:1531:orabpel" >
            <property name=3D"driver" =value=3D"oracle.lite.poljdbc.POLJDBCDriver"/>
            <property name=3D"user" value=3D"system"/>
            <property name=3D"password" value=3D"manager"/>
         </connection>
        </provider>
      </configuration>
   </configurations>
</ISConfiguration>
```

In the proceeding example, the configuration defines a provider named `DBProvider` that is implemented by the `is.custom.plugin.CustomIdentityService` class. The provider uses custom connection properties for driver, user, and password.

In addition to the existing provider properties, you can define custom property elements that can be added to `provider`, `connection`, `userControls`, `roleControls`, and `search` elements in the configuration file to extend provider definitions.

Setting Up Group Ownership

The workflow rules for groups can be created and updated by users who are either the group owner or have the `BPMWorkflowAdmin` role. This user can own the group directly or indirectly. The user owns the group or role indirectly if they are a grantee for another role or group that owns the role or group.

Defining Group Ownership for JAZN XML-Based Providers

Group ownership is an extended role property that is stored in the *`SOA_Oracle_Home`*`\bpel\system\services\config\users-properties.xml` file in the `<owners>` element of the `groupObject`.

An owner can be either a user, application role, or group. The `<owners>` element value can have several owners separated by commas. For example:

```
<groupObject>
```

```
        <name>LoanAgentGroup</name>
        <email>user1@dlsun4254.us.oracle.com</email>
        <owners>fkafka, jcooper, BPMAnalyst</owners>
</groupObject>
```

Defining Group Ownership for JAZN Oracle Internet Directory-Based and LDAP-Based Providers

Group ownership is defined through the `owner` attribute. To create an owner of the group in Oracle Internet Directory, you must use Oracle Delegated Administration Services. For third-party LDAP servers, use the server's tools or the `ldapmodify` utility to set values to the group's `owner` attribute. The attribute name for owners can be overwritten with a property element of `roleControls` in the `is_config.xml` identity service configuration file:

```
<property name="roleOwnerAtribute"
  value="customOnwerAttribute" />
```

Configuring the Notification Services

The notification service in Oracle BPEL Process Manager enables you to send notifications from a BPEL process using a variety of channels. Oracle BPEL Process Manager can deliver these notifications by e-mail, voice message, fax, pager, or short message service (SMS). In addition to configuring notification channels in Oracle JDeveloper, you must also perform additional configuration procedures.

This section contains the following topics:

- Configuring the E-mail Server
- Configuring the Wireless Provider for Voice
- Configuring the Wireless Provider for SMS
- Configuring the Wireless Provider for Fax
- Configuring the Wireless Provider for Pager
- Configuring the Pluggable Notification Service

> **See Also:** *Oracle BPEL Process Manager Developer's Guide* for details about configuring notification channels in Oracle JDeveloper

Configuring the E-mail Server

The file `ns_emails.xml` in the directory *`SOA_Oracle_Home`*`\bpel\system\services\config` contains the configuration for e-mail accounts. Each `EmailAccount` element sets the configuration of a specific e-mail account. The `name` attribute in the `EmailAccount` element is the name of the account.

A default e-mail account is specified in the e-mail configuration file. This account is used when there is no account specified to send an e-mail notification. This account is also used to send task-related notifications. A default e-mail account must always be specified in the configuration file.

The `EmailAccount` element contains the `OutgoingServerSettings` and `IncomingServerSettings` attributes. For actionable notifications in a workflow, both `IncomingServerSettings` and `OutgoingServerSettings` are required.

Table 2–1 describes the XML elements for the e-mail notification configuration stored in the `ns_emails.xml` file.

Table 2–1 XML Elements for the E-mail Notification Configuration File

Name	Description
EmailAccount/Name	Name of the account. This can be any name, but must be unique within this server.
EmailAccount/GeneralSettings/FromName	Name of the From e-mail address
EmailAccount/GeneralSettings/FromAddress	E-mail address for the From e-mail address
EmailAccount/OutgoingServerSettings/SMTPHost	Name of the outgoing SMTP server
EmailAccount/OutgoingServerSettings/SMTPPort	Port of the outgoing SMTP server
EmailAccount/OutgoingServerSettings/AuthenticationRequired	Optional element to specify that authentication is required for the SMTP server
EmailAccount/OutgoingServerSettings/UserName	Optional element to specify the user name for the SMTP account
EmailAccount/OutgoingServerSettings/Password	Optional element to specify the password for the SMTP account
EmailAccount/OutgoingServerSettings/Password[encrypted]	Encrypted attribute of the password. It is true if the password is encrypted and false if it is not. You generally set this to false when you first enter the password. The server automatically encrypts the password the first time it reads the configuration file and sets the attribute to true.
EmailAccount/IncomingServerSettings/Server	Name of the incoming e-mail server
EmailAccount/IncomingServerSettings/Port	Port of the incoming e-mail server
EmailAccount/IncomingServerSettings/UserName	User ID of the e-mail address
EmailAccount/IncomingServerSettings/Password	User password
EmailAccount/IncomingServerSettings/Password[encrypted	Encrypted attribute of the password. It is true if the password is encrypted and false if it is not. Generally, you should set this to false when you first enter the password. The server automatically encrypts the password the first time it reads the configuration file and sets the attribute to true.
EmailAccount/IncomingServerSettings/UseSSL	Secure sockets layer (SSL) attribute. It is true if the incoming server requires SSL and false if it does not.
EmailAccount/IncomingServerSettings/Folder	Name of the folder from which to read the incoming messages
EmailAccount/IncomingServerSettings/PollingFrequency	Polling interval for reading messages from the incoming messages folder
EmailMimeCharset	MIME charset to be used to encode the e-mail from the address and the subject

Table 2–1 (Cont.) XML Elements for the E-mail Notification Configuration File

Name	Description
`NotificationMode`	The notification mode of the notification service. It is expected that the notification mode is set to either `ALL` or `EMAIL` after configuring the notification service for e-mail and other channels. By default, this value is set to `NONE` and therefore no notifications are sent. The possible values for this attribute are: ■ `ALL` – the e-mail, SMS, voice, fax, and pager channels are configured and notification is sent through any channel. ■ `EMAIL` – Only the e-mail channel is configured for sending notification messages. ■ `NONE` – No channel is configured for sending notification messages. This is the default setting.

Example ns_emails.xml File

```
<EmailAccounts
xmlns="http://xmlns.oracle.com/ias/pcbpel/NotificationService"
               EmailMimeCharset=""
               NotificationMode="NONE">
   <EmailAccount>
      <Name>Default</Name>
      <GeneralSettings>
         <FromName>Oracle BPM</FromName>
         <FromAddress>accountId@yourdomain.com</FromAddress>
      </GeneralSettings>
      <OutgoingServerSettings>
         <SMTPHost>yourdomain.com</SMTPHost>
         <SMTPPort>25</SMTPPort>
          <AuthenticationRequired>true</AuthenticationRequired>
          <UserName>userId</UserName>
          <Password encrypted="false"
             xmlns:ns0="http://xmlns.oracle.com/ias/pcbpel/NotificationService">
             password</Password>
      </OutgoingServerSettings>
      <IncomingServerSettings>
         <Server>yourdomain.com</Server>
         <Port>110</Port>
         <Protocol>pop3</Protocol>
         <UserName>accountId</UserName>
         <Password ns0:encrypted="false"
            xmlns:ns0="http://xmlns.oracle.com/ias/pcbpel/NotificationService">
            password</Password>
         <UseSSL>false</UseSSL>
         <Folder>Inbox</Folder>
         <PollingFrequency>1</PollingFrequency>
         <PostReadOperation>
            <MarkAsRead/>
         </PostReadOperation>
      </IncomingServerSettings>
   </EmailAccount>
</EmailAccounts>
```

Configuring the Wireless Provider for Voice

The configuration for the wireless service provider is stored in an XML file, `ns_iaswconfig.xml`, which is in

SOA_Oracle_Home\bpel\system\services\config

Table 2–2 describes the XML elements for the voice notification configuration stored in `ns_iaswconfig.xml` on the *SOA_Oracle_Home* server.

Table 2–2 XML Elements for the Voice Notification Configuration File

Name	Description
/IASWConfiguration/SoapURL	URL of the wireless service provider
/IASWConfiguration/UserName	Name of the user account with the wireless service provider
/IASWConfiguration/Password	User password
/IASWConfiguration/Password[encrypted	Encrypted attribute of the password. It is `true` if the password is encrypted and `false` if it is not. Generally, you should set this to `false` when you first enter the password. The server automatically encrypts the password the first time it reads the configuration file and sets the attribute to `true`.
/IASWConfiguration/ProxyHost	Name of the proxy server
/IASWConfiguration/ProxyPort	Port number of the proxy server

Note: The username and password are intentionally left blank at installation. If a username or password is not specified, the wireless server allows up to 50 notifications from a specific IP address. After 50 notifications, you must get a paid account from

`http://messenger.oracle.com`

You then specify the appropriate username and password in the configuration file, `ns_iaswconfig.xml`, or by using Oracle Enterprise Manager 10*g* Application Server Control Console.

Example ns_iaswconfig.xml File

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<!--This XML file stores the details of the IAS Wireless Notification Service-->
<IASWConfiguration xmlns="http://xmlns.oracle.com/ias/pcbpel/NotificationService">
  <!-- URL to the SOAP Service -->
  <SoapURL>http://messenger.oracle.com/xms/webservices</SoapURL>

  <!-- UserName - this username should exist in iAS Wireless schema -->
  <UserName>username</UserName>

  <Password ns0:encrypted="false"
xmlns:ns0="http://xmlns.oracle.com/ias/pcbpel/NotificationService">password
</Password>
</IASWConfiguration>
```

Configuring the Wireless Provider for SMS

As with the voice notification channel, the configuration for the wireless service provider for SMS is stored in the XML file `ns_iaswconfig.xml`, which is in

```
SOA_Oracle_Home\bpel\system\services\config
```

See "Configuring the Wireless Provider for Voice" on page 2-23 to configure a wireless service provider for SMS.

Configuring the Wireless Provider for Fax

As with the voice notification channel, the configuration for the wireless service provider for fax is stored in the XML file `ns_iaswconfig.xml`, which is in

```
SOA_Oracle_Home\bpel\system\services\config
```

See "Configuring the Wireless Provider for Voice" on page 2-23 to configure a wireless service provider for fax. Note that you cannot use the Oracle Enterprise Manager 10*g* Application Server Control Console to configure a fax provider for this release.

Configuring the Fax Cover Page

To add cover pages for a fax, you must edit the *SOA_Oracle_Home*`\bpel\system\services\config\ns_faxcoverpages.xml`. Use the cover page name to specify the cover page in the fax message.

```
<FaxCoverPages xmlns="http://xmlns.oracle.com/ias/pcbpel/NotificationService">
  <FaxCoverPage>
     <Name>legal</Name>
     <MimeType>application/pdf</MimeType>
<FileLocation>C:\orabpel\bpel\runtime\config\faxcoverpages\003288.pdf</FileLocatio
n>
  </FaxCoverPage>
</FaxCoverPages>
```

Configuring the Wireless Provider for Pager

As with the voice notification channel, the configuration for the wireless service provider for pager is stored in the XML file `ns_iaswconfig.xml`, which is in

```
SOA_Oracle_Home\bpel\system\services\config
```

See "Configuring the Wireless Provider for Voice" on page 2-23 to configure a wireless service provider for pager. Note that you cannot use the Oracle Enterprise Manager 10*g* Application Server Control Console to configure a pager provider for this release.

Configuring the Pluggable Notification Service

Custom notification service implementations can be plugged in and used instead of the default notification service providers. You can plug in a custom notification service for all channels or selectively for specific channels. For example, the notification service provides the ability to plug in an existing SMS implementation instead of the default SMS notification service.

Pluggable Notification Service Implementation

To plug in a notification service, you implement the `oracle.tip.pc.services.notification.ICustomNotificationService` interface. This interface has methods for the following channels:

- E-mail
- Voice
- Fax
- SMS
- Instant messaging (IM)
- Pager

The plugged-in notification service can override the default providers for one or more channels. When the custom notification service is overriding the default implementation for a subset of channels, the methods corresponding to the other channels (channels that are not overridden) are not called by the notification service. Those methods can just return null. Alternatively, the implementation can extend the abstract class `oracle.tip.pc.services.notification.AbstractCustomNotificationServiceImpl`, which provides empty implementations for each of the channels. In that case, the implementation can just extend the methods for the interested channels.

Pluggable Notification Service Registration

Once the implementation is available, you register it in the *`SOA_Oracle_Home`*`\system\services\config\ns_iaswconfig.xml` file. The section in `ns_iaswconfig.xml` to set is shown below. By default, all elements are empty.

```
 <!-- Specify any custom implementation for sending notification via Voice, Fax,
Pager, SMS or IM channels.
      'All' refers to an implementation for all of the above specified channels.
 -->
 <CustomNotificationServices>
   <All/>
   <Email/>
   <Voice/>
   <Fax/>
   <Pager/>
   <SMS/>
   <IM/>
 </CustomNotificationServices>
```

If you are overriding the default implementation for all channels, set the `All` element with the complete class name of your implementation:

```
<CustomNotificationServices>
  <All>com.xyz.test.NotificationService</All>
  <Email/>
  <Voice/>
  <Fax/>
  <Pager/>
  <SMS/>
  <IM/>
</CustomNotificationServices>
```

If you are overriding the default implementation for only the e-mail channel, set the `Email` element with the complete class name of your implementation:

```
<CustomNotificationServices>
  <All/>
  <Email>com.xyz.test.NotificationService</Email>
  <Voice/>
  <Fax/>
```

```
    <Pager/>
    <SMS/>
    <IM/>
  </CustomNotificationServices>
```

The override for other channels is configured the same way as the e-mail channel.

Note that the implementation and its dependent classes must be available in the classpath of Oracle BPEL Server.

Configuring the Workflow Service

The configuration for all workflow services is performed in the `SOA_Oracle_Home\bpel\system\services\config\wf_config.xml` file.

```
<workflowConfigurations
  xmlns="http://xmlns.oracle.com/pcbpel/humanworkflow/configurations"
  xmlns:user="http://xmlns.oracle.com/bpel/workflow/userMetadata">
  <taskAutoReleaseConfigurations>
     <taskAutoRelease priority="1" default="P1D" percentageOfExpiration="30"/>
     <taskAutoRelease priority="2" default="P2D" percentageOfExpiration="40"/>
     <taskAutoRelease priority="3" default="P3D" percentageOfExpiration="50"/>
     <taskAutoRelease priority="4" default="P4D" percentageOfExpiration="60"/>
     <taskAutoRelease priority="5" default="P5D" percentageOfExpiration="70"/>
  </taskAutoReleaseConfigurations>
  <worklistApplicationURL>http://mlkenned-pc.us.oracle.com:8888/integration/
      worklistapp/TaskDetails?taskId=PC_HW_TASK_ID_TAG</worklistApplicationURL>
  <actionableEmailAccountName/>
  <pushbackAssignee>INITIAL_ASSIGNEES</pushbackAssignee>
  <assigneeDelimiter><![CDATA[,]]></assigneeDelimiter>
  <shortHistoryActions>
    <action>ACQUIRE</action>
    <action>INFO_REQUEST</action>
    <action>INFO_SUBMIT</action>
    <action>RELEASE</action>
  </shortHistoryActions>
  <workflowServiceSessionTimeoutInMinutes>60</workflowServiceSessionTimeout
     InMinute>
  <user:ruleRepositoryInfo>
    <user:ruleEngine>ORACLE</user:ruleEngine>
    <user:repositoryLocation>WFRepository</user:repositoryLocation>
    <user:dictionaryName>WFDictionary</user:dictionaryName>
    <user:reposProperty name="reposType">jar</user:reposProperty>
  </user:ruleRepositoryInfo>
    <property name="worklist.redirectpage" value="TaskDetails" />
    <property name="worklist.loginpage" value="Login.jsp" />
    <property name="worklist.errorpage" value="Error.jsp" />
</workflowConfigurations>
```

This section describes the configuration parameters in this file:

- taskAutoReleaseConfigurations
- worklistApplicationURL
- actionableEmailAccountName
- pushbackAssignee
- assigneeDelimiter
- shortHistoryActions

- workflowServiceSessionTimeoutInMinutes
- user:ruleRepositoryInfo

> **See Also:** *Oracle BPEL Process Manager Developer's Guide* for additional details about the workflow services

taskAutoReleaseConfigurations

If a task is assigned to groups or multiple users, one of the users in the group or the list of users must acquire the task before acting on it. After the task is acquired, none of the initial assignees can see the task if the task was assigned to them. If the user does not act within a given time, the task is automatically released so that all other users in the group or list of users can see it. A particular business process can disable the autorelease by making autorelease a restricted action. The release duration is configurable in the `wf_config.xml` file.

The configurations for the autorelease durations are in the element `taskAutoReleaseConfigurations`. The release durations can be configured for tasks of each priority. For each priority, the autorelease duration can be specified as a percentage of the expiration (the `percentageOfExpiration` attribute) duration or a default value (the `default` attribute). The default values are used when the task does not have an expiration duration. The datatype of default is `xsd:duration`, whose format is defined by ISO 8601 under the form `PnYnMnDTnHnMnS`. The capital letters are delimiters and can be omitted when the corresponding member is not used. Examples include `PT1004199059S`, `PT130S`, `PT2M10S`, `P1DT2S`, `-P1Y`, or `P1Y2M3DT5H20M30.123S`.

For example, if the task of priority `3` is acquired at `3/24/2005 10:00 AM` and the task expires at `3/31/2005 10:00 AM`, then the time left for expiration is `7` days. If the `percentageOfExpiration` for priority `3` tasks was `50`, then the task is released at `3/37/2005 10:00 PM` (`3 1/2` days from when it was acquired).

worklistApplicationURL

In the e-mails that are sent for tasks, the link to the Oracle BPEL Worklist Application is read from the `worklistApplicationURL` XML element in the `wf_config.xml` file.

The element `worklistApplicationURL` identifies the URL. Configuring this is useful if the custom Oracle BPEL Worklist Application is built. The tag `PC_HW_TASK_ID_TAG` in this URL is replaced with the task ID when constructing the URL for the e-mail.

actionableEmailAccountName

Task actions can be performed through e-mail. The actionable e-mail account is the account in which task action-related e-mails are received and processed. This e-mail account name is identified by the XML element `actionableEmailAccountName` in the `wf_config.xml` file.

pushbackAssignee

A task can be pushed back to the previous approver or previous original assignees. The original assignees do not need to be the approver, as they may have reassigned the task, escalated the task, and so. The XML element `pushbackAssignee` in the `wf_config.xml` file controls whether the task is pushed back to the original assignees or the approvers. The possible values for this element are as follows:

- APPROVER
- INITIAL_ASSIGNEES

assigneeDelimiter

Task assignees in the routing slip can be specified as a delimited string. For example, the following two are equivalent.

- Listing 1:

```
<participant name="Loan Agent">
      <resource isGroup="false" type="STATIC">jcooper, jstein</resource>
    </participant>
```

- Listing 2:

```
<participant name="Loan Agent">
      <resource isGroup="false" type="STATIC">jcooper </resource>
      <resource isGroup="false" type="STATIC">jstein</resource>
    </participant>
```

In the above example, a comma (,) was used as a delimiter. If a different delimiter is used in a particular environment, then the delimiter can be specified in the XML element assigneeDelimiter in the wf_config.xml file. Note that the dynamic assignee names are also interpreted for delimited strings. In the example below, if the XPath expression /task:task/task:payload/payload:assignee returns jcooper, stein, then this participant is the same as listing 2 above.

```
<participant name="Loan Agent">
      <resource isGroup="false"
 type="XPATH">/task:task/task:payload/payload:assignee</resource>
    </participant>
```

shortHistoryActions

The workflow service maintains two types of task history:

- Detailed history
- Short history

The detailed history contains all the changes made to the task. The short history contains only versions created by certain actions. By default, task initiation, reinitation, outcome update, complete, expiration, and withdrawn result in the version being in the short history. You can add other actions to the short history list in the XML element shortHistoryActions in the wf_config.xml file. The possible actions that can be added to the short history actions are listed below.

Action	Action	Action	Action
ACQUIRE	INFO_REQUEST	RENEW	OUTCOME_UPDATE_ ROUTE
AUTO_RELEASE	INFO_SUBMIT	RESUME	
ADHOC_ROUTE	OVERRIDE_ ROUTING_SLIP	SKIP_CURRENT_ ASSIGNMENT	
DELEGATE	PUSH_BACK	SUSPEND	
ERROR	REASSIGN	UPDATE	

Action	Action	Action	Action
ESCALATE	RELEASE	OUTCOME_UPDATE	

workflowServiceSessionTimeoutInMinutes

This is the length of time a workflowContext remains valid. If the client does not perform any activity for longer than the specified time, the workflowContext is marked as invalid, and a new authenticated context must be created.

For the Oracle BPEL Worklist Application, this means that if a user remains logged into the application, but does not perform any activity for a time greater than the value specified in workflowServiceSessionTimeoutInMinutes, they are required to log into the application again.

user:ruleRepositoryInfo

The user metadata service stores the workflow rules for users and groups in an Oracle Business Rules repository file. The ruleRepositoryInfo section of the wf_config.xml file configures how to look up this file. The repository file can be accessed from the file system, or from a HTTP server through the WebDAV protocol. Accessing the repository through WebDAV is useful if you have several instances of the user metadata service on separate hosts that must access the same rule information. The separate instances can all point to the same WebDAV URL.

By default, the rule repository is the file WFRepository, located in the same directory as wf_config.xml.

When specifying a repository file on the file system, set the following properties for ruleRepositoryInfo:

- ruleEngine: ORACLE (only the Oracle Business Rules Rules Engine is currently supported)
- repositoryLocation: file path to the repository file, relative to the directory *SOA_Oracle_Home*\bpel\system\services\config.
- dictionaryName: WFDictionary
- reposProperty
  - name: reposType, value: jar

To host the workflow rules repository using WebDAV, set up a WebDAV Oracle Business Rules repository. Use the import utility in the Oracle Business Rules Rule Author to import the dictionary WFDictionary from the WFRepository file-based repository into your WebDAV-based repository.

Set the following properties for ruleRepositoryInfo:

- ruleEngine: ORACLE (only the Oracle Business Rules Rules Engine is currently supported)
- repositoryLocation: URL for the WebDAV repository
- dictionaryName: WFDictionary
- reposProperty
  - name: reposType, value: webDAV
  - name: proxyHost, value: Web proxy to use when accessing the WebDAV repository (this property is optional)

- name: proxyPort, value: Web proxy port (this property is optional)
- name: wallet, value: path (on local file system) to the wallet file containing the credentials for connecting to a secure WebDAV URL (this property is optional)

See Also:

- *Oracle Business Rules User's Guide* for instructions on setting up a WebDAV repository
- *Oracle BPEL Process Manager Developer's Guide* for details about BPEL process integration with business rules

Integrating Oracle BPEL Process Manager with the Oracle Application Server Service Registry

You can integrate Oracle BPEL Process Manager with Oracle Application Server Service Registry (OracleAS Service Registry).

OracleAS Service Registry is a version 3-compliant implementation of the Universal Description, Discovery and Integration (UDDI) specification, and is a key component of a Service Oriented Architecture (SOA). A UDDI registry provides a standards-based foundation for locating published services, invoking services, and managing metadata about services (security, transport, or quality of service). Consumers can browse and select published services that meet their needs.

Oracle BPEL Process Manager integration with OracleAS Service Registry also insulates Oracle BPEL Process Manager processes from any changes to service endpoints (physical hosted location, implementation, and so on).

The integration between Oracle BPEL Process Manager and OracleAS Service Registry is achieved by passing a reference to the service key (for the registered service in OracleAS Service Registry) at runtime. The registry service inquiry URI is identified at the Oracle BPEL Process Manager domain level by a configuration property. If the service endpoint for the published service changes, the new physical endpoint is discovered with the service key reference (which does not change).

This section provides an overview of how to integrate the RapidDistributors Web service of the OrderBooking tutorial with OracleAS Service Registry. This section contains the following topics:

Task 1: Installing the Oracle Application Server SOA Suite and OracleAS Service Registry

1. Install an Oracle Application Server SOA Advanced install type (recommended). If you install the Oracle Application Server SOA Basic install type, you must perform additional configuration steps described in Step 3.
2. Follow the instructions in *Oracle Application Server Administrator's Guide* to create a second OC4J instance through one of the following methods:
   - With the `createinstance` utility
   - With Oracle Enterprise Manager 10*g* Application Server Control Console

   For the procedures in this section, the second OC4J container is named `registry`.

   > **Notes:**
   >
   > - Do *not* start the second OC4J instance.
   > - The OracleAS Service Registry must *not* reside in the same OC4J container as your Oracle Application Server SOA install type.

3. If you installed the Oracle Application Server SOA Basic install type, perform the following procedure:
   1. Open the *`SOA_Oracle_Home`*`/opmn/config/opmn.xml` file.
   2. Locate the `registry` component.
   3. Change `default-web-site` as follows:

      ```
      <port id="default-web-site" range="8889" protocol="http"/>
      ```

      This assigns a fixed port (`8889`).
4. Follow the instructions in *Oracle Application Server Administrator's Guide* to start the second OC4J instance created in Step 2.
5. Download OracleAS Service Registry from the following Oracle Technology Network location:

   `http://www.oracle.com/technology/tech/webservices/htdocs/uddi/index.html`

   > **Note:** These instructions assume you are familiar with OracleAS Service Registry and have read and used the following documentation included with the downloaded product:
   >
   > - *OracleAS Service Registry 10.1.3 Product Documentation*
   > - *Registry Step By Step Guide About Oracle Registry* (online tutorial)

6. Install OracleAS Service Registry by following the instructions in *OracleAS Service Registry 10.1.3 Product Documentation*, which is included in the downloaded ZIP file. Ensure that you use the same HTTP port (`8889`) specified for the OC4J container.

   Note that the OracleAS Service Registry must *not* be installed in the same OC4J container as Oracle BPEL Process Manager and Oracle Enterprise Service Bus. Depending on the Oracle Application Server SOA install type that you select, the container location is either in `home` or `OC4J_SOA`. The OC4J container name appears in your installation directory path. For example:

- `home` — is the container name if you install the Oracle Application Server SOA Basic install type
- `OC4J_SOA` — is the container name if you accept the default value when installing an Oracle Application Server SOA Advanced install type.

7. Shut down your Oracle Application Server SOA install type:

```
cd SOA_Oracle_Home/opmn/bin
opmnctl stopall
```

8. Copy the following library files from *`SOA_Oracle_Home`*`/j2ee/registry/applications/registry/registry/WEB-INF/lib` to *`SOA_Oracle_Home`*`/j2ee/`*`OC4J_Home`*`/lib/api-ext`:

   - `security-ng.jar`
   - `security3-ng.jar`

   where *`OC4J_Home`* is the name of the OC4J container for your install type:

   - `home` — for the Oracle Application Server SOA Basic install type
   - `OC4J_SOA` — for the Oracle Application Server SOA Advanced install types

   Note: If the `api-ext` directory does not exist, create it.

9. Copy the following library files from the OracleAS Service Registry *`Registry_Installation_Home`*`\lib` directory to the *`SOA_Oracle_Home`*`/bpel/registry/lib` directory:

   - `security2-ng.jar`
   - `security_providers_client.jar`

10. Back up the *`SOA_Oracle_Home`*`/j2ee/`*`OC4J_Home`*`/config/server.xml` file.
11. Open the `server.xml` file.
12. Locate the shared library named `orabpel.common` in the `server.xml` file.
13. Add the following lines at the bottom:

```
<code-source path="SOA_Oracle_Home/bpel/registry/lib/security2-ng.jar"/>
<code-source path="SOA_Oracle_Home/bpel/registry/lib/security_providers_
client.jar"/>
```

   Ensure that you replace *`SOA_Oracle_Home`* with the real path.

14. Back up the *`SOA_Oracle_Home`*`/j2ee/`*`OC4J_Home`*`/config/system-jazn-data.xml` file.
15. Open the `system-jazn-data.xml` file.
16. Add new login modules to the `<jazn-loginconfig>` section:

```
<application>
   <name>NamePasswordNoAN</name>
   <login-modules>
       <login-module>
          <class>com.idoox.security.jaas.NamePasswordLoginModuleNoAuth</class>
          <control-flag>required</control-flag>
          <options>
             <option>
                <name>debug</name>
                <value>true</value>
             </option>
          </options>
       </login-module>
   </login-modules>
</application>

<application>
   <name>NamePasswordAN</name>
   <login-modules>
       <login-module>
           <class>
 com.systinet.uddi.security.jaas.NamePasswordLoginModule</class>
           <control-flag>required</control-flag>
           <options>
               <option>
                   <name>debug</name>
                   <value>true</value>
               </option>
           </options>
       </login-module>
   </login-modules>
</application>
```

17. Back up the *`SOA_Oracle_Home`*`/j2ee/registry/config/system-jazn-data.xml` file.

18. Open the `system-jazn-data.xml` file.

19. Add new login modules to the `<jazn-loginconfig>` section:

```
 <application>
    <name>HttpRequest</name>
    <login-modules>
        <login-module>
            <class>com.systinet.uddi.security.jaas.SmLoginModule</class>
            <control-flag>required</control-flag>
            <options>
                <option>
                    <name>debug</name>
                    <value>true</value>
                </option>
            </options>
        </login-module>
    </login-modules>
 </application>

<application>
   <name>NamePasswordNoAN</name>
   <login-modules>
       <login-module>
           <class>com.idoox.security.jaas.NamePasswordLoginModuleNoAuth</class>
           <control-flag>required</control-flag>
              <options>
                 <option>
                    <name>debug</name>
                    <value>true</value>
                 </option>
              </options>
       </login-module>
   </login-modules>
</application>

<application>
   <name>NamePasswordAN</name>
   <login-modules>
       <login-module>
           <class>
 com.systinet.uddi.security.jaas.NamePasswordLoginModule</class>
           <control-flag>required</control-flag>
           <options>
               <option>
                   <name>debug</name>
                   <value>true</value>
               </option>
           </options>
       </login-module>
   </login-modules>
</application>

<application>
   <name>IdentityAsserter</name>
   <login-modules>
       <login-module><class>
            com.systinet.uddi.security.jaas.IdentityAsserterLoginModule</class>
           <control-flag>required</control-flag>
           <options>
               <option>
                   <name>debug</name>
                   <value>true</value>
               </option>
           </options>
       </login-module>
   </login-modules>
</application>
```

20. Follow the instructions in section 8, "Authentication Configuration" of *OracleAS Service Registry 10.1.3 Product Documentation* to enable OracleAS Service Registry basic authentication.

21. Open `SOA_Oracle_Home/j2ee/registry/application-deployments/registry/registry/orion-web.xml` and change the value of `search-local-classes-first` to `false`.

22. Start your Oracle Application Server SOA install type:

```
cd SOA_Oracle_Home/opmn/bin
opmnctl startall
```

Task 2: Deploying Web Services

You are now ready to deploy the RapidDistributors Web service.

1. Select **Start** > **All Programs** > **Oracle -** ***Oracle_Home*** > **Oracle BPEL Process Manager** > **Developer Prompt** to open up an operating system command prompt at the `SOA_Oracle_Home\bpel\samples` directory.

2. Change directories to the `tutorials\127.OrderBookingTutorial` subdirectory:

   ```
   cd tutorials\127.OrderBookingTutorial
   ```

3. Enter the following command:

   ```
   ant
   ```

   This deploys and starts the required services, including RapidDistributors. If successful, a message appears at the end:

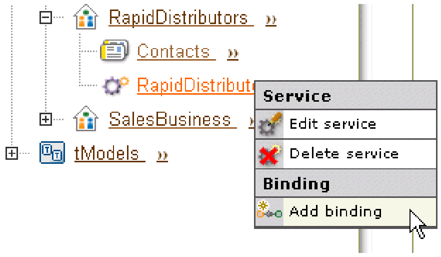
   ```
   BUILD SUCCESSFUL
   ```

Task 3: Publishing a Service and Adding Bindings

1. Go to OracleAS Service Registry.
2. Publish the RapidDistributors service to OracleAS Service Registry by following the instructions for creating a provider and publishing a service in "Publishing a Service" of *OracleAS Service Registry 10.1.3 Product Documentation*. The *Registry Step By Step Guide About Oracle Registry* also provides an example.

   After publishing a service (named **RapidDistributors** for this example), you are ready to add bindings. Bindings represent a Web service instance from which you obtain the access point of an instance. See "Publishing a Binding Template" of *OracleAS Service Registry 10.1.3 Product Documentation* for details.

3. Right-click the published service and select **Add binding**.



   The Add Binding page appears.

4. Enter the Web service access point for the RapidDistributors service:

   ```
   http://hostname:port/orabpel/default/RapidDistributors/1.0/
   ```

5. Select an entry from the **Use type** list:

   - **wsdlDeployment** (This is a mandatory selection and refers to the physical endpoint of the service.)

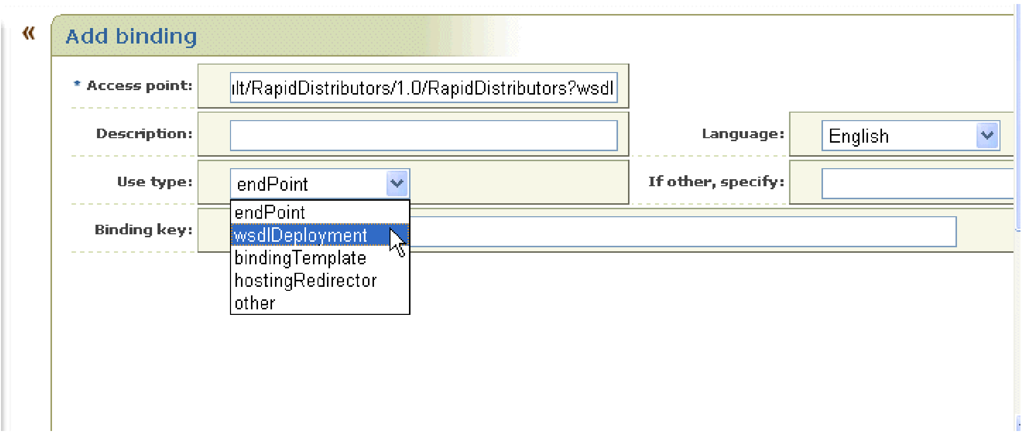
     ```
     http://hostname:port/orabpel/default/RapidDistributors/1.0/RapidDistributor
     s?wsdl
     ```

   - **endPoint** (This is an optional selection and refers to the WSDL endpoint.)

     ```
     http://hostname:port/orabpel/default/RapidDistributors/1.0/
     ```

   - **other** (This is an optional selection and refers to the abstract location of the service.)

     ```
     http://hostname:port/RapidDistributors?wsdl
     ```



Add binding

* Access point: ılt/RapidDistributors/1.0/RapidDistributors?wsdl

Description:

Language: English

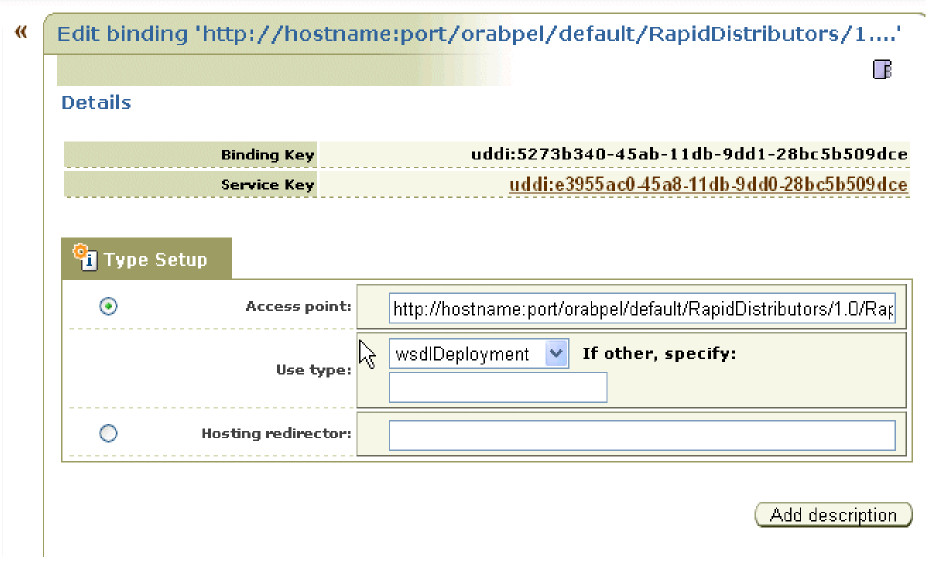
Use type: endPoint

If other, specify:

Binding key:

6. Click **Add binding** when complete.

   The bindings are added to the service. Note the service key value. You later specify this value in a deployment descriptor property in the `bpel.xml` file in "Task 7: Specifying the OracleAS Service Registry Service Key" on page 2-38.



Edit binding 'http://hostname:port/orabpel/default/RapidDistributors/1....'

Details

Binding Key	uddi:5273b340-45ab-11db-9dd1-28bc5b509dce
Service Key	uddi:e3955ac0-45a8-11db-9dd0-28bc5b509dce

Type Setup

● Access point: http://hostname:port/orabpel/default/RapidDistributors/1.0/Rap

Use type: wsdlDeployment

If other, specify:

○ Hosting redirector:

Task 4: Specifying the Registry Service Inquiry URL in Oracle BPEL Control

1. Access Oracle BPEL Control through one of the following methods:

   - Select **Start** > **All Programs** > **Oracle -** ***Oracle_Home*** > **Oracle BPEL Process Manager** > **BPEL Control**
   - Go to the following URL:

     ```
     http://localhost:port/BPELConsole
     ```

2. Select **Manage BPEL Domain** > **Configuration**.
3. Enter a value for the **uddiLocation** property:

   ```
   http://hostname:port/registryname/uddi/inquiry
   ```

   This property must refer to the inquiry WSDL URL of the OracleAS Service Registry. For example:

```
http://hostname.us.oracle.com:42461/registryrc7/uddi/inquiry?wsdl
```

> **Note:** There can only be one OracleAS Service Registry reference in a Oracle BPEL Process Manager Installation at any point in time.

Task 5: Creating a Connection to the UDDI Registry

You now create a connection to OracleAS Service Registry in Oracle JDeveloper.

1. Select **Connection Navigator** from the **View** main menu in Oracle JDeveloper.
2. Right click **UDDI Registry**.
3. Select **New UDDI Registry Connection**.
4. Click **Next** on the Welcome page.
5. Enter the following connection information:

Field	Description
Connection Name	Enter a name for connecting to the registry.
Inquiry Endpoint URL	Enter the URL of the inquiry endpoint. For example: `http://hostname.us.oracle.com:42461/registryrc7/uddi/inquiry?wsdl` **Note:** The value you enter here is the same as the value specified in Oracle BPEL Control in Step 3 of "Task 4: Specifying the Registry Service Inquiry URL in Oracle BPEL Control" on page 2-36.

6. Click **Next**.
7. Test the connection by clicking **Test Connection**. If the connection was successful, the following message appears:

```
Successfully contacted UDDI inquiry endpoint
```

8. Click **Finish**.

   The required endpoint service can now be selected in the Service Explorer window by browsing services under the **UDDI Registry** folder while a creating a partner link.

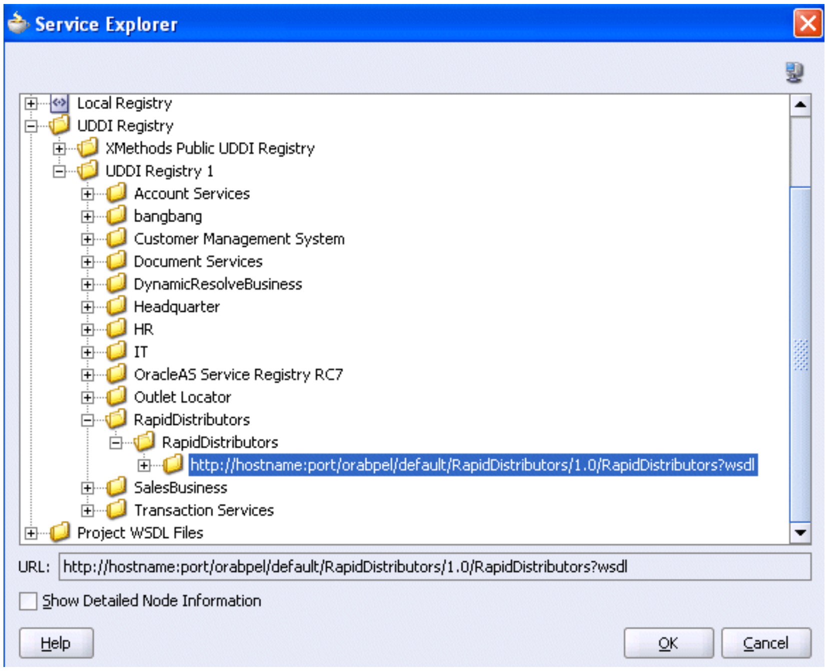
Task 6: Configuring the RapidDistributors Partner Link

You now configure the RapidDistributors partner link and select the Web service access point in OracleAS Service Registry.

1. Open the *SOA_Oracle_Home*/bpel/samples/tutorials/127.OrderBookingTutorial/OrderBooking/OrderBooking.jpr file in Oracle JDeveloper.
2. Double-click the **RapidDistributors** partner link in the designer window.

   The Edit Partner Link window appears.
3. Click the **flashlight** (the second icon from the left named **Service Explorer**) to access the Service Explorer window for selecting the RapidDistributors service deployed in "Task 1: Installing the Oracle Application Server SOA Suite and OracleAS Service Registry" on page 2-31.

4. Expand the navigation tree and select the **RapidDistributors** service under the **UDDI Registry**.



URL: http://hostname:port/orabpel/default/RapidDistributors/1.0/RapidDistributors?wsdl

□ Show Detailed Node Information

Task 7: Specifying the OracleAS Service Registry Service Key

You now configure the service key value with the `registryServiceKey` property. This enables a partner link to invoke the Web service through OracleAS Service Registry.

1. Add the following `registryServiceKey` property to the partner link for RapidDistributors in the respective `partnerLinkBinding` section of the `bpel.xml` file:

```
<property
name="registryServiceKey">uddi:e3955ac0-45a8-11db-9dd0-28bc5b509dce</property>
```

The service key value was created in Step 6 on page 2-36.

At runtime, you can verify that the service for RapidDistributors is invoked from the service endpoint retrieved from OracleAS Service Registry.

Task 8: Securing the Client with Basic Authentication (Optional)

If you want to secure the client side with basic authentication, follow these procedures.

1. Enable secure HTTP basic authentication by adding the following two properties to your `bpel.xml` file in the respective `partnerLinkBinding` section.

```
<property name="registryUsername"> registry_username </property>
<property name="registryPassword"> registry_password </property>
```

where:

- *`registry_username`* — the name of a registry user
- *`registry_password`* — the password for this registry user

You can also set these properties at the domain level in Oracle BPEL Control under **Manage BPEL Domain** > **Configuration** (as you set the **uddiLocation** property). If they are also set in the `bpel.xml` file, the settings in that file overwrite the ones set in Oracle BPEL Control.

Troubleshooting

This section describes troubleshooting procedures for OracleAS Service Registry and Oracle BPEL Process Manager integration.

- A one-time binding fault occurs when OracleAS Service Registry is deployed on a remote Oracle Application Server SOA instance while the BPEL processes are deployed on another SOA instance of a different host. The following error displays the first time the calling BPEL process invokes the registry published service. All subsequent invocations of the service are performed successfully without error (after deployment).

  ```
  <bindingFault xmlns="http://schemas.oracle.com/bpel/extension"><part
  name="code"><code>GenericError</code>
  </part><part name="summary"><summary>http_client transport doesn't support
  nonProxyHosts with wildcards</summary>
  </part></bindingFault>
  ```

- If your WSDL URLs point to hosts using the DHCP communication protocol instead of static IP addresses, ensure that you update the *`SOA_Oracle_Home`*`/opmn/conf/opmn.xml` file to include this information. For example:

  ```
  -Dhttp.proxyHost=www-proxy.us.oracle.com -Dhttp.proxyPort=80
  -Dhttp.nonProxyHosts=122.39.159.106|*acmecorp.com|*.acme.com|localhost|
     ser vices.xmethods.net|xmethods.net|*idc.acme.com|dhcp-idc-towers-122-
     39-159-106.idc.acme.com"/>
  ```

  Otherwise, you can receive deployment errors.

Summary

This chapter describes how to configure the identity service, worklist service, and notification service.

3 Creating a Custom Identity Service Plug-in

This chapter describes how to create a custom identity service plug-in to integrate into specific third-party repositories.

This chapter contains the following topics:

- Creating a Custom Identity Service Plug-in
- Summary

Creating a Custom Identity Service Plug-in

You can create a custom identity service plug-in to integrate into specific third-party repositories. After you create a custom plug-in, you perform the following tasks:

- Assemble all classes into a custom plug-in JAR file
- Deploy the JAR file
- Include the JAR file in the Oracle BPEL Process Manager classpath
- Configure the `is_config.xml` file
- Create system (demo) users and roles

This section contains the following topics:

- Description of Oracle BPEL Process Manager Interfaces
- Implementing the Identity Service Plug-in
- Specifying the Service Factory Methods
- Specifying the Provider Configuration
- Implementing the BPMProvider Interface
- Deploying the Identity Service Plug-in
- Registering and Configuring the Identity Service for the Custom Plug-in
- Creating Users and Groups

Description of Oracle BPEL Process Manager Interfaces

Table 3–1 describes the available interfaces.

Table 3–1 Interfaces

Interface	Description
`BPMIdentityService`	This is the main interface defining the Oracle BPEL Process Manager identity service. For any plug-in, this is the main class that is instantiated by the service factory. This interface extends the `BPMAuthorizationService` and `BPMAuthenticationService` interfaces.
`BPMAuthorizationService`	This defines the Oracle BPEL Process Manager authorization service provider interface.
`BPMAuthenticationService`	This defines the Oracle BPEL Process Manager authentication service provider interface.
`BPMUser`	This defines an Oracle BPEL Process Manager user. A user is identified by a logical name and has various associated attributes such as first name, last name, e-mail, and so on.
`BPMRole`	This defines an Oracle BPEL Process Manager role. A role is defined as the permissions or privileges required by a user to perform the tasks related to their job. Roles are granted to users and groups of users. Roles can be nested (that is, a role itself can be granted to another role, and so on). `BPMRole` defines methods to find all users and roles that are granted a role.
`BPMAppRole`	This defines an Oracle BPEL Process Manager application role. Derived from `BPMRole`, application roles are defined for tasks supported by an application. This enables users and groups that are granted these roles to perform all of the tasks related to the application. `BPMAppRole` instances must define a set of workflow `BPMActivities` that users can have.
`BPMGroup`	This defines an Oracle BPEL Process Manager group. Derived from `BPMRole`, `BPMGroup` is a convenient way to assign rights and permissions to several users at one time. All users who belong to a user group are granted all of that group's privileges.
`BPMIdentity`	This defines the common set of methods and APIs for roles and users. Both `BPMUser` and `BPMRole` are derived from this interface.
`BPMPrincipal`	This represents an identity and defines the common methods for any identity in the Oracle BPEL Process Manager identity service. `BPMIdentity` derives from the `BPMPrincipal` interface.
`BPMProvider`	This defines the set of methods and APIs that must be supported for any third-party repository. This is a convenience interface. This interface makes it possible to have one identity service implementation that can plug into different custom providers. The interface helps to localize all functionality required for repository access in implementing the interface class. You can use this interface for each of your custom repositories.

See Also: The Javadoc for the different interfaces of the identity service located in

SOA_Oracle_Home`\bpel\docs\workflow`

Implementing the Identity Service Plug-in

The entry point for identity service implementation is `BPMIdentityService`.

```
package is.custom.plugin;

import java.util.*;
import javax.servlet.http.HttpServletRequest;
import org.w3c.dom.Element;
import oracle.tip.pc.infra.exception.*;
import oracle.tip.pc.services.common.*;
import oracle.tip.pc.services.identity.*;

public class CustomIdentityService extends BPMServiceBase implements
 BPMIdentityService {


/**
  * Constructor
  */
  private CustomIdentityService(ProviderCfg provCfg) throws ServiceException {
     super(provCfg);
  }
/**
  * Factory Method
  */
  public static Service getInstance(String realmName)  throws ServiceException {
    try {
      BPMIdentityConfigService cfgSrv  =
                                ServiceFactory.getIdentityConfigServiceInstance();
      if ( realmName == null )  {
           realmName = cfgSrv.getDefaultRealmName();
       }
       Configuration config = cfgSrv.getConfigurationInstance(realmName);
       ProviderCfg  providerCfg = config.getProviderCfg(
                                   ProviderCfg.IDENTITY_SERVICE);
       CustomIdentityService service =  new  CustomIdentityService(providerCfg);

       return service;
    } catch (Exception e) {
      throw new ServiceException(e, DiagnosticService.SERVICESCOMPONENT);
    }
  }
…

  /**
  * @see oracle.tip.pc.services.identity. BPMAuthorizationService
     #lookupUser(String)
  */
  public BPMUser lookupUser(String userName)  throws BPMIdentityException,
    BPMIdentityNotFoundException {
    Logger.debugLog("CustomIdentityService::lookupUser() begin");
    if(userName == null)
       throw new BPMIdentityException(
                 PCExceptionIndex.IDENTITYSERVICE_NAME_IS_NULL);

    BPMUser user =  getProvider().lookupUser(userName);
    if(user==null) {
       throw new BPMIdentityNotFoundException(
              PCExceptionIndex.IDENTITYSERVICE_USER_NOT_FOUND,
              new String[] {userName, getRealmName()});
    }
    Logger.debugLog("CustomIdentityService::lookupUser() end");
    return user;
 }
 ….
}
```

In the proceeding example, CustomIdentityService is a class that implements the BPMIdentityService interfaces. You must implement the BPMAuthenticationService or BPMAutorizationService interface if the configuration specifies CUSTOM providerType for only the authentication or authorization service providers:

```
<?xml version = '1.0' encoding = 'UTF-8'?
<ISConfiguration xmlns="http://www.oracle.com/pcbpel/identityservice/isconfig"
   <configurations
      <configuration realmName="jazn.com"
          <provider providerType="JAZN" name="xml" service="Identity"
             <property name="userPropertiesFile" value="users-properties.xml"/
          </provider
          <provider providerType="CUSTOM
             name="CustomPlugIn" service="Authentication
             class="package.name.CustomAuthenticationService" /
      </configuration
   </configurations
</ISConfiguration
```

In the preceding example, only the authentication provider uses the custom implementation while all other inquiries use the JAZN XML-based provider. The class CustomAuthenticationService must implement the BPMAuthenticationService interface.

If the *SOA_Oracle_Home*\bpel\system\services\config\is_config.xml file specifies CUSTOM as the providerType for the identity service, then the BPMIdentityService interface must be implemented:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<ISConfiguration xmlns="http://www.oracle.com/pcbpel/identityservice/isconfig">
   <configurations>
      <configuration realmName="jazn.com">
         <provider providerType="CUSTOM"
                   name="CustomPlugIn" service="Identity"
                   class="package.name.CustomIdentityService">
         <property name="customProperty" value="customValue" />
      </configuration>
   </configurations>
</ISConfiguration>
```

All three service classes implementing the BPMIdentityService, BPMAuthenticationService, or BPMAuthorization interface can extend the oracle.tip.pc.services.identity.BPMServiceBase class to leverage basic service functionality.

Specifying the Service Factory Methods

All custom service classes must have the static factory getInstance() method to create the instance of the service provider:

```
public static Service getInstance(String realmName)  throws ServiceException;
```

The realmName is a context of the business process. If realmName is null, it assumes the default realm is used. You can get the default realm name by using the configuration service API:

```
BPMIdentityConfigService cfgService =
 ServiceFactory.getIdentityConfigServiceInstance();
String defaultRealmName = cfgService.getDefaultRealmName();
```

The service factory `oracle.tip.pc.service.common.ServiceFactory` class defines the following public static methods:

```
BPMIdentityService         getIdentityServiceInstance(String realmName);
BPMAutorizationService     getAuthorizationServiceInstance(String  realmName);
BPMAuthenticationService   getAuthenticationServiceInstance(String realmName);
BPMIdentityService         getIdentityServiceInstance();
BPMAutorizationService     getAutorizationServiceInstance();
BPMAuthenticationService   getAuthenticationServiceInstance();
```

These methods are used by Java clients to receive the instances of the corresponding services providers. The service factory methods create custom service instances at run time if the `CUSTOM` provider type is specified for the service. These methods invoke the custom service `getInstance(String realmName)` API.

If the service provider class extends the `oracle.tip.pc.services.BPMServiceBase` class, the service factory method must get an instance of the `ProviderCfg` class, which defines properties for a given configuration. The service can then be created:

```
CustomIdentityService service = new CustomIdentityService(providerCfg);
```

Specifying the Provider Configuration

The `BPMIdentityConfigService` instance provides access to the `Configuration` object by the given `realmName`:

```
Configuration conf = cfgService.getConfigurationInstance(realmName);
```

The specific provider configuration can then be fetched:

```
ProviderCfg providerCfg = conf.getProviderCfg(serviceName);
```

where `serviceName` is one of the following values:

- `ProviderCfg.IDENTITY_SERVICE` — for identity service provider
- `ProviderCfg.AUTHENTICATION_SERVICE` — for authentication service provider
- `ProviderCfg.AUTHORIZATION_SERVICE` — for authorization service provider

> **See Also:** Identity service configuration Javadoc located in
>
> *SOA_Oracle_Home*`\bpel\docs\workflow`

Implementing the BPMProvider Interface

The easiest way to write authentication and authorization services for a custom plug-in is to implement the `BPMProvider` interface for the custom repository. The `BPMProvider` interface defines the minimum set of essential operations that must be defined over a custom repository for the identity service to function correctly. All instances of the implementing object can delegate calls to the provider that is responsible for dealing with the repository-specific calls.

For example, the `BPMServiceBase` class is an abstract class that your service providers may extend. If they do, the custom service classes must implement the `init()` method:

```
public void init() throws BPMIdentityException {
       m_provider = CustomerProvider.getInstance(m_providerCfg);
       m_status = new ServiceStatus(true, "Service is available", -1,
}
```

where m_provider is a protected member of the BPMServiceBase to reference the BPMProvider object. CustomerProvider is a class that implements the BPMProvider interface. Optionally, you can load the provider class dynamically. Get the provider class name for the CUSTOM providerType from the is_config.xml file. The name attribute of the provider element is reserved for that purpose. The m_status member is a protected member of the BPMServiceBase class. It stores the ServiceStatus object.

Deploying the Identity Service Plug-in

For the workflow services to instantiate and invoke the plug-in, it must be deployed to a location in the library path of Oracle BPEL Process Manager.

The plug-in code must be compiled, the classes assembled into a JAR file (for example, isplugin.jar), and the JAR file deployed to the *SOA_Oracle_Home*\services\lib directory.

The library path must also point to the deployed JAR file. For example, isplugin.jar must be added to the library path by adding the following lines to the application.xml file in the *SOA_Oracle_Home*\j2ee\home\config directory:

```
 <library path="SOA_Oracle_Home\bpel\system\services\lib\isplugin.jar"/>
```

The same deployment scheme can be used for any custom plug-in, with minor changes such as the name of the provider JAR file.

Registering and Configuring the Identity Service for the Custom Plug-in

Identity service configuration in defined in the is_config.xml file. The file must be located in a directory that is in the classpath of Oracle BPEL Process Manager. By default, this file is stored in *SOA_Oracle_Home*\bpel\system\services\config.

The following configuration file sample describes the database plug-in implemented as a custom provider for the identity service:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<ISConfiguration mlns="http://www.oracle.com/pcbpel/identityservice/isconfig">
    <configurations>
       <configuration realmName="realm">
          <provider providerType="CUSTOM" service="Identity"
                    name="package.name.CustomerProvider"
                    class="package.name.CustomIdentityService">
              <property name="dataSource" value="jdbc/BPELSamplesDataSource"/>
          </provider>
    </configuration>
  </configurations>
</ISConfiguration>
```

The name attribute of the provider element can load the BPMProvider class. The class attribute defines the implementation class for BPMIdentityService. The custom property with the name dataSource gets the data source name for the JDBC connection.

> **See Also:** "Configuring the Identity Service" on page 2-1 for identity service configuration procedures

Creating Users and Groups

The identity service does not define any of the administrative APIs, including the creation of users and roles or the granting of roles to users. You must use tools specific to the user repository to accomplish this task.

See Also: Appendix A, "Demo User Community" for details about users, roles, and groups

Summary

This chapter describes how to create a custom identity service plug-in to integrate into specific third-party repositories. Details are provided for assembling all classes into a custom plug-in JAR file, deploying the JAR file, including the JAR file in the Oracle BPEL Process Manager classpath, configuring the `is_config.xml` file, and creating system users and groups.

4

Configuring and Viewing BPEL Process Logs

This chapter describes how to configure Oracle BPEL Process Manager logging levels and view logging results.

This chapter contains the following topics:

Logging Overview

Oracle BPEL Process Manager uses the log4j tool to generate log files containing messages that describe startup and shutdown information, errors, warning messages, access information on HTTP requests, and additional information.

The log4j tool enables logging at runtime without modifying the application binary. Instead, logging behavior is controlled by editing properties in Oracle BPEL Control and Oracle BPEL Admin Console.

Two logging levels are supported in Oracle BPEL Process Manager:

- Domain — Manages logging information within specific domains
- System — Manages logging information on a system-wide level

The default format of log files created by log4j cannot be read by the log loader and written to the log repository in Oracle Enterprise Manager 10*g* Application Server Control Console. To do this, you must change the appender format of the messages written to log files. Make this change in the `log4j-config.xml` file, which is located in the following directories:

- For domain level logging, change `log4j-config.xml` in the *`SOA_Oracle_Home`*`\bpel\domains\`*`domain_name`*`\config` directory
- For system level logging, change `log4j-config.xml` in the *`SOA_Oracle_Home`*`\bpel\system\config` directory

See Also:

- "Configuring Logging for Application Server Control" of *Oracle Application Server Administrator's Guide* for details about changing the appender format for log files in Oracle Enterprise Manager 10*g* Application Server Control Console
- `http://logging.apache.org/log4j/docs` for details about log4j
- *Oracle Process Manager and Notification Server Administrator's Guide* for details about Oracle Process Manager and Notification Server (OPMN) log files located in *`SOA_Oracle_Home`*`\opmn\logs`

Domain Level Logging

Domain logging enables you to log and troubleshoot issues for a specific BPEL domain. You set domain logging levels in Oracle BPEL Control. The domain log files are located in *`SOA_Oracle_Home`*`\bpel\domains\`*`domain_name`*`\logs`.

Follow these procedures to set domain logging levels.

1. Access Oracle BPEL Control through one of the following methods:
   - Selecting **Start** > **All Programs** > **Oracle -** ***Oracle_Home*** > **Oracle BPEL Process Manager** > **BPEL Control**
   - Going to the following URL:

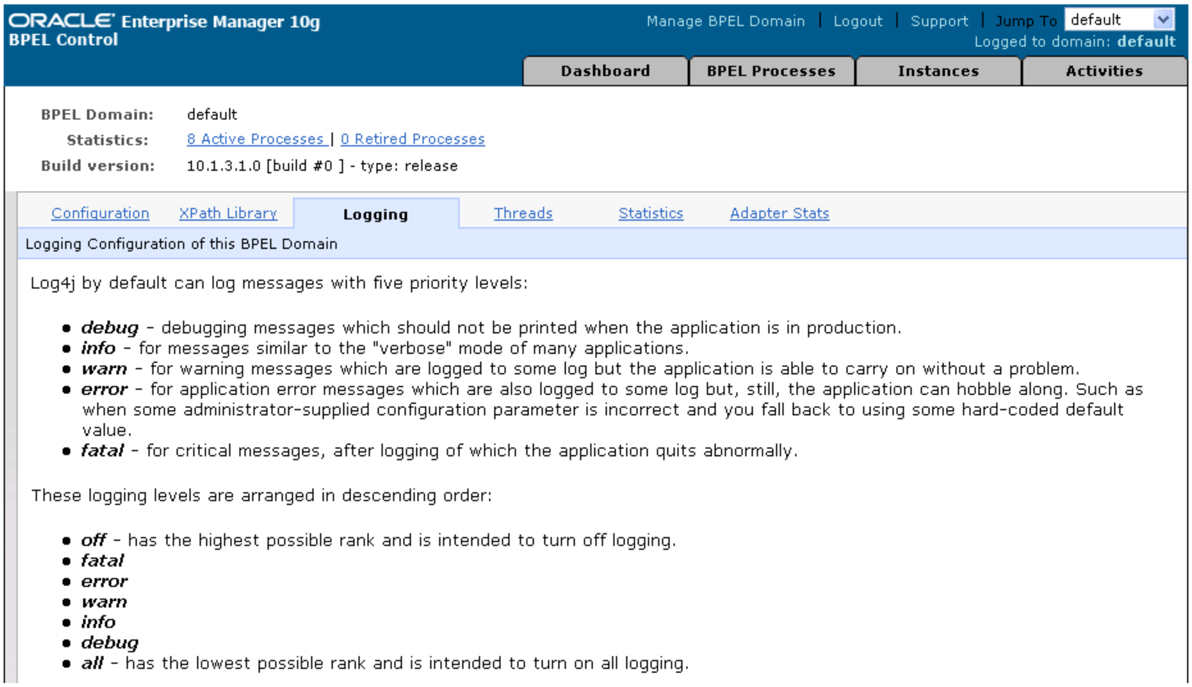
     ```
     http://localhost:port/BPELConsole
     ```

     where *`port`* is:

     – `8888` if you installed Oracle BPEL Process Manager from the Oracle Application Server SOA software CD.

     – `9700` if you installed the Oracle BPEL Process Manager for Developers or Oracle BPEL Process Manager for OracleAS Middle Tier install type from the Oracle BPEL Process Manager software CD.
2. Enter the `oc4jadmin` user name and password when prompted.
3. Select the domain for which to set domain logging levels from the list in the upper right corner.
4. Select **Manage BPEL Domain** > **Logging**.

   The Logging window appears.



BPEL Domain: default
Statistics: 8 Active Processes | 0 Retired Processes
Build version: 10.1.3.1.0 [build #0] - type: release

Logging Configuration of this BPEL Domain

Log4j by default can log messages with five priority levels:

- ***debug*** - debugging messages which should not be printed when the application is in production.
- ***info*** - for messages similar to the "verbose" mode of many applications.
- ***warn*** - for warning messages which are logged to some log but the application is able to carry on without a problem.
- ***error*** - for application error messages which are also logged to some log but, still, the application can hobble along. Such as when some administrator-supplied configuration parameter is incorrect and you fall back to using some hard-coded default value.
- ***fatal*** - for critical messages, after logging of which the application quits abnormally.

These logging levels are arranged in descending order:

- ***off*** - has the highest possible rank and is intended to turn off logging.
- ***fatal***
- ***error***
- ***warn***
- ***info***
- ***debug***
- ***all*** - has the lowest possible rank and is intended to turn on all logging.

The following logging levels are available and listed here from highest priority to lowest priority. When a logging level is specified, all messages with a lower priority level than the one selected are ignored.

Logging Level	This Selection...
Off	Disables logging. This selection is the highest priority.
Fatal	Logs critical messages. After logging occurs, the application quits abnormally.
Error	Logs application error messages to a log; the application continues to run (for example, an administrator-supplied configuration parameter is incorrect and you default to using a hard-coded value).
Warn	Logs warning messages to a log; the application continues to run without problems.
Info	Logs messages in a format similar to the verbose mode of many applications.
Debug	Logs debugging messages that should not be printed when the application is in a production environment.
All	Enables all logging. This selection is the lowest priority.

The lower part of the Logging window displays the types of loggers you can set.

5. Review the levels and descriptions.

Logger Name	Description
`*domain_name*.collaxa.cube`	General BPEL logging (system)
`*domain_name*.collaxa.cube.activation`	Activation agent logging (inbound adapters)
`*domain_name*.collaxa.cube.bpeltest`	BPEL unit test logging
`*domain_name*.collaxa.cube.console.reports`	Oracle BPEL Control reports logging

Logger Name	Description
domain_name.collaxa.cube.engine	XML message logging
domain_name.collaxa.cube.engine.agents	Internal agent framework logging (for example, expiration agents)
domain_name.collaxa.cube.engine.archive	JAR file archive deployment logging in pre-10.1.3 releases. Archiving is not supported in 10.1.3.
domain_name.collaxa.cube.engine.bpel	BPEL process logging. If enabled, each executed BPEL activity logs messages.
domain_name.collaxa.cube.engine.data	Persistence and dehydration layer logging
domain_name.collaxa.cube.engine.delivery	Delivery service and manager logging; this logger is responsible for callbacks and initiating delivery
domain_name.collaxa.cube.engine.deployment	Deployment of BPEL suitcases logging
domain_name.collaxa.cube.engine.dispatch	Asynchronous message logging
domain_name.collaxa.cube.engine.transaction	Transaction-related logging of the execution of process steps. This logger is not used in 10.1.3.
domain_name.collaxa.cube.messaging	Messaging layer logging (as Oracle BPEL Process Manager uses messaging services for scaling)
domain_name.collaxa.cube.security	Server-side security logging (message header authentication layer)
domain_name.collaxa.cube.sensor	Sensor publisher layer logging
domain_name.collaxa.cube.services	Services logging (for example, notifications or human workflow)
domain_name.collaxa.cube.translation	Adapter translation layer logging (the transformation between adapter protocol and inbound XML documents)
domain_name.collaxa.cube.ws	Communication-related logging (for example, WSIF layer (inbound and outbound), SOAP, and adapters).
domain_name.collaxa.cube.xml	XML processing and transformation, XPath, and XML documents (BPEL variables) logging
domain_name.oracle.bpel.security	Inside validator logging

6. Select a level from the **Logging Level** list for a specific logger name or select a single level for all logger names from the **Change All** list.
7. Click **Apply**.
8. Rerun the process to collect logging data.
9. Review the domain.log file located in *SOA_Oracle_Home*\bpel\domains\domain_name\logs.

Note: These parameters can also be configured by editing log4j-config.xml in the *SOA_Oracle_Home*\bpel\domains\domain_name\config directory.

System Level Logging

System level logging is provided for infrastructure, AXIS, and WSIF issues. You set system logging levels in Oracle BPEL Admin Console. The system log files are located in *SOA_Oracle_Home*\bpel\system\logs.

Follow these procedures to set system logging levels.

1. Access Oracle BPEL Admin Console:

   ```
   http://localhost:port/BPELAdmin
   ```

   where *port* is:

   - 8888 if you installed Oracle BPEL Process Manager from the Oracle Application Server SOA software CD.
   - 9700 if you installed the Oracle BPEL Process Manager for Developers or Oracle BPEL Process Manager for OracleAS Middle Tier install type from the Oracle BPEL Process Manager software CD.

2. Enter the oc4jadmin user name and password when prompted.
3. Click **Logging**.

   The Logging window appears.

   The logging levels that display are the same as those described in Step 4 on page 4-3. The lower part of the Logging window displays the types of loggers you can set.

Logger Name	Description
collaxa	General Oracle BPEL Process Manager logging
collaxa.cube.cluster	Cluster-related logging
collaxa.cube.infrastructure	Infrastructure logging issues such as database connectors
collaxa.cube.services	All Oracle BPEL Process Manager service logging
org.collaxa.thirdparty.apache.axis	General AXIS-related logging
org.collaxa.thirdparty.apache.axis.enterprise	AXIS-related logging
org.collaxa.thirdparty.apache.axis.transport	Axis-related logging (to see what AXIS is sending over the network)
org.collaxa.thirdparty.apache.wsif	System-wide WSIF logging.
org.collaxa.thirdparty.apache.wsif.logging	WSIF logging
org.collaxa.thirdparty.jgroups	JGroups related-logging
org.quartz	Quartz scheduler-related logging
wsif	WSIF logging

4. Review the levels and descriptions.
5. Select a level from the **Logging Level** list for a specific logger name or select a single level for all logger names from the **Change All** list.
6. Click **Apply**.
7. Rerun the process to collect logging data.

8. Review the `orabpel.log` file located in *`SOA_Oracle_Home`*`\bpel\system\logs`.

> **Note:** These parameters can also be configured by editing `log4j-config.xml` in the *`SOA_Oracle_Home`*`\bpel\system\config` directory.

System and Domain Level Logging Examples

This section provides examples of the logger names to set to the **Debug** logging level to troubleshoot problems.

Example 1: Process Invokes an External Web Service

Your process invokes an external Web service, and the data retrieved by the Web service is incorrect. Check the following loggers:

- *`domain_name`*`.collaxa.cube.ws` — to see if something went wrong before sending
- `org.collaxa.thirdparty.apache.axis` and `org.collaxa.thirdparty.apache.axis.transport` — to see what is currently being sent

Example 2: Java Class Invoked through WSIF Binding

A Java class is invoked through WSIF binding and a binding fault occurs. Check the following loggers:

- *`domain_name`*`.collaxa.cube.ws` — to see the complete stack
- `org.collaxa.thirdparty.apache.wsif` — to examine the WSIF portion that performs the real invocation

Example 3: Process Invokes an Asynchronous Service

Your process invokes an asynchronous service and never receives a callback (it times out, or waits forever). Check the following loggers:

- *`domain_name`*`.collaxa.cube.ws` — this is related to the outbound direction
- `org.collaxa.thirdparty.apache.axis` and `org.collaxa.thirdparty.apache.axis.transport` — to see what was sent and to see the outgoing WSA header needed for correlation
- *`domain_name`*`.collaxa.cube.engine.delivery` — to see what the delivery handler does, and if a message is retrieved that can be correlated

Example 4: Process Sends a Notification

Your process sends a notification through an e-mail activity, but the e-mail does not arrive at the intended location. Check the following logger:

- *`domain_name`*`.collaxa.cube.services` — to see what occurred during delivery attempt

Logging with Sensors

You can use sensors to generate application logging activity. Note that logging with sensors impacts performance because sensor data objects are built even when logging is disabled.

You add sensors to specific activities and then extract data from variables. To do this, you must implement a custom sensor publishing action to do the log4j logging. For example, you can create a sensor on an invoke activity and create a message that is sent to a JMS queue.

> **See Also:** *Oracle BPEL Process Manager Developer's Guide* for details about sensors

Logging with bpelx:exec in a Java Embedding Activity

You can also log messages by adding custom Java code to a BPEL process using the Java BPEL `exec` extension `bpelx:exec` inside a Java Embedding activity in Oracle JDeveloper.

The method `addAuditTrailEntry(String):void` enables you to add an entry to the audit trail.

Summary

This chapter describes how to configure and view BPEL process logs at the domain and system levels. Examples of logger names to set in order to view troubleshooting information are provided. Alternative methods for generating logging information (with sensors and with `bpelx:exec`) are also described.

A

Demo User Community

This appendix describes the demo user community for task assignments in Oracle BPEL Process Manager.

This appendix contains the following topics:

- Setting Up JAZN Demo Users
- Summary

Setting Up JAZN Demo Users

Demo users are included with Oracle BPEL Process Manager. See "Demo Users and Roles" on page A-1 for a list of users and roles seeded with the product.

See Also:

- Chapter 2, "Service Configuration"
- *Oracle BPEL Process Manager Developer's Guide*
- *Oracle Containers for J2EE Configuration and Administration Guide*
- *Oracle Containers for J2EE Security Guide*
- *Oracle Containers for J2EE Services Guide*

Demo Users and Roles

For the LDAP-based JAZN provider (Oracle Internet Directory), users are seeded under the realm you select when prompted during BPEL Process Manager for OracleAS Middle Tier installation.

The default password is `welcome1` for all LDAP-based JAZN providers (Oracle Internet Directory, Active Directory, Sun Directory Server, and so on) and for the XML-based JAZN provider.

For the XML-based JAZN provider, the Oracle BPEL Process Manager demo community is defined under the default realm `jazn.com`. The Oracle BPEL Process Manager demo community includes:

- System users (`oc4jadmin`, `bpeladmin`, `guest`, and `default`)
- Demo users (see Table A–1)
- Application roles (see Table A–2)
- Enterprise groups (see Table A–3)
- Demo role and group owners (see Table A–4)

The seventeen Oracle BPEL Process Manager demo users and demo role owners are shown in Table A–1.

Table A–1 Demo User Community

User	Name	First Name	Middle Name	Last Name	Title	Manager	E-mail
1	achrist	Agatha	--	Christie	Loan Consultant	sfitzger	user3@dlsun1313.us.oracle.com
2	cdickens	Charles	--	Dickens	CEO	--	user1@dlsun1313.us.oracle.com
3	cdoyle	Conan	--	Doyle	Loan Agent 2	rsteven	user4@dlsun1313.us.oracle.com
4	fkafka	Franz	--	Kafka	Manager 1	ltolsoy	user2@dlsun1313.us.oracle.com
5	istone	Irving	--	Stone	Loan Agent 2	sfitzger	user3@dlsun1313.us.oracle.com
6	jausten	Jane	--	Austen	Loan Consultant	fkafka	user3@dlsun1313.us.oracle.com
7	jcooper	James	--	Cooper	Loan Agent 1	jstein	user3@dlsun1313.us.oracle.com
8	jlondon	Jack	--	London	Loan Agent 1	sfitzger	user3@dlsun1313.us.oracle.com
9	jstein	John	--	Steinbeck	Manager 2	wfaulk	user2@dlsun1313.us.oracle.com
10	ltolsoy	Leo	--	Tolstoy	Director	wfaulk	user1@dlsun1313.us.oracle.com
11	mmitch	Margaret	Munnerlyn	Mitchell	Loan Analyst	fkafka	user3@dlsun1313.us.oracle.com
12	mtwain	Mark	--	Twain	Loan Agent 2	jstein	user3@dlsun1313.us.oracle.com
13	rsteven	Robert	Louis	Stevenson	Manager 3	jstein	user4@dlsun1313.us.oracle.com
14	sfitzger	Scott	--	Fitzgerald	Manager 1	wfaulk	user2@dlsun1313.us.oracle.com
15	szweig	Stefan	--	Zweig	Loan Analyst	fkafka	user3@dlsun1313.us.oracle.com
16	wfaulk	William	--	Faulkner	Vice President	cdickens	user1@dlsun1313.us.oracle.com
17	wshake	William	--	Shakespeare	Loan Consultant	rsteven	user4@dlsun1313.us.oracle.com

Notes:

- The system users (oc4jadmin, bpeladmin, guest, and default) are also part of the demo user community shown in Table A–1.
- All users have the languagePreference property defined as en-US (U.S. English) and the notificationPreference property set to Mail.

Table A–2 and Table A–3 list the Oracle BPEL Process Manager application roles and enterprise groups for the users shown in Table A–1.

Table A–2 Oracle BPEL Process Manager Application Roles

Role	Demo Users in Role	System Users in Role	Granted Roles	Direct Role/Group Grantee to Role
BPMAnalyst	sfitzger, fkafka, jstein, szweig, mmitch, wshake	default, guest, oc4jadmin, bpeladmin	--	BPMWorkflowAdmin, EasternRegion
BPMSystemAdmin	--	oc4jadmin, bpeladmin	BPMWorkflowSuspend, BPMWorkflowReassign, BPMAnalyst, BPMDefaultDomainAdmin, BPMWorkflowViewHistory, BPMWorkflowAdmin, rule-administrators	--
BPMWorkflowAdmin	--	oc4jadmin, bpeladmin	BPMWorkflowSuspend, BPMWorkflowReassign, BPMAnalyst, BPMWorkflowViewHistory, rule-administrators	BPMSystemAdmin
BPMWorkflowReassign	jstein, wfaulk, sfitzger,	oc4jadmin, bpeladmin	--	BPMWorkflowAdmin
BPMWorkflowSuspend	jstein, wfaulk, sfitzger,	oc4jadmin, bpeladmin	--	BPMWorkflowAdmin
BPMWorkflowViewHistory	jcooper, wshake, wfaulk, mtwain, fkafka, szweig, jlondon, mmitch, cdoyle, istone	oc4jadmin, bpeladmin	--	BPMWorkflowAdmin, RegionalOffices
BPMDefaultDomainAdmin	--	default	--	BPMSystemAdmin

See Also:

- "Oracle BPEL Control and Oracle BPEL Admin Console Users and Roles" on page 1-22 for additional details about the BPMDefaultDomainAdmin role
- The "Oracle BPEL Process Manager Workflow Services" chapter of *Oracle BPEL Process Manager Developer's Guide* for additional details about other roles listed in Table A–2

Table A–3 Oracle BPEL Process Manager Enterprise Groups

Group	Demo Users in Group	System Users in Role	Granted Roles and Group	Direct Role and Group Grantee to Group
Supervisor	jcooper,mtwain, rsteven	--	--	--
LoanAnalyticGroup	fkafka, szweig, mmitch	--	LoanAgentGroup	--
LoanAgentGroup	jlondon,istone, jcooper,mtwain, cdoyle, wshake, mmitch, fkafka, szweig	--	--	LoanAnalyticGroup
RegionalOffices	jlondon,istone, jcooper,mtwain, cdoyle, wshake, mmitch, fkafka, szweig	--	BPMWorkflowViewHistory	EasternRegion, CentralRegion, WesternRegion
EasternRegion	wshake, fkafka, szweig, mmitch	--	BPMAnalyst, RegionalOffices, BPMWorkflowViewHistory	--
CentralRegion	jlondon, mtwain	--	RegionalOffices, BPMWorkflowViewHistory	--
WesternRegion	cdoyle,jcooper, istone	--	RegionalOffices, BPMWorkflowViewHistory	California
California	istone, jcooper	--	RegionalOffices, WesternRegion, BPMWorkflowViewHistory	--

Oracle BPEL Process Manager declares the role PUBLIC, which is implicitly granted to all registered Oracle BPEL Process Manager users.

The workflow rules for groups and roles can be created by users who are either the group owner or have the BPMWorkflowAdmin role. The user can own the group directly or indirectly. The user owns the group and role indirectly if they are a grantee for another role and group that owns the role and group. Table A–4 shows the list of roles and groups with owners. The bpeladmin system user owns the BPMWorkflowAdmin role.

Table A–4 Oracle BPEL Process Manager Demo Role and Group Owners

Role/Group	Direct Owners	Owners
LoanAnalyticGroup	jstein, BPMAnalyst	bpeladmin, default, guest, oc4jadmin, fkafka, jstein, mmitch, sfitzger, szweig, wshake
LoanAgentGroup	fkafka, jcooper, BPMAnalyst	bpeladmin, default, guest, oc4jadmin, fkafka, jstein, jcooper, mmitch, sfitzger, szweig, wshake, wfaulk
Supervisor	jstein	jstein
California	fkafka	fkafka
WesternRegion	jstein	jstein

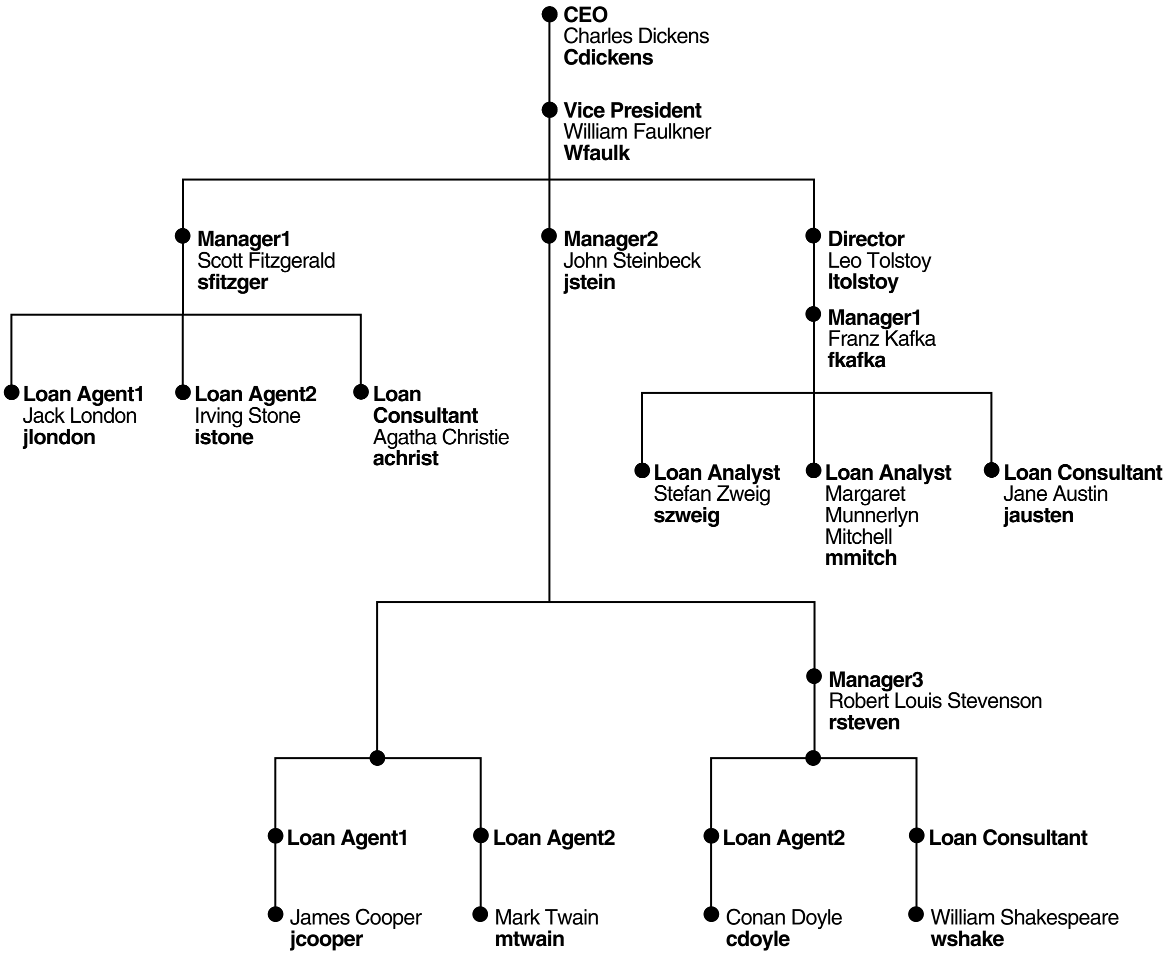
Table A–4 (Cont.) Oracle BPEL Process Manager Demo Role and Group Owners

Role/Group	Direct Owners	Owners
`EasternRegion`	`jstein`	`jstein`
`BPMAnalyst`	`jstein, jcooper, LoanAgentGroup`	`cdoyle, jstein, jcooper, szweig, istone, jlondon, fkafka, mmitch, mtwain, sfitzger, wshake`
`BPMWorkflowAdmin`	`bpeladmin`	`bpeladmin`

Note: Every demo community user who belongs to the `BPMAnalyst` role is an indirect owner of the `LoanAgentGroup` and `LoanAnalyticGroup` roles. Every user who belongs to the `LoanAgentGroup` is an indirect owner of the `BPMAnalyst` role.

Figure A–1 shows the organizational hierarchy of the demo users.

Figure A–1 Demo Users Organizational Hierarchy



Using the Demo User Community in the Order Booking Tutorial

The OrderBooking tutorial provides an example in which you use the demo user community. In this tutorial, you assign a group of users to a task in the Human Task editor. When using this editor, you perform the following tasks:

- Select **Management Chain** as the participant type, which enables a chain of management to sequentially review the task.
- Assign a task to a group named **Supervisor**.
- Select **1** for the number of levels in the management chain to sequentially review this task.

When the BPEL process is deployed, you log in to the Oracle BPEL Worklist Application with the user **jcooper**, acquire the task, and approve it. As shown in Table A–1, the supervisor of **jcooper** is **jstein**. Because you specified **1** as the number of levels in the management chain to sequentially review this task, **jstein** (the supervisor of **jcooper**) must also review this task. You then log in as **jstein** and approve the task.

> **See:** *Oracle BPEL Process Manager Order Booking Tutorial* for instructions on performing these tasks

Summary

This appendix describes the demo user community for task assignments in Oracle BPEL Process Manager.

Index