

Oracle® Fusion Middleware Application Adapters

Application Adapter Best Practices Guide for

12c (12.2.1.0.0)

E80129-01

July 2016

Provides best practices for Oracle Application Adapters for Oracle WebLogic Server.

Oracle Fusion Middleware Application 12c (12.2.1.0.0) Best Practices Guide for Oracle WebLogic Server, 12c (12.2.1.0.0)

E80129-01

Copyright © 2001, 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author: Stefan Kostial

Contributors: Vikas Anand, Marian Jones, Sunil Gopal, Bo Stern

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	v
Audience	v
Documentation Accessibility	v
Related Documents	v
Conventions	vi
1 General Best Practices	
Understanding Managed Connections	1-1
Configuring Oracle Application Adapters in a High Availability Cluster Environment for SOA, BPM, and OSB	1-3
Installing and Configuring SOA or OSB in a Clustered Environment Containing Two Systems .	
1-4	
Deploying Oracle Application Adapters	1-5
Deploying and Configuring the iBSE Connector Application	1-6
Deploying the iBSE configuration	1-15
Configuring a BSE Configuration in Application Explorer	1-15
Key Points to Consider	1-17
Configuring the High Availability (HA) File Adapter for an Outbound Process	1-17
Configuring the Outbound Process	1-22
Configuring Oracle HTTP Server for the Inbound Process	1-23
Configuring the Inbound Process	1-26
Singleton Testing	1-26
Preferred Repository Type	1-28
Generating Input XML Documents	1-28
Prerequisites	1-28
Creating a WSDL Document Using Application Explorer	1-28
Generating an Input XML File From a Request Schema	1-29
Using the Generated Input XML File in the Oracle Enterprise Manager Console	1-32
2 Oracle Application Adapter for SAP R/3 Best Practices	
Understanding Connection Pools	2-1
Security	2-2
Load Balancing	2-2
Inbound to SAP Load Balancing With a SAP Message Server	2-2
Outbound From SAP Load Balancing With the SAP Gateway	2-3

Encoding.....	2-4
SAP Server Behavior.....	2-4
IDoc Configuration.....	2-5
IDoc Metadata.....	2-5
3 Oracle Application Adapter for Siebel Best Practices	
Best Practices for Siebel.....	3-1
Configuring Connection Pooling for Siebel.....	3-2
4 Oracle Application Adapter for PeopleSoft Best Practices	
PeopleSoft Events (Inbound Processing).....	4-1
5 Oracle Application Adapter for J.D. Edwards OneWorld Best Practices	
Connection Pooling and Load Balancing.....	5-1
Security.....	5-2
Multiple Connections and Multiple Users.....	5-2
Unicode Support.....	5-2
Delivery Confirmation and Error Messages.....	5-2
Index	

Preface

Welcome to *Oracle Fusion Middleware Application Adapter Best Practices Guide for Oracle WebLogic Server*. This document provides general best practices that are common to all Oracle Application Adapters for Oracle WebLogic Server.

Audience

This document is intended for system administrators and application developers who are using Oracle Application Adapters for Oracle WebLogic Server.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 12c (12.2.1.0.0) documentation set:

- *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for SAP R/3 User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for Siebel User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for PeopleSoft User's Guide for Oracle WebLogic Server*

- *Oracle Fusion Middleware Application Adapter for J.D. Edwards OneWorld User's Guide for Oracle WebLogic Server*
- Oracle's Unified Method (OUM)

A wealth of additional Governance information can be found within Oracle's Unified Method (OUM). OUM can be used by Oracle employees, Oracle Partner Network Certified Partners or Certified Advantage Partners, and Clients who either participate in the OUM Customer Program or are engaged on projects where Oracle provides consulting services. OUM is a web-deployed toolkit for planning, executing and controlling software development and implementation projects.

For more information about OUM, see the OUM FAQ at

http://my.oracle.com/portal/page/myo/ROOTCORNER/KNOWLEDGEAREAS1/BUSINESS_PRACTICE/Methods/Learn_about_OUM.html

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.
<i>italic</i>	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.

General Best Practices

This chapter lists and describes general best practices that are common to all Oracle Application Adapters for Oracle WebLogic Server.

Note: Throughout this document, `<ORACLE_HOME>` refers to the 12c (12.2.1.0.0) SOA/OSB installed home location.

`<ADAPTER_HOME>` refers to the following:

- For SOA:

`<ORACLE_HOME>\soa\soa\thirdparty\ApplicationAdapters`

- For OSB:

`<ORACLE_HOME>\osb\3rdparty\ApplicationAdapters`

This chapter contains the following sections:

- [Section 1.1, "Understanding Managed Connections"](#)
- [Section 1.2, "Configuring Oracle Application Adapters in a High Availability Cluster Environment for SOA, BPM, and OSB"](#)
- [Section 1.3, "Preferred Repository Type"](#)
- [Section 1.4, "Generating Input XML Documents"](#)

1.1 Understanding Managed Connections

Oracle WebLogic Server (WLS) creates and maintains a pool of connections to the Oracle J2CA resource adapter. These are called "managed connections". Each managed connection maintains a distinct set of connections to a back-end Enterprise Information System (EIS) systems (exposed as an adapter target). The set is initially empty and becomes populated only as specific adapter target connections are requested by an executing SOA process that is using the managed connection.

A managed connection maintains only a single connection to any given adapter target and reuses it each time a connection to that target is requested. When WLS destroys a managed connection, the managed connection closes all its internal EIS connections.

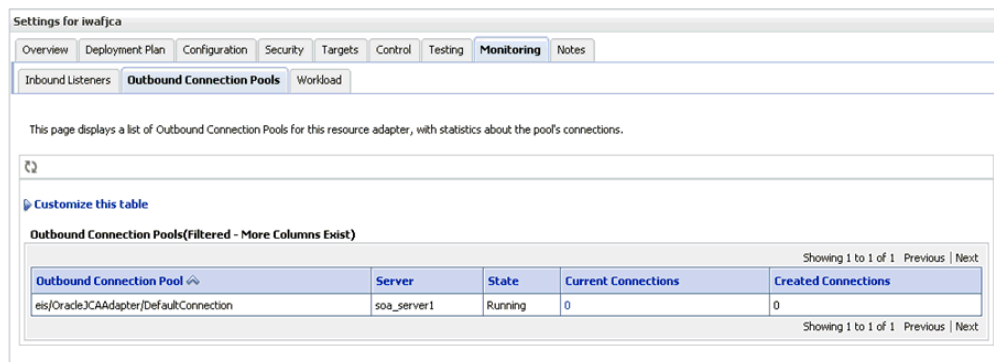
By way of illustration, the following steps detail a scenario in which a SOA process containing references to multiple Oracle adapter targets is used.

1. SOA process requests a managed connection to the Oracle resource adapter ("eis/OracleJCAAdapter/DefaultConnection") and receives a new, empty, managed connection called "Managed1".
2. SOA process invokes a service with adapter target MySAP/SAP1 using Managed1.
3. Managed1 creates a new MySAP adapter instance, activates it for target SAP1, and adds it to its internal EIS connection set.
4. Service is invoked, response returned, and SOA process continues.
5. SOA process invokes a service with adapter target MySAP/SAP2 using Managed1.
6. Managed1 creates a new MySAP adapter instance, activates it for target SAP2, and adds it to its internal EIS connection set.
7. Service is invoked, response returned, and SOA process continues.
8. SOA process invokes a service with adapter target JDEdwards/JDE1 using Managed1.
9. Managed1 creates a new JDEdwards adapter instance, activates it for target JDE1, and adds it to its internal EIS connection set.
10. SOA process terminates. WLS returns Managed1 to the connection pool.

After termination of the SOA process, Managed1 is available to be used by some other SOA process. Its internal EIS connections (SAP1, SAP2, JDE1) remain open. They are reused the next time a process requests connections to those targets from Managed1. They are closed only when WLS destroys Managed1.

You can monitor the managed connection pool in the WLS admin console, as shown in [Figure 1-1](#).

Figure 1-1 Outbound Connection Pools Tab



The "current connections" column is the number of managed connections that are currently in use, presumably by executing SOA processes. This number should go up when a SOA process is started and should go down after a period of inactivity.

When a SOA process needs a managed connection, it tries to retrieve one from the connection pool. If a managed connection is not available, and the number of current connections does not yet exceed the maximum size of the pool, then it creates a new one. If the maximum size of the pool has been reached and no free connection is available, then a resource availability error is generated. When a process is finished, the connection is returned to the pool.

It is important to remember that these numbers represent connections to the Oracle resource adapter and not to any specific EIS. A single Oracle managed connection may contain any number of EIS connections depending upon its usage history. Consequently, there is no way to introspect the number of EIS connections in use by the Oracle resource adapter.

The maximum size of the managed connection pool, the minimum size, and other pool parameters can be set in the *weblogic-ra.xml* file and viewed in the admin console, as shown in [Figure 1-2](#).

Figure 1-2 Connection Pool Tab

Settings for `javax.resource.cci.ConnectionFactory`

General Properties Transaction Authentication **Connection Pool** Logging

Save

This page allows you to view and modify the pool parameters of this outbound connection.

Initial Capacity:

Max Capacity:

Capacity Increment:

Shrinking Enabled:

Shrink Frequency Seconds:

There are many pool parameters, but because they control the pool of connections to the Oracle resource adapter and not any specific EIS many of these are immaterial. Setting an initial pool size, for example, accomplishes nothing except creating a set of empty resource adapter connections. There is no reason to do this. For similar reasons, the "capacity increment" parameter is also not particularly useful.

You have to set the initial capacity to zero always to work with the Oracle adapters.

Some settings have implicit effects for all adapter targets. For example, setting a maximum size for the managed connection pool implicitly establishes the maximum number of connections that can be made to any single target since each managed connection can hold at most one connection to a unique adapter target. Similarly, defining the lifetime of a managed connection implicitly defines the lifetime for all EIS connections created by the connection. However, there is no way to control connection pooling for specific EIS targets using the Oracle "universal" resource adapter.

1.2 Configuring Oracle Application Adapters in a High Availability Cluster Environment for SOA, BPM, and OSB

Application adapters integrate Oracle WebLogic Server with various packaged applications, which includes Oracle Application Adapters for PeopleSoft, SAP R/3, Siebel, and J.D. Edwards OneWorld.

This section describes the steps required to successfully deploy the 12c (12.2.1.0.0) Oracle Application Adapters in a cluster environment (in all available SOA or OSB

nodes, which in this case are two systems) and working with inbound and outbound processes.

Note: All of the nodes in the cluster environment use the same database. As a result, the database repository must be configured for the specific database and all of the nodes must point to this database. For more information on how to configure BSE and J2CA database repository settings, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server* (Section 2.7.4, "Configuring the Oracle Database Repository").

This section includes the following topics:

- [Section 1.2.1, "Installing and Configuring SOA or OSB in a Clustered Environment Containing Two Systems"](#)
- [Section 1.2.2, "Deploying Oracle Application Adapters"](#)
- [Section 1.2.3, "Deploying and Configuring the iBSE Connector Application"](#)
- [Section 1.2.4, "Key Points to Consider"](#)
- [Section 1.2.5, "Configuring the High Availability \(HA\) File Adapter for an Outbound Process"](#)
- [Section 1.2.6, "Configuring the Outbound Process"](#)
- [Section 1.2.7, "Configuring Oracle HTTP Server for the Inbound Process"](#)
- [Section 1.2.8, "Configuring the Inbound Process"](#)
- [Section 1.2.9, "Singleton Testing"](#)

1.2.1 Installing and Configuring SOA or OSB in a Clustered Environment Containing Two Systems

To install and configure SOA or OSB in a clustered environment containing two systems:

1. Install SOA or OSB suite on both the systems. For more information on installing SOA or OSB suite, see the following websites:
 - For SOA:
<http://docs.oracle.com/middleware/1221/core/INSOA/GUID-D5AFD830-8A7D-42CC-8C22-CE68C452CF4A.htm#INSOA369>
 - For OSB:
<http://docs.oracle.com/middleware/1221/core/INOSB/GUID-D5AFD830-8A7D-42CC-8C22-CE68C452CF4A.htm#INSOA369>
2. Create and configure SOA or OSB clustered domains. For more information, see the following websites:
 - For SOA:
<http://docs.oracle.com/middleware/1221/core/INSOA/GUID-36DFF16B-4891-46EB-9554-436A3CCF85BB.htm#INSOA377>
 - For OSB:
<http://docs.oracle.com/middleware/1221/core/INOSB/GUID-36DFF16B-4891-46EB-9554-436A3CCF85BB.htm#INSOA377>

554-436A3CCF85BB.htm#INSOA377

Note: The above links contain steps for configuring a cluster environment in a single system. Modify the steps accordingly to configure cluster environments in two systems (for example, providing an IP of the second system for the second Managed Server, creating two machines instead of a single machine, and pointing the second managed server to the second machine, and so on).

3. Pack the domain in the first system and unpack it in the second system. For more information, see:

http://docs.oracle.com/middleware/1221/core/ASHIA/scale_out_prototype.htm#ASHIA134

1.2.2 Deploying Oracle Application Adapters

To deploy the J2CA Connector Application:

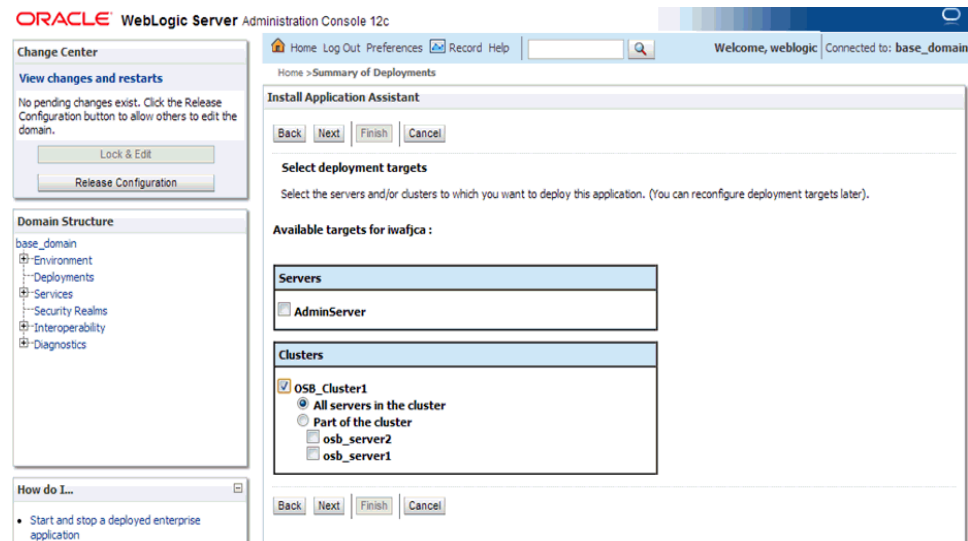
1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that has been configured.
2. Start the SOA or OSB managed servers in both systems.
3. Open the Oracle WebLogic Server Administration Console in a web browser by entering the following URL:

`http://host name:port/console`

Where:

- *host name* - Is the name of the system where Oracle WebLogic Server is running.
 - *port* - Is the port number for the domain you are using.
4. Log in to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.
 5. In the Domain Structure section in the left pane, click **Deployments**.
 6. Click **Lock & Edit** in the Change Center.
 7. Click **Install**.
 8. Browse to the Adapters installed directory, select the option next to `iwafjca.rar`, and click **Next**.
 9. Accept the default selection (**Install this deployment as an application**) and click **Next**.
 10. In the Clusters section, select **SOA_Cluster** or **OSB_Cluster**, followed by **All servers in the cluster**, and then click **Next**, as shown in [Figure 1-3](#).

Figure 1–3 Clusters Section



The Optional Settings page is displayed.

11. Accept the default values and click **Next**.

The Summary page is displayed.

12. Click **Finish**.

The Settings page for the J2CA (iwafjca) Connector Application appears.

13. Click **Save**.

14. Click **Activate Changes** in the Change Center.

15. In the Domain Structure section in the left pane, click **Deployments** and navigate through the table that lists all the deployed applications and find the J2CA (iwafjca) connector application.

16. Select the Control tab and then select the check box next to iwafjca.

17. Click the **Start** submenu (down arrow) and select **Servicing all requests**.

The Start Application Assistant page is displayed.

18. Click **Yes**.

19. Verify that the application has successfully started in the Deployments page.

20. Similarly, repeat steps 5 through 19 for the iwafjca.war deployment.

To deploy and configure the iBSE Connector Application:

1.2.3 Deploying and Configuring the iBSE Connector Application

This section contains the following topics:

- [Section 1.2.3.1, "Deploying the iBSE configuration"](#)
- [Section 1.2.3.2, "Configuring a BSE Configuration in Application Explorer"](#)

Note: Before continuing with the steps below, ensure that the cluster domain is created and configured already. If it is not created, see [Section 1.2.1, "Installing and Configuring SOA or OSB in a Clustered Environment Containing Two Systems"](#) on page 1-4 for further details, and create the cluster domain before proceeding with these steps.

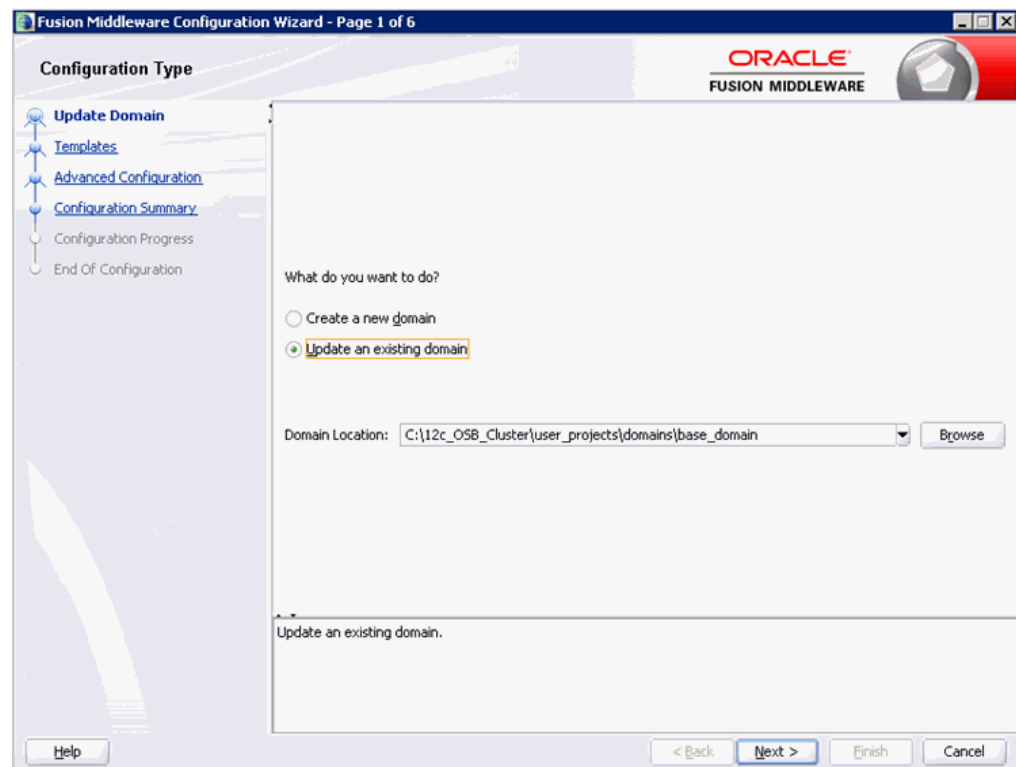
To extend the domain with HTTP Proxy Server for iBSE:

1. Navigate to the following directory and execute `config.cmd`.

```
<ORACLE_HOME>\oracle_common\common\bin
```

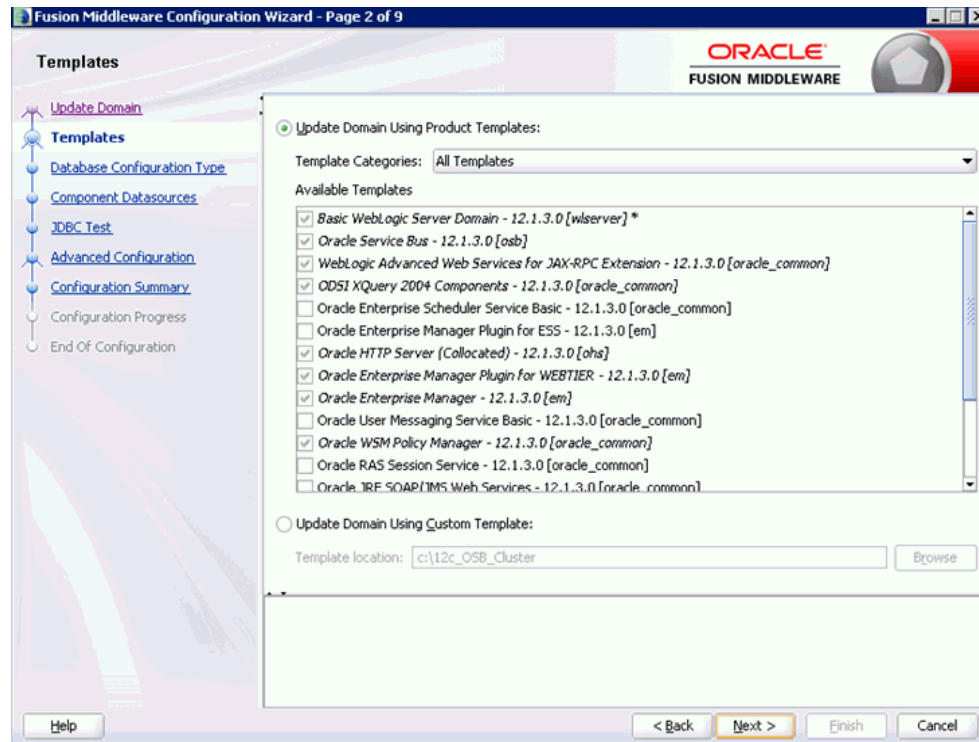
2. In the displayed Configuration Wizard, select **Update an existing domain**, browse to the location of the created cluster domain, and click **Next**, as shown in [Figure 1-4](#).

Figure 1-4 Configuration Type Pane



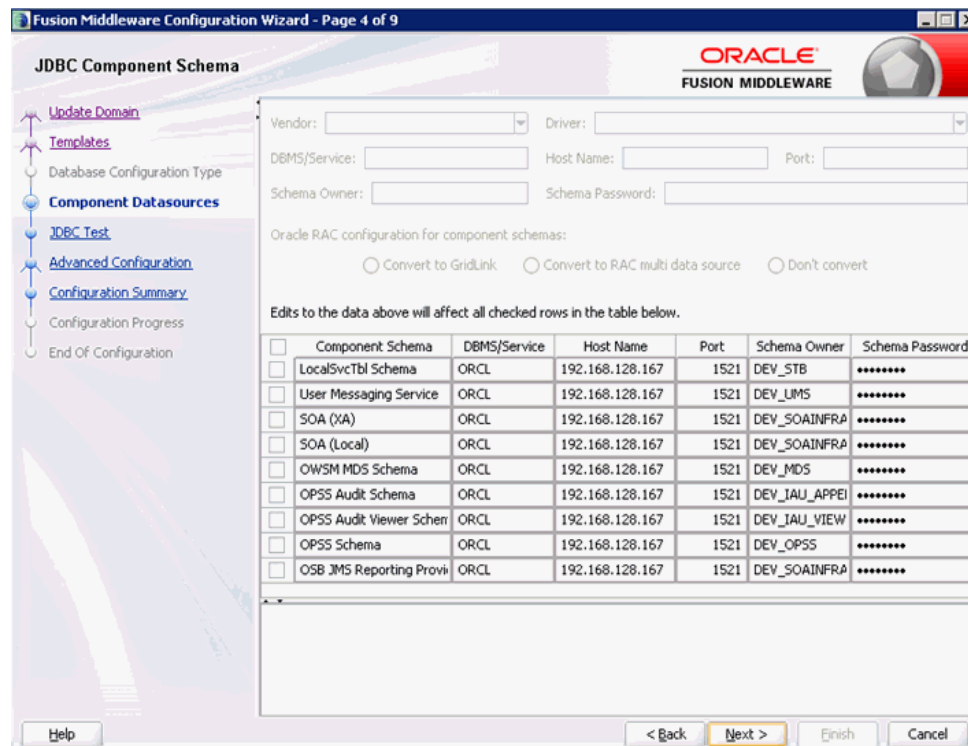
3. In the Templates pane that appears, leave the default values and click **Next**, as shown in [Figure 1-5](#).

Figure 1-5 Templates Pane



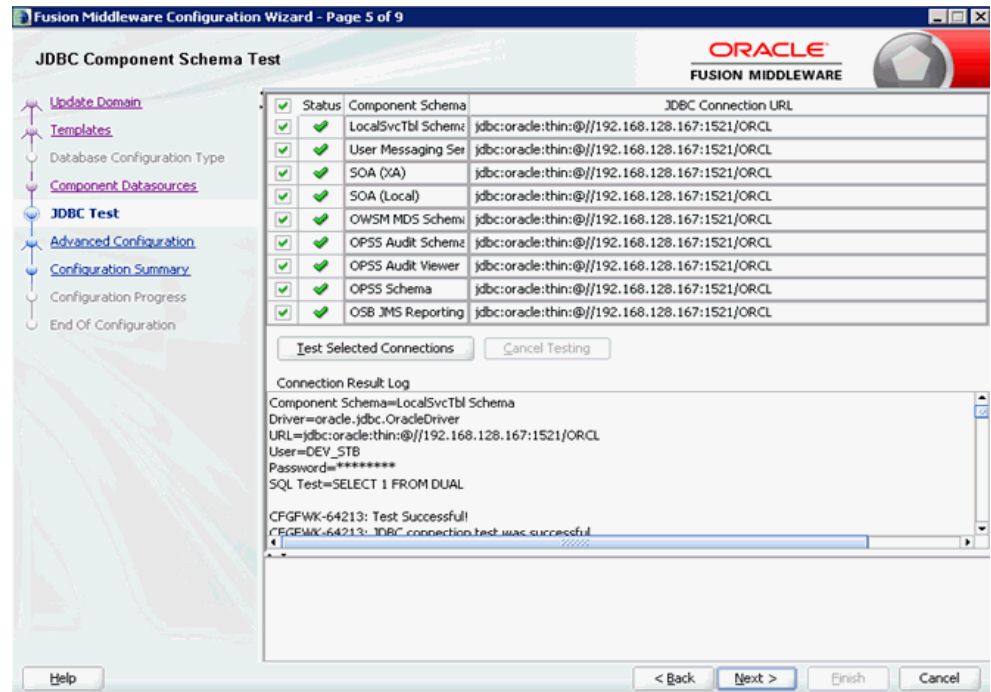
- In the JDBC Component Schema pane that appears, leave the default values and click **Next**, as shown in Figure 1-6.

Figure 1-6 JDBC Component Schema Pane



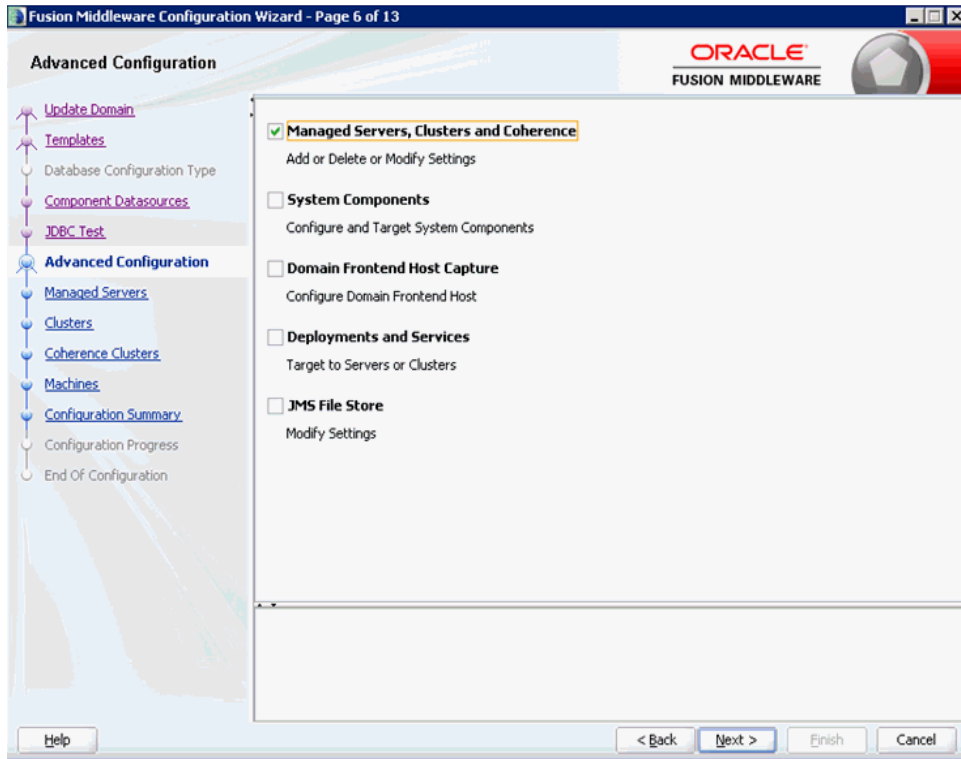
- In the JDBC Component Schema Test pane, make sure all the tests have passed, and then click Next, as shown in Figure 1-7.

Figure 1-7 JDBC Component Schema Test Pane



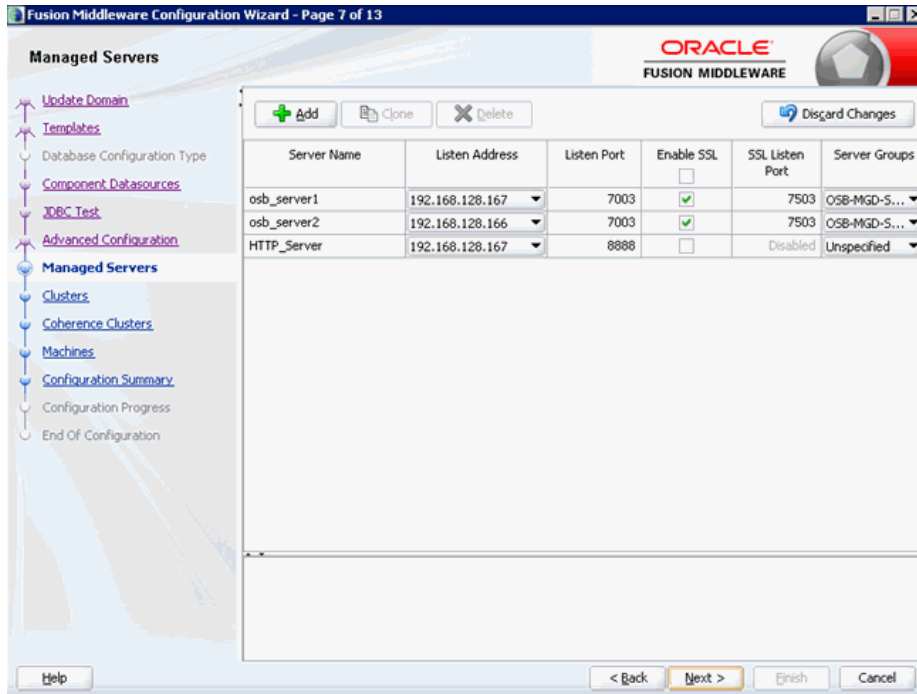
- Select the **Managed Servers, Cluster and Coherence** check box, and then click **Next**, as shown in Figure 1-8.

Figure 1–8 Advanced Configuration Pane



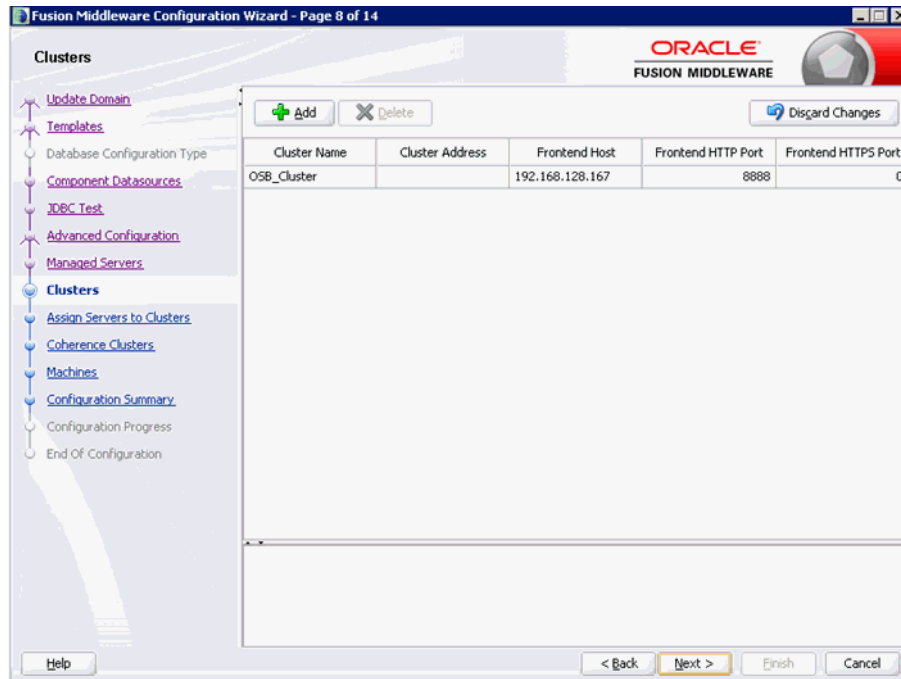
7. Click **Add**, create a new server (for example, HTTP_Server), and provide a Listen port (for example, 8888), as shown in Figure 1–9.

Figure 1–9 Managed Servers Pane



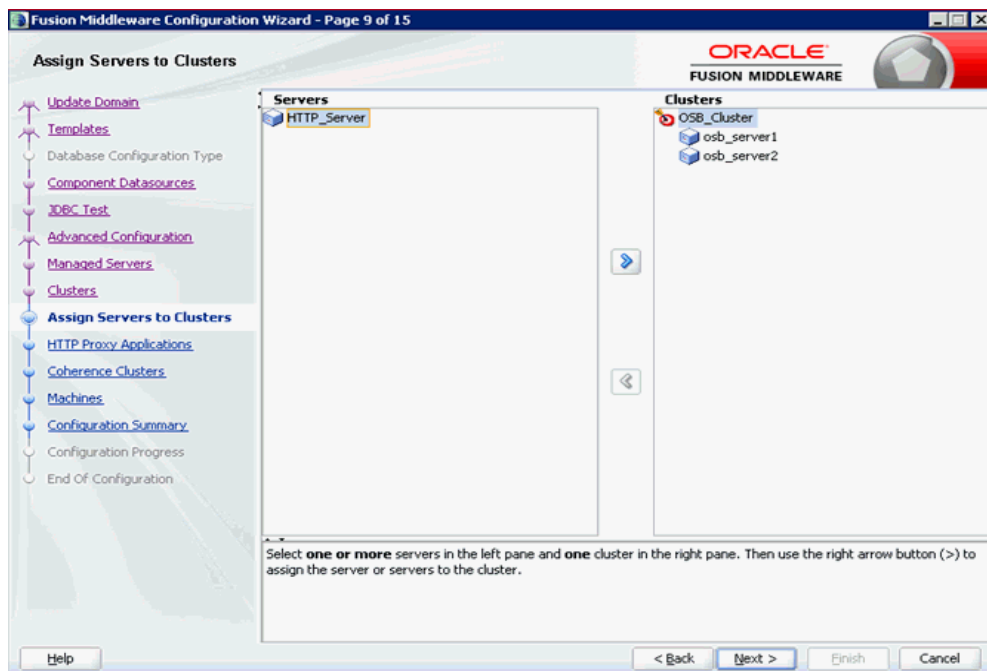
- In the Clusters pane, leave the available cluster, and click **Next**, as shown in Figure 1-10.

Figure 1-10 Clusters Pane



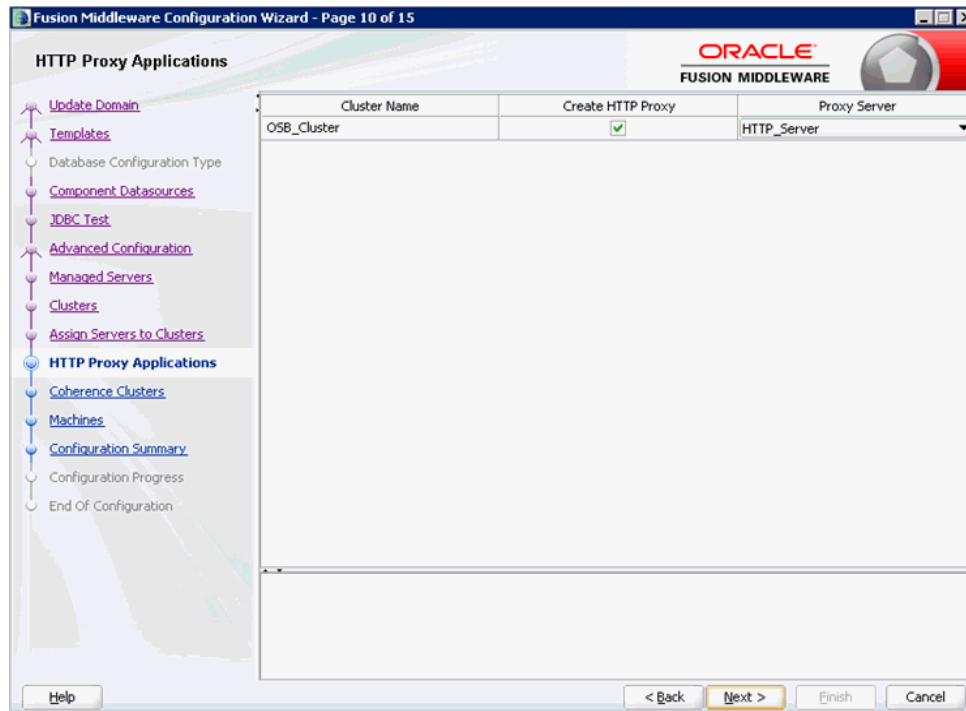
- In the Assign Servers to Clusters pane, make sure that the newly created HTTP_Server is not assigned to the cluster, and click **Next**, as shown in Figure 1-11.

Figure 1-11 Assign Servers to Clusters Pane



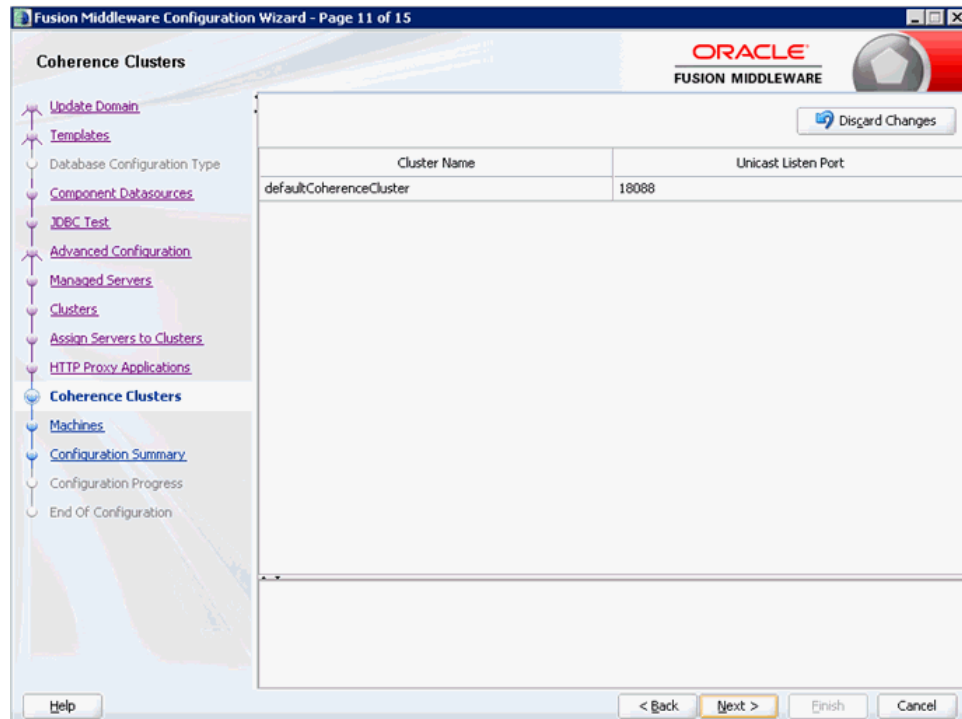
- In the HTTP Proxy Applications pane, select the **Create HTTP Proxy** check box, select **HTTP_Server** from the drop-down list, and then click **Next**, as shown in Figure 1–12.

Figure 1–12 HTTP Proxy Applications Pane



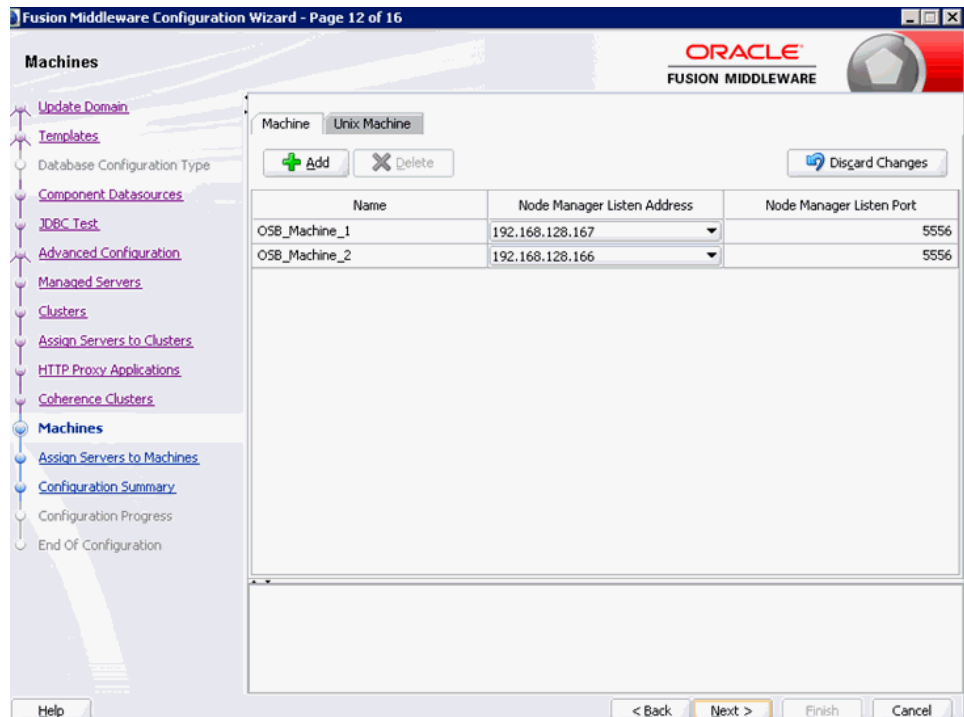
- In the Coherence Clusters pane, leave the default values and click **Next**, as shown in Figure 1–13.

Figure 1–13 Coherence Clusters Pane



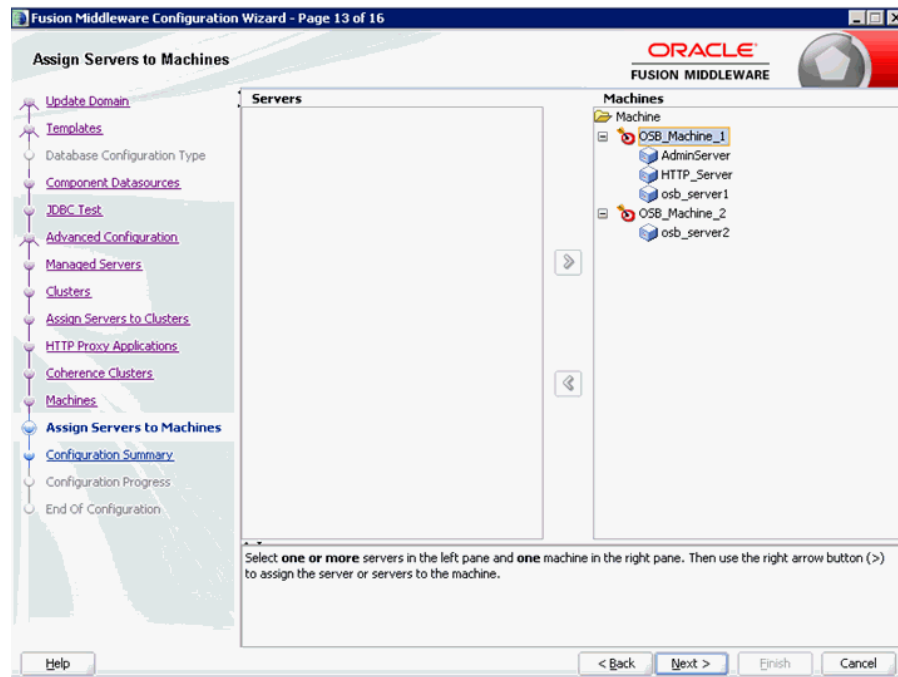
12. In the Machines pane, leave the available machines, and click **Next**, as shown in Figure 1–14.

Figure 1–14 Machines Pane



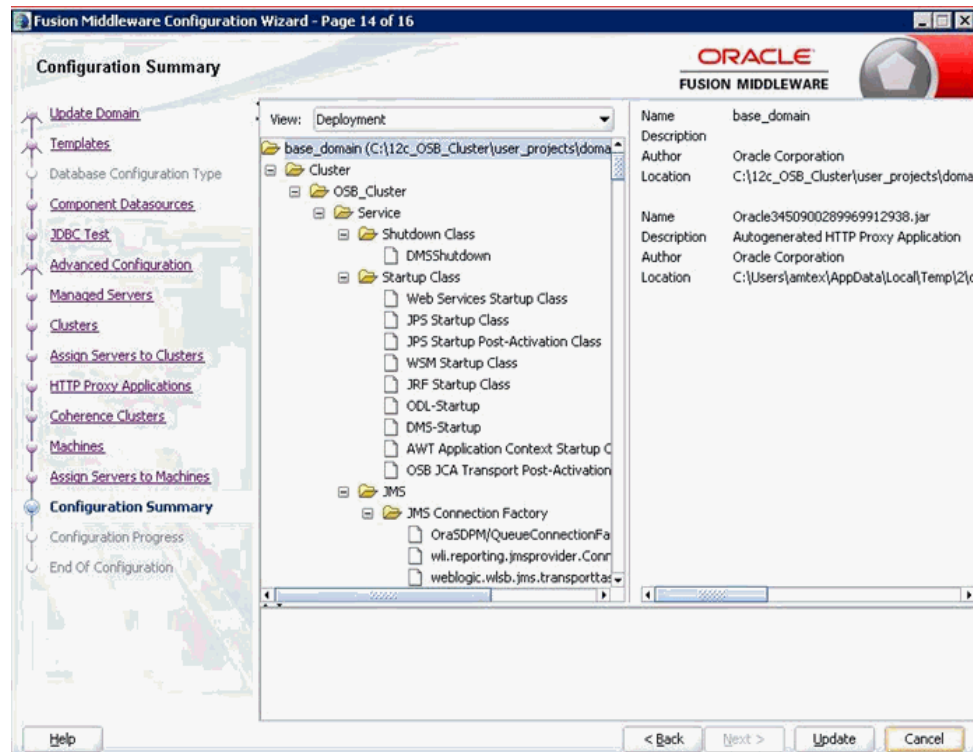
13. In the Assign Servers to Machines pane, assign the created HTTP server to the first node (where Admin Server is available), and click **Next**, as shown in Figure 1–15.

Figure 1–15 Assign Servers to Machines Pane



14. In the Confirmation Summary pane, check the summary and click **Update**, as shown in Figure 1–16.

Figure 1–16 Configuration Summary Pane



1.2.3.1 Deploying the iBSE configuration

To deploy `ibse.war`, repeat steps 5 through 19 from the J2CA deployment steps given in [Section 1.2.2, "Deploying Oracle Application Adapters"](#) on page 1-5 with the following changes:

- Select `ibse.war` in step 8.
- Select the created `HTTP_Server` at step 10 (do not select any other servers).

1.2.3.2 Configuring a BSE Configuration in Application Explorer

Perform the following steps to configure a BSE configuration in Application Explorer.

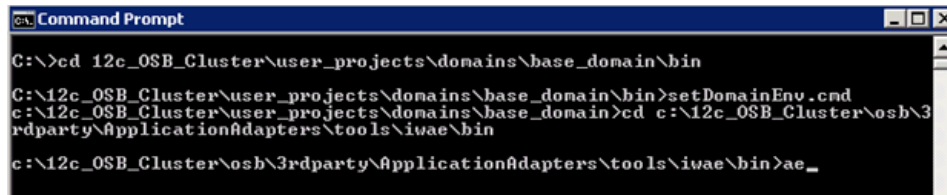
1. Open a terminal and navigate to the following directory, and execute either `setDomainEnv.sh` (on UNIX platforms) or `setDomainEnv.cmd` (on Windows platforms):

```
<ORACLE_HOME>\user_projects\domains\created_cluster_domain\bin
```

2. In the same terminal, as shown in [Figure 1-17](#), navigate to the following directory and execute `ae.exe` (on Windows platforms) or `iwae.sh` (on UNIX platforms):

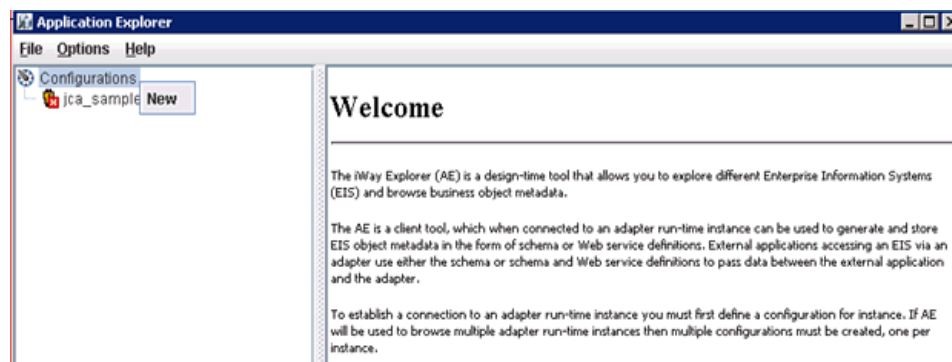
```
<ADAPTER_HOME>\tools\iwae\bin
```

Figure 1-17 Command Prompt



3. Right-click **Configurations**, and select **New**, as shown in [Figure 1-18](#).

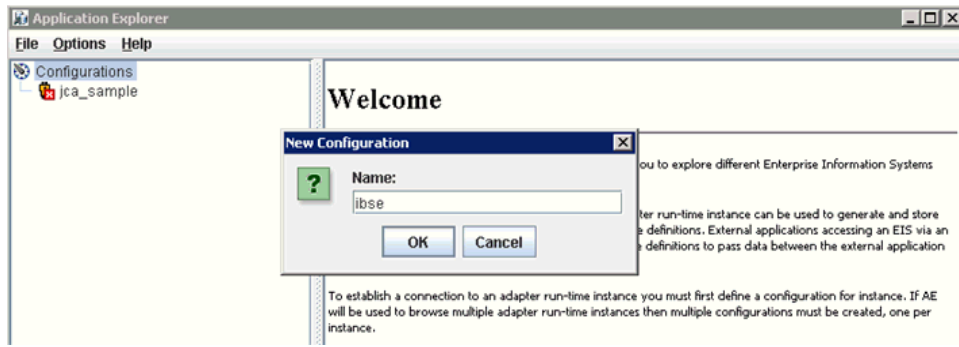
Figure 1-18 Application Explorer



The New Configuration dialog is displayed.

4. Enter a name for the new configuration (for example, `ibse`) and then click **OK**, as shown in [Figure 1-19](#).

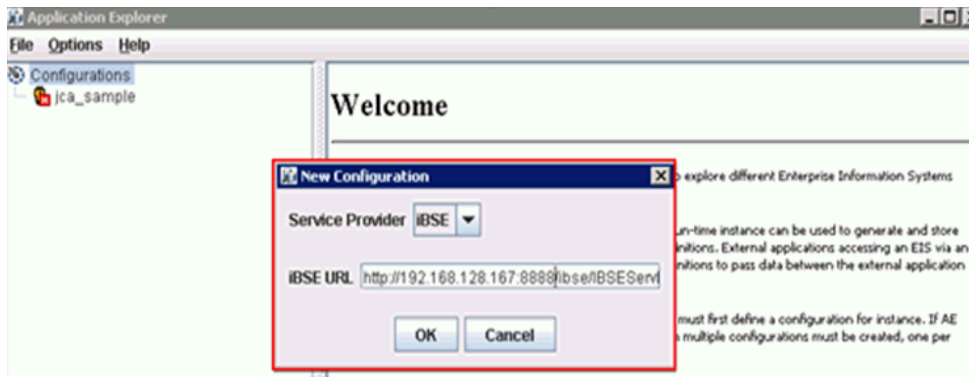
Figure 1–19 New Configuration Dialog



Note: The name of the BSE configuration that is specified here is used during the BSE deployment process.

5. From the Service Provider list, select **ibSE**.
6. In the **ibSE URL** field, accept the default URL or replace it with a different URL with the following format and click **OK**, as shown in [Figure 1–20](#).

Figure 1–20 New Configuration Dialog



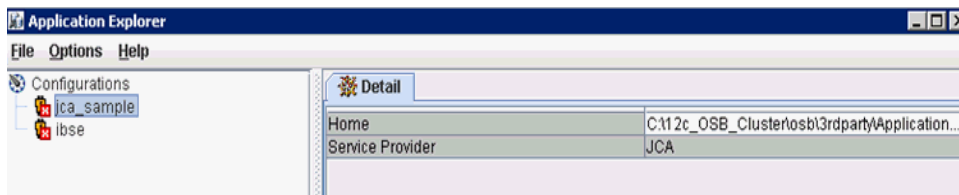
`http://host name:port/ibse/IBSEServlet`

where:

- *host name* - Is the system where the HTTP server resides.
- *port* - Is the HTTP port number where the HTTP server is listening.

A node representing the new configuration is displayed beneath the root Configurations node, as shown in [Figure 1–21](#).

Figure 1–21 New Configuration



Note: 1. Configure iBSE for database using the iBSE config page at:

```
http:// HTTP_PROXY_SERVER_IP:HTTP_PROXY_Server_PORT /ibse
```

For more information, see the *Installation Guide for Oracle WebLogic Server*.

2. Restart the servers.

3. Use the IP and port of the HTTP Proxy server to access ibse. For example:

```
http://HTTP_PROXY_SERVER_IP:HTTP_PROXY_Server_
PORT/ibse/IBSEServlet
```

1.2.4 Key Points to Consider

The outbound process configuration is identical for all Oracle Application Adapters for Oracle WebLogic Server (SAP R/3, Siebel, PeopleSoft, and J.D. Edwards).

For more information, see [Section 1.2.5, "Configuring the High Availability \(HA\) File Adapter for an Outbound Process"](#) on page 1-17 and [Section 1.2.6, "Configuring the Outbound Process"](#) on page 1-22.

For inbound processing, the steps that are described in [Section 1.2.7, "Configuring Oracle HTTP Server for the Inbound Process"](#) on page 1-23 must be followed only for Siebel, PeopleSoft, and J.D. Edwards adapters, and not for the SAP R/3 adapter. For PeopleSoft and J.D. E adapters, only the HTTP listener must be used in High Availability (HA) environments. The TCP listener should not be used.

1.2.5 Configuring the High Availability (HA) File Adapter for an Outbound Process

The following configuration steps must be performed in the Oracle WebLogic Server Administration Console to work with the outbound process:

1. Open the Oracle WebLogic Server Administration Console in a Web browser by entering the following URL:

```
http://host name:port/console
```

Where *host name* is the name of the system where Oracle WebLogic Server is running and *port* is the port number for the domain you are using.

The Oracle WebLogic Server Administration Console logon page is displayed.

2. Log in to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.

The Oracle WebLogic Server Administration Console home page is displayed.

3. In the Domain Structure section in the left pane, click **Deployments**.

The Deployments page is displayed, as shown in [Figure 1-22](#).

Figure 1–22 Deployments Page

Name	State	Health	Type	Deployment Order
adf.oracle.domain(1.0,11.1.1.2.0)	Active		Library	100
adf.oracle.domain.webapp(1.0,11.1.1.2.0)	Active		Library	100
AqAdapter	Active	OK	Resource Adapter	324
b2bui	Active	OK	Enterprise Application	313
composer	Active	OK	Enterprise Application	315
DbAdapter	Active	OK	Resource Adapter	322
DefaultToDoTaskFlow	Active	OK	Enterprise Application	314
DMS Application (11.1.1.1.0)	Active	OK	Web Application	5
em	Active	OK	Enterprise Application	400
emai	Active		Library	100
emas	Active		Library	100
emcore	Active		Library	100
FileAdapter	Active	OK	Resource Adapter	321
FMW Welcome Page Application (11.1.0.0.0)	Active	OK	Enterprise Application	5
FtpAdapter	Active	OK	Resource Adapter	325

- Click FileAdapter.

The Settings for FileAdapter page is displayed, as shown in Figure 1–23.

Figure 1–23 Settings for File Adapter Page

Settings for FileAdapter

Overview | Deployment Plan | **Configuration** | Security | Targets | Control | Testing | Monitoring | Notes

General | Properties | **Outbound Connection Pools** | Admin Objects | Workload | Instrumentation

This page displays a table of Outbound Connection Pool groups and instances for this resource adapter. The top level entries in the table represent Outbound Connection Pool groups. Groups are listed by connection factory interface and the instances are listed by their JNDI names. Expand a group to obtain configuration information for a Connection Pool instance within an Outbound Connection Pool group. Click the name of a group or instance to configure it. Automatically generated Connection Pools are not displayed in the table below.

Outbound Connection Pool Configuration Table

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

New | Delete | Showing 1 to 1 of 1 | Previous | Next

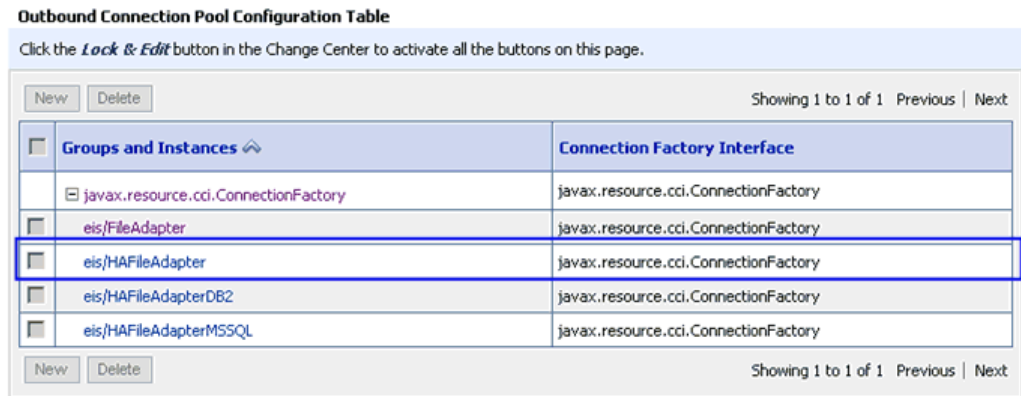
Groups and Instances	Connection Factory Interface
javax.resource.cci.ConnectionFactory	javax.resource.cci.ConnectionFactory

New | Delete | Showing 1 to 1 of 1 | Previous | Next

- Click the Configuration tab followed by the Outbound Connection Pools tab.

The Outbound Connection Pool Configuration Table page is displayed, as shown in Figure 1–24.

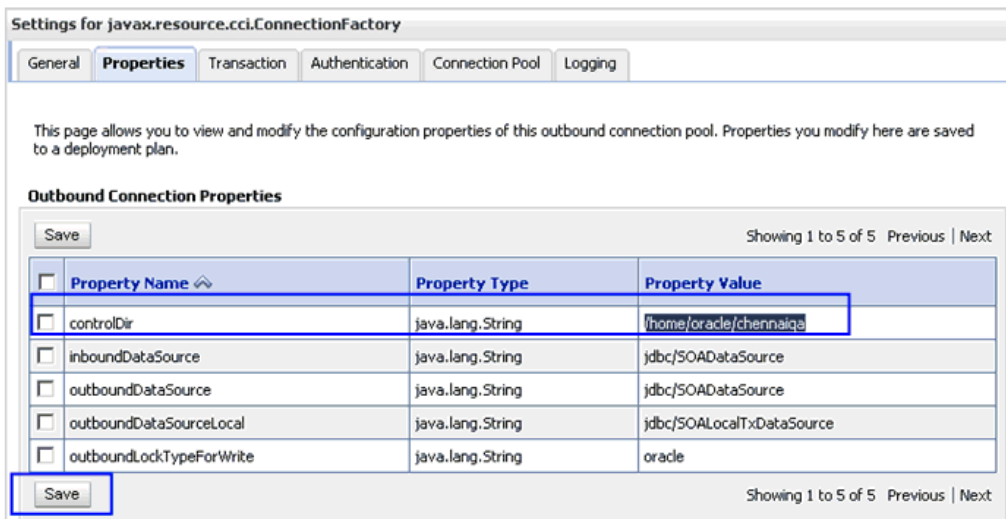
Figure 1–24 Outbound Connection Pool Configuration Table Page



- Expand `javax.resource.cci.ConnectionFactory` and click `eis/HAFileAdapter`.

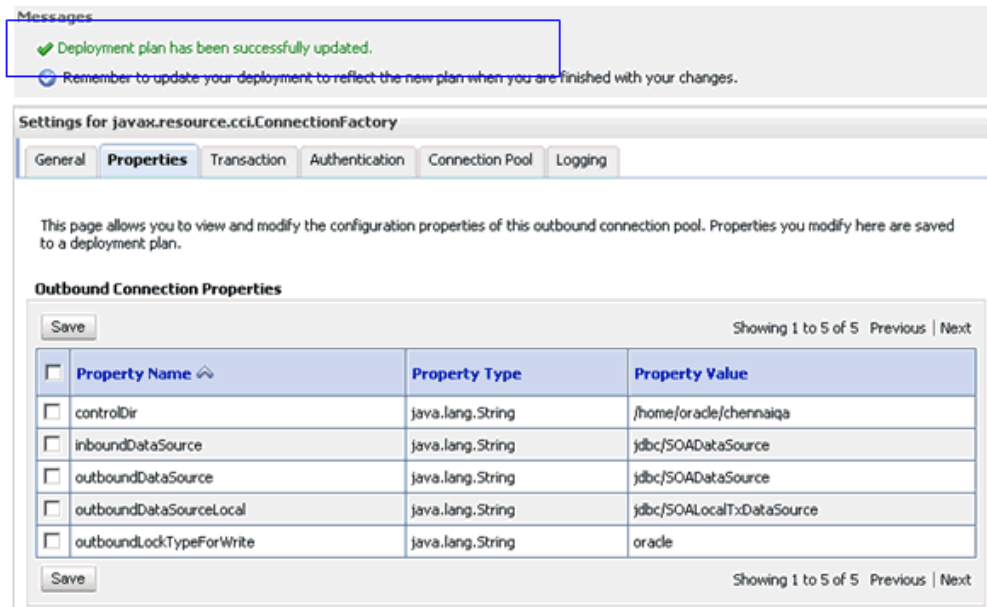
The settings for `javax.resource.cci.ConnectionFactory` page is displayed, as shown in [Figure 1–25](#).

Figure 1–25 Javax.resource.cci.ConnectionFactory Page



- Click the **Properties** tab.
- Provide a valid location for the `controlDir` property and click **Save**.
- Check if the success message is displayed, as shown below in [Figure 1–26](#).

Figure 1–26 Deployment Plan Success Message



10. In the Domain Structure section in the left pane, click **Deployments**, as shown in Figure 1–27.

Figure 1–27 Deployments Page



11. Select **FileAdapter** from the deployments list and click **Update**.
The Update Application Assistant page is displayed, as shown in Figure 1–28.

Figure 1–28 Update Application Assistant Page

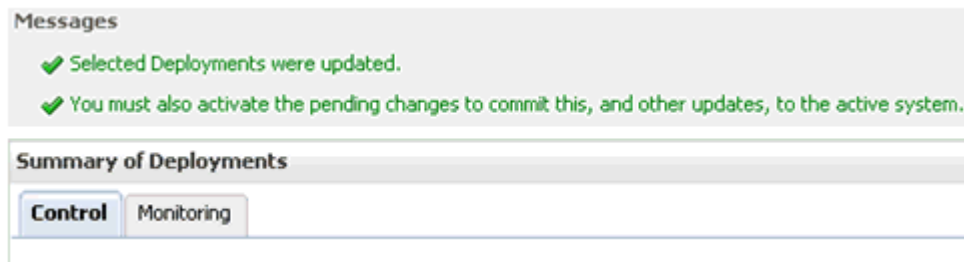
12. Select **Update this application in place with new deployment plan changes** and click **Next**.

The Summary page is displayed, as shown in [Figure 1–29](#).

Figure 1–29 Summary Page

13. Click **Finish**.
14. Check if the success message is displayed, as shown in [Figure 1–30](#).

Figure 1–30 Deployment Success Message



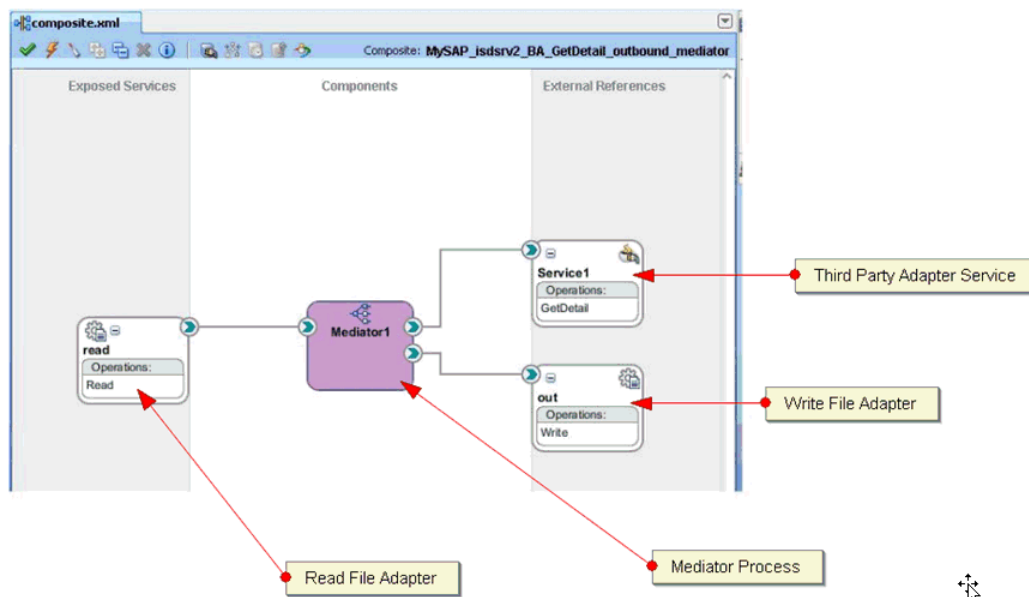
15. Copy the generated **plan.xml** file from the first system (first node in the cluster configuration) to the same location (Oracle_Home\Middleware\Oracle_SOA1\soa) in the second system (second node in the cluster configuration).
16. Restart the servers (Admin_Server, SOA_Server1, SOA_Server2).
17. Create a shared input location that is accessible by both systems.
For example, if the inputs are pasted in this location, they can be consumed by system1 and system2.

1.2.6 Configuring the Outbound Process

To configure the outbound process:

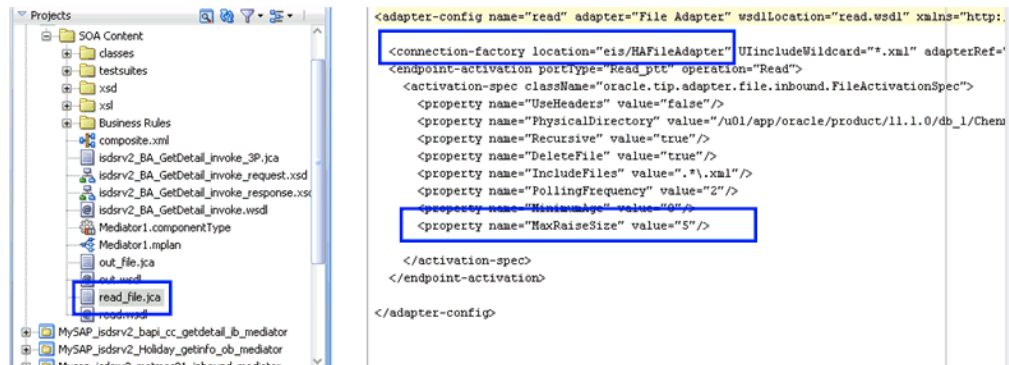
1. Create a target using Application Explorer on the first system.
2. Connect to the target (on the first system) and create an outbound WSDL for the Oracle Application Adapter for SAP R/3 (MySAP node).
3. On the second system, create a target using Application Explorer with the same name as specified on the first system.
4. Restart the Admin Server and the soa_servers on both systems.
5. As shown below in [Figure 1–31](#), create an outbound Mediator process using Oracle JDeveloper with the following configuration:

Figure 1–31 Outbound Mediator Process in Oracle JDeveloper



6. Select the `Read_file_adapter.jca` file and make the following changes:
 - a. Change `<connection-factory location="eis/FileAdapter" UIincludeWildcard="*.xml" adapterRef="" />` to `<connection-factory location="eis/HFileAdapter" UIincludeWildcard="*.xml" adapterRef="" />`
 - b. Add the value `<property name="MaxRaiseSize" value="5" />` to the end, as shown in [Figure 1–32](#).

Figure 1–32 JCA Properties Source



7. Save the process and deploy the process to both SOA servers.
8. Provide the input files in the input location configured in the Read file adapter and check if the outputs are shared and placed in the output location (in two systems) configured in the Write file adapter.

1.2.7 Configuring Oracle HTTP Server for the Inbound Process

The following configuration steps must be completed (for PeopleSoft, Siebel, and J.D. Edwards OneWorld Application Adapters) before executing an inbound process:

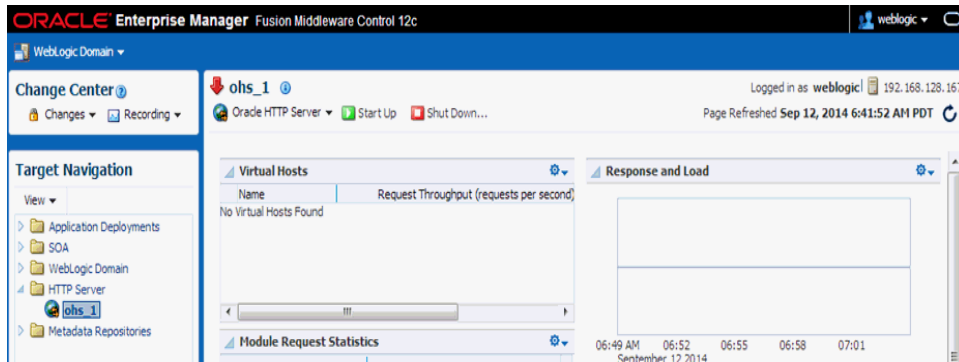
1. Open the Oracle EM Console in a web browser by entering the following URL:

`http://host name:port/em`

Where:

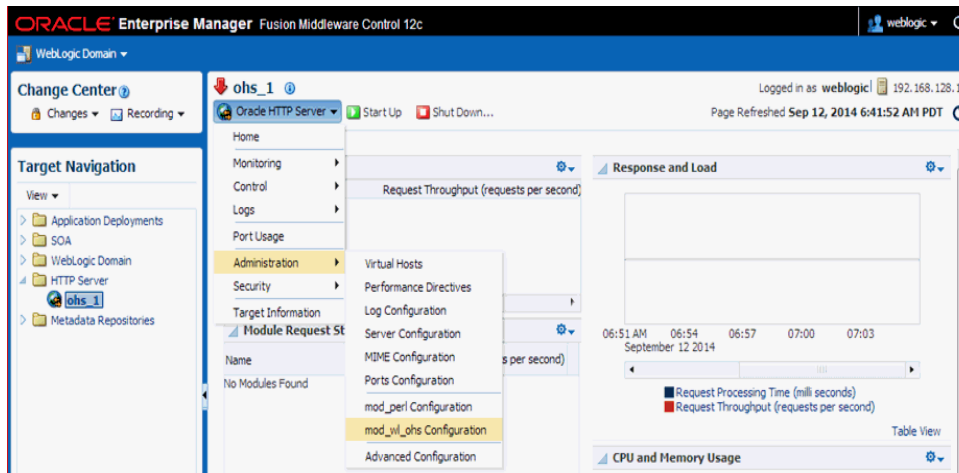
- `host name` - Is the name of the system where Oracle WebLogic Server is running.
 - `port` - Is the port number for the domain you are using.
2. Log in to the Oracle EM Console using an account that has administrator privileges.
 3. On the left pane, expand **HTTP Server** and select the created OHS instance, as shown in [Figure 1–33](#).

Figure 1–33 OHS Instance



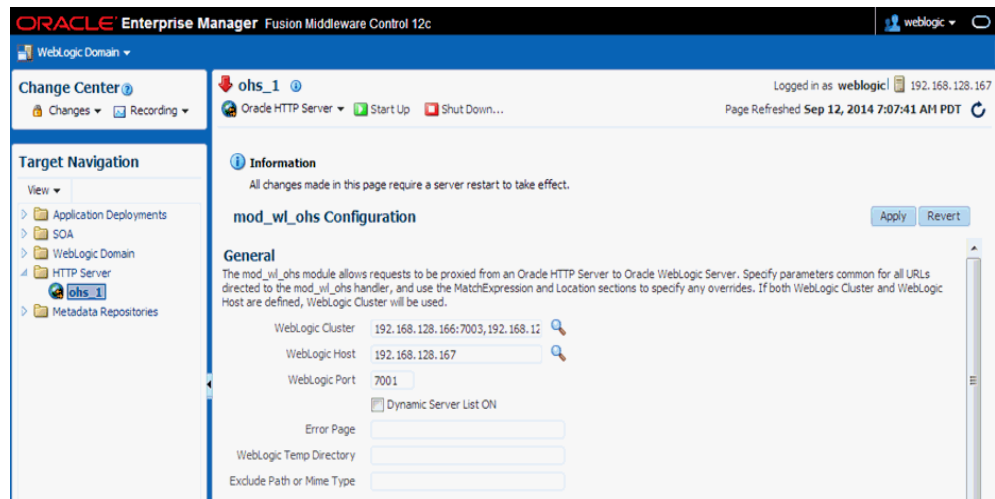
4. On the right pane, click the **Oracle HTTP Server** drop-down list, select **Administration**, and then click **mod_wl_ohs Configuration**, as shown in Figure 1–34.

Figure 1–34 OHS Configuration



5. In the Change Center, click **Create**, and then select **Lock & Edit**.
6. In the top of the screen, under the General configuration, select the created WebLogic Cluster and WebLogic Host using the Browse button, and then provide the WebLogic port (if not done already), as shown in Figure 1–35.

Figure 1–35 General Configuration Pane



7. In the bottom of the screen, click **Add** and provide the following parameters:

- Location - *Siebel (Any appropriate name)*
- WebLogic Cluster - *system1_ip:port, system2_ip:port*

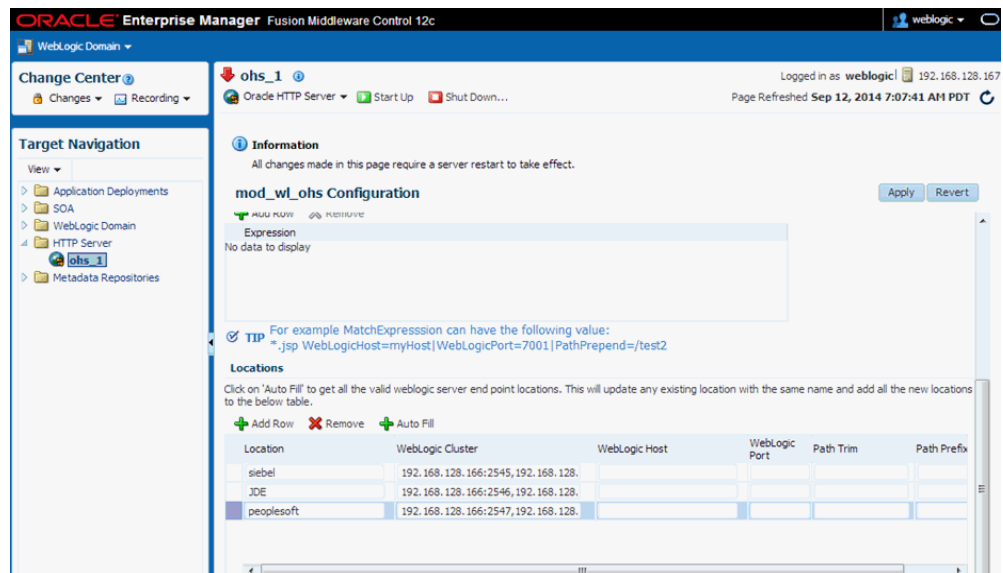
Where:

- Location - Is any appropriate name.
- *system1_ip* and *system2_ip* - Are the system IP addresses where SOA or OSB managed servers are configured.
- *port* - The port number that is configured in the channel configuration for system1 and system2.

8. In the right pane, click **Apply**, and then in the Change Center, click **Changes**, and select **Activate Changes**.

9. Restart the OHS server, as shown in Figure 1–36.

Figure 1–36 OHS Server

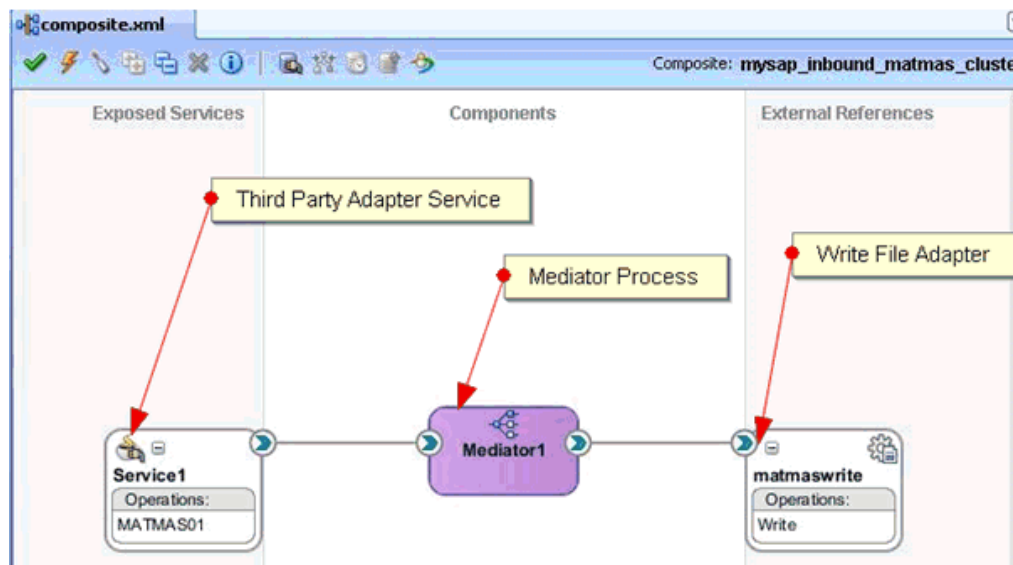


1.2.8 Configuring the Inbound Process

To configure the inbound process:

1. Create a target and channel using Application Explorer on the first system.
2. Connect to the target (on the first system) and create an inbound WSDL for the Oracle Application Adapter for SAP R/3 (MySAP node).
3. On the second system, create a target and channel using Application Explorer with the same name as specified on the first system.
4. Restart the Admin Server and the soa_servers on both systems.
5. As shown in [Figure 1–37](#), create an inbound Mediator process using Oracle JDeveloper with the following configuration:

Figure 1–37 Inbound Mediator Process in Oracle JDeveloper

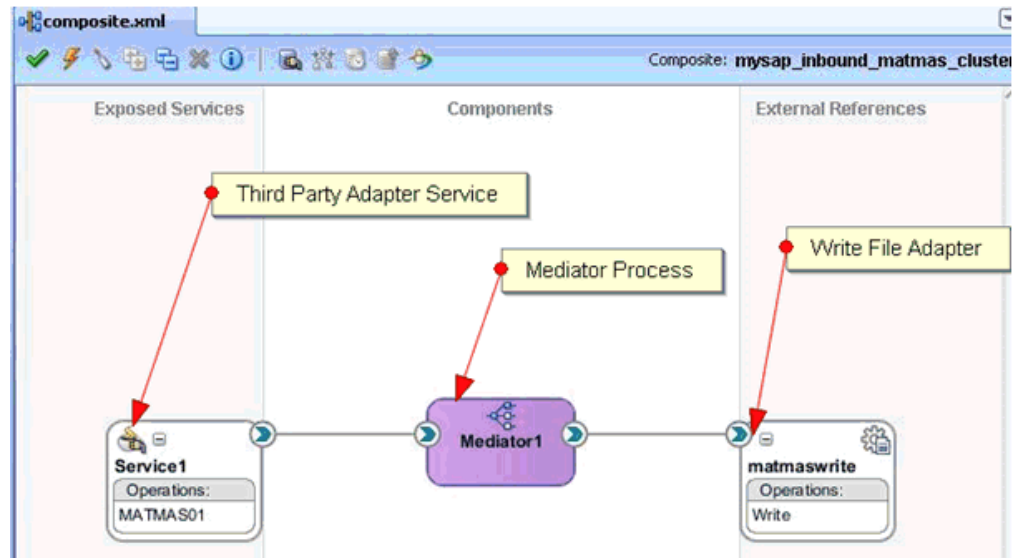


6. Save the process and deploy the process (to both SOA servers).
7. Trigger from the back end or use HTTP publisher and check if the output is shared and placed in the output locations (in both systems) that are configured in the Write File adapter.

1.2.9 Singleton Testing

To perform Singleton testing:

1. On the first system, create a target and channel using Application Explorer.
2. Connect to the target (on the first system) and create an inbound WSDL for the Oracle Application Adapter for SAP R/3 (MySAP node).
3. On the second system, create a target and channel using Application Explorer with the same name as specified on the first system.
4. Restart the Admin Server and the soa_servers on both systems.
5. As shown in [Figure 1–38](#), create an inbound Mediator process using Oracle JDeveloper with the following configuration:

Figure 1–38 Inbound Mediator Process in Oracle JDeveloper

- For singleton testing, open the **composite.xml** file (source view) and add the following property in the Service section:

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

For example:

```
<service name="Service1" ui:wSDLLocation="MATMAS01_receive_cluster.wsdl">
<interface.wSDL
interface="http://xmlns.oracle.com/pcbpel/iWay/wSDL/MySAP/isdsrv2_
cluster/MATMAS01#wsdl.interface(MATMAS01PortType)"/>
<binding.jca config="MATMAS01_receive_cluster_3P.jca"/>
<property name="singleton">true</property>
</service>
```

- Save the inbound Mediator process and deploy the process to both SOA servers.
- Trigger from the back end or use HTTP publisher and check if the messages are received by any system in the output location.
- Stop the `soa_server` of the system that is receiving the messages.
- Check whether the messages are being received by the second system.

When you are using the Singleton testing feature with Oracle Application Adapter for SAP R/3, there are some loss of messages (messages are dumped in the SAP GUI). This is caused because the end-point activation of the second system is in progress after the first system is down.

To retrieve the lost messages, perform the following steps:

- Login to the SAP GUI.
- Enter the `/sm58` transaction and navigate to the dumped messages.
- Right-click a dumped message, and then select **Execute LUW**.
- Repeat step 3 for all the dumped messages.

All the lost messages are received in the Oracle Enterprise Manager console.

1.3 Preferred Repository Type

As a best practice, it is recommended to use only a database repository (for example, Oracle) for adapters in development, test, and production environments. Do not use the File repository, which is provided by default only for initial startup purposes. The File repository is not supported for troubleshooting any issues.

1.4 Generating Input XML Documents

This section describes how to generate input XML documents that can be used as payloads for outbound BPEL and Mediator processes. It includes the following topics:

- [Section 1.4.1, "Prerequisites"](#)
- [Section 1.4.2, "Creating a WSDL Document Using Application Explorer"](#)
- [Section 1.4.3, "Generating an Input XML File From a Request Schema"](#)
- [Section 1.4.4, "Using the Generated Input XML File in the Oracle Enterprise Manager Console"](#)

1.4.1 Prerequisites

Before continuing, ensure that the following components and applications are available:

- Outbound WSDL document created using Application Explorer.
- XML editor (for example, Oracle JDeveloper or Altova XML Spy, which is used as an example in this section).
- Oracle WebLogic Server 12c (12.2.1.0.0) with Oracle JDeveloper.

1.4.2 Creating a WSDL Document Using Application Explorer

To create a WSDL document using Application Explorer:

1. Ensure that Oracle WebLogic Server is started, which is where Application Explorer is deployed.
2. Open a command prompt window and navigate to the following directory:
`<ORACLE_HOME>\user_projects\domains\base_domain\bin`
3. Run `setDomainEnv.cmd` (Windows) or `./setDomainEnv.sh` (UNIX/Linux).
The `setDomainEnv` command sets the class path and other environment variables for Application Explorer in the Oracle WebLogic Server environment.
4. Do not close the command prompt window.
5. Navigate to the following directory:
`<ADAPTER_HOME>\tools\iwae\bin`
6. Run `ae.bat` (Windows) or `iwae.sh` (UNIX/Linux) to start Application Explorer.
7. Select an available J2CA configuration.
8. Select an appropriate adapter (for example, MySAP).
9. Create a new target or connect to an existing target.
10. Expand the created target adapter and select the appropriate object.

11. Right-click the object and select **Create Outbound JCA Service(Request/Response)**.

The Export WSDL dialog is displayed.

12. Accept the default location in the Name field and click **OK** to export the WSDL document to the default location.

You can also click **Browse** to provide a different location and then click **OK**, which exports the WSDL document to your defined location.

13. Navigate to the location where the WSDL document was exported and verify that the WSDL, JCA, Request, and Response schema files are exported and available.

1.4.3 Generating an Input XML File From a Request Schema

To generate an input XML file from a request schema:

1. Open an XML editor, as shown in [Figure 1–39](#) (for example, Altova XML Spy, which is used as an example in this section).

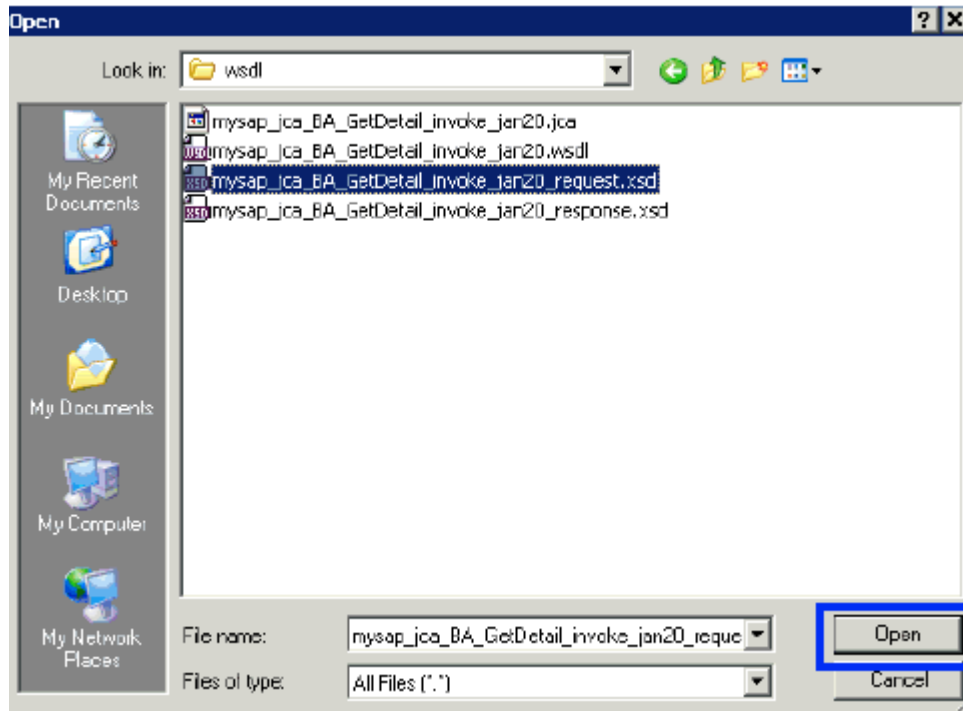
Figure 1–39 *The Open Option*



2. Click **File**, and then select **Open**.

The Open dialog is displayed, as shown in [Figure 1–40](#).

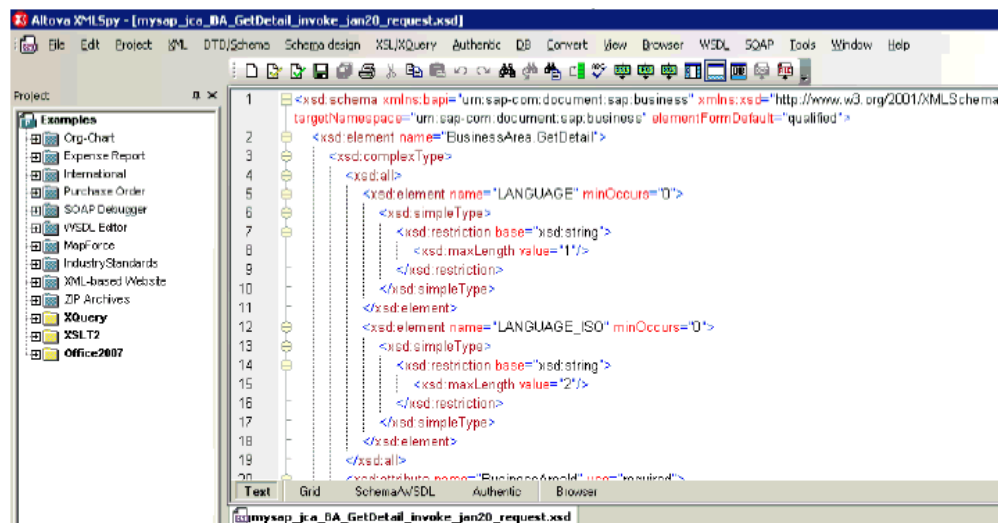
Figure 1–40 The Open Dialog



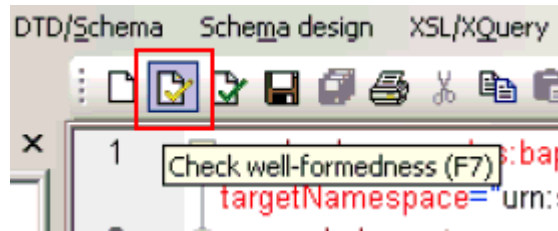
3. Navigate to the location on your file system where the XML request schema is exported, select the schema file and click **Open**.

The XML request schema file is opened and displayed in Altova XML Spy, as shown in [Figure 1–41](#).

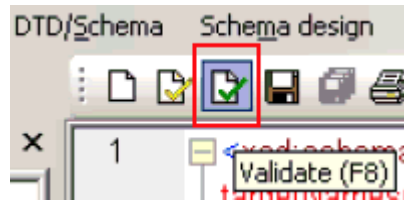
Figure 1–41 XML Request Schema File Displayed in Altova XML Spy



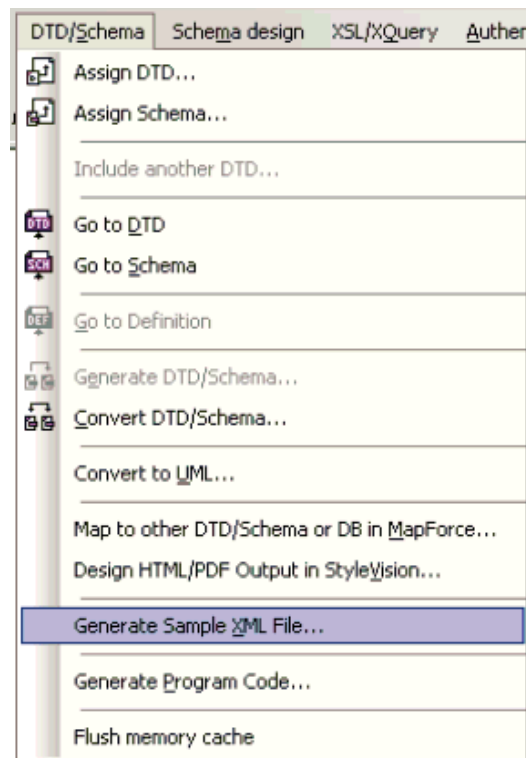
4. Check if the schema is well formed by clicking **Check well-formedness** or pressing **F7**, as shown in [Figure 1–42](#).

Figure 1–42 Check well-formedness Icon

5. Validate the schema by clicking **Validate** or pressing **F8**, as shown in [Figure 1–43](#).

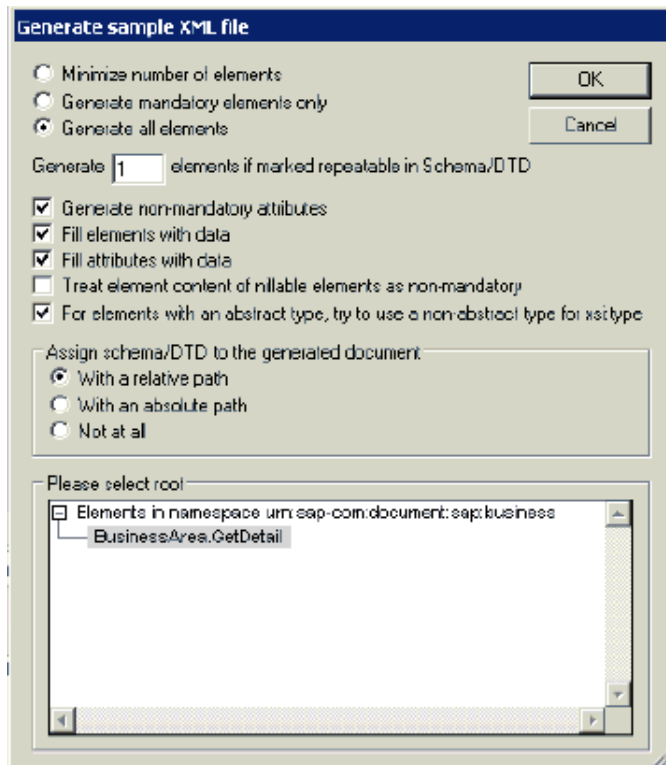
Figure 1–43 The Validate Icon

6. Once you have confirmed that the schema is well-formed and valid, click **DTD/Schema**, and then select **Generate Sample XML File**, as shown in [Figure 1–44](#).

Figure 1–44 Generate Sample XML File Option

As shown in [Figure 1–45](#), the Generate Sample XML File dialog is displayed.

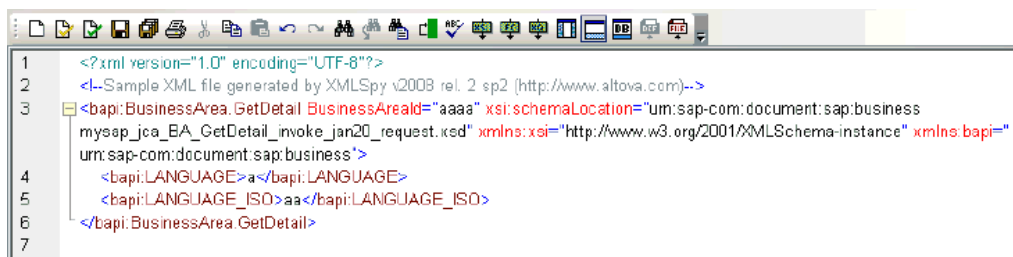
Figure 1–45 Generate Sample XML File Dialog



7. Select the appropriate parameters for your sample XML file and click **OK** when you are ready.

As shown in [Figure 1–46](#), the sample XML file is generated in Altova XML Spy.

Figure 1–46 Sample XML File Generated in Altova XML Spy



8. Verify that the generated input XML file is well-formed and valid.

You can use the generated input XML file to invoke a BPM, BPEL, or Mediator process after providing the required values in the file.

1.4.4 Using the Generated Input XML File in the Oracle Enterprise Manager Console

Before you can use the generated input XML file in the Oracle Enterprise Manager console, verify that the following prerequisites are available:

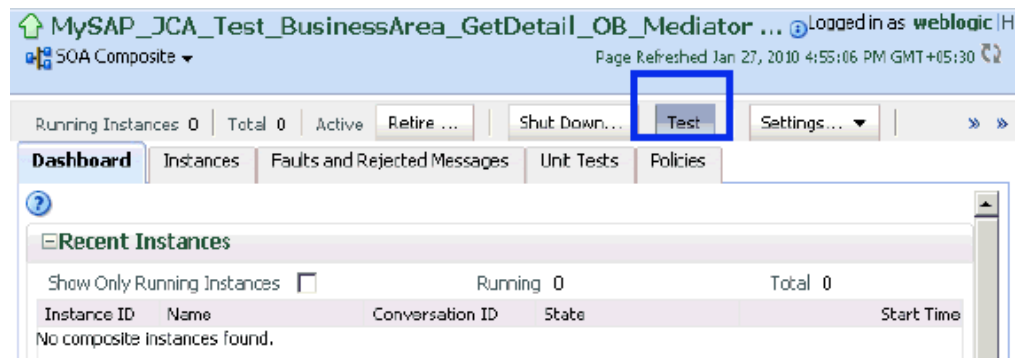
- BPM Outbound process created in JDeveloper and deployed in SOA_Server1
- Mediator Outbound process created in JDeveloper and deployed in SOA_Server1
- BPEL Outbound process created in JDeveloper and deployed in SOA_Server1

Mediator Outbound Process

Perform the following steps to use the input XML file in an outbound Mediator process:

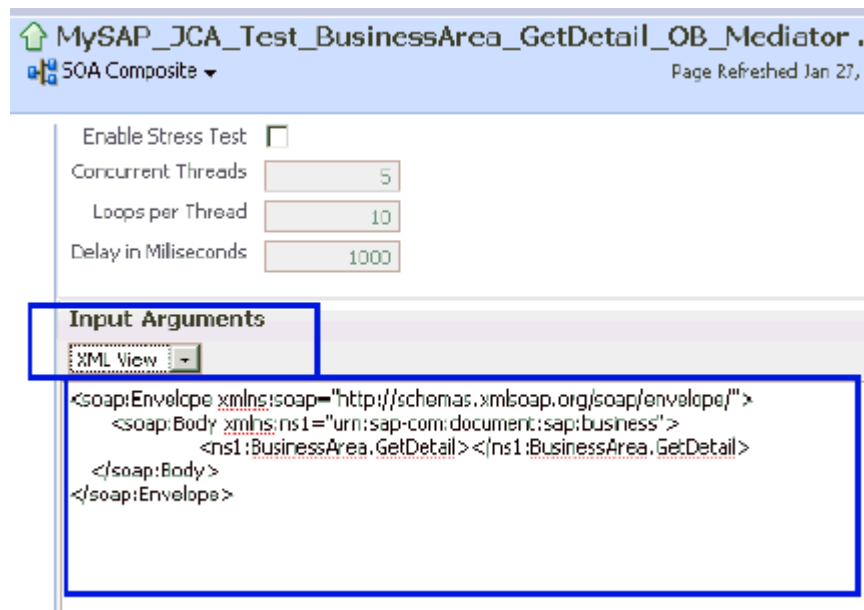
1. Log in to the Oracle Enterprise Manager console by using the following URL:
http://localhost:7001/em
2. In the left pane, expand **SOA**, **soa-infra (soa_server_name)**, and then **default**.
3. Select an outbound Mediator project (for example, MySAP_JCA_Test_BusinessArea_GetDetail_OB_Mediator).
4. Click the **Test** button on the top right-hand corner of the console, as shown in [Figure 1-47](#).

Figure 1-47 Test Button



5. In the Input Arguments section, select **XML View** from the list and verify that the input XML is displayed, as shown in [Figure 1-48](#).

Figure 1-48 XML View Option in Input Arguments Section



Note: For Mediator processes, it is mandatory for the input XML to be used with a namespace.

6. The displayed XML in the Oracle Enterprise Manager console can be altered and used as follows:

XML Displayed in the Oracle Enterprise Manager Console

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
<ns1:BusinessArea.GetDetail> </ns1:BusinessArea.GetDetail> </soap:Body>
</soap:Envelope>
```

Altered XML

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
<ns1:BusinessArea.GetDetail BusinessAreaId="1000">
</ns1:BusinessArea.GetDetail> </soap:Body> </soap:Envelope>
```

7. The sample input XML that was generated using Altova XML Spy can be used after making the following required modifications:
 - a. Remove XML headers.
 - b. Add necessary inputs.
 - c. Add the soap headers as displayed in the Oracle Enterprise Manager console.
 - d. Modify the namespaces to match the namespaces in the input XML.

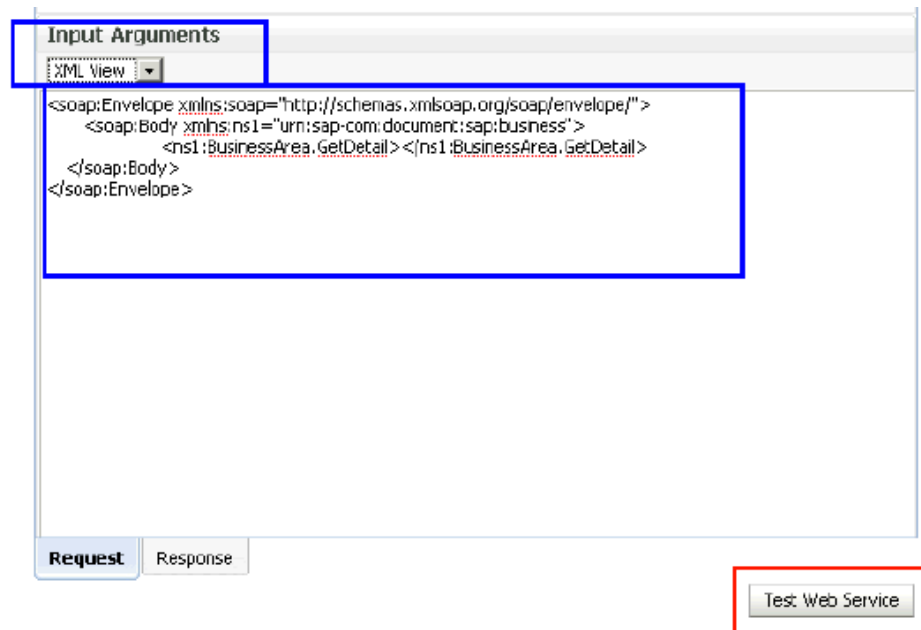
XML Generated Using Altova XML Spy

```
<?xml version="1.0" encoding="UTF-8"?> <!--Sample XML file generated by XMLSpy
v2008 rel. 2 sp2 (http://www.altova.com)--> <bapi:BusinessArea.GetDetail
BusinessAreaId="aaaa" xsi:schemaLocation="urn:sap-com:document:sap:business
mysap_jca_BA_GetDetail_invoke_jan20_request.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:bapi="urn:sap-com:document:sap:business">
<bapi:LANGUAGE>a</bapi:LANGUAGE> <bapi:LANGUAGE_ISO>aa</bapi:LANGUAGE_ISO>
</bapi:BusinessArea.GetDetail>
```

Altered XML

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
<ns1:BusinessArea.GetDetail BusinessAreaId="1000"> <ns1:LANGUAGE>D</
ns1:LANGUAGE> < ns1:LANGUAGE_ISO>EN</ ns1:LANGUAGE_ISO>
</ns1:BusinessArea.GetDetail> </soap:Body> </soap:Envelope>
```

8. Click **Test Web Service** after providing the input XML, as shown in [Figure 1–49](#).

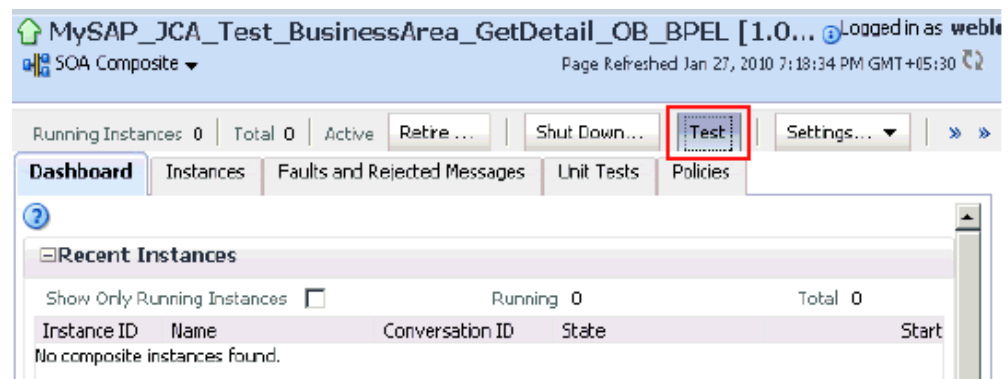
Figure 1–49 Test Web Service Button

The output is displayed in the Response tab.

BPEL Outbound Process

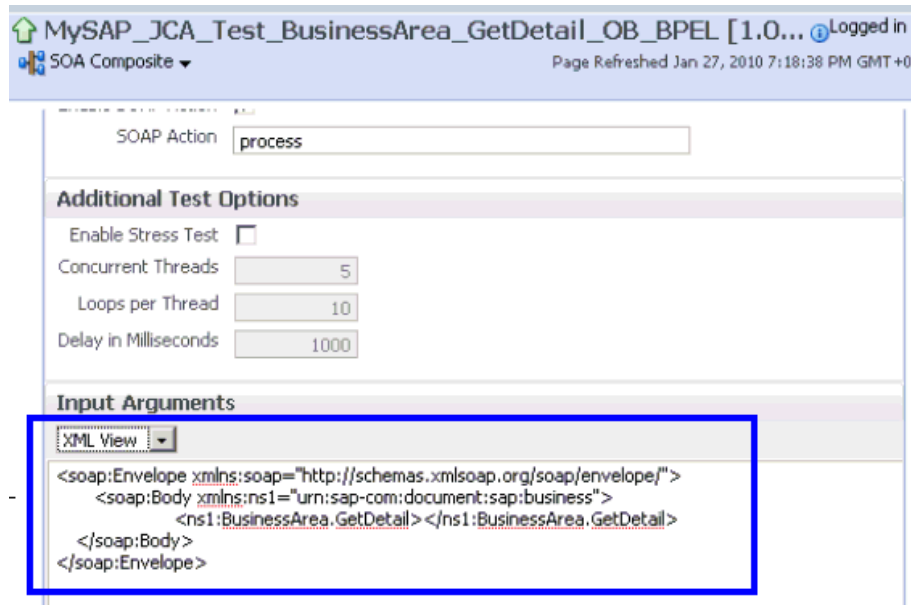
Perform the following steps to use the input XML file in an outbound BPEL process:

1. Log in to the Oracle Enterprise Manager console by using the following URL:
<http://localhost:7001/em>
2. In the left pane, expand **SOA**, **soa-infra (soa_server_name)**, and then **default**.
3. Select an outbound BPEL project (for example, **MySAP_JCA_Test_BusinessArea_GetDetail_OB_BPEL**).
4. Click the **Test** button on the top right-hand corner of the console, as shown in [Figure 1–50](#).

Figure 1–50 Test Button

5. In the Input Arguments section, select **XML View** from the list and verify that the input XML is displayed, as shown in [Figure 1–51](#).

Figure 1–51 Input XML Document Inside Input Arguments Section



Note: For BPEL processes, it is not mandatory for the input XML to be used with a namespace.

6. The displayed XML in the Oracle Enterprise Manager console can be altered and used as follows:

XML Displayed in the Oracle Enterprise Manager Console

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
<ns1:BusinessArea.GetDetail> </ns1:BusinessArea.GetDetail> </soap:Body>
</soap:Envelope>
```

Altered XML With Namespace

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
<ns1:BusinessArea.GetDetail BusinessAreaId="1000">
</ns1:BusinessArea.GetDetail> </soap:Body>
</soap:Envelope>
```

Altered XML Without Namespace

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body xmlns="urn:sap-com:document:sap:business">
<BusinessArea.GetDetail BusinessAreaId="1000">
</BusinessArea.GetDetail> </soap:Body> </soap:Envelope>
```

7. The sample input XML that was generated using Altova XML Spy can be used after making the following required modifications:
 - a. Remove XML headers.
 - b. Add necessary inputs.
 - c. Add the soap headers as displayed in the Oracle Enterprise Manager console.

- d. Modify or remove the namespaces.

XML Displayed in XML View in the Oracle Enterprise Manager Console

```
<?xml version="1.0" encoding="UTF-8"?> <!--Sample XML file generated by XMLSpy
v2008 rel. 2 sp2 (http://www.altova.com)--> <bapi:BusinessArea.GetDetail
BusinessAreaId="aaaa" xsi:schemaLocation="urn:sap-com:document:sap:business
mysap_jca_BA_GetDetail_invoke_jan20_request.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:bapi="urn:sap-com:document:sap:business">
<bapi:LANGUAGE>a</bapi:LANGUAGE> <bapi:LANGUAGE_ISO>aa</bapi:LANGUAGE_ISO>
</bapi:BusinessArea.GetDetail>
```

Altered XML With Namespace

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
<ns1:BusinessArea.GetDetail BusinessAreaId="1000"> <ns1:LANGUAGE>D</
ns1:LANGUAGE> < ns1:LANGUAGE_ISO>EN</ ns1:LANGUAGE_ISO>
</ns1:BusinessArea.GetDetail> </soap:Body> </soap:Envelope>
```

Altered XML Without Namespace

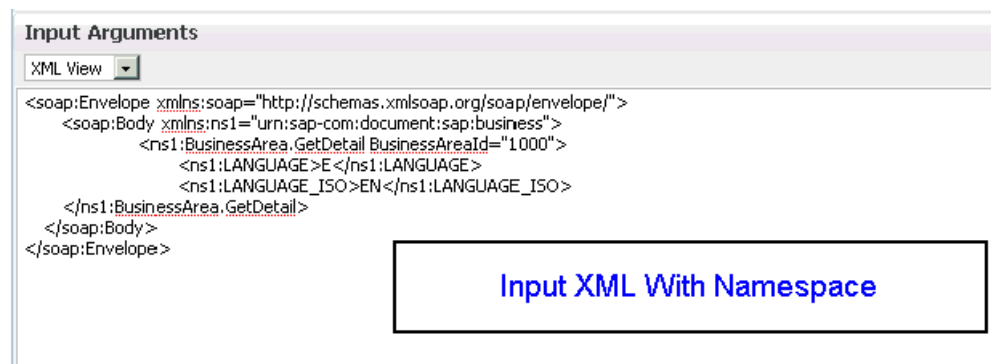
```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body xmlns="urn:sap-com:document:sap:business"> <BusinessArea.GetDetail
BusinessAreaId="1000"> <LANGUAGE>D</ LANGUAGE> < LANGUAGE_ISO>EN</ LANGUAGE_
ISO> </BusinessArea.GetDetail> </soap:Body> </soap:Envelope>
```

8. Click Test Web Service after providing the input XML with or without a namespace.

The output is displayed in the Response tab, as shown in [Figure 1–52](#).

Input XML With Namespace

Figure 1–52 Input XML Document With Namespace



Response

Figure 1–53 Response Tab

Request **Response**

Test Status Passed

Response Time (ms) 890

XML View

Launch Message Flow Trace

```

<BusinessArea.GetDetail.Response xmlns="urn:sap-com:document:sap:business:response">
  <BUSINESSAREA_DETAIL xmlns="">
    <BUS_AREA>1000</BUS_AREA>
    <BUS_AR_DES>Mechanical engineering</BUS_AR_DES>
    <CONS_BA>1000</CONS_BA>
  </BUSINESSAREA_DETAIL>
  <RETURN xmlns="">
    <TYPE/>
    <CODE/>
    <MESSAGE/>
    <LOG_NO/>
    <LOG_MSG_NO>000000</LOG_MSG_NO>
    <MESSAGE_V1/>
    <MESSAGE_V2/>
    <MESSAGE_V3/>
  </RETURN>
</BusinessArea.GetDetail.Response>
    
```

Input XML Without Namespace

Figure 1–54 Input XML Document Without Namespace

Input Arguments

XML View

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns="urn:sap-com:document:sap:business">
    <BusinessArea.GetDetail BusinessAreaId="1000">
      <LANGUAGE>E</LANGUAGE>
      <LANGUAGE_ISO>EN</LANGUAGE_ISO>
    </BusinessArea.GetDetail>
  </soap:Body>
</soap:Envelope>
    
```

Input XML Without Namespace

Response

Figure 1–55 XML Response Document Being Returned

Request **Response**

Test Status Passed

Response Time (ms) 781

XML View

Launch Message Flow Trace

```

<BusinessArea.GetDetail.Response xmlns="urn:sap-com:document:sap:business:response">
  <BUSINESSAREA_DETAIL xmlns="">
    <BUS_AREA>1000</BUS_AREA>
    <BUS_AR_DES>Mechanical engineering</BUS_AR_DES>
    <CONS_BA>1000</CONS_BA>
  </BUSINESSAREA_DETAIL>
  <RETURN xmlns="">
    <TYPE/>
    <CODE/>
    <MESSAGE/>
    <LOG_NO/>
    <LOG_MSG_NO>000000</LOG_MSG_NO>
    <MESSAGE_V1/>
    <MESSAGE_V2/>
    <MESSAGE_V3/>
  </RETURN>
</BusinessArea.GetDetail.Response>

```

Using the Input XML File in an Outbound BPM Process

Perform the following steps to use the input XML file in an outbound BPM process:

1. Log in to the Oracle Enterprise Manager Console by using the following URL:
`http://localhost:7001/em`
2. In the left pane, expand **SOA**, **soa-infra** (**soa_server_name**), and then **default**.
3. Select an outbound BPM project (for example, **SAP2_isdsrv2_BA_GD_jca_ob**).
4. Click **Test** on the top right-hand corner of the console, as shown in [Figure 1–56](#).

Figure 1–56 Test Button

sap2_isdsrv2_BA_GD_jca_ob [1.0] ⓘ

SOA Composite

Page Refreshed Sep 23, 2010 8:02:51 PM GMT+05:30

Running Instances 0 | Total 0 | Active | Retire ... | Shut Down... | **Test** | Settings... | Related Links

Dashboard | Instances | Faults and Rejected Messages | Unit Tests | Policies | **Test Service**

Recent Instances

Show Only Running Instances Running 0 Total 0

Instance ID	Name	Conversation ID	State	Start Time
No composite instances found.				

5. In the Input Arguments section, select **XML View** from the list and verify that the input XML is displayed, as shown in [Figure 1–57](#).

Figure 1–57 XML View Option Selected in the Input Arguments Section

Additional Test Options	
Enable Stress Test	<input type="checkbox"/>
Concurrent Threads	<input type="text" value="5"/>
Loops per Thread	<input type="text" value="10"/>
Delay in Milliseconds	<input type="text" value="1000"/>

Input Arguments
XML View <input type="button" value="v"/>
<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body xmlns:ns1="http://xmlns.oracle.com/bpmn/bpmnProcess/Process"> <ns1:operation xmlns:ns2="urn:sap-com:document:sap:business"> <ns2:BusinessArea.GetDetail></ns2:BusinessArea.GetDetail> </ns1:operation> </soap:Body> </soap:Envelope></pre>

Note: For BPM processes, it is required for the input XML to be used with a namespace.

- The displayed XML in the Oracle Enterprise Manager console can be altered and used as follows:

XML Displayed in the Oracle Enterprise Manager Console

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="http://xmlns.oracle.com/bpmn/bpmnProcess/Process">
    <ns1:operation xmlns:ns2="urn:sap-com:document:sap:business">
      <ns2:BusinessArea.GetDetail/>
    </ns1:operation>
  </soap:Body>
</soap:Envelope>
```

Altered XML

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="http://xmlns.oracle.com/bpmn/bpmnProcess/Process">
    <ns1:operation xmlns:bapi="urn:sap-com:document:sap:business">
      <bapi:BusinessArea.GetDetail BusinessAreaId="0001">
        <bapi:LANGUAGE>D</bapi:LANGUAGE>
        <bapi:LANGUAGE_ISO>DE</bapi:LANGUAGE_ISO>
      </bapi:BusinessArea.GetDetail>
    </ns1:operation>
  </soap:Body>
</soap:Envelope>
```

- The sample input XML that was generated using Altova XML Spy can be used after making the following required modifications:
 - Remove the XML headers.
 - Add the necessary inputs.
 - Add the SOAP headers as displayed in the Oracle Enterprise Manager Console.

- d. Modify the namespaces to match the namespaces in the input XML.

XML Generated Using Altova XML Spy

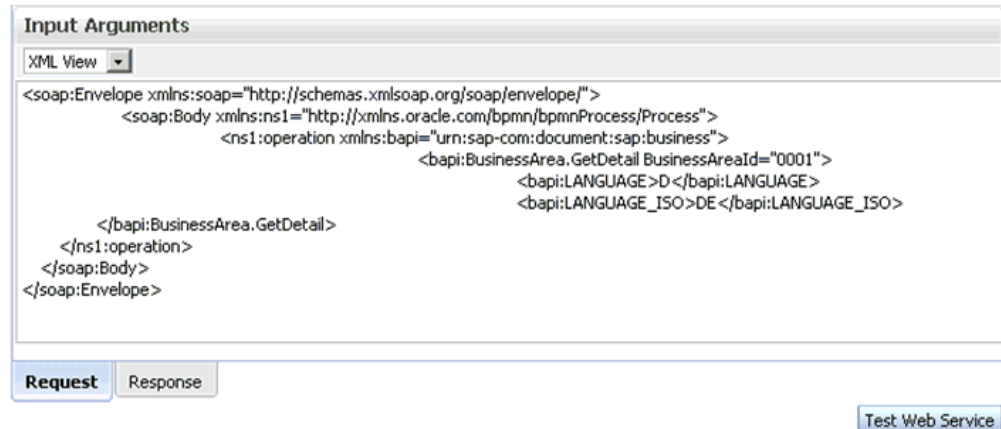
```
<?xml version="1.0" encoding="UTF-8"?> <!--Sample XML file generated by
XMLSpyv2008 rel. 2 sp2 (http://www.altova.com)--> <bapi:BusinessArea.GetDetail
BusinessAreaId="aaaa" xsi:schemaLocation="urn:sap-com:document:sap:business
mysap_jca_BA_GetDetail_invoke_jan20_request.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:bapi="urn:sap-com:document:sap:business">
  <bapi:LANGUAGE>a</bapi:LANGUAGE>
  <bapi:LANGUAGE_ISO>aa</bapi:LANGUAGE_ISO>
</bapi:BusinessArea.GetDetail>
```

Altered XML

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="http://xmlns.oracle.com/bpmn/bpmnProcess/Process">
    <ns1:operation xmlns:bapi="urn:sap-com:document:sap:business">
      <bapi:BusinessArea.GetDetail BusinessAreaId="0001">
        <bapi:LANGUAGE>D</bapi:LANGUAGE>
        <bapi:LANGUAGE_ISO>DE</bapi:LANGUAGE_ISO>
      </bapi:BusinessArea.GetDetail>
    </ns1:operation>
  </soap:Body>
</soap:Envelope>
```

8. Click **Test Web Service** after providing the input XML, as shown in [Figure 1–58](#).

Figure 1–58 Input XML



The output is received in the destination folder.

Oracle Application Adapter for SAP R/3 Best Practices

This chapter lists and describes best practices that are specific to the Oracle Application Adapter for SAP R/3 (SAP JCo 3.0xx). It contains the following topics:

- [Section 2.1, "Understanding Connection Pools"](#)
- [Section 2.2, "Security"](#)
- [Section 2.3, "Load Balancing"](#)
- [Section 2.4, "Encoding"](#)
- [Section 2.5, "SAP Server Behavior"](#)
- [Section 2.6, "IDoc Configuration"](#)
- [Section 2.7, "IDoc Metadata"](#)

2.1 Understanding Connection Pools

A connection pool is a set of client-connections to a certain destination. The pool may automatically create new connections to the specified remote system or return an existing connection. It also provides methods to return a connection back to the pool when it is no longer needed. The following list describes several key points:

- Each adapter target has its own unique connection pool with size and timing parameters. Ensure that when the target is created it has enough connections to handle all the tasks that will be allocated to the connection pool.
- The maximum number of connections specified can never be increased while the application is running. As a best practice, you must choose a number that is large enough for your application.
- If a target and consequently the connection pool is shared by different applications, then the applications share the SAP authorization permissions of the underlying user ID that created the connection pool.
- When a target is created using Application Explorer during design time, the specified connection parameters are used during run time for all objects created using the target. Ensure that the connection properties are capable of supporting your environment.
- The general parameters for calculating a connection pool size are: $\text{TIME} = (\text{SAP Application Server execution time for called function}) + (\text{size of document}) + (\text{network lag})$. It is recommended to check with a SAP Gateway administrator about the maximum size of allowed connection pools for a particular landscape.

- Larger documents or long running transactions need a larger value for the Connection Timeout parameter. Extremely long running documents should not be run in the foreground. It is recommended to check with a system administrator about establishing a batch job that can be called by the Oracle Application Adapter for SAP R/3.
- By default, SAP Java Connector (SAP JCo) opens one initial connection for a connection pool. If that initial connection is busy when another request arrives, then SAP JCo opens an additional connection, leading up to the maximum pool size. When the maximum pool size is reached, the Connection Wait Time parameter is called, which determines how long SAP JCo must wait for a free connection without aborting the task.

To monitor the number of connections that are being made to SAP from the Oracle Application Adapter for SAP R/3, you can use the SMGW transaction from your SAP GUI.

Optimizing a connection pool effectively requires knowledge about the application and its users. The following questions should be discussed:

- How much time does it normally take to execute this function?
- How much data is returned by this function?
- How many users are going to use this function?

It is not unreasonable to configure multiple connection targets with different connection pools for different functions. For example, a sales order inquiry that checks if an item exists may execute quickly, it does not need maximum resources. However, a sales order creation that creates a header and line items that is executed by many people may require a greater execution time, so configuring a larger connection pool is recommended.

2.2 Security

The following list describes several key points regarding security best practices:

- The SAP Java Connector (SAP JCo) default is plain text. If the SAP JCo communication has an insecure network path, then it is recommended that the SAP Router technology with RFC encryption is used to protect the data privacy.
- Restrict the ability of users to monitor or access the SAP Gateway and its functions. Ensure that the run time user ID for SAP JCo communications is not an SAP dialog user but a communications type user.
- As a best practice, consider always using the SAP router mechanism for communications between systems that are behind firewalls.

2.3 Load Balancing

This section provides best practices pertaining to inbound and outbound load balancing using SAP. It contains the following topics:

- [Section 2.3.1, "Inbound to SAP Load Balancing With a SAP Message Server"](#)
- [Section 2.3.2, "Outbound From SAP Load Balancing With the SAP Gateway"](#)

2.3.1 Inbound to SAP Load Balancing With a SAP Message Server

Inbound load balancing is restricted to logons through an SAP Message Server. The Message Server distributes logons to SAP Application Servers or distributes logons by

specific application groups to specific servers. When configuring the connection parameters for the adapter, additional information about the Message Server and Group to be used with the adapter is required. The Oracle Application Adapter for SAP R/3 supports connections to SAP using Message Servers. Select the *Message Server* option when creating adapter targets or channels using Application Explorer.

2.3.2 Outbound From SAP Load Balancing With the SAP Gateway

Load balancing for registered programs is a technique that is used when a large amount of data is being sent from SAP to remote destinations. The adapter channel is registered in the SAP application by a SAP administrator through a Registered Program ID.

The mechanism for outbound message load balancing is configured on the SAP Gateway Server by the administrator. However, it usually follows one of the following rules:

- 0: No load balancing, the first free registered program is used.
- 1: The program with the lowest counter is used. Every time a registered program is assigned a request, the counter is increased by one.
- 2: The program with the least load is used, which is determined by SAP.

The adapter channel connects to the SAP Gateway and exposes the same Program ID to the SAP Gateway. At this point, one or more servers accept the connection.

When a Registered Server (the adapter channel) connects to the gateway, it is to the gateway and not to a specific SAP application server. All messages sent to a Program ID arrive at a channel configured to listen to a SAP.

Gateway and Program ID

To increase load balancing for messages, more channels must expose the same Program ID. SAP Java Connector 3 (JCo 3.0xx) restricts the combination of Program ID and Gateway Server credentials in a single instance to a one-to-one correspondence. However, load balancing is available by using additional threads in the same instance or additional servers with the same Program ID on distributed servers. Perform the following steps to increase load balancing for messages:

1. In the Advanced tab of the Add Channel dialog, the *Thread count* parameter is set to three threads by default.
2. As a best practice, it is recommended to increase the *Thread count* parameter by multiples of three, which is the default value.
3. Ensure that the number of threads in the channel configuration matches a corresponding amount in the application server in a 1:3 ratio to avoid wasted resources or a SAP JCo automatic downcast of threadpool size.

Any destinations configured to receive messages from a channel receive all messages from that channel. This may have important implications when configuring business processes. Consider using different Program IDs for different messages, message filters, or message splitters, as coding techniques to route messages based on type or content.

If servers are configured with the same Program ID, then messages can be duplicated or never arrive, depending if load balancing is enabled on the SAP system. Use caution when deploying and using Program IDs and assign them in a logical and coordinated manner (for example, by department or by message type). Usually, binding the schema of a message to a channel is done to avoid the duplication of messages while reusing a Program ID across multiple servers.

Receiving Messages From SAP

The types of messages that are sent to the server are determined by the Interface document style (IDoc, RFC, BAPI) and the RFC destination. The RFC destination is used to hold the program ID inside SAP, and is used to route all messages to the adapter channel. For this reason, it is recommended that the SAP SM59 transaction that controls the RFC destination is locked to hide the IP address of the remote server inside SAP. Inside SAP, RFC function modules are routed to the adapter channel by specifying the DESTINATION parameter in the CALL FUNCTION invocation, passing the RFC destination that holds the server. For example:

```
CALL FUNCTION 'RFC_GET_SYSTEM_INFO DESTINATION ' DESTINATION 'MYDEST'
```

Where *MYDEST* is defined as a remote TCP (T) destination in the SM59 transaction and holds a Registered Server Program as one its parameters.

Receiving IDocs

SAP IDocs require additional configuration to define the sending and receiving systems. They are contained in a SAP Logical System. All IDocs routed through SAP ALE use a Logical System linked to a RFC destination to process messages through the channel. For the complete list of IDoc configuration steps, see [Section 2.6, "IDoc Configuration"](#) on page 2-5.

Asynchronous BAPI

It is possible to receive BAPI output in IDoc format. For more information, see the SAP help topic *Distribution Using BAPIs*. Once configured in the SAP system, asynchronous BAPIs are received the same way as ALE IDocs.

2.4 Encoding

Adapter channels on Unicode systems work only in Unicode mode. In the SM59 transaction, ensure that the RFC Destination Parameter, *Unicode*, is selected when creating the destination in SAP GUI.

The RFC components of SAP Java Connector (SAP JCo) automatically determine the target code page of the sending (client) and adjust the code page translation between the client and server accordingly. The only way to modify this is to change the target system code page, in Windows using Regional and Language Options in the Control Panel. For other systems, it is recommended to consult the administrator. In general, a system must have a code page and a language pack. Encoding is used to map one code page to another. This is straightforward in Unicode systems, but can be difficult or not possible on non-Unicode systems. As a general rule, in non-Unicode systems, you are limited to the code page and language pack you have on the system for the display of data.

Receiving document from SAP, especially IDocs, can have segments containing multiple languages. In non-Unicode systems, SAP ensures that the language used during logon is correct at output. For other languages, especially multibyte, it is usually not possible to get all languages correct in one document. For example, setting the Java encoding variable to ISO-8859-2 correctly sends German umlauts, but distorts Japanese Kanjii. The only solution for this situation is sending multiple text segments and combining them into one result, or using Unicode.

2.5 SAP Server Behavior

The following are key points related to SAP server behavior:

- If a connection cannot be opened or is dropped, then SAP Java Connector (JCo) tries to reconnect automatically.
- JCo sets the maximum number of threads in the listener configuration automatically. JCo manages the worker threads used for dispatching of requests on demand. The maximum number of the threads is defined by the maximum connection count.
- JCo puts all events from SAP in an internal queue. A cached request is dispatched when at least one worker thread is available.
- If a document is sent from SAP and cannot be delivered because of communication or application failure, then the SAP server will wait for a certain amount of time and then try to resend the document. The SAP server will do this up to a configured number of times before timing out. This behavior can be configured using SAP transaction SM59, specifically the *trfc options* setting for the RFC Destination.

2.6 IDoc Configuration

This section provides SAP system configuration guidelines for sending and receiving IDocs.

The following objects are required to transmit IDocs to SAP:

- Logical System - Place holder in SAP for remote system configuration.
- Distribution Model - Configures the sending and receiving partners for messages by Logical system and IDoc message type.
- Partner Profile - For each entry in a Distribution Model, this configures the physical parameters of how SAP will process the message.

The following objects are required to receive IDocs from SAP:

- RFC Destination with Registered Program ID - Defines the sending setup.
- Logical System - Placeholder in SAP for remote system configuration.
- Distribution Model - Configures the sending and receiving partners for messages by Logical system and IDoc message type.
- Partner Profile - For each entry in a Distribution Model, this configures the physical parameters of how SAP will process the message.
- Port - Configures where an IDoc is sent, directly links to the RFC Destination.

2.7 IDoc Metadata

An IDoc message type has an *IDoc Type* or basic type, consisting of a tree of segments. Each segment has as release independent *Segment Type* and a release dependent *Segment Definition*. There can be only one active or released IDoc Type or dependent Segment Definition for each SAP version. One part of IDoc processing is matching the segment information in the IDoc to the correct segment information in SAP. This information is obtained by matching the IDoc Type and SAP release of the incoming IDoc to the matching SAP metadata version of the IDoc Type and SAP release, and then obtaining the correct segment and field data based on the matching metadata. If additional Segment Definitions are added to a Segment Type, they cannot be released until the next version of SAP. However, this information does impact the stored Segment Definitions metadata information on SAP, and so, can impact current IDoc processing.

For example, Segment Type *ZSEG* has a Segment Definition, *ZSEG* that has a SAP version of 000, an SAP release of 700 and 74 fields. If an additional version is added, *ZSEG001* it has no release status, and 91 fields. It cannot be released until SAP version 701 is released.

If the ABAP program generating outbound SAP IDocs does not check for released or not released segments in sending IDocs, then IDocs sent will have the *ZSEG001* definition, and that will cause a problem when retrieving IDoc metadata because the IDoc will contain the new segment definition but the metadata will not, causing a Segment Unknown exception.

In the adapter channel configuration, if the IDoc release provider parameter is set to *SAP release*, then when checking for the segment *ZSEG001* metadata, an exception is generated by the *Segment Unknown* metadata layer, because the segment is not released in version 700. In the adapter channel configuration, if the IDoc release provider parameter is set to *IDoc Release*, and the release of the IDoc is not overridden in the SAP port, then the release will still be 700, leading to the same error. To receive the correct metadata in this situation, set the IDoc release provider parameter to *user input* and leave the *IDoc release* field blank. A blank value is a specific value to the metadata layer of SAP. It signifies that the highest version of the IDoc metadata, released or not, is returned. In this case, the metadata for *ZSEG001* will be retrieved, it will match the IDoc, and the document will be successfully processed.

Oracle Application Adapter for Siebel Best Practices

This chapter lists and describes best practices that are specific to the Oracle Application Adapter for Siebel. It contains the following topics:

- [Section 3.1, "Best Practices for Siebel"](#)
- [Section 3.2, "Configuring Connection Pooling for Siebel"](#)

3.1 Best Practices for Siebel

This section lists and describes best practices for the Oracle Application Adapter for Siebel.

- If you are using the native Siebel Java Data Bean (JDB) API to connect with a Siebel system, then integration should be performed using Siebel Business Objects and Business Services. If the HTTP protocol is required, then you must use HTTP. As a best practice, it is recommended to use the Siebel JDB API for connectivity when possible.
- If large amounts of transaction volumes need processing, then as a best practice, it is recommended to use connection pooling by configuring the `siebel.properties` file. For more information, see "[Configuring Connection Pooling for Siebel](#)" on page 3-2.
- For secure connectivity during outbound (services) processing only, it is recommended to enable RSA encryption using Application Explorer during the adapter target configuration stage.
- As a best practice, it is recommended to use XML schemas (.xsd) files that are generated through Siebel for Integration Objects and Business Services nodes.
- If any changes to the Siebel repository (.srf file) are made, then it is a best practice to recompile the .srf file and restart Siebel services to reflect the changes in the adapter.
- Even though Siebel Versions 6.2 and lower (using COM connectivity) are still supported, as a best practice, it is recommended that your integration project use Siebel 6.3 or higher with the Siebel JDB API.
- When sending an Insert, Update, or Query request, it is a best practice to include only the elements in the payload that has data to be inserted, updated, or retrieved. Any blank elements should be removed.
- As a best practice, do not include any special characters in the name when creating an Integration Object or Business Services node.

3.2 Configuring Connection Pooling for Siebel

To configure connection pooling for Siebel, you must create a **siebel.properties** file for use with the Oracle Application Adapter for Siebel. You can use this file to provide default parameters for applications connecting to Siebel using the Java Data Bean (JDB) API. The **siebel.properties** file must be added to your system classpath.

The following table lists and describes the property values that can be added in the **siebel.properties** file.

Property Type	Property	Description
Request Timeout	siebel.conmgr.txtimeout	The number of milliseconds to wait after sending a request to the Siebel Server. Must be a positive integer. Other values are ignored. The default value is 600000 milliseconds (10 minutes). The maximum value is 2,147,483,647 ms (approximately 25 days).
Poolsize	siebel.conmgr.poolsize	For each Application Object Manager process, a pool of open connections is maintained and shared by all users of that process. This parameter specifies the maximum number of connections that will be stored in the pool. Its value must be a positive integer less than 500. Other values are ignored. The default is 2.
Session Timeout	siebel.conmgr.sesstimeout	The number of seconds the Siebel Server will wait before disconnecting an idle client session. Its value must be a nonnegative integer. The default is 2700 seconds (45 minutes). The maximum value is 2,147,483,647 s (approximately 68 years).

Oracle Application Adapter for PeopleSoft Best Practices

This chapter lists and describes best practices that are specific to the Oracle Application Adapter for PeopleSoft.

4.1 PeopleSoft Events (Inbound Processing)

For Release 8.4, Oracle Software recommends using the TCP/IP and HTTP target connectors that are delivered by PeopleSoft for the PeopleTools 8.4 series. Do not use the target connectors that are supplied by Oracle Software for the PeopleTools 8.1 series. They are only packaged by Oracle Software for the PeopleTools 8.4 series to assist existing users who are migrating from Release 8.1 to Release 8.4.

Oracle Application Adapter for J.D. Edwards OneWorld Best Practices

This chapter lists and describes best practices that are specific to the Oracle Application Adapter for J.D. Edwards OneWorld. It contains the following topics:

- Section 5.1, "Connection Pooling and Load Balancing"
- Section 5.2, "Security"
- Section 5.3, "Multiple Connections and Multiple Users"
- Section 5.4, "Unicode Support"
- Section 5.5, "Delivery Confirmation and Error Messages"

5.1 Connection Pooling and Load Balancing

Connection pooling is established for a session by setting the session attribute of the standard `jdeRequest` element. When the session attribute is an empty string, a new session is started. On the server, the `SessionManager` singleton class creates a new instance of a session object given the user name, password, and environment name. The session can be reused before it expires to avoid the overhead of session initialization. You can specify the session ID in the session attribute for an established session in an earlier request.

Session expiration is addressed by the `sessionidle` attribute of the standard `jdeRequest` element. This attribute, when given on a session creation request, specifies the amount of time in seconds that a session is allowed to be idle. If the `SessionManager` determines that a session has not had any requests processed in this amount of time, then it terminates the session and frees all associated resources. For more information, see the *J.D. Edwards OneWorld Interoperability Guide*.

Run time settings in the `jde.ini` file on the server control the load balancing configuration. These settings are in the `[JDENET]` and applicable `[JDENET_KERNEL_DEFx]` sections. All relevant settings are listed and described in the preceding section. You can control two types of processes:

- Network jobs (`JDENET_n`)
- Dedicated kernel process jobs (`JDENET_k`)

For more information, see the *J.D. Edwards OneWorld System Administration Guide*.

5.2 Security

The Oracle Application Adapter for J.D. Edwards OneWorld supports standard JDE security. Security is managed by user profiles defined in the system. For more information, see the *J.D. Edwards OneWorld Server and Workstation Administration Guide*.

5.3 Multiple Connections and Multiple Users

Multiple connections and multiple users depends on the user profiles that are defined in J.D. Edwards OneWorld. For more information, see the *J.D. Edwards OneWorld Server and Workstation Administration Guide*.

5.4 Unicode Support

The J.D. Edwards XML solution supports well-formed XML documents. It also supports UTF8 and UTF16 Unicode standards for inbound processing and UTF8 standards for outbound processing.

5.5 Delivery Confirmation and Error Messages

Each transaction is assigned session ID by J.D. Edwards OneWorld. As a best practice, use application P92002 to investigate server-side error messages.

Glossary

adapter

Provides universal connectivity by enabling an electronic interface to be accommodated (without loss of function) to another electronic interface.

agent

Supports service protocols in listeners and documents.

business service

Also known as a Web service. A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity.

channel

Represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by an adapter.

listener

A component that accepts requests from client applications.

port

Associates a particular business object exposed by the adapter with a particular disposition. A disposition is a URL that defines the protocol and location of the event data. The port defines the end point of the event consumption.

B

best practices

- general, 1-1
- Oracle Application Adapter for J.D. Edwards OneWorld, 5-1
- Oracle Application Adapter for PeopleSoft, 4-1
- Oracle Application Adapter for SAP R/3, 2-1
- Oracle Application Adapter for Siebel, 3-1

G

- general best practices, 1-1

I

input XML documents

- generating, 1-28

M

- managed connections, 1-1

O

Oracle Application Adapter for Siebel

- connection pooling, 3-2
- Oracle's Unified Method (OUM), vi

