Oracle® Application Server Integration InterConnect

Adapter for JCA Installation and User's Guide

10*g* Release 2 (10.1.2)

B19182-01

December 2005

ORACLE®

Oracle Application Server Integration InterConnect Adapter for JCA Installation and User's Guide, 10*g* Release 2 (10.1.2)

B19182-01

Primary Author: Rima Dave

Contributing Author: Vimmy Raj

Contributor: Ashwin Patel, Harish Sriramalu

Contents

A Frequently Asked Questions

Index

Preface

This guide describes how to use the JCA adapter that are provided with Oracle Application Server Integration InterConnect.

This preface contains the following topics:

Audience

Oracle Application Server Integration InterConnect Adapter for JCA Installation and User's Guide is intended for anyone who is interested in using this adapters.

To use this document, you need to know how to install and configure Oracle Application Server Integration InterConnect.

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, refer to these Oracle resources:

- *Oracle Application Server Integration InterConnect User's Guide*
- *Oracle Application Server Integration InterConnect Installation Guide*

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.

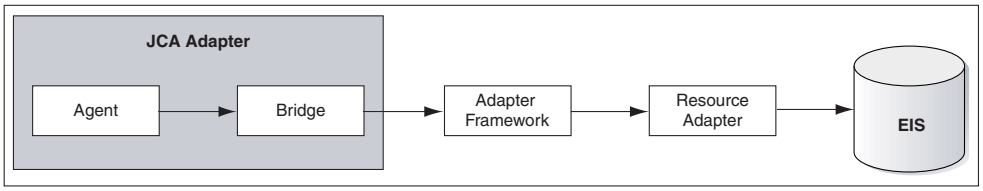
1 Overview

This chapter provides an overview of Oracle Application Server Integration InterConnect (OracleAS Integration InterConnect) adapter for JCA (JCA adapter). It contains the following topics:

- Section 1.1, "JCA Adapter Overview"
- Section 1.2, "JCA Adapter System Requirements"

1.1 JCA Adapter Overview

The JCA adapter consists of a bridge and a run-time agent. The bridge communicates with the resource adapter through adapter framework. Adapter framework is a WSDL JCA provider that can communicate with any JCA compliant resource adapter. It makes the WSIF invocation to resource adapter and pass the information coming from bridge.



1.2 JCA Adapter System Requirements

The following sections describe the JCA adapter system requirements:

- Hardware Requirements
- Software Requirements

1.2.1 Hardware Requirements

Table 1–1 lists the hardware requirements for installing the JCA adapter.

Table 1–1 Hardware Requirements

Hardware	Windows 2000	UNIX
Disk Space	400 MB	400 MB
Memory	512 MB	512 MB

1.2.2 Software Requirements

The following sections describe the JCA adapter software requirements:

- Operating System Requirements
- JRE Requirements

Operating System Requirements

Table 1–2 lists the operating system requirements for installing the JCA adapter.

Table 1–2 Operating System Requirements

Operating System	Version
HP Tru64	HP Tru64 UNIX (Alpha) 5.1b
HP-UX	HP-UX (PA-RISC) 11.11, 11.23
IBM AIX	AIX (POWER) version 5.2
Linux (x86)	Red Hat Enterprise Linux 2.1, 3.0 SuSE SLES8, SLES9
Sun SPARC Solaris	Sun SPARC Solaris 2.8 and 2.9
Microsoft Windows	Windows XP Professional, Windows 2000 (SP3 or higher)

JRE Requirements

OracleAS Integration InterConnect uses Java Runtime Environment (JRE) 1.4, which is installed with its components.

2

Installation

This chapter describes how to install and configure the JCA adapter. It contains the following topics:

- Section 2.1, "Installing the JCA Adapter"
- Section 2.3, "Configuring the JCA adapter"

2.1 Installing the JCA Adapter

The JCA adapter must be installed in an existing Oracle home Middle Tier for OracleAS Integration InterConnect 10*g* Release 2 (10.1.2).

This section contains the following topics:

- Preinstallation Tasks
- Installation Tasks

2.1.1 Preinstallation Tasks

Refer to following guides before installing the JCA adapter:

- *Oracle Application Server Installation Guide* for information about Oracle Universal Installer startup.
- *Oracle Application Server Integration InterConnect Installation Guide* for information on software, hardware, and system requirements for OracleAS Integration InterConnect.

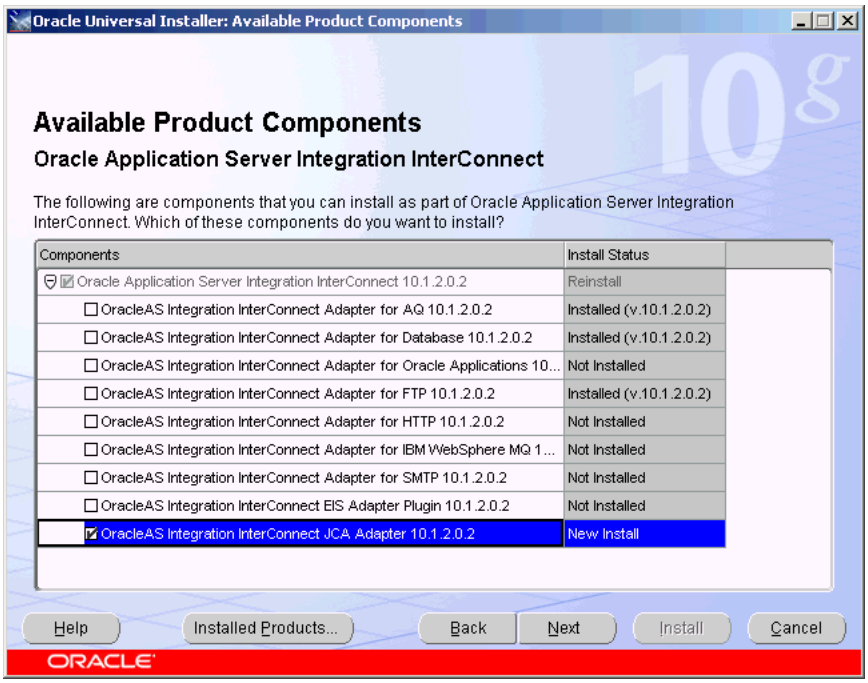
> **Note:** OracleAS Integration InterConnect Hub is installable through the OracleAS Integration InterConnect Hub installation type. You must install the OracleAS Integration InterConnect Hub before proceeding with the JCA adapter installation.

2.1.2 Installation Tasks

To install the JCA adapter:

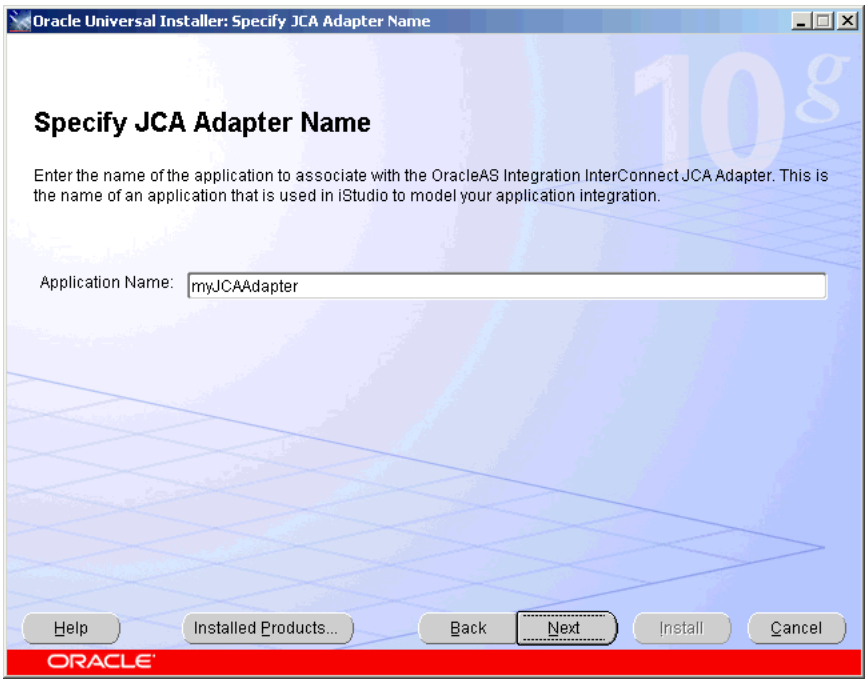
1. In the Available Product Components window of the OracleAS Integration InterConnect installation, select **OracleAS Integration InterConnect JCA Adapter** and click **Next**.

Figure 2–1 Available Product Components Window



2. The Set Oracle Wallet Password window is displayed. Enter and confirm the password, which will be used to manage OracleAS Integration InterConnect installation. Click **Next**.
   - Go to step 3, if installing the JCA adapter in an OracleAS Middle Tier Oracle home that does not have an InterConnect component already installed. Ensure that the OracleAS Integration InterConnect hub has been installed.
   - Go to step 4, if installing the JCA adapter in an OracleAS Middle Tier Oracle home that has an existing InterConnect component. Ensure that it is a home directory to an OracleAS Integration InterConnect component.
3. The Specify Hub Database Connection window is displayed. Enter information in the following fields:
   - Host Name: The host name of the computer where the hub database is installed.
   - Port Number: The TNS listener port for the hub database.
   - Database SID: The System Identifier (SID) for the hub database.
   - Password: The password for the hub database user.
4. Click **Next**. The Specify HTTP Adapter Name window is displayed.
5. Enter the application to be defined. Blank spaces are not permitted. The default value is `myJCAAdapter`.

Figure 2–2 Specify JCA Adapter Name Window



6. Click **Next**. The Summary window is displayed.
7. Click **Install** to install the JCA adapter. The adapter is installed in the following directory, depending on the operating system:

Platform	Directory
UNIX	*ORACLE_HOME*/integration/interconnect/adapters/*Application*
Windows	*ORACLE_HOME*\integration\interconnect\adapters*Application*

8. Click **Exit** on the Installation window to exit the JCA adapter installation.

2.2 Installing Multiple JCA Adapters in the Same Oracle Home

To install multiple instances of the JCA adapter in same Oracle home, use the `copyAdapter` script located in the *ORACLE_HOME*`/integration/interconnect/bin` directory.

Usage: `copyAdapter` *app1 app2*

For example, you have one instance of JCA adapter with name `myJCAApp` installed on a computer. To install another instance of the JCA adapter with name `myJCAApp1` in the same Oracle home, use the following command:

```
copyAdapter myJCAApp myJCAApp1
```

The `copyAdapter` script is copied to the following `bin` directory only during Hub installation:

- UNIX: `ORACLE_HOME/integration/interconnect/bin`

- Windows: ORACLE_HOME\integration\interconnect\bin

If you need to use this script to create multiple adapters on a spoke computer, then copy the script to the bin directory on the spoke computer, and edit the script to reflect the new Oracle home.

After running the copyAdapter script, If you want to manage or monitor the newly installed adapter through Oracle Enterprise Manager 10*g* Application Server Control Console, then you need to modify the opmn.xml file by adding information about the new instance. For example, you have created a new instance of the JCA adapter myJCAApp1 by using the copyAdapter script. To manage the myJCAApp1 adapter through Enterprise Manager, perform the following:

1. Navigate to the *MiddleTier*\bin directory and run the following command to stop the Enterprise Manager:

   ```
   emctl stop iasconsole
   ```

2. Next, specify the information about this new instance in the opmn.xml file located in the *ORACLEMIDDLETIER_HOME*/opmn/conf directory as follows:

   ```
   <process-type id="myJCAApp1" module-id="adapter" working-dir="$ORACLE_
   HOME/integration/interconnect/adapters/myJCAApp1" status="enabled">
          <start timeout="600" retry="2"/>
          <stop timeout="120"/>
          <port id="icadapter_dmsport_range" range="15701-15800"/>
          <process-set id="myJCAApp1" restart-on-death="true" numprocs="1">
               <module-data>
                   <category id="start-parameters">
                       <data id="java-parameters" value="-Xms8M"/>
                       <data id="class-name"
                        value="oracle.oai.agent.service.AgentService"/>
                   </category>
                   <category id="stop-parameters">
                       <data id="java-parameters" value="-mx64m"/>
                       <data id="class-name"
                        value="oracle.oai.agent.proxy.ShutdownAgent"/>
                       <data id="application-parameters"
                        value="persistence/Agent.ior"/>
                   </category>
               </module-data>
          </process-set>
   </process-type>
   ```

   The opmn.xml file would appear like this:

   ```
   <process-type id="myJCAApp" module-id="adapter" working-dir="$ORACLE
   _HOME/integration/interconnect/adapters/myJCAApp" status="enabled">
          <start timeout="600" retry="2"/>
          <stop timeout="120"/>
          <port id="icadapter_dmsport_range" range="15701-15800"/>
          <process-set id="myJCAApp" restart-on-death="true" numprocs="1">
               <module-data>
                   <category id="start-parameters">
                       <data id="java-parameters" value="-Xms8M"/>
                       <data id="class-name"
                        value="oracle.oai.agent.service.AgentService"/>
                   </category>
                   <category id="stop-parameters">
                       <data id="java-parameters" value="-mx64m"/>
                       <data id="class-name"
                    value="oracle.oai.agent.proxy.ShutdownAgent"/>
   ```

```
            <data id="application-parameters"
             value="persistence/Agent.ior"/>
         </category>
      </module-data>
  </process-set>
</process-type>

<process-type id="myJCAApp1" module-id="adapter" working-dir="$ORACLE
_HOME/integration/interconnect/adapters/myJCAApp1" status="enabled">
       <start timeout="600" retry="2"/>
       <stop timeout="120"/>
       <port id="icadapter_dmsport_range" range="15701-15800"/>
       <process-set id="myJCAApp1" restart-on-death="true" numprocs="1">
        <module-data>
          <category id="start-parameters">
              <data id="java-parameters" value="-Xms8M"/>
              <data id="class-name"
               value="oracle.oai.agent.service.AgentService"/>
          </category>
          <category id="stop-parameters">
               <data id="java-parameters" value="-mx64m"/>
               <data id="class-name"
                value="oracle.oai.agent.proxy.ShutdownAgent"/>
               <data id="application-parameters"
                value="persistence/Agent.ior"/>
          </category>
        </module-data>
       </process-set>
</process-type>
```

3. Save the `opmn.xml` file.
4. Navigate to the *`MiddleTier`*`\opmn\bin` directory and run the following command to reload the OPMN:

   ```
   opmnctl reload
   ```

5. You can start the `myJCAApp1` adapter by using the following command

   ```
   opmnctl startproc ias-component="InterConnect" process-type="myJCAApp1"
   ```

6. Navigate to the *`MiddleTier`*`\bin` directory and run the following command to start the Enterprise Manager:

   ```
   emctl start iasconsole
   ```

7. Login to the Oracle Enterprise Manager 10*g* Application Server Control Console to view and manage the newly installed or copied adapter. For information about how to use Oracle Enterprise Manager 10*g* Application Server Control Console , refer to the *Oracle Application Server Integration InterConnect User's Guide*

Note: While installing multiple adapters in the same computer, the copyadapter script does not create entries for the new adapter's password in the Oracle Wallet. You need to manually create a password for this new adapter using the Oracle Wallet Manager. To store the password in Oracle Wallet, use the following format:

```
ApplicationName/password
```

The number of entries is dependent on the type of adapter. For example, Database Adapter needs two entries whereas AQ Adapter needs only one entry. For more information about how to manage your passwords in Oracle Wallet, refer to Section A.8, "How do I secure my passwords?"in Appendix A, "Frequently Asked Questions"

2.3 Configuring the JCA adapter

After installing the JCA adapter, you can configure it for your needs. The following tables describe the location and details of the configuration files.

Table 2–1 describes the location where the adapter is installed.

Table 2–1 JCA Adapter Directory

Platform	Directory
UNIX	*ORACLE_HOME*/integration/interconnect/adapters/Application
Windows	*ORACLE_HOME*\integration\interconnect\adapters\Application

Table 2–2 describes the various executable files available for the JCA adapter.

Table 2–2 OA Executable Files

File	Description
`start` (UNIX)	Does not take parameters; starts the adapter.
`start.bat` (Windows)	Does not take parameters; starts the adapter.
`stop` (UNIX)	Does not take parameters; stops the adapter.
`stop.bat` (Windows)	Does not take parameters; stops the adapter.

Table 2–3 describes the JCA adapter configuration files.

Table 2–3 OA Configuration Files

File	Description
`adapter.ini` (UNIX)	Consists of all the initialization parameters that the adapter reads at startup.
`adapter.ini` (Windows)	Consists of all the initialization parameters that the adapter reads at startup.

Table 2–4 describes the directories used by the JCA adapter.

Table 2–4 JCA Directories

Directory	Description
`logs`	The logging of adapter activity is done in subdirectories of the logs directory. Each new run of the adapter creates a new subdirectory in which logging is done in an `log.xml` file.
`persistence`	The messages are persisted (made available) in this directory. Do not edit this directory or its files.

2.3.1 JCA Adapter Ini File Settings

The following are the `.ini` files used to configure the JCA adapter:

- hub.ini Parameters
- adapter.ini Parameters

2.3.1.1 hub.ini Parameters

The JCA adapter connects to the hub database using the parameters in the `hub.ini` file located in the `hub` directory. Table 2–5 gives a description and an example for each parameter of the `hub.ini` file.

Table 2–5 hub.ini Parameters

Parameter	Description	Example
`hub_host`	The name of the computer hosting the hub database. There is no default value. The value is set during installation.	`hub_host=mpscottpc`
`hub_instance`	The SID of the hub database. There is no default value. The value is set during installation.	`hub_instance=orcl`
`hub_port`	The TNS listener port number for the hub database instance. There is no default value. The value is set during installation.	`hub_port=1521`
`hub_username`	The name of the hub database schema (or user name). The default value is `ichub`.	`hub_username=ichub`
`repository_name`	The name of the repository that communicates with the adapter. The default value is `InterConnectRepository`.	`repository_name=InterConnectRepository`

Oracle Real Application Clusters hub.ini Parameters

When a hub is installed on a Oracle Real Application Cluster database, the parameters listed in Table 2–6 represent information about additional nodes used for connection and configuration. These parameters are in addition to the default parameters for the primary node. In Table 2–6, `x` represents the node number which can range from 2 to total number of nodes in a cluster. For example, if the cluster contains 4 nodes, then `x` can be a value between 2 and 4.

Table 2–6 Oracle Real Application Clusters hub.ini Parameters

Parameter	Description	Example
`hub_hostx`	The host where the Real Application Clusters database is installed.	`hub_host2=dscott13`

Table 2–6 (Cont.) Oracle Real Application Clusters hub.ini Parameters

Parameter	Description	Example
`hub_instancex`	The instance on the respective node.	`hub_instance2=orcl2`
`hub_num_nodes`	The number of nodes in a cluster.	`hub_num_nodes=4`
`hub_portx`	The port where the TNS listener is listening.	`hub_port2=1521`

2.3.1.2 adapter.ini Parameters

The agent component of the JCA adapter reads the `adapter.ini` file at run time to access information on configuring the JCA adapter parameter. Table 2–7 gives a description and an example for each parameter of the `adapter.ini` file.

Table 2–7 adapter.ini Parameters

Parameter	Description	Example
`agent_admin_port`	Specifies the port through which the adapter can be accessed through firewalls. Possible value: A valid port number Default value: None	`agent_admin_port=15005`
`agent_delete_file_cache_at_startup`	Specifies whether to delete the cached metadata during startup. If any agent caching method is enabled, then metadata from the repository is cached locally on the file system. Set the parameter to `true` to delete all cached metadata on startup. Possible values: `true` or `false` Default value: `true` **Note:** After changing metadata or DVM tables for the adapter in iStudio, you must delete the cache to guarantee access to new metadata or table information.	`agent_delete_file_cache_at_startup=false`
`agent_dvm_table_caching`	Specifies the Domain Value Mapping (DVM) table caching algorithm. Possible values: ■ `startup`: Cache all DVM tables at startup. This may be time-consuming if there are many tables in the repository. ■ `demand`: Cache tables as they are used. ■ `none`: No caching. This slows down performance. Default value: `demand`.	`agent_dvm_table_caching=demand`
`agent_log_level`	Specifies the amount of logging necessary. Possible values: `0`=errors only `1`=status and errors `2`=trace, status, and errors Default value: `1`	`agent_log_level=2`

Table 2–7 (Cont.) adapter.ini Parameters

Parameter	Description	Example
`agent_lookup_table_caching`	Specifies the lookup table caching algorithm. Possible values: ■ `startup`: Cache all lookup tables at startup. This may be time-consuming if there are many tables in the repository. ■ `demand`: Cache tables as they are used. ■ `none`: No caching. This slows down performance Default value: `demand`	`agent_lookup_table_caching=demand`
`agent_max_ao_cache_size`	Specifies the maximum number of application object metadata to cache. Possible value: An integer greater than or equal to `1` Default value: `200`	`agent_max_ao_cache_size=200`
`agent_max_co_cache_size`	Specifies the maximum number of common object metadata to cache. Possible value: An integer greater than or equal to `1` Default value: `100`	`agent_max_co_cache_size=100`
`agent_max_dvm_table_cache_size`	Specifies the maximum number of DVM tables to cache. Possible value: An integer greater than or equal to `1` Default value: `200`	`agent_max_dvm_table_cache_size=200`
`agent_max_lookup_table_cache_size`	Specifies the maximum number of lookup tables to cache. Possible value: Any integer greater than or equal to `1` Default value: `200`	`agent_max_lookup_table_cache_size=200`
`agent_max_message_metadata_cache_size`	Specifies the maximum number of message metadata (publish/subscribe and invoke/implement) to cache. Possible value: An integer greater than or equal to `1` Default value: `200`	`agent_max_message_metadata_cache_size=200`
`agent_max_queue_size`	Specifies the maximum size internal OracleAS Integration InterConnect message queues can grow. Possible value: An integer greater than or equal to `1` Default value: `1000`	`agent_max_queue_size=1000`
`agent_message_selector`	Specifies conditions for message selection when the adapter registers its subscription with the hub. Possible value: A valid Oracle Advanced Queue message selector string (like `'%,%s_adaptername%,%'`) Default value: None	`agent_message_selector=%,aqapp,%`
`agent_metadata_caching`	Specifies the metadata caching algorithm. Possible values: ■ `startup`: Cache everything at startup. This may be time-consuming if there are many tables in the repository. ■ `demand`: Cache metadata as it is used. ■ `none`: No caching. This slows down performance Default value: None	`agent_metadata_caching=demand`

Table 2–7 (Cont.) adapter.ini Parameters

Parameter	Description	Example
agent_ persistence_ cleanup_interval	Specifies how often to run the persistence cleaner thread in milliseconds. Possible value: An integer greater than or equal to 30000 milliseconds Default value: 60000	agent_persistence_ cleanup_ interval=60000
agent_ persistence_ queue_size	Specifies the maximum size of internal OracleAS Integration InterConnect persistence queues. Possible value: An integer greater than or equal to 1 Default value: 1000	agent_persistence_ queue_size=1000
agent_ persistence_ retry_interval	Specifies how often the persistence thread retries when it fails to send an OracleAS Integration InterConnect message. Possible value: An integer greater than or equal to 5000 milliseconds Default value: 60000	agent_persistence_ retry_interval=60000
agent_pipeline_ from_hub	Specifies whether to activate the pipeline for messages from the hub to the bridge. If you set the pipeline to false, then the file persistence is not used in that direction. Possible value: true, false Default value: false	agent_pipeline_from_ hub=false
agent_pipeline_ to_hub	Specifies whether to activate the pipeline for messages from the bridge to the hub. If you set the pipeline to false, then the file persistence is not used in that direction. Possible value: true, false Default value: false	agent_pipeline_to_ hub=false
agent_reply_ message_selector	Specifies the application instance to which the reply must be sent. This parameter is used if multiple adapter instances exist for the given application and given partition. Possible value: A string built using the application name (parameter:application) concatenated with the instance number (parameter:instance_number). Default value: None	If application=aqapp, instance_number=2, then agent_reply_message_ selector=recipient_list like'%,aqapp2,%'
agent_reply_ subscriber_name	Specifies the subscriber name used when multiple adapter instances are used for the given application and given partition. This parameter is optional if only one instance is running. Possible value: The application name (parameter:application) concatenated with the instance number (parameter:instance_number). Default value: None	If application=oaapp and instance_number=2, then agent_reply_ subscriber_ name=oaapp2
agent_ subscriber_name	Specifies the subscriber name used when this adapter registers its subscription. Possible value: A valid Oracle Advanced Queue subscriber name Default value: None	agent_subscriber_ name=oaapp

Table 2–7 (Cont.) adapter.ini Parameters

Parameter	Description	Example
agent_ throughput_ measurement_ enabled	Specifies if the throughput measurement is enabled. Set this parameter to true to turn on throughput measurements. Default value: true	agent_throughput_ measurement_ enabled=true
agent_tracking_ enabled	Specifies if message tracking is enabled. Set this parameter to false to turn off tracking of messages. Set this parameter to true to track messages with tracking fields set in iStudio. Default value: true	agent_tracking_ enabled=true
agent_use_ custom_hub_dtd	Specifies whether to use a custom DTD for the common view message when handing it to the hub. By default, adapters use a specific OracleAS Integration InterConnect DTD for all messages sent to the hub. Set this parameter to true to have the adapter use the DTD imported for the message of the common view instead of the OracleAS Integration InterConnect DTD. Default value: false	agent_use_custom_hub_ dtd=false
application	Specifies the name of the application to which this adapter connects. This must match the name specified in iStudio while creating metadata. Possible value: An alphanumeric string. Default value: None	application=oaapp
instance_number	Specifies the instance number to which this adapter corresponds. Specify a value only if you have multiple adapter instances for the given application with the given partition. Possible value: An integer greater than or equal to 1. Default value: None	instance_number=1
partition	Specifies the partition this adapter handles as specified in iStudio. Possible value: An alphanumeric string. Default value: None	partition=germany
service_class	Specifies the entry class for the Windows service. Possible value: oracle/oai/agent/service/AgentService. Default value: None	service_ class=oracle/oai/agen t/service/AgentServic e
service_ classpath	Specifies the class path used by the adapter JVM. If a custom adapter is developed and the adapter is to pick up any additional jar files, then add the files to the existing set of jar files. Possible value: A valid PATH setting Default value: None This parameter is for Microsoft Windows only.	service_ classpath=D:\oracle\o raic\integration\inte rconnect\lib\oai.jar; D:\oracle\oraic\jdbc\ classes12.zip
service_jdk_dll	Specifies the Dynamic Link Library(DLL) that the adapter JVM should use. Possible value: A valid jvm.dll Default value: jvm.dll This parameter is for Microsoft Windows only.	service_jdk_ dll=jvm.dll

Table 2–7 (Cont.) adapter.ini Parameters

Parameter	Description	Example
service_jdk_ version	Specifies the JDK version that the adapter JVM should use. Possible value: A valid JDK version number Default value: 1.4.2 This parameter is for Microsoft Windows only.	service_jdk_ version=1.4
service_max_ heap_size	Specifies the maximum heap size for the adapter JVM. Possible value: A valid JVM heap size Default value: 536870912 This parameter is only for Microsoft Windows.	service_max_heap_ size=536870912
service_max_ java_stack_size	Specifies the maximum size the JVM stack can grow. Possible value: A valid JVM maximum stack size Default value: Default value for the JVM This parameter is only for Microsoft Windows.	service_max_java_ stack_size=409600
service_max_ native_stack_ size	Specifies the maximum size the JVM native stack can grow. Possible value: The valid JVM maximum native stack size Default value: Default value for the JVM This parameter is only for Microsoft Windows.	service_max_native_ size=131072
service_min_ heap_size	Specifies the minimum heap size for the adapter JVM. Possible value: The valid JVM heap size Default value: 67108864 This parameter is only for Microsoft Windows.	service_min_heap_ size=67108864
service_num_vm_ args	Specifies the number of service_vm_arg*number* parameters specified in JVM. Possible value: The number of service_vm_ arg*number* parameters Default value: 2 This parameter is only for Microsoft Windows.	service_num_vm_args=1
service_path	Specifies the environment variable PATH. The PATH variable is set before starting the Java Virtual Machine (JVM). Typically, list all directories that contain necessary DLLs. Possible value: The valid PATH environment variable setting Default value: None This parameter is only for Microsoft Windows.	service_ path=%JREHOME%\bin;D: \oracle\oraic\bin
service_vm_ arg*number*	Specifies any additional arguments to the JVM. For example, to retrieve line numbers in any stack traces, set service_vm_arg1=java.compiler=NONE. If a list of arguments exists, then use multiple parameters as shown in the example, by incrementing the last digit by 1. Possible value: A valid JVM arguments Default value: None This parameter is only for Microsoft Windows.	service_vm_ arg1=java.compiler= NONE service_vm_ arg2=oai.adapter=.aq

JCA Adapter-specific Parameters

Table 2–8 lists the parameters specific to the JCA adapter.

Table 2–8 JCA Adapter-specific Parameters

Parameter	Description	Example
`connection1_name`	Specifies EIS-specific connection factory instance. Possible value: `connection1_name=eis/DatabaseAdapter`	`connection1_name=eis/DatabaseAdapter`
`connection1_mcf_class`	Specifies the managed connection factory class. Possible value: `oracle.tip.adapter.db.DBManagedConnectionFactory`	`connection1_mcf_class=oracle.tip.adapter.db.DBManagedConnectionFactory`
`connection1_mcf_params`	Specifies the managed connection factory class parameters such as `ConnectionString`, `DriverClassname`, `UserName`, and `Password`.	`connection1_mcf_params={ConnectionString=jdbc:oracle:thin:@apatel-pc.us.oracle.com:1521:orcl, DriverClassName=oracle.jdbc.driver.OracleDriver, UserName=scott, Password=tiger}`

3

Design-Time and Run-Time

This chapter describes the design-time and run-time concepts for the JCA adapter. It contains the following topics:

- Design-Time Concepts
- Run-Time Concepts
- Starting the JCA Adapter
- Stopping the JCA Adapter

3.1 Design-Time Concepts

The design time for JCA adapter consists of following three steps:

- Generating a WSDL file
- Modifying the generated WSDL file
- Importing the generated WSDL in iStudio by using the WSDL browser to create the common view and application view

These design-time steps are explained by using a scenario in the following sections.

- Scenario Overview
- Generating the WSDL Files
- Modifying the Generated WSDL Files
- iStudio Design-Time Steps

3.1.1 Scenario Overview

This section describes how to design and execute the following integration scenario

1. The publishing JCA adapter polls two database tables `Departments` and `Employees` for new record inserts. The `DeptID` column of the `Employees` table references the `DeptID` column of the `Departments` table.
2. When you insert a new record in any of these tables, the publishing JCA adapter publishes the message to the InterConnect Hub. This message contains the data that was inserted in the table.
3. The subscribing JCA adapter polls the InterConnect hub queue for messages and receives the new message.
4. The subscribing JCA adapter inserts the data contained in the message into the `NewEmployees` or `NewDepartments` tables.

Pre-Requisites

This example assumes that you are familiar with basic BPEL constructs, such as activities and partner links, and JDeveloper environment for creating a BPEL process.

You need following tables for this scenario:

- DEPARTMENTS

  Use the following code to create the table:

```
CREATE TABLE DEPARTMENTS
(
  DEPTNO NUMBER(2) CONSTRAINT PK_DEPT PRIMARY KEY,
  DNAME VARCHAR2(14),
  LOC VARCHAR2(13)
);
```

- NewDEPARTMENTS

  Use the following code to create the table:

```
CREATE TABLE NewDEPARTMENTS
(
   DEPTNO NUMBER(2) CONSTRAINT PK_NewDEPT PRIMARY KEY,
   DNAME VARCHAR2(14),
   LOC VARCHAR2(13)
) ;
```

- EMPLOYEES

  Use the following code to create the table:

```
CREATE TABLE EMPLOYEES
(
   EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
   ENAME VARCHAR2(10),
   JOB VARCHAR2(9),
   MGR NUMBER(4),
   HIREDATE DATE,
   SAL NUMBER(7,2),
   COMM NUMBER(7,2),
   DEPTNO NUMBER(2) CONSTRAINT FK_DEPTNO REFERENCES DEPARTMENTS
);
```

- NewEMPLOYEES

  Use the following code to create the table:

```
CREATE TABLE NewEMPLOYEES
(
   EMPNO NUMBER(4) CONSTRAINT PK_NewEMP PRIMARY KEY,
   ENAME VARCHAR2(10),
   JOB VARCHAR2(9),
   MGR NUMBER(4),
   HIREDATE DATE,
   SAL NUMBER(7,2),
   COMM NUMBER(7,2),
   DEPTNO NUMBER(2) CONSTRAINT FK_NewDEPTNO REFERENCES NewDEPARTMENTS
);
```

3.1.2 Generating the WSDL Files

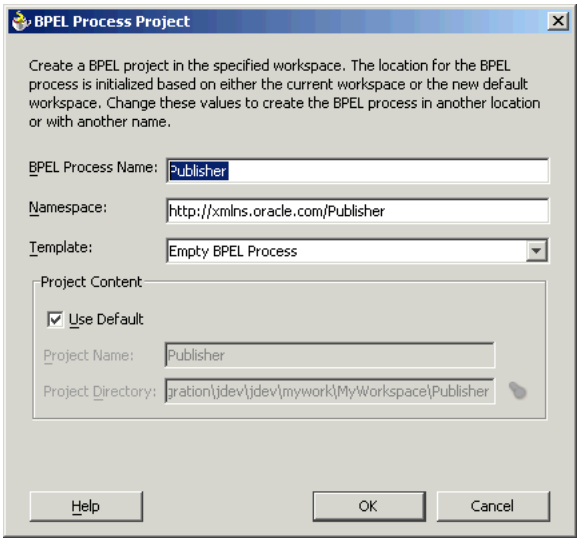
JCA adapter uses the WSDL file to perform the WSIF invocation. You can generate a WSDL file in many ways. This section explains how to generate WSDL files by using the Oracle JDeveloper.

Generating the JCAPublish.wsdl File

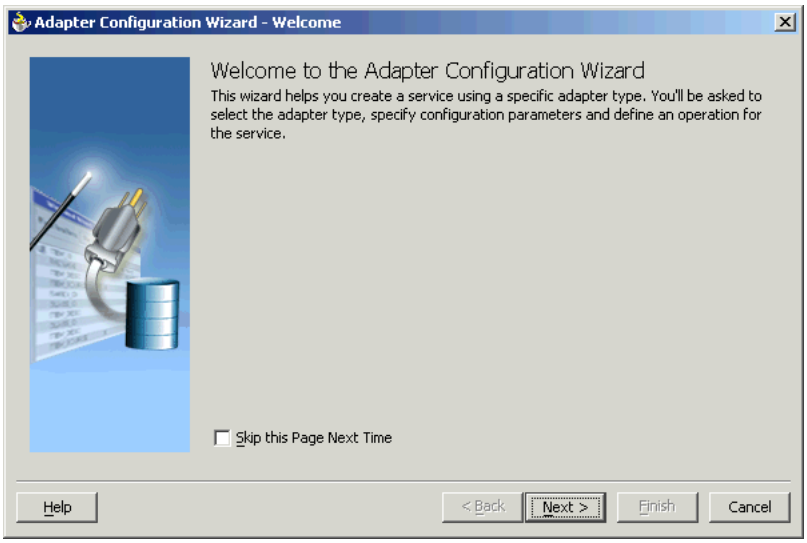
Follow these steps to generate the `JCAPublish.wsdl` file in JDeveloper:

1. From the **File** menu, select **New**. The New Gallery dialog box is displayed.
2. Select **Workspaces** from Categories and then **Workspace** from Items.
3. Click **OK**. The Create Workspace dialog box is displayed.
4. Enter `MyWorkspace` as **Workspace Name**.
5. Deselect **Add a Empty New Project** and click **OK**.
6. Right-click **MyWorkspace** and select **New Project**. The New Gallery dialog box is displayed.
7. Select **Project** from Categories and select **BPEL Process Project** from Items.
8. Click **OK**. The BPEL Process Project dialog box is displayed.
9. Enter `Publisher` as the **BPEL Process Name**.
10. Select **Empty BPEL Process** from Templates.

Figure 3–1 BPEL Process Project dialog box

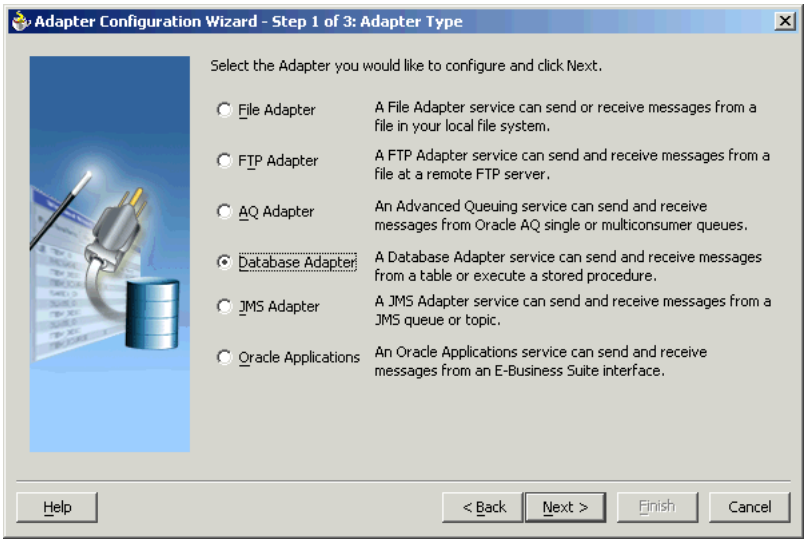


11. Click **OK**.
12. Drag a **Partner Link** activity from the components palette. The Create Partner Link dialog box is displayed.
13. Enter `PublishProcess` in the **Name** field.
14. Click the **Define Adapter Service** icon. The Adapter Configuration Wizard Welcome window is displayed.



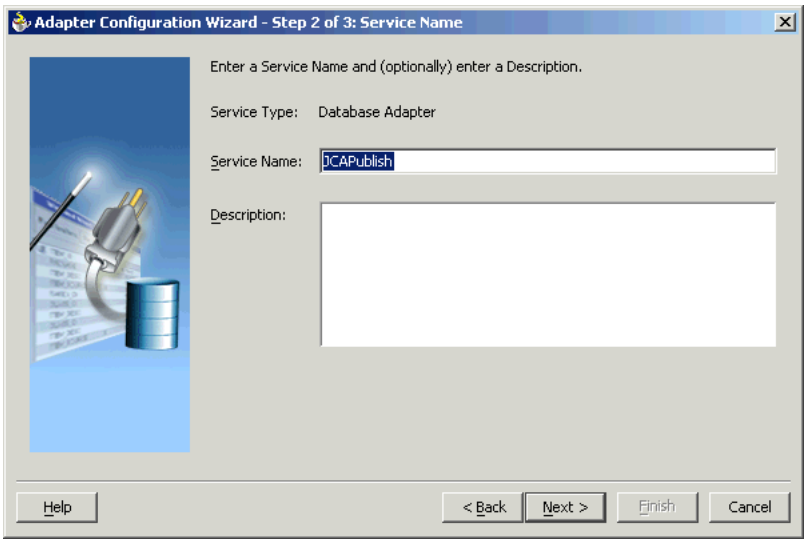
15. Click **Next**. The Adapter Type window is displayed
16. Select **Database adapter**.

Figure 3–2 Selecting as Adapter Type



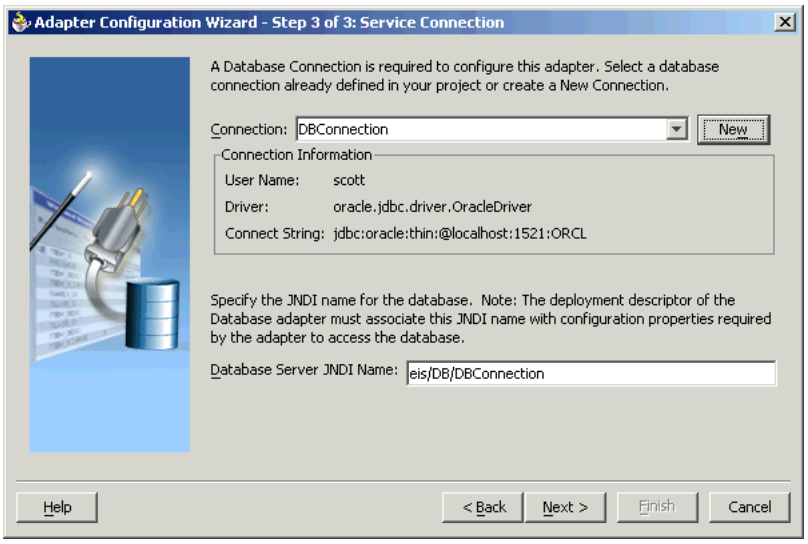
17. Click **Next**. The Service Name window is displayed.
18. Enter a `JCAPublish` in the **Service Name** field.

Figure 3–3 Specifying the Service Name



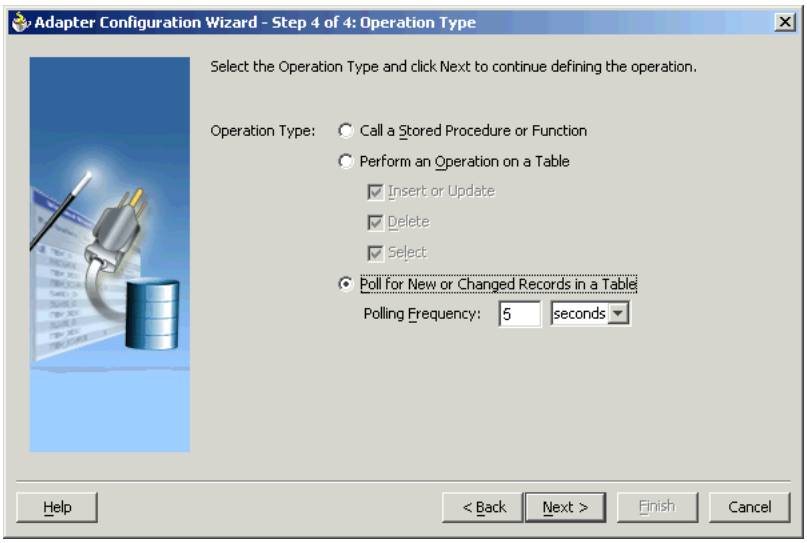
19. Select a database connection from **Connection**. If no connection exists, you can also create a new connection. To create a new connection, click **New** and use the Create Database Connection wizard to create a new connection.

Figure 3–4 Specifying the Service Connection



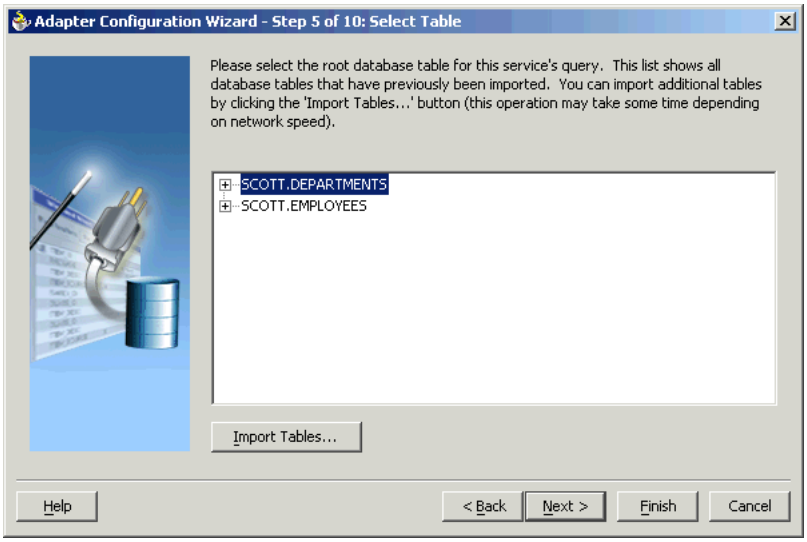
20. Click **Next**. The Operation Type window is displayed.
21. Select **Poll for New or Changed Records in a Table**.

Figure 3–5 Selecting the Operation Type



22. Click **Next**. The Select Table window is displayed.
23. Click **Import Tables**. The Import tables window is displayed.
24. Select **Employees** and **Departments** table from Available and click the right arrow.
25. Click **OK**.
26. Select **Departments** table as the root database table.

Figure 3–6 Selecting a Root Table



27. Click **Next**. The Relationships window is displayed.
28. Click **Next**. The Object Model window is displayed.
29. Click **Finish**.
30. Click **OK** in the Create Partner Link dialog box.
31. Copy the following files from the project directory to the publishing adapter directory:

- toplink_mappings.xml
- JCAPublish.wsdl
- Departments_table.xsd

Generated JCAPublish.wsdl file

After performing these steps, following JCAPublish.wsdl file is generated:

```
<?xml version="1.0" encoding="UTF-8"?>
 <definitions
   name="JCAPublish"
   targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/db/JCAPublish/"
   xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/db/JCAPublish/"
   xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
   xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
   xmlns:pc="http://xmlns.oracle.com/pcbpel/"
   xmlns:top="http://xmlns.oracle.com/pcbpel/adapter/db/top/Publisher"
   xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
      <schema xmlns="http://www.w3.org/2001/XMLSchema">
       <import namespace="http://xmlns.oracle.com/pcbpel/adapter/db/top/Publisher"
        schemaLocation="Departments_table.xsd"/>
      </schema>
  </types>
  <message name="DepartmentsCollection_msg">
        <part name="DepartmentsCollection" element="top:DepartmentsCollection"/>
  </message>
  <portType name="JCAPublish_ptt">
      <operation name="receive">
          <input message="tns:DepartmentsCollection_msg"/>
      </operation>
  </portType>
  <binding name="JCAPublish_binding" type="tns:JCAPublish_ptt">
        <pc:inbound_binding/>
        <operation name="receive">
            <jca:operation
                ActivationSpec="oracle.tip.adapter.db.DBActivationSpec"
                DescriptorName="Publisher.Departments"
                QueryName="JCAPublish"
                PollingStrategyName="DeletePollingStrategy"
                MaxRaiseSize="1"
                MaxTransactionSize="unlimited"
                PollingInterval="5"
                MappingsMetaDataURL="toplink_mappings.xml"
        <input/>
        </operation>
  </binding>
  <!-- Your runtime connection is declared in
        J2EE_HOME/application-deployments/default/DbAdapter/oc4j-ra.xml
        These 'mcf' properties here are from your design time connection and
        save you from having to edit that file and restart the application server
        if eis/DB/DBConnection is missing.
        These 'mcf' properties are safe to remove.
  -->
    <service name="JCAPublish">
    <port name="JCAPublish_pt" binding="tns:JCAPublish_binding">
    <jca:address location="eis/DB/DBConnection"
      UIConnectionName="DBConnection"
      ManagedConnectionFactory="oracle.tip.adapter.db.DBManagedConnectionFactory"
      mcf.DriverClassName="oracle.jdbc.driver.OracleDriver"
```

```
      mcf.PlatformClassName="oracle.toplink.internal.databaseaccess.
       DatabasePlatform
      mcf.ConnectionString="jdbc:oracle:thin:@localhost:1521:ORCL"
                mcf.ConnectionString="jdbc:oracle:thin:@localhost:1521:ORCL"
                mcf.UserName="scott"
                mcf.Password="3E20F8982C53F4ABA825E30206EC8ADE"
/>
    </port>
    </service>
    <plt:partnerLinkType name="JCAPublish_plt" >
        <plt:role name="JCAPublish_role" >
            <plt:portType name="tns:JCAPublish_ptt" />
        </plt:role>
    </plt:partnerLinkType>
 </definitions>
```

Generating the JCASubscribe.wsdl File

Follow these steps to generate another WSDL with name `JCASubscribe`:

1. Click **Applications** and then right-click **MyWorkspace**.
2. Select **New Project**.
3. Select **Project** from Categories and select **BPEL Process Project** from Items.
4. Select **Empty BPEL Process** from Templetes.
5. Click **OK**. The BPEL Process Project dialog box is displayed.
6. Enter `Subscriber` as the **BPEL Process Name**.
7. Click **OK**.
8. Drag a **Partner Link** activity from the components palette. The Create Partner Link dialog box is displayed.
9. Enter `SubscribeProcess` in the **Name** field.
10. Click the **Define Adapter Service** icon. The Adapter Configuration Wizard Welcome window is displayed.
11. Click **Next**. The Adapter Type dialog box is displayed.
12. Select **Databse adapter** and click **Next**. The Service Name dialog box is displayed.
13. Enter a `JCASubscribe` in the **Service Name** field and click **Next**. The Connection dialog box is displayed.
14. Select a database connection from **Connection**. If no connection exists, you can also create a new connection. To create a new connection, click **New** and use the Create Database Connection wizard to create a new connection.
15. Click **Next**. The Operation Type dialog box is displayed.
16. Select **Perform an Operation on a Table**.
17. Select **Insert or Update**.
18. Deselect **Delete** and **Select**.
19. Click **Next**. The Select Table dialog box is displayed
20. Click **Import Tables**. The Import tables dialog box is displayed.
21. Select **NewEmployees** and **NewDepartments** table from Available and click the right arrow.

22. Click **OK**.
23. Select **NewDepartments** table as the root database table and click **Next**. The Relationships dialog box is displayed.
24. Click **Next**.
25. Click **Finish**.
26. Click **OK** in the Create Partner Link dialog box.
27. Copy the following files from the project directory to the subscribing adapter directory:
    - `toplink_mappings.xml`
    - `JCASubscribe.wsdl`
    - `Newdepartments_table.xsd`

    Note: the generated files should be copied to the home directory of the adapter only.

Generated JCASubscribe.wsdl File

After performing these steps, following `JCASubscribe.wsdl` file is generated:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
   name="JCASubscribe"
   targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/db/JCASubscribe/"
   xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/db/JCASubscribe/"
   xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
   xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
   xmlns:pc="http://xmlns.oracle.com/pcbpel/"
   xmlns:top="http://xmlns.oracle.com/pcbpel/adapter/db/top/Subscriber"
   xmlns="http://schemas.xmlsoap.org/wsdl/">
   <types>
     <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://xmlns.oracle.com/pcbpel/adapter/db/top/Subscriber"
                schemaLocation="Newdepartments_table.xsd"/>
     </schema>
   </types>

   <message name="NewdepartmentsCollection_msg">
    <part name="NewdepartmentsCollection" element="top:NewdepartmentsCollection"/>
   </message>

  <message name="Newdepartments_msg">
     <part name="Newdepartments" element="top:Newdepartments"/>
  </message>

  <portType name="JCASubscribe_ptt">
     <operation name="merge">
           <input message="tns:NewdepartmentsCollection_msg"/>
     </operation>
     <operation name="insert">
           <input message="tns:NewdepartmentsCollection_msg"/>
     </operation>
     <operation name="update">
           <input message="tns:NewdepartmentsCollection_msg"/>
     </operation>
```

```
      <operation name="write">
            <input message="tns:NewdepartmentsCollection_msg"/>
      </operation>
      <operation name="queryByExample">
            <input message="tns:Newdepartments_msg"/>
            <output message="tns:NewdepartmentsCollection_msg"/>
      </operation>
   </portType>
   <binding name="JCASubscribe_binding" type="tns:JCASubscribe_ptt">
      <jca:binding />
      <operation name="merge">
        <jca:operation
          InteractionSpec="oracle.tip.adapter.db.DBWriteInteractionSpec"
          DescriptorName="Subscriber.Newdepartments"
          DmlType="merge"
          MappingsMetaDataURL="toplink_mappings.xml" />
        <input/>
      </operation>
         <operation name="insert">
             <jca:operation
                 InteractionSpec="oracle.tip.adapter.db.DBWriteInteractionSpec"
                 DescriptorName="Subscriber.Newdepartments"
                 DmlType="insert"
                 MappingsMetaDataURL="toplink_mappings.xml" />
             <input/>
         </operation>
         <operation name="update">
             <jca:operation
                 InteractionSpec="oracle.tip.adapter.db.DBWriteInteractionSpec"
                 DescriptorName="Subscriber.Newdepartments"
                 DmlType="update"
                 MappingsMetaDataURL="toplink_mappings.xml" />
             <input/>
         </operation>
         <operation name="write">
             <jca:operation
                 InteractionSpec="oracle.tip.adapter.db.DBWriteInteractionSpec"
                 DescriptorName="Subscriber.Newdepartments"
                 DmlType="write"
                 MappingsMetaDataURL="toplink_mappings.xml" />
             <input/>
         </operation>
         <operation name="queryByExample">
             <jca:operation
                 InteractionSpec="oracle.tip.adapter.db.DBReadInteractionSpec"
                 DescriptorName="Subscriber.Newdepartments"
                 IsQueryByExample="true"

MappingsMetaDataURL="toplink_mappings.xml" />
             <input/>
         </operation>
    </binding>
    <!-- Your runtime connection is declared in
        J2EE_HOME/application-deployments/default/DbAdapter/oc4j-ra.xml
        These 'mcf' properties here are from your design time connection and
        save you from having to edit that file and restart the application server
        if eis/DB/DBConnection is missing.
        These 'mcf' properties are safe to remove.
    -->
    <service name="JCASubscribe">
```

```
        <port name="JCASubscribe_pt" binding="tns:JCASubscribe_binding">
            <jca:address location="eis/DB/DBConnection"
                UIConnectionName="DBConnection"

 ManagedConnectionFactory="oracle.tip.adapter.db.DBManagedConnectionFactory"
                mcf.DriverClassName="oracle.jdbc.driver.OracleDriver"

 mcf.PlatformClassName="oracle.toplink.internal.databaseaccess.DatabasePlatform"
                mcf.ConnectionString="jdbc:oracle:thin:@localhost:1521:ORCL"
                mcf.UserName="scott"
                mcf.Password="3E20F8982C53F4ABA825E30206EC8ADE"
            />
        </port>
    </service>
    <plt:partnerLinkType name="JCASubscribe_plt" >
        <plt:role name="JCASubscribe_role" >
            <plt:portType name="tns:JCASubscribe_ptt" />
        </plt:role>
    </plt:partnerLinkType>
</definitions>
```

3.1.3 Modifying the Generated WSDL Files

In the generated WSDL files, you need to specify the abosolute parh of the `toplink_mappings.xml` file in the `MappingsMetaDataURL` parameter such as:

```
MappingsMetaDataURL="file:///C:/Oracle/midtier/integration/interconnect/adapters/myJCAAdapter/toplink_mappings.xml"/>
```

3.1.4 iStudio Design-Time Steps

This section describes design-time steps to be performed in iStudio for implementing the scenario. it contains the following steps:

- Creating a Common View
- Publishing the JCA Event
- Subscribing the JCA Event

3.1.4.1 Creating a Common View

To create a common view:

1. From the **File** menu, select **New Project**. The New Project dialog box is displayed.
2. Enter the `JCAProject` in the **Project Name** field.
3. Click **OK**. The Hub Information dialog box is displayed.
4. Enter information in the following fields:
   - Hub database username: The name of the hub database user.
   - Hub database password: The password associated with the hub database user.
   - Hub database url: The URL of the database in the following format:

     ```
     machine name:port number:database sid
     ```

5. In iStudio, click **JCAProject**, **Common View**, and then click **Business Object**.

6. Right-click **Business Object** and select **New**. The Create Business Object dialog box is displayed.
7. Enter the `JCABusObj` in the **Business Object Name** field.
8. Click **OK**.
9. In the iStudio list, click **JCABusObj** and then click **Event**.
10. Right-click **Event** and select **New**. The Create Event dialog box is displayed.
11. Enter the `CVPublishEvent` in **Event Name** field.
12. Click **Import**, and then **JCA**.
13. Click **Browse**.
14. Navigate to the `JCAPublish.wsdl` file and click **Open**. The Choose Operation from WSDL dialog box is displayed.
15. Select **Receive** Operation.
16. Click **OK**.
17. Click **Save** in the Create Event dialog box.
18. Repeat Step 9 to Step 16 to create an event `CVSubscribeEvent` by using the `JCASubscribe.wsdl` file.

3.1.4.2 Publishing the JCA Event

To publish the JCA event:

1. In iStudio, click **JCAProject**, and then right-click **Applications**.
2. Select **New**. The Create Application dialog box is displayed.
3. Enter the `JCAPublisher` in the **Application Name** field and click **OK**.
4. In iStudio, click **Applications**, **JCAPublisher**, and then **Published Events**.
5. Right-click **Published Events** and select **New**. The Publish Wizard- Select an event dialog box is displayed.
6. Perform the following steps:
   a. Select **JCA** from the Message Type list.
   b. Click **Business Objects**, **JCABusObj**, and then **CVPublishEvent** from the Select an Event list.
   c. Click **Next**. The Publish Wizard- Define Application View dialog box is displayed.
7. Click **Import** and then **WSDL**. The Choose Operations From WSDL dialog box is displayed.
8. Click **Browse** and select `JCAPublish.wsdl`.
9. Click **Open**. The Choose Operation from WSDL dialog box is displayed.
10. Select **Receive** operation.
11. Click **Next**. The Publish Wizard- Define Mapping dialog box is displayed.
12. Click **Define Mapping**. The Mapping Parameters dialog box is displayed.
13. Perform the following mappings:
    a. In JCAPublisher view, select **Departments[ ]**.

b. In Transformations, select **ObjectCopy**.

c. In Common View, select **Departments[ ]**.

d. Click **Add**.

e. Click **OK**.

14. Click **Finish**.

3.1.4.3 Subscribing the JCA Event

To subscribe to the JCA event:

1. In iStudio, click **JCAProject**, and then right-click **Applications**.
2. Select **New**. The Create Application dialog box is displayed.
3. Enter the `JCASubscriber` in the **Application Name** field and click **OK**.
4. In iStudio, click **Applications**, **JCASubscriber**, and then **Subscribed Events**.
5. Right-click **Subscribed Events** and select **New**. The Subscribe Wizard- Select an event dialog box is displayed.
6. Perform the following steps:
   1. Select **JCA** from the **Message Type** list.
   2. Click **Business Objects**, **JCABusObj**, and then **CVSubscribeEvent** from the **Select an Event** list.
   3. Click **Next**. The Subscribe Wizard- Define Application View dialog box is displayed.
7. Click **Import** and then **WSDL**. The Choose Operations From WSDL dialog box is displayed.
8. Click **Browse** and select `JCASubscribe.wsdl`.
9. Click **OK**. The Choose Operation from WSDL dialog box is displayed.
10. Select **Insert** operation.
11. Click **Next**. The Publish Wizard- Define Mapping dialog box is displayed.
12. Click **Define Mapping**. The Mapping Parameters dialog box is displayed.
13. Perform the following mappings:
    a. In Common View, select **Newdepartments[ ]**.
    b. In Transformations, select **ObjectCopy**.
    c. In JCASubscriber view, select **Newdepartments[ ]**.
    d. Click **Add**.
    e. Click **OK**.
14. Click **Next**.
15. Click **Finish**.

3.2 Run-Time Concepts

This section describes how JCA adapter works at run time. It contains the following sections:

- JCA Adapter as the Publishing Adapter
- JCA Adapter as the Subscribing Adapter

3.2.1 JCA Adapter as the Publishing Adapter

When JCA adapter acts as a publishing adapter, it communicates with the JCA compliant resource adapter through adapter framework. When the adapter is started, the bridge performs endpoint activation to register a callback with the adapter framework. Once this callback is registered, the bridge is ready to start processing the messages it receives from the adapter framework through the registered callback. The resource adpater, in turn, is getting the messages from the Enterprise Information System (EIS).

3.2.2 JCA Adapter as the Subscribing Adapter

When the JCA adapter acts as a subscribing adapter, the communication between the bridge and the resource adapter happens through Web Services Activation Framework (WSIF). The bridge uses the WSDL file to perform the WSIF invocation. The bridge obtains the information about the WSDL file to be used from the metadata that was defined in iStudio during design time. This metadata also contains information about which port type and operation is to be used from the WSDL file. When the bridge receives the message from agent, it uses this information for WSIF invocation.

3.3 Starting the JCA Adapter

The process for starting the adapter varies based on the operating system.

- To start the JCA adapter on UNIX:
  1. Change to the directory containing the start script.

     ```
     cd ORACLE_HOME/integration/interconnect/adapters/Application
     ```

  2. Type **start** and press **Enter**.
- To start the JCA adapter from Services on Windows:
  1. Access the Services window from the Start menu. The Services window is displayed.
  2. Select the *OracleHomeOracleASInterConnectAdapter-Application* service.
  3. Start the service based on the operating system. For example, in Windows 2000, right-click the service and select **Start** from the context menu.

Note: You can also start and stop the JCA adapter using the IC Manager or Enterprise Manager. Refer to *Oracle Application Server Integration InterConnect User's Guide* for more details.

3.3.1 Log File of the JCA Adapter

You can verify the start up status of the JCA adapter by viewing the `log.xml` files. The files are located in the time-stamped subdirectory of the `log` directory of the JCA adapter. Subdirectory names have the following form:

```
timestamp_in_milliseconds
```

3.4 Stopping the JCA Adapter

The process for stopping the adapter varies based on the operating system.

- To stop the JCA adapter on UNIX:
  1. Change to the directory containing the stop script.

     ```
     cd ORACLE_HOME/integration/interconnect/adapters/Application
     ```

  2. Type **stop** and press **Enter**.
- To stop the JCA adapter from Services on Windows:
  1. Access the Services window from the Start menu. The Services window is displayed.
  2. Select the *OracleHomeOracleASInterConnectAdapter-Application* service.
  3. Stop the service based on the operating system.

  You can verify the stop status of the JCA adapter by viewing the `log.xml` files. These files are located in the time-stamped subdirectory of the `log` directory of the JCA adapter.

A

Frequently Asked Questions

This appendix provides answers to frequently asked questions about the JCA adapter.

A.1 While starting the adapter, I get the following message. What could be the reason?

Error Message:

```
Error happened when reading wsdl at "Datatypes_Read.wsdl", because "WSDLExceptio
n: faultCode=PARSER_ERROR: Error reading file at: file:/C:/Oracle/MidHome/integr
ation/interconnect/adapters/JCAAPP/Datatypes_Read.wsdl: C:\Oracle\MidHome\integr
ation\interconnect\adapters\JCAAPP\Datatypes_Read.wsdl (The system cannot find t
he file specified)".
Make sure wsdl exists at that URL and is valid.
        at com.collaxa.cube.ws.wsdl.WSDLUtils.readWSDL(WSDLUtils.java:164)
        at oracle.oai.agent.adapter.service.ServiceBridge.start(ServiceBridge.ja
va:210)
        at oracle.oai.agent.proxy.AgentProxy.go(AgentProxy.java:672)
        at oracle.oai.agent.proxy.AgentProxy.start(AgentProxy.java:706)
        at oracle.oai.agent.proxy.AgentProxy.<init>(AgentProxy.java:415)
        at oracle.oai.agent.service.AgentService.run(AgentService.java:43)
        at java.lang.Thread.run(Thread.java:534)
Service started successfully.
```

Solution

In iStudio, you imported the WSDL while defining the application view in the publish, subscribe, invoke, and implement wizard. The imported WSDl file should be present

in the home directory of the adapter. If the WSDL file is not present in the home directory of the adapter, you get this error message.

A.2 Adapter gives the Address Location Exception. What could be the reason?

The value of the `jca:address location` parameter in the WSDL file and value of the `connection1_name` parameter in the `adapter.ini` file is not same. These values should be same. For example:

WSDL File

```
<service name="Datatypes_Read">
  <port name="Datatypes_Read_pt" binding="tns:Datatypes_Read_binding">
    <jca:address location="eis/DB/DB" UIConnectionName="DB"
     ManagedConnectionFactory="oracle.tip.adapter.db.DBManagedConnectionFactory"
     mcf.DriverClassName="oracle.jdbc.driver.OracleDriver"
     mcf.PlatformClassName="oracle.toplink.internal.databaseaccess.
     DatabasePlatform"
     mcf.ConnectionString="jdbc:oracle:thin:@localhost:1521:orcl"
     mcf.UserName="dbapp"
     mcf.Password="7347B141D0FBCEA0B136BF4D7C3600BC"
    />
  </port>
</service>
```

Adapter.ini File

```
connection1_name=eis/DB/DB
connection1_mcf_class=oracle.tip.adapter.db.DBManagedConnectionFactory
connection1_mcf_params={ConnectionString=jdbc:oracle:thin:@localhost:1521:orcl,
DriverClassName=oracle.jdbc.driver.OracleDriver, UserName=scott, Password=tiger}
```

A.3 Adapter fails while processing messages after I changed input part name (or output part name or operation name) in the WSDL file.

If you modify the values of input part, output part, operation name or the xsd in the WSDL file, then you have to re-define the design time by importing the edited WSDL again.

A.4 How do I know the JCA adapter started properly?

View the `log.xml` file located in the time-stamped subdirectory of the JCA adapter `logs` directory.

On...	Change to...
UNIX	*ORACLE_HOME*/integration/interconnect/adapters/*Application*/logs/*timestamp_in_milliseconds*
Windows	*ORACLE_HOME*\integration\interconnect\adapters*Application*\logs*timestamp_in_milliseconds*

A.5 The JCA adapter did not start properly: what is wrong?

View the exceptions in the adapter log file (`log.xml`).

The exceptions should provide information about what went wrong. It is possible that the JCA adapter is unable to connect to the repository. Ensure the repository is started properly. The JCA adapter will connect to the repository once it is started properly. You do not need to restart the Adapter.

A.6 Is it possible to edit the JCA adapter configuration settings created during installation?

Yes, edit the parameters in the `adapter.ini` file in the following directory:

Platform	Directory
UNIX	*ORACLE_HOME*`/integration/interconnect/adapters/`*Application*`/`
Windows	*ORACLE_HOME*`\integration\interconnect\adapters\`*Application*`\`

Note: All configuration parameters with the exception of `bridge_class` can be edited more than once.

See Also: "hub.ini Files" on page 2-8 for parameter information

A.7 When I change an element in iStudio, such as mappings, it seems like the JCA adapter uses old information. What is happening?

The JCA adapter caches information from iStudio. The information is stored in the repository locally. If you change something in iStudio and want to view the change in the runtime, then you need to stop the JCA adapter, delete the JCA adapter cache files, and restart the JCA adapter.

The JCA adapter has a persistence directory which is located in the JCA adapter directory. Deleting this directory when the JCA adapter has been stopped should make it obtain the new metadata from the repository when started.

A.8 How do I secure my passwords?

OracleAS Integration InterConnect uses Oracle Wallet Manager to maintain system passwords. When you install OracleAS Integration InterConnect, Oracle Wallet Manager is also installed and a password store is created. All passwords used by OracleAS Integration InterConnect components are stored in the password store. The password is stored in the Oracle Wallet in the following format:

```
ApplicationName/password
```

The `ApplicationName` is the name of the application, which is extracted from the `adapter.ini` file of the corresponding adapter. In the `adapter.ini` file, the `application` parameter specifies the `ApplicationName` to which this adapter

connects. The password for the application is also retrieved from the `adapter.ini` file.

The number of entries is dependent on the type of adapter. For example, DB Adapter needs two entries whereas AQ Adapter needs only one entry. The following table lists the entries that will be created for each adapter:

Adapter	Entry In Oracle Wallet
AQ	*ApplicationName*/aq_bridge_password
HTTP	*ApplicationName*/http.sender.password
HTTP	*ApplicationName*/sender.wallet_password
SMTP	*ApplicationName*/smtp.receiver.password
MQ	*ApplicationName*/mq.default.password
FTP	*ApplicationName*/file.sender.password
FTP	*ApplicationName*/file.receiver.password
DB	*ApplicationName*/db_bridge_schema1_password
DB	*ApplicationName*/db_bridge_schema1_writer_password

You can create, update, and delete passwords using the `oraclewallet` command. When you run the command, it prompts you for the admin password.

You can use the following commands to manage your passwords:

- List all passwords in the store

  ```
  oraclewallet -listsecrets
  ```

- Create a password

  ```
  oraclewallet -createsecret passwordname
  ```

  For example, to create a password for the hub schema:

  ```
  oraclewallet -createsecret hub_password
  ```

- View a password

  ```
  oraclewallet -viewsecret passwordname
  ```

  For example, to view the password for the hub schema:

  ```
  oraclewallet -viewsecret hub_password
  ```

- Update a password

  ```
  oraclewallet -updatesecret passwordname
  ```

  For example, to update the password for the hub schema:

  ```
  oraclewallet -updatesecret hub_password
  ```

- Delete a password

  ```
  oraclewallet -deletesecret passwordname
  ```

  For example, to delete the password for the hub schema:

```
oraclewallet -deletesecret hub_password
```

Index

A

C

D

E

H

I

J

L

O

P

R

S

T