





Note

Before using this information and the product it supports, read the information in "Notices" on page 95.

16 January 2007

This edition applies to version 6, release 1, modification 0 of IBM WebSphere Adapter for Oracle E-Business Suite and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email <mailto://doc-comments@us.ibm.com>. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2006, 2007. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Overview of WebSphere

Adapter for Oracle E-Business Suite . . . 1

| | |
|---|---|
| New in this release | 1 |
| Hardware and software requirements | 1 |
| Technical overview of the Adapter for Oracle E-Business Suite | 2 |
| Adapter architecture | 2 |
| Outbound processing sample applications | 4 |
| Inbound processing sample application | 7 |
| Standards compliance | 8 |
| Accessibility | 8 |
| Internet Protocol Version 6 (IPv6) | 9 |

Chapter 2. Planning for adapter

implementation 11

| | |
|--|----|
| Before you begin | 11 |
| Deployment options | 11 |
| WebSphere Adapters in clustered environments | 14 |

Chapter 3. Samples 17

| | |
|--|----|
| Database and applications user account requirements | 17 |
| Accessing the sample files | 17 |
| Sample 1: Sending outbound data to the Oracle database | 18 |
| Preparing for outbound processing | 18 |
| Creating the project. | 19 |
| Configuring the module for outbound processing | 19 |
| Deploying the module for testing | 30 |
| Testing the Create operation | 31 |
| Testing the Retrieve operation | 38 |
| Clearing the sample content | 39 |
| Sample 2: Receiving inbound data from the database | 39 |
| Preparing for inbound processing | 40 |
| Setup within Oracle E-Business Suite | 41 |

| | |
|--|----|
| Creating the project. | 44 |
| Configuring the module for inbound processing | 45 |
| Creating an inbound component | 54 |
| Deploying the module for testing | 57 |
| Testing the assembled adapter application | 57 |
| Clearing the sample content | 59 |
| Sample 3: Calling an Oracle E-Business Suite API for outbound processing | 60 |
| Creating the project. | 60 |
| Configuring the module for outbound processing for API calls | 60 |
| Deploying the module for testing | 66 |
| Testing the assembled adapter application | 67 |
| Clearing the sample content | 69 |
| Sample 4: Importing a business object using the Business Event System and workflow | 69 |
| Preparing for outbound processing | 70 |
| Setup within Oracle E-Business Suite | 71 |
| Creating the project. | 74 |
| Configuring the module for outbound processing | 74 |
| Deploying the module for testing | 84 |
| Testing the Create operation | 84 |
| Clearing the sample content | 89 |
| Troubleshooting information for the sample applications | 90 |
| Viewing the sample adapter artifacts | 90 |

Chapter 4. Reference information . . . 93

| | |
|-------------------------------|----|
| Related information | 93 |
|-------------------------------|----|

Notices 95

| | |
|---|----|
| Programming interface information | 97 |
| Trademarks and service marks | 97 |

Index 99

Chapter 1. Overview of WebSphere Adapter for Oracle E-Business Suite

WebSphere Adapter for Oracle E-Business Suite is a solution that uses existing WebSphere components to enable bidirectional communication between Oracle E-Business Suite and the WebSphere Business Integration system. This solution uses WebSphere Adapter for JDBC to interact with the database components in Oracle E-Business Suite to process data to and from external sources.

WebSphere Adapter for Oracle E-Business Suite solution is not a traditional resource adapter. The solution uses Adapter for JDBC to communicate with the Oracle E-Business Suite database, and it shows how to set up the interaction between them through the use of sample applications. Instructions explain how to enable information flows between Oracle E-Business Suite and WebSphere Process Server or WebSphere Enterprise Service Bus, both for inbound event and outbound request processing.

Sample content and artifacts that compose this solution help speed up the process of enabling integration with Oracle E-Business Suite. Some of the sample applications provide a guideline for database integration using the Receivables module with an Oracle Customer and an Oracle API. You can apply what you learn from the samples to different Oracle E-Business Suite modules that you choose, for example, General Ledger or Payables.

New in this release

WebSphere Adapter for Oracle E-Business Suite, Version 6.1.0 provides enhancements to the adapter.

The following new or enhanced features are provided:

- Support for Oracle E-Business Suite Version 12.
- The sample applications have been changed as needed based on updates to the Adapter for JDBC, Version 6.1.0. The changes include a capability of the external service wizard to create parent-child relationships between business objects. Previously this was performed by using the business object editor of WebSphere Integration Developer.

Updates to this information are made available at the WebSphere Adapters product support Web site. To read updated or additional information, see:
<http://www.ibm.com/software/integration/wbiadapters/support/>.

Hardware and software requirements

The hardware and software requirements for WebSphere Adapters are documented on the IBM® Web site at the location below.

Hardware and software requirements for WebSphere Adapters:
<http://www.ibm.com/support/docview.wss?uid=swg27006249>

Additional information

The following links provide additional information you might need to configure and deploy your adapter:

- The compatibility matrix for WebSphere Business Integration Adapters and WebSphere Adapters identifies the supported versions of required software for your adapter. To view this document, go to the WebSphere Adapters support page and click the link for the compatibility matrix under **Planning upgrades**: <http://www.ibm.com/software/integration/wbiadapters/support/>.
- Technotes for WebSphere Adapters document workarounds and additional information not included in the product documentation. To view the technotes for your adapter, go to the following Web page, select the name of your adapter from the **Product category** list, and click the search icon: <http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>.

Technical overview of the Adapter for Oracle E-Business Suite

WebSphere Adapter for Oracle E-Business Suite enables connectivity between the WebSphere system and Oracle E-Business Suite for both outbound request and inbound event processing. It uses WebSphere Adapter for JDBC and Oracle E-Business Suite database mechanisms together for processing data. The adapter comprises four sample applications, three for outbound request processing and one for an inbound event.

Adapter architecture

The event processing mechanism for Oracle E-Business Suite is used with WebSphere Adapter for JDBC to process inbound events from the Oracle database. The Adapter for JDBC processes outbound requests to create, update or retrieve data in the Oracle database.

Overview

Oracle E-Business Suite is designed to work with the Oracle database. Oracle E-Business Suite has its own event processing mechanism, the Business Event System (BES). An event could be, for example, the creation or update of a customer record in the database, resulting from an occurrence in an Internet or intranet application or program in the business integration system. The Business Event System enables raising events in the Oracle database and then using subscriptions that consume the events and perform an action, which is based on your preset configuration. To raise an event identifies the event to the Oracle event manager, which can initiate subscriptions related to that event. Subscriptions identify what actions to take. Actions can include populating a table or an Oracle AQ queue with information from the event, or running some custom SQL or Java™ code.

When an event is raised, the Business Event System searches for subscriptions that use that event type, and it processes the actions defined in those subscriptions. The Oracle user has flexibility in defining what actions can be taken. For example, an event could be designed to trigger based on updates to a Customer business object in the Oracle database. The action could be to populate the key values for the changed object into an event table for consumption by another Service Component Architecture (SCA) component.

WebSphere Adapter for JDBC is used to handle interfacing with database artifacts. It enables direct SQL table calls and interaction with database stored procedures

and APIs. This solution uses the built-in event processing support inherent in Oracle E-Business Suite to process events that can be consumed by the Adapter for JDBC. This solution also uses the Adapter for JDBC to populate information in the Oracle database from applications or components, and to call stored procedures to process data into the Oracle system.

Inbound architecture

In Sample 2, a sample application for inbound event processing, the Adapter for JDBC is used to poll an event table created in the Oracle application's database. The adapter uses the event information from the event table, such as the object key and type, to retrieve business object information from the Oracle base tables. For information on the architecture of the Adapter for JDBC, refer to *WebSphere Adapters: Adapter for JDBC User Guide*.

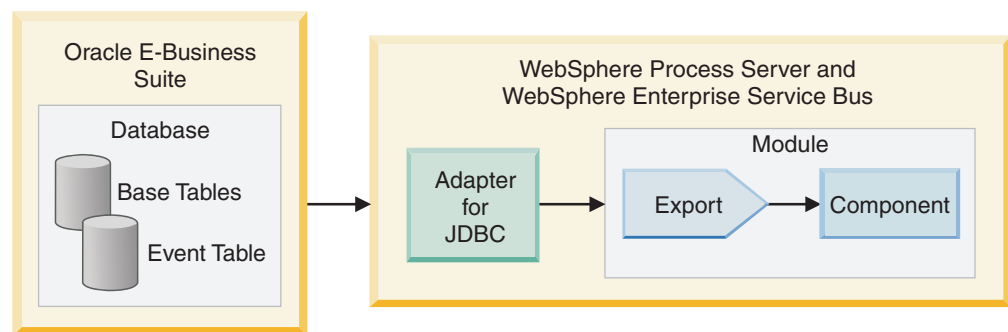


Figure 1. Inbound event processing

Oracle's internal Business Event System enables an Oracle process to recognize when a business object is changed and to populate that information to the JDBC event store. See the sample application for specific details on use of the Business Event System. For more information on the Business Event System, refer to your documentation for Oracle Applications, specifically the *Workflow Administrator's Guide*.

Outbound architecture

For outbound request processing, as in Sample 1, the Adapter for JDBC is used to populate object information to the interface tables in the Oracle application. The interface tables are standard tables used to enable outbound information to be processed into the Oracle base tables. Oracle supports this process to help prevent undesirable content changes that could occur from direct updates to the base tables.

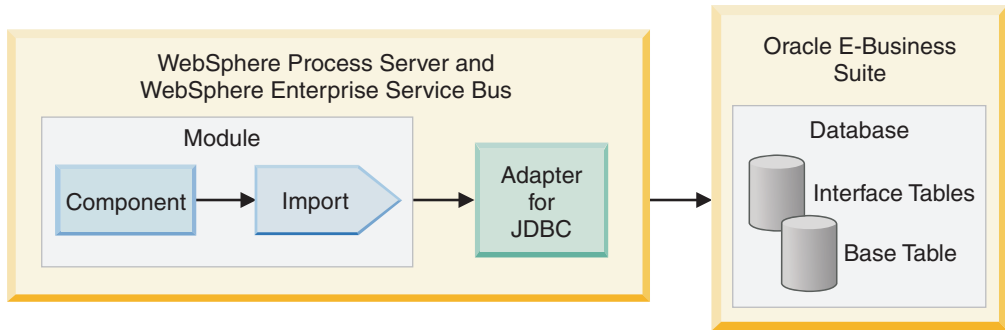


Figure 2. Outbound request processing

The Adapter for JDBC populates the interface tables, then it is used to call a stored procedure that invokes a standard function call included in the Oracle application. The Oracle database processes the data in the interface tables and populates it to the base tables.

To retrieve data from the Oracle application, a Retrieve operation using the Adapter for JDBC can be performed against the base tables, because no data is being modified in the tables.

In addition, as in Sample 3, the Adapter for JDBC can be used to directly call Oracle’s built-in database application programming interfaces (APIs). However, the JDBC driver limits parameters to simple, or non-record, data types when an API is called directly from the Adapter for JDBC. Because the majority of the Oracle APIs use record parameters, these APIs can be wrapped inside of stored procedures, which can then be called by the Adapter for JDBC.

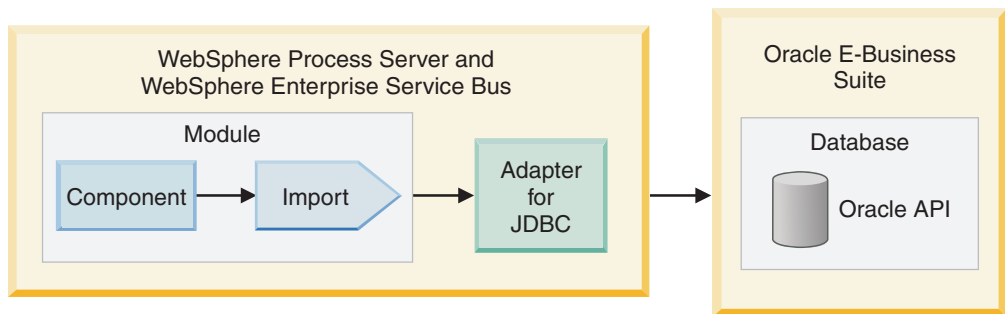


Figure 3. Outbound processing with an API call

In Sample 4, a business object is imported using the Business Event System and a workflow. The Adapter for JDBC sends an outbound request to Oracle E-Business Suite to create customer data in an Oracle custom event table. The Oracle Business Event System triggers a workflow from that event. The workflow creates the customer record in Oracle E-Business Suite, and also handles errors during the process.

Outbound processing sample applications

Some outbound processing sample applications are provided with this solution. Sample 1 uses Oracle database interface tables, which is the standard outbound scenario for the Oracle E-Business Suite. Sample 3 uses API calls, while Sample 4 makes use of a workflow.

Outbound processing using Interface tables

The first outbound request processing sample, Sample 1, has two parts:

- Creating a business object by using a Create operation
- Retrieving an object by using a Retrieve operation

The Oracle database permits the retrieval of data from the applications's base tables. The business objects used for the Retrieve operation reflect the base tables' representation of the data. The Oracle database does not permit the manual modification of data in the base tables. Thus, this sample uses the interface tables when the Adapter for JDBC sends business objects that change the content in the Oracle database. This is the standard outbound request processing scenario. After the interface tables are updated with the business object data, a function is called by the adapter, through a stored procedure, that moves the data from the interface to the base tables.

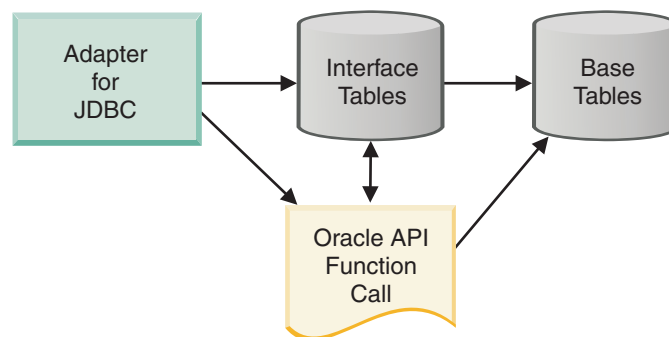


Figure 4. Create scenario

The Adapter for JDBC is used to populate the interface tables with business object data. Because updates cannot be made directly to the Oracle base tables, the Update operation cannot be used. The Create operation is used to process both new and changed data through the Oracle interface tables. The business object data can be moved into the Oracle base tables by invoking an internal, standard Oracle function. The function is called through a stored procedure to move the data to the base tables using AfterCreateSP application-specific information on the business object.

The Create and Retrieve operations use different sets of business objects because they represent different tables in the Oracle database. This sample application includes the wrapper stored procedure for the function call to move the data to the base tables, plus the sample content to use for the procedure.

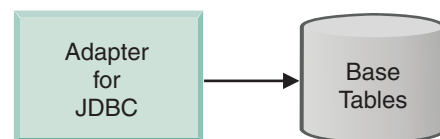


Figure 5. Retrieve scenario

Outbound processing using API call

Sample 3 shows how the Adapter for JDBC can call an Oracle E-Business Suite API directly. The JDBC driver limits parameters to simple, non-record data types when the Adapter for JDBC calls an API directly. Most Oracle APIs include record

parameters. API calls that use record type parameters can be called through the Adapter for JDBC only if they are wrapped so that the call to the wrapper stored procedure does not use the record type parameter. The API call for this sample uses simple data type parameters.

This sample uses the Adapter for JDBC to call an Oracle API directly, which creates a concurrent program executable in the Oracle database. The content for this sample includes examples of the API call.

Outbound processing using the Business Event System and a workflow

Sample 4 demonstrates use of a workflow to create the customer record in the database, based on an event in the Business Event System. The Adapter for JDBC uses a Create operation to insert customer data into an Oracle custom event table. The adapter invokes a stored procedure to create the custom event in the Business Event System. The Business Event System has a built-in subscription to this event. It builds a workflow that creates a customer record in Oracle E-Business Suite.

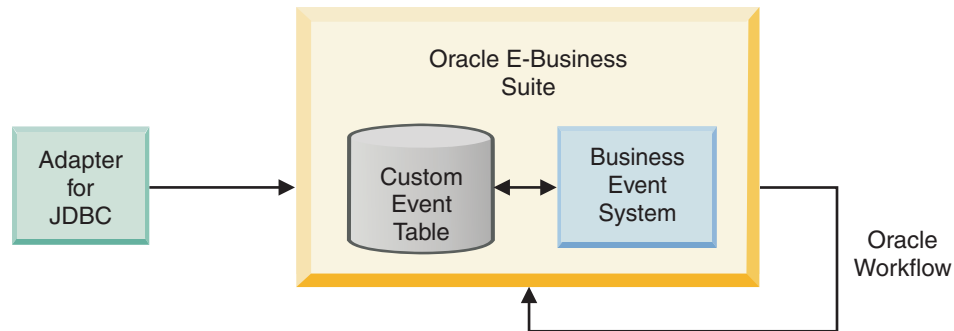


Figure 6. Outbound processing using a workflow

Related tasks

“Sample 1: Sending outbound data to the Oracle database” on page 18
With WebSphere Adapter for Oracle E-Business Suite, you can create data in an Oracle application and retrieve data from the application. In this scenario, you use WebSphere Adapter for JDBC as the integration method to create a customer record in the Oracle E-Business Suite Financials database and to retrieve that customer information from the Financials database.

“Sample 3: Calling an Oracle E-Business Suite API for outbound processing” on page 60

The adapter can call directly any Oracle E-Business Suite API that uses simple data parameters. Follow the steps in this sample to call an API that creates a concurrent program executable for outbound processing. By performing this scenario, you can see how to structure simple data API calls between the adapter and an Oracle E-Business Suite database.

“Sample 4: Importing a business object using the Business Event System and workflow” on page 69

The Adapter for JDBC sends an outbound request to Oracle E-Business Suite to create customer data in Oracle’s custom event table. From that event, the Oracle Business Event System triggers a workflow that creates the customer record in Oracle E-Business Suite. The workflow also handles errors during this process.

Inbound processing sample application

Sample 2 uses the Oracle Business Event System to send inbound customer data from Oracle E-Business Suite to an event table where the Adapter for JDBC can read it. The Adapter for JDBC picks up the event from the event table and retrieves the business object from the Oracle database based on the information provided in the event. The Adapter for JDBC then processes the business object from the Oracle database. The content included for this sample is described in this section.

This sample shows the processing of inbound customer data from Oracle E-Business Suite. The Oracle built-in Business Event System raises the event and populates the necessary values (new and updated customer data) to the Adapter for JDBC event table. The custom event table is created within Oracle and is used by the Business Event System.

A custom concurrent program searches for updates to the Customer table (Oracle base table) and raises events if any new or modified customer records are found. You can configure the concurrent program to run at set intervals or be manually invoked, or be invoked through a stored procedure. For this sample, the concurrent program is configured for set intervals. To save time while the sample steps are performed, the concurrent program is invoked manually. Once the data reaches the event table, the Adapter for JDBC polls the table for the object_key, object_name, and object_function. Then it processes the business object from the Oracle database application.

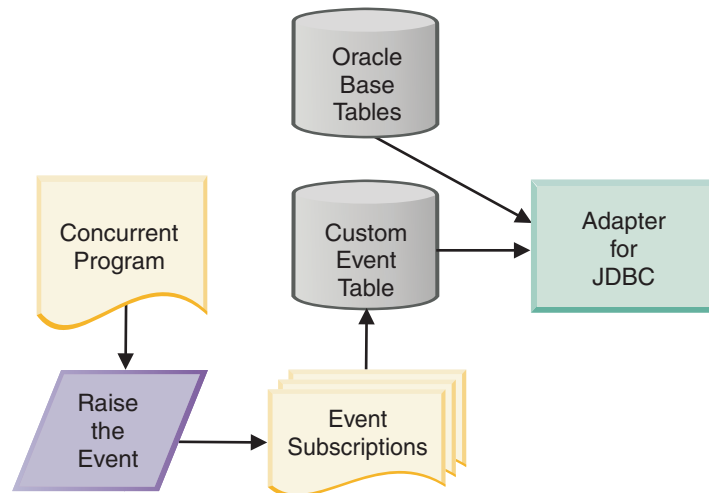


Figure 7. Inbound event processing

For the Business Event System, an event is defined for updates to the Customer table. When changes are saved, the event is raised, and the Oracle Event Manager searches for subscriptions for that event type. The sample uses a custom subscription that recognizes the change to the customer data. The event is passed to the subscription, which processes the data based on the action defined in the subscription. The event information populates the Adapter for JDBC's event table. The adapter queries that table and uses the event information to perform a Retrieve operation from the business object's base tables.

The content for this sample includes the stored procedures necessary to set up the action and the event tables.

Note: Although it is possible to use a trigger on the customer table to handle the population of the event table, this is not recommended by Oracle.

Related tasks

“Sample 2: Receiving inbound data from the database” on page 39

In this sample, you use the event system within Oracle E-Business Suite to populate an event table with customer events, and you also use WebSphere Adapter for JDBC to poll for events and retrieve customer data. By performing this scenario, you can see how to set up information flows between Oracle E-Business Suite and WebSphere Process Server or WebSphere Enterprise Service Bus for inbound events.

Standards compliance

This product is compliant with several government and industry standards, including accessibility standards and Internet protocol standards.

Accessibility

IBM strives to provide products with usable access for everyone, regardless of age or ability. WebSphere Adapters are fully accessible and section 508-compliant. Accessibility features enable users with physical disabilities, such as restricted mobility or limited vision, to operate software products successfully. These features are built into the installation and administration features of WebSphere Adapters.

Administration

The run time administrative console is the primary interface for deployment and administration of enterprise applications. The console is displayed within a standard Web browser. By using an accessible Web browser, such as Microsoft® Internet Explorer or Netscape Browser, you are able to:

- Use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen
- Use voice recognition software, such as IBM ViaVoice®, to enter data and to navigate the user interface
- Operate features by using the keyboard instead of the mouse

You can configure and use product features by utilizing standard text editors and scripted or command-line interfaces instead of the graphical interfaces that are provided.

When appropriate, the documentation for specific product features contains additional information about the accessibility of the features.

External service wizard

The external service wizard is the primary component used to create modules. This wizard, which is implemented as an Eclipse plug-in that is available through WebSphere Integration Developer, is fully accessible.

Keyboard navigation

This product uses standard Microsoft Windows® navigation keys.

IBM and accessibility

See the *IBM Accessibility Center* web site <http://www.ibm.com/able/> for more information about the commitment that IBM has to accessibility.

Internet Protocol Version 6 (IPv6)

WebSphere Process Server and WebSphere Enterprise Service Bus rely on WebSphere Application Server for Internet Protocol Version 6 (IPv6) compatibility.

IBM WebSphere Application Server, version 6.1.0 and later support dual-stack Internet Protocol Version 6.0 (IPv6).

For more information about this compatibility in WebSphere Application Server, see IPv6 support in the <http://www.ibm.com/software/webservers/appserv/was/library/>.

For more information about IPv6, see <http://www.ipv6.org>.

Chapter 2. Planning for adapter implementation

Before using WebSphere Adapter for Oracle E-Business Suite, make sure you understand the experience you need and the server environment in which it runs. Learn the considerations for deploying the adapter in your server environment, and find out how to improve the performance and availability of the adapter by using a clustered server environment.

Before you begin

Before you begin to set up and use the adapter, you should possess a thorough understanding of business integration concepts, the Oracle E-Business Suite environment, WebSphere Adapter for JDBC, and the features and capabilities of WebSphere Integration Developer and WebSphere Process Server or WebSphere Enterprise Service Bus.

To use WebSphere Adapter for Oracle E-Business Suite you should understand and have experience with the following concepts, tools, and tasks:

- The business requirements of the solution you are building.
- The Oracle E-Business Suite environment in which you are working.
- Business integration concepts and models, including the Service Component Architecture (SCA) programming model.
- The capabilities and requirements of the server you plan to use for the integration solution. You should know how to configure and administer the host server and how to use the administrative console to set and modify property definitions, configure connection factories, and manage events.
- The tools and capabilities provided by WebSphere Integration Developer. You should know how to use these tools to create modules, wire and test components, and complete other integration tasks.
- The capabilities provided by WebSphere Adapter for JDBC, which is used to communicate with the Oracle database. Become familiar with the Adapter for JDBC; for example, you can read the documentation and try using it with some simple database tables.

Deployment options

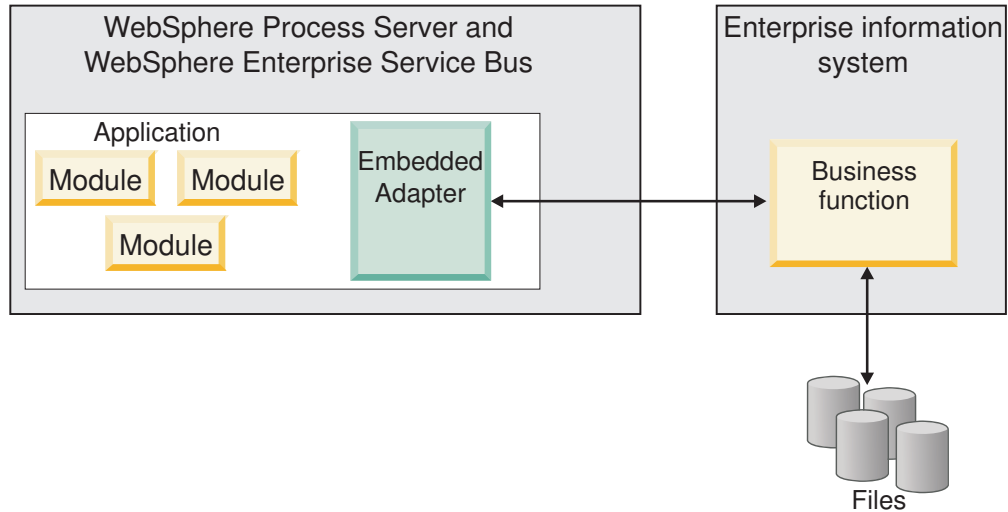
You can choose to embed the adapter to be part of the deployed application or you can choose to deploy the RAR file stand-alone.

The deployment options are described below:

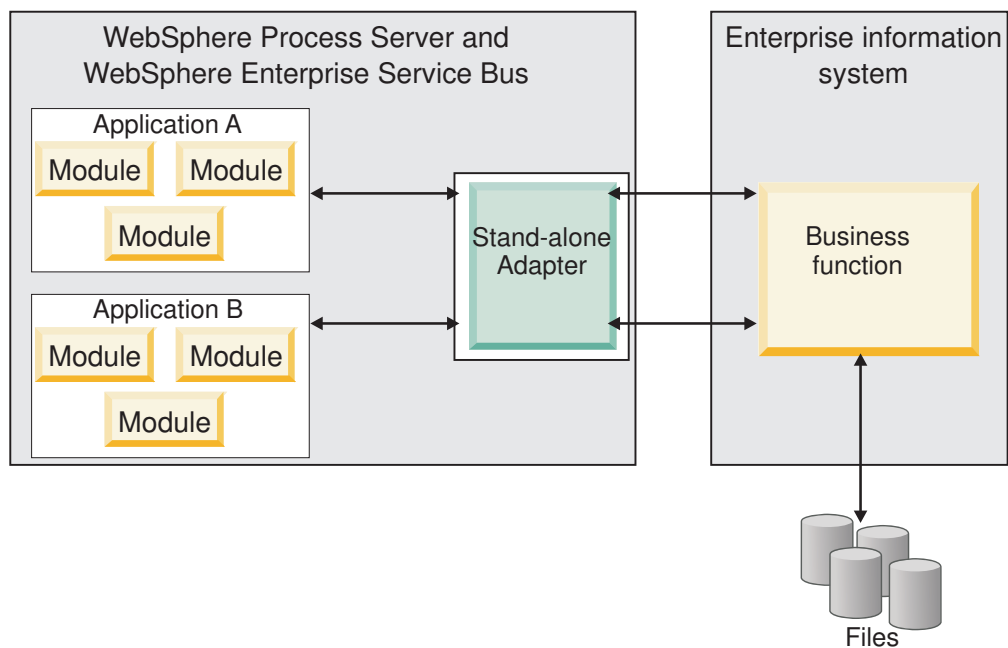
- **With module for use by single application.** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications.** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want

to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

An embedded adapter is bundled within an enterprise archive (EAR) file and is available only to the application with which it is packaged and deployed.



A stand-alone adapter is represented by a stand-alone resource adapter archive (RAR) file, and when deployed, it is available to all deployed applications in the server instance.



While creating the project for your application using WebSphere Integration Developer, you can choose how to package the adapter [either bundled with the (EAR) file or as a stand-alone (RAR) file]. Your choice will affect how the adapter is used in the runtime environment, as well as how the properties for the adapter are displayed on the administrative console.

Choosing either to embed an adapter with your application or to deploy the adapter as a stand-alone module depends on how you want to administer the adapter. If you want a single copy of the adapter and do not care about disruption to multiple applications when you upgrade the adapter, then you would be more likely to deploy the adapter as a stand-alone module.

If you plan on running multiple versions, and if you care more about potential disruption when you upgrade the adapter, you would be more likely to embed the adapter with the application. Embedding the adapter with the application allows you to associate an adapter version with an application version and administer it as a single module.

Considerations for embedding an adapter in the application

Take into consideration the following items if you plan on embedding the adapter with your application:

- An embedded adapter has class loader isolation.
A class loader affects the packaging of applications and the behavior of packaged applications deployed on runtime environments. *Class loader isolation* means the adapter cannot load classes from another application or module. Class loader isolation prevents two similarly named classes in different applications from interfering with each other.
- Each application in which the adapter is embedded must be administered separately.

Considerations for using a stand-alone adapter

Take into consideration the following items if you plan on using a stand-alone adapter:

- Stand-alone adapters have no class loader isolation.
Because stand-alone adapters have no class loader isolation, only one version of any given Java artifact is run and the version and sequence of that artifact is undetermined. For example, when you use a stand-alone adapter there is only *one* resource adapter version, *one* adapter foundation class (AFC) version, or *one* third-party JAR version. All adapters deployed as stand-alone adapters share a single AFC version, and all instances of a given adapter share the same code version. All adapter instances using a given third-party library must share that library.
- If you update any of these shared artifacts, all applications using the artifacts are affected.
For instance, if you have an adapter that is working with server version X, and you update the version of the client application to version Y, your original application might stop working.
- AFC is compatible with previous versions, but the latest AFC version must be in every RAR file that is deployed in a stand-alone manner.
If more than one copy of any JAR file is in the classpath in a stand-alone adapter, the one that is used is random; therefore, they all must be the latest version.

WebSphere Adapters in clustered environments

You can improve adapter performance and availability by deploying the module to a clustered server environment. The module is replicated across all servers in a cluster, regardless of whether you deploy the module using a stand-alone or embedded adapter.

WebSphere Process Server, WebSphere Application Server Network Deployment, and WebSphere Extended Deployment support clustered environments. Clusters are groups of servers that are managed together to balance workloads and to provide high availability and scalability. When you set up a server cluster, you create a Deployment Manager profile. The HAManager, a subcomponent of the Deployment Manager, notifies the JCA (Java EE Connector Architecture) container to activate the adapter instance. The JCA container provides a runtime environment for adapter instances. For information about creating clustered environments, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_v61.html.

Using WebSphere Extended Deployment, you can optionally enhance the performance of adapter instances in your clustered environment. WebSphere Extended Deployment extends the WebSphere Application Server Network Deployment capabilities by using a dynamic workload manager instead of a static workload manager, which is used by WebSphere Application Server Network Deployment. The dynamic workload manager can optimize the performance of adapter instances in the cluster by dynamically balancing the load of the requests. This means that application server instances can be automatically stopped and started based on the load variations, allowing machines with different capacities and configurations to evenly handle load variations. For information on the benefits of WebSphere Extended Deployment, see the following link: <http://publib.boulder.ibm.com/infocenter/wxdinfo/v6r1/index.jsp>.

In clustered environments, adapter instances can handle both inbound and outbound processes.

High availability for inbound processes

Inbound processes are based on events triggered as a result of updates to data in the Oracle database. WebSphere Adapter for Oracle E-Business Suite is configured to detect updates by polling an event table. The adapter then publishes the event to its endpoint.

When you deploy a module to a cluster, the JCA (Java EE Connector Architecture) container checks the enableHASupport resource adapter property. If the value for the enableHASupport property is true, which is the default setting, all of the adapter instances are registered with the HAManager with a policy 1 of N. This policy means that only one of the adapter instances starts polling for events. Although other adapter instances in the cluster are started, they remain dormant with respect to the active event until the active adapter instance finishes processing the event. If the server on which the polling thread was started shuts down for some reason, an adapter instance that is running on one of the backup servers is activated.

Important: Do not change the setting of the enableHASupport property.

High availability for outbound processes

In clustered environments, multiple adapter instances are available to perform outbound process requests. Accordingly, if your environment has multiple applications that interact with WebSphere Adapter for Oracle E-Business Suite for outbound requests, then you might improve performance by deploying the module to a clustered environment. In a clustered environment, multiple outbound requests can be processed simultaneously, as long as they are not attempting to process the same record.

If multiple outbound requests are attempting to process the same record, such as a Customer address, the workload management capability in WebSphere Application Server Network Deployment distributes the requests among the available adapter instances in the sequence they were received. As a result, these types of outbound requests in a clustered environment are processed in the same manner as those in a single server environment: one adapter instance processes only one outbound request at a time. For more information on workload management, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html.

Chapter 3. Samples

The sample content and artifacts that compose WebSphere Adapter for Oracle E-Business Suite demonstrate methods for structuring outbound and inbound integration between the Oracle E-Business Suite database and the WebSphere Business Integration system.

Database and applications user account requirements

To use the samples, you must use a database account that gives you rights to the artifacts needed to run the sample content, and use an Oracle E-Business Suite account that allows you to perform responsibilities of the Workflow Administrator for Web Applications, System Administrator, and Receivables Manager.

For the purposes of these samples, the user account running all scripts is assumed to be the APPS user for the Oracle database. This user has the following rights:

- To change and create content in the applications (APPS) schema
- To add and remove data from tables
- To run the required executable code in the APPS schema

Check with your Oracle database administrator to determine the account that will be used to run the sample content. If you want to choose a different user account, work with your database administrator to ensure that the user has rights to all of the database artifacts needed to run the sample content.

Oracle E-Business Suite requires you to have an account with rights to the following responsibilities:

- Workflow Administrator Web Applications
- System Administrator
- Receivables Manager

Note: If the Oracle E-Business Suite account that you are using does not have access to these responsibilities, they can be added. To change the responsibilities assigned to the account you will use, log onto an account that has System Administrator responsibility rights and go to the Security->User->Define menu option. For specific information about changing user responsibilities, refer to your documentation for Oracle Applications.

Accessing the sample files

When WebSphere Adapter for Oracle E-Business Suite is installed, a samples directory is created. The samples directory has the content you use to run the sample applications.

Procedure

1. Find the Sample folder under the installation directory for WebSphere Adapter for Oracle E-Business Suite.
2. Unzip the oracleEBSSamples.zip file. The samples files are described in the following table.

Table 1. Sample files

| Filename | Description |
|--------------------------------------|--|
| lbn_websphere_event_table_create.sql | Event table creation script |
| lbn_websphere_events_s.sql | Event ID sequence creation script |
| lbn_customer_event_pkg.pls | Package creation script for the procedures used in the customer object creation samples |
| lbn_customer_event_key_s.sql | Event Key sequence creation script |
| lbn_submit_request.sql | SQL script to insert a special procedure to be called after event entries have been made to the interface tables to move the data to the base tables |
| Raise_inbound_event.sql | SQL script to raise the customer inbound event |
| lbn_create_synonyms.sql | SQL script to create the synonym names used in creating business objects for the samples |
| IMPCUST.wft | Workflow file used in Sample 4 |

Additional samples: The WebSphere® Integration Developer online samples gallery includes additional samples and tutorials to help you use WebSphere Adapters. You can access the online samples and tutorials gallery from the welcome page that opens when you start WebSphere Integration Developer. To see samples for WebSphere Adapter for Oracle E-Business Suite, click **Retrieve**. Then navigate through the displayed directories to make your selections.

Sample 1: Sending outbound data to the Oracle database

With WebSphere Adapter for Oracle E-Business Suite, you can create data in an Oracle application and retrieve data from the application. In this scenario, you use WebSphere Adapter for JDBC as the integration method to create a customer record in the Oracle E-Business Suite Financials database and to retrieve that customer information from the Financials database.

About this task

Performing this scenario shows you how to set up information flows between Oracle E-Business Suite and WebSphere Process Server or WebSphere Enterprise Service Bus for outbound requests, regardless of the Oracle E-Business Suite module you use; for example, you could use Payables or General Ledger, rather than Receivables.

Related concepts

“Outbound processing sample applications” on page 4

Some outbound processing sample applications are provided with this solution. Sample 1 uses Oracle database interface tables, which is the standard outbound scenario for the Oracle E-Business Suite. Sample 3 uses API calls, while Sample 4 makes use of a workflow.

Preparing for outbound processing

You must run SQL script files to prepare for outbound processing. You need to create synonyms to save you time later when you generate business objects from the Oracle database. You also need to insert artifacts into Oracle E-Business Suite before you can process objects using the database tables.

Procedure

1. Create synonyms

Open the SQL*Plus tool (or similar program for processing SQL statements). Log on to SQL*Plus using a database account that has been granted database administrator (DBA) rights.

Run the SQL script `ibm_create_synonyms.sql`.

For example: `SQL> @C:\samplecontent\ibm_create_synonyms.sql;`

The size of the database and number of database artifacts in the Oracle database can lengthen the process of searching through large lists when running the external service wizard to generate objects. Creating the synonyms helps to make the search quicker. In addition, the synonyms help to resolve an issue related to the Oracle database driver. The issue is documented in IBM Technote 1218775. Refer to “Related information ” in the “Reference” section for a link to WebSphere Adapters Technotes.

2. Insert artifacts into Oracle E-Business Suite

Before you process business objects, run the SQL script `ibm_submit_request.sql`.

This places a stored procedure called `IBM_WEBSPHERE_CUSTOMER_IMP` into the Oracle APPS schema. This procedure is used to move customer data from the Oracle interface tables to the base tables. You will use the interface tables when you construct the Create business object.

Creating the project

To begin the process of creating and deploying a module to communicate with an Oracle E-Business Suite application, you start the external service wizard in WebSphere Integration Developer. The wizard creates an Adapter for JDBC project, which is used to organize the files associated with the adapter.

Before you begin

Ensure that you have access to the documentation for the Adapter for JDBC. To view it, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

Procedure

1. Review the information in “Configuring the module for deployment” in the *WebSphere Adapters: Adapter for JDBC User Guide*.
2. Perform the task “Creating the project” in the *WebSphere Adapters: Adapter for JDBC User Guide*. Follow the steps in this section to set up the project in your workspace on WebSphere Integration Developer.

Configuring the module for outbound processing

You must create the business objects necessary to process data into Oracle E-Business Suite. To accomplish this, use the external service wizard in WebSphere Integration Developer for WebSphere Adapter for JDBC. First, you initialize the wizard by configuring values of the properties that enable the adapter to set up a communication channel to a specific database. Then, run a query to discover business objects. Finally, you must specify the operations and other properties for the selected business objects.

Setting connection properties for the external service wizard

Set values of the connection properties for your database instance. These properties enable the external service wizard in the Adapter for JDBC to connect to the Oracle database for discovery and for creating the service description.

Before you begin

This task is performed within the external service wizard, which you already started for the tasks under “Creating the project.”

Procedure

1. In the Required Files and Libraries window, next to the top pane for JDBC driver JAR files, click **Add**. Browse to the location of the external JDBC driver, select the driver JAR file, and click **Open**.

The JDBC driver that the wizard needs to add to your project is the external JAR file that is provided with your database, which you can obtain from your database administrator or from the Web site for the database.

2. Click **Next**.

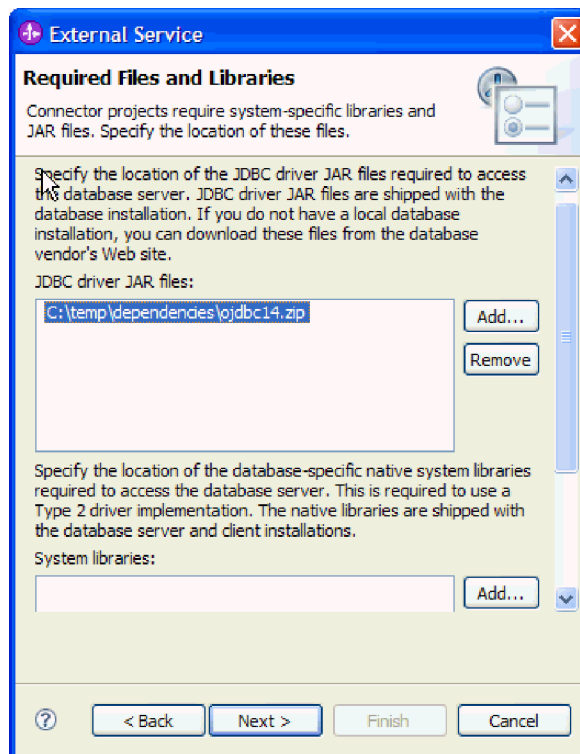


Figure 8. Add JDBC driver JAR file to project

3. In the Processing Direction window, click **Outbound** and click **Next**.
4. In the Discovery Configuration window, in the left pane, select your database vendor, driver, and version.
5. Enter the information for your database that is shown in the following table.
The table lists the properties required for the sample and their descriptions. For details on these properties, see “Connection properties for the external service wizard” in the “Reference” section of the *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html

Table 2. Connection properties for the external service wizard

| Property | Description |
|------------------------|--|
| Database | The database name. For Oracle databases this is the system ID (SID). Database is used along with host name and port number to generate the Database URL that is used to connect to the database. |
| Host name | The host name or IP address of the database server. Host name is used along with database name and port number to generate the Database URL that is used to connect to the database. |
| Port number | The port number, which, along with database name and host name, generates the Database URL that is used to connect to the database. |
| JDBC driver class name | The class name of the JDBC driver that is used to connect to the database. |
| User name | The database account you are using. It is used during the discovery process. Later in the wizard, you can specify a different name and password to use at run time. |
| Password | The password for the account you are using. |

Click Next.

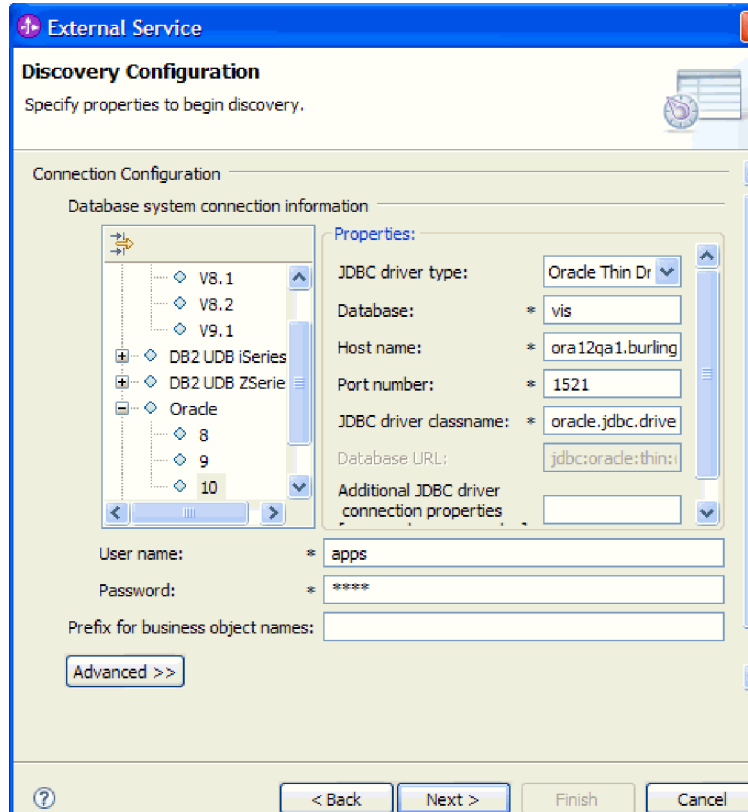


Figure 9. Add connection property values

Results

The wizard uses the connection properties to connect to the database.

Create operation business object selection and linking

This sample uses one set of primary and child business objects to populate customer data into the Oracle E-Business Suite interface tables. The Create operation is used to post new customer data and updates to existing customer data. You must configure five business objects for the Create operation.

Business objects in the database tables

The Customer Profile object is the primary object, and the rest of the objects are structured as child business objects. You link the business objects by setting the application-specific information foreign key relationship between the parent and child objects. For more information about the nature of business objects and about setting the application-specific information values, refer to the “Business objects” section of *WebSphere Adapters: Adapter for JDBC User Guide*.

The following table lists the database tables for the Accounts Receivable (AR) schema including the table name, suggested synonym, and a brief description of the information that each object contains.

Note: For more information about the table data, refer to your documentation for Oracle Applications, particularly the section of the Receivables user guide that describes the customer interface.

Table 3. Database tables for the AR schema

| Table name | Synonym | Description |
|---------------------------------|------------------------|--|
| AR.RA_CUSTOMER_PROFILES_INT_ALL | ArIbm_C_Cust_Profile | Customer Profile interface table, which contains high-level information about the customer |
| AR.RA_CUSTOMERS_INTERFACE_ALL | ArIbm_C_Cust_Interface | Primary table for customers, as well as customer addresses |
| AR.RA_CONTACT_PHONES_INT_ALL | ArIbm_C_Contacts | Contact and phone information |
| AR.RA_CUST_PAY_METHOD_INT_ALL | ArIbm_C_Pay_Methods | Payment methods associated with the customer |
| AR.RA_CUSTOMER_BANKS_INT_ALL | ArIbm_C_Cust_Banks | Banks associated with the customer |

Next you will select business objects by running a query on the AR schema.

Selecting business objects and services

After setting connection properties, run a query to search for database objects. Browse the metadata tree of discovered objects to understand the structure of the objects in the Oracle database and select objects that you want to include in the service description.

About this task

For this sample, you run a query on the accounts receivable (AR) schema, and select synonyms that represent data in the Oracle E-Business Suite interface tables.

Procedure

1. Specify filter properties
 - a. In the Object Discovery and Selection window, click **Edit Query**.
 - b. In the Query Properties window, type AR in the **Schema name pattern** field to display the accounts receivable (AR) schema.
 - c. Select the check box **Prompt for additional configuration settings when adding business object**, and click **OK**.

Now whenever you select one of the discovered objects for import, you are prompted to enter application-specific information for the object.

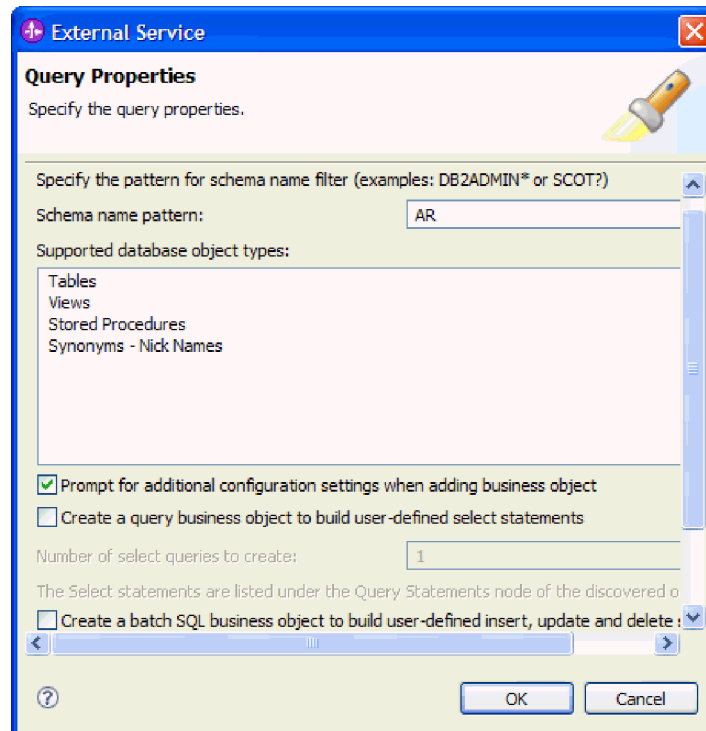


Figure 10. Specify query properties

2. Run the metadata query
 - a. Display objects discovered by the query
Click **Run Query**. The AR schema and data elements of the schema are displayed.
 - b. Select the object for import
Expand the **AR** schema. Expand **Synonyms - Nicknames**.
Highlight the synonym **IBM_C_CUST_PROFILE**, and click > (the Add icon) to select this object to be imported.
 - c. Add business object application-specific information
In the Configuration Properties for **IBM_C_CUST_PROFILE** window, click **Add** to select the primary key for the table related to **IBM_C_CUST_PROFILE**.
Select **REQUEST_ID** and click **OK**.
Click **Add** to select and configure a stored procedure that will be associated with this synonym.
In the Add window, select **AfterCreateSP** and click **OK**.

In the **AfterCreateSP** area of the Configuration Properties for **IBM_C_CUST_PROFILE** window, set the following values:

- Type APPS in the **Schema name pattern** field.
- Select APPS from the **Schema name** list.

Note: There may be long delay after selecting the APPS schema before the **Stored procedure name pattern** field is active.

- Type IBM_WEBSHERE_CUSTOMER_IMP in the **Stored procedure name pattern** field.
- Select IBM_WEBSHERE_CUSTOMER_IMP from the **Stored procedure name** list.

Click **OK**.

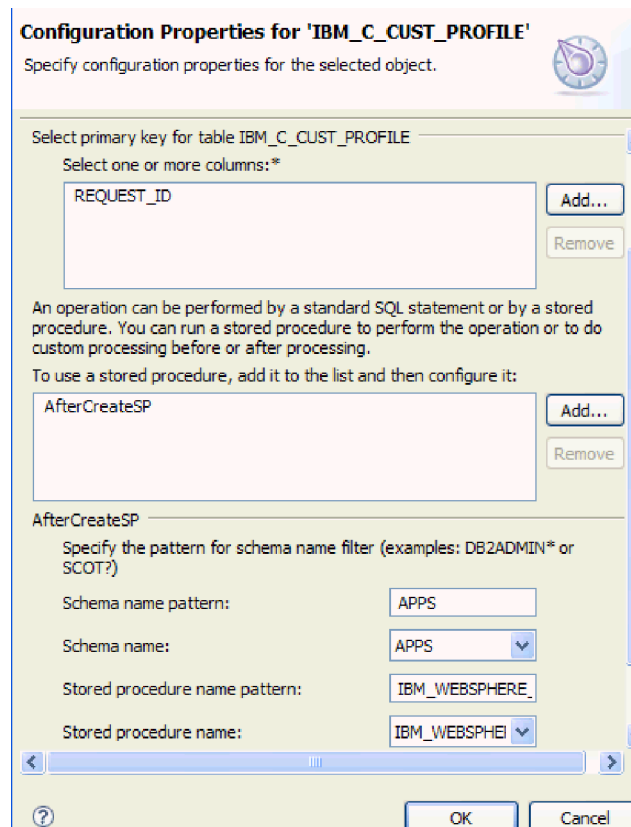


Figure 11. Add business object application-specific information

After a delay, the selected object is displayed in the Selected objects pane.

In this sample, you process data using Oracle's built-in interface and base tables. Only the Create operation is necessary to post both new customer data and updates to existing customer data. As part of calling the Create operation of the object, the stored procedure **IBM_WEBSHERE_CUSTOMER_IMP** is run using the **AfterCreateASI** value for the object to process. This stored procedure moves the data from the interface to the base tables.

- Select another object for import and link child object to parent

In the Object Discovery and Selection window, highlight the synonym **IBM_C_CUST_INTERFACE**, and click > (the Add icon) to select this object to be imported.

In the Configuration Properties for IBM_C_CUST_INTERFACE window, click **Add** to select the primary key.

Select **CUSTOMER_KEY** and click **OK**.

Set the parent for the table related to IBM_C_CUST_INTERFACE by selecting **IBM_C_CUST_PROFILE**.

Under **Build a foreign key relationship by selecting a parent table column for each child column**, link the child to the parent's attributes by selecting **request_id** as the foreign key attribute value for the **REQUEST_ID** foreign key attribute.

Check **Parent object owns child object (cascade delete)** and click **OK**.

e. Select remaining synonyms

Repeat step 2d for each of the remaining synonyms in the following table.

The primary key is specified for each synonym.

Table 4. Remaining synonyms to select for import and to link to the parent object

| Synonym | Primary Key |
|----------------------|------------------|
| AR.IBM_C_CONTACTS | CONTACT_KEY |
| AR.IBM_C_CUST_BANKS | BANK_ACCOUNT_NUM |
| AR.IBM_C_PAY_METHODS | ORG_ID |

When you are finished, click **Next**.

For information on the object-level, operation, and attribute application-specific information, refer to "Business object information" in the "Reference" section of *IBM WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

Retrieve operation business object selection and linking

This sample uses a second set of objects to retrieve customer data directly from the Oracle E-Business Suite base tables. Because the Retrieve operation does not modify the values in the base tables, you can retrieve the data from those tables directly. You do not have to use interface tables. You must configure the business objects for the Retrieve operation.

Base tables for the Accounts Receivable schema

The following table lists the base tables for the Accounts Receivable (AR) schema, including the table name, suggested synonym, and description. The next section shows a diagram of the relationship of the business objects that you configure for the Retrieve operation.

Table 5. Base tables for the AR schema

| Table name | Synonym | Description |
|-------------------------|-----------------------|--|
| AR.HZ_CUST_ACCOUNTS | ArIbm_R_Cust_Accounts | Customer number, type, and other customer information. |
| AR.HZ_CUSTOMER_PROFILES | ArIbm_R_Cust_Profiles | High-level profile information for the customer. |

Table 5. Base tables for the AR schema (continued)

| Table name | Synonym | Description |
|---------------------------|---------------------------|---|
| AR.HZ_PARTIES | ArIbm_R_Parties | Customer name and if it is a person, not a business, the person's details. |
| AR.HZ_PARTY_SITES | ArIbm_R_Party_Sites | Links between customer's address IDs and address location IDs. |
| AR.HZ_CUST_ACCT_SITES_ALL | ArIbm_R_Acct_Sites | Address IDs. |
| AR.HZ_CUST_SITE_USES_ALL | ArIbm_R_Site_Uses | Customer business address purposes (BILL_TO, SHIP_TO, etc.). |
| AR.HZ_LOCATIONS | ArIbm_R_Locations | Address details. |
| AR.HZ_CONTACT_POINTS | ArIbm_R_Contacts_Sites | Contact and phone details. The table is listed twice because both sites and customers (PARTY_SITES and PARTIES) use the same table for contact details. |
| AR.HZ_CONTACT_POINTS | ArIbm_R_Contacts_Customer | Contact and phone details. The table is listed twice because both sites and customers (PARTY_SITES and PARTIES) use the same table for contact details. |

Relationship of the business objects

For the Retrieve operation to succeed, the links connecting the customer object to its child objects must match those in the Oracle database. The following diagram shows the relationship of the objects you must configure for the Retrieve operation.

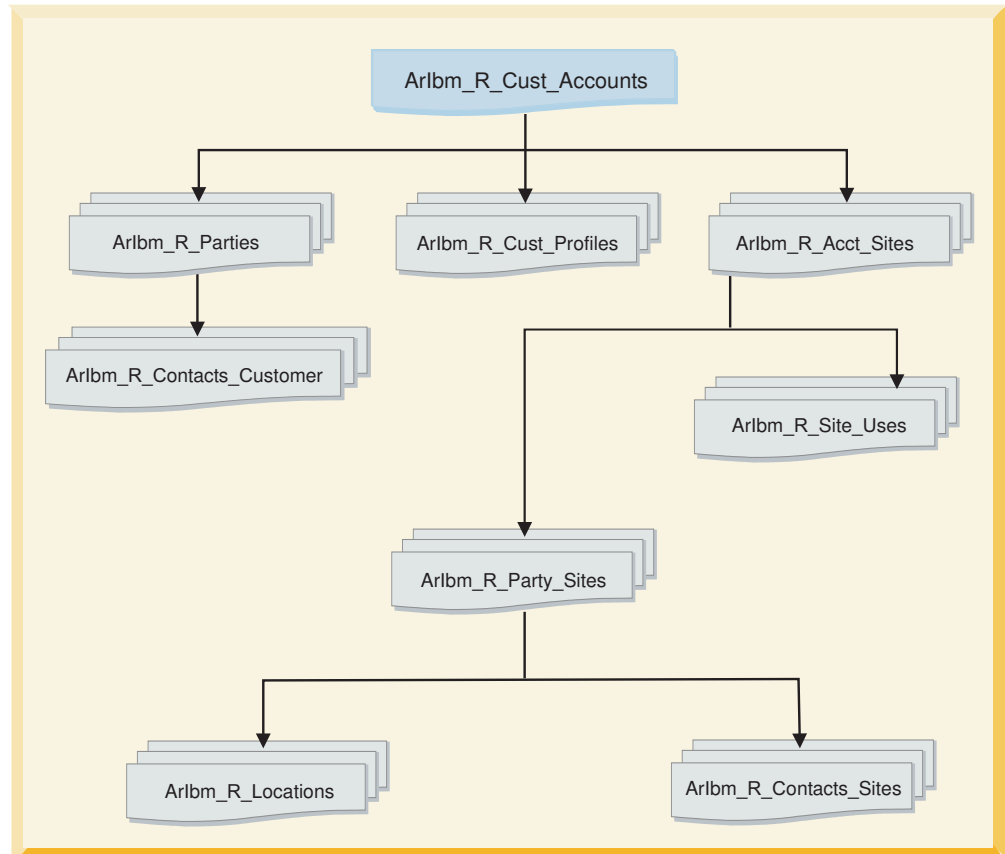


Figure 12. Relationship of objects for the Retrieve operation

To link the objects, you create the child objects and set the application-specific information foreign key relationship between the parent and child objects. For more information on the nature of business objects, refer to the “Business objects” section of the “Technical overview” in *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

Selecting and linking business objects for Retrieve

After you have run a query on the accounts receivable (AR) schema, you select synonyms that represent data in the Oracle E-Business Suite base tables. You also configure the objects in a parent and child hierarchy.

Before you begin

You have already set connection properties, specified filter properties, and run the query to display discovered objects.

Procedure

1. Select the object for import

In the Object Discovery and Selection window, after you have expanded the **AR** schema and expanded **Synonyms - Nicknames**, highlight the synonym **IBM_R_CUST_ACCOUNTS** and click > (the Add icon) to select the object to be imported.

2. Add business object application-specific information

In the Configuration Parameters for IBM_R_CUST_ACCOUNTS window, click **Add** to select the primary key for the table related to IBM_R_CUST_ACCOUNTS.

Select **CUST_ACCOUNT_ID** and click **OK**.

3. Select another object for import and link child object to parent
 - a. Highlight the synonym **IBM_R_CUST_PROFILES** and click > (the Add icon).
 - b. In the Configuration Parameters for IBM_R_CUST_PROFILES window, click **Add** to select the primary key for the table related to IBM_R_CUST_PROFILES.
Select **APPLICATION_ID** and click **OK**.
 - c. Set the parent for the table related to IBM_R_CUST_PROFILES by selecting **IBM_R_CUST_ACCOUNTS**.
 - d. Under **Build a foreign key relationship by selecting a parent table column for each child column**, link the child to the parent's attributes by selecting a foreign key attribute and a foreign key value for that attribute.
For the foreign key attribute **CUST_ACCOUNT_ID**, select the foreign key attribute value **CUST_ACCOUNT_ID**.
Check **Parent object owns child object (cascade delete)** and click **OK**.
4. Select remaining synonyms
Repeat step 3 for each of the remaining synonyms in the order the synonyms are presented in the following table. Use the primary key, parent, foreign key attribute, and foreign key attribute value shown for each synonym in the table.
When you are finished, click **Next**.

Table 6. Objects requiring foreign key values and child objects

| Synonym | Primary key | Parent | Foreign key attribute | Foreign key attribute value |
|---------------------------|-------------------------------------|-----------------------|-----------------------|-----------------------------|
| ArIbm_R_Parties | party_id | Aribm_R_Cust_Accounts | party_id | party_id |
| ArIbm_R_Acct_Sites | party_site_id and cust_acct_site_id | Aribm_R_Cust_Accounts | cust_account_id | cust_account_id |
| ArIbm_R_Party_Sites | location_id and party_site_id | Aribm_R_Acct_Sites | party_site_id | party_site_id |
| ArIbm_R_Site_Uses | application_id | Aribm_R_Acct_Sites | cust_acct_site_id | cust_acct_site_id |
| ArIbm_R_Locations | address_key | Aribm_R_Party_Sites | location_id | location_id |
| ArIbm_R_Contacts_Sites | application_id | Aribm_R_Party_Sites | owner_table_id | party_site_id |
| ArIbm_R_Contacts_Customer | application_id | Aribm_R_Parties | owner_table_id | party_id |

Configuring the selected objects

After you have selected and linked database objects, specify operations and other properties that apply to all of the business objects.

Procedure

1. Select operations

In the Configure Composite Properties window of the external service wizard, the Operations pane lists the operations that the adapter supports for the outbound service type. Remove the following operations by highlighting them and clicking **Remove**.

- **Update**
- **Delete**
- **RetrieveAll**
- **ApplyChanges**

The following operations remain:

- **Create**
 - **Retrieve**
2. Leave the default values for these fields:
 - **Maximum records for RetrieveAll operation**
 - **Business object namespace**
 - **Folder**
 - **Generate a business graph for each business object**
 3. Click **Next**.

What to do next

Next, you will specify deployment information to use at run time and information for saving the service as a module.

Setting deployment properties and generating the service

Use the external service wizard to configure the properties the adapter uses to set up a communication channel to a specific database for outbound processing. Then create a business integration module where all of the artifacts and property values can be saved.

Procedure

1. Set configuration property values
 - a. In the Service Generation and Deployment Configuration window, in the **Deploy connector project** field, ensure that this option is selected: **With module for use by single application**.
 - b. Click **Advanced** to see all of the properties under **Connection properties, Database system connection information**.
For outbound processing, the property fields are displayed for the managed connection factory and resource adapter properties.
 - c. Set the values of the required connection properties as shown in the following table.
The fields are initialized with the connection information you specified when you started the wizard. You can edit the existing values for Database vendor and User name, as required. You must type your Password.

Table 7. Database system connection information

| Properties | Values |
|-----------------|--|
| User name | The database account you are using |
| Password | The password for the account you are using |
| Database vendor | Oracle |

d. Click **Next**.

For information about these properties, refer to the “Reference” section of *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

2. Create a module

- a. In the Service Location Properties window, click **New** next to the **Module** field.
- b. In the Integration Project window, confirm that **Create a module project** is selected and click **Next**.
- c. In the Module window, type `Oracle_Outbound`, and click **Finish**.
- d. In the Service Location Properties window, click **Finish**.

Results

This module will hold all of the generated artifacts.

Saving the module

Save the business integration module for outbound processing that contains the artifacts and property values you created.

Procedure

1. In the Business Integration view, click the **Oracle_Outbound** module.
2. Select **File > Save**.

Note: If the option to **Save** is inactive (grey), then the module has already been saved.

Deploying the module for testing

To test the adapter application, you need to install the module for outbound processing in the WebSphere Integration Developer integration test client.

Before you begin

The module that contains an Oracle database import file is produced by running the external service wizard.

Procedure

1. In WebSphere Integration Developer, click the **Servers** tab to open the Servers view.

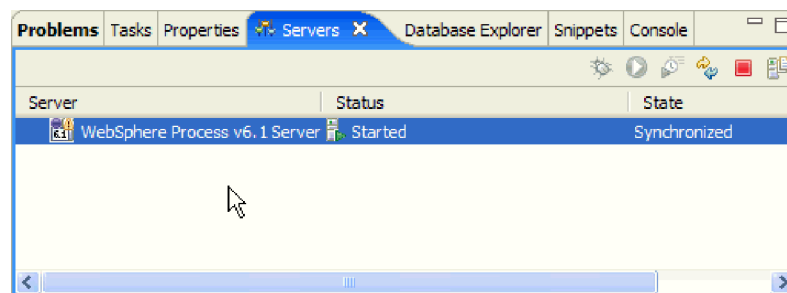


Figure 13. Servers view

2. If your server is not displayed in the Servers view, right-click the Servers view and select **New** → **Server**. Select your WebSphere Process Server or WebSphere Enterprise Service Bus.
Click **Next** and click **Finish**.
3. If your WebSphere Process Server or WebSphere Enterprise Service Bus is not running, right-click the name of it and select **Start**. The **Status** entry changes to **Started**.
4. Right-click the name of your WebSphere Process Server or WebSphere Enterprise Service Bus, and select **Add and remove projects**.
5. From the **Available Projects** list on the left, select the **Oracle_OutboundApp** module and click **Add**. Click **Finish**.

What to do next

Next, you can test the assembled application in the integration test client.

Testing the Create operation

After you have deployed your outbound processing application to WebSphere Process Server or WebSphere Enterprise Service Bus, test the assembled adapter application using the WebSphere Integration Developer integration test client. You test the Create operation first, and then test the Retrieve operation.

About this task

To test the Create operation, you create a customer entry that has two addresses and one phone number.

Procedure

1. Select the module to test
In WebSphere Integration Developer, go to the Business Integration view of the Business Integration perspective. Right-click the **Oracle_Outbound** module, and select **Test** > **Test Module**.
2. Leave the default value for **verb**.
The value may be different from the one shown in the “Set attribute values” figure below.
3. Set the test values
In the Oracle_Outbound_Test window, use the arrows to set the test values shown in the following table.

Table 8. Test values for the Create operation

| Item | Test value |
|---------------|------------------------------|
| Configuration | Default Module Test |
| Module | Oracle_Outbound |
| Component | JDBCOutboundInterface |
| Interface | JDBCOutboundInterface |
| Operation | createArIbm_C_Cust_ProfileBG |

Note: The test values for **Component** and **Interface** may include a number, for example, JDBCOutboundInterface1. WebSphere Integration Developer appends numbers so that each name in the workspace is unique.

4. Set attribute values for the input business object

Set the attribute values for the Customer Profile object, which is the top-level object `ArIbm_C_Cust_Profile`. Because this example uses two addresses and one contact, you also need to set values for two child objects of the Addresses type (`ArIbm_C_Cust_Interfaceobj`) and one child object of the Contacts type (`ArIbm_C_Contactsobj`). The following tables provide the values for each object and indicate any restrictions, such as whether the value has to be unique.

a. Create and record unique ID values.

Oracle requires ID values for outbound processing. These values uniquely identify the records inserted into Oracle E-Business Suite and are required for successful retrieval of the corresponding inserted data. The ID values are used in multiple objects, so you need to write down the values you will use to be able to enter them consistently for the different objects. For each of the following five values, create unique string values.

- CustomerReference#
- AddressReference#1
- AddressReference#2
- TelephoneReference#
- CustomerName

Note: For more information on required values and other allowed values for data in the Oracle interface tables, refer to your documentation for Oracle Applications, specifically the Oracle Receivables User Guide.

b. In the Events window, under **Initial request parameters**, click the **Value** column across from the name of each attribute whose value you need to set, and enter the value from the following tables.

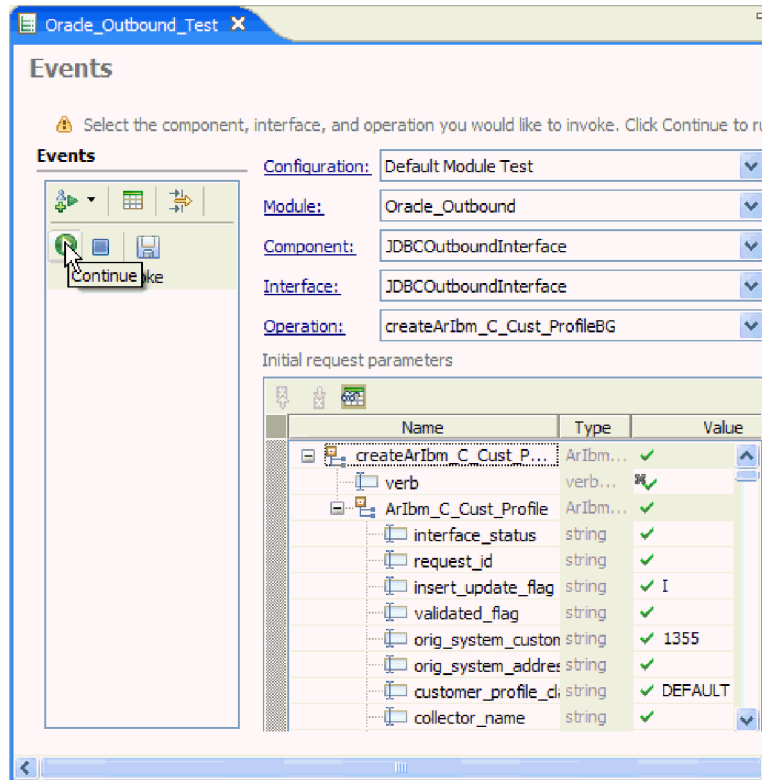


Figure 14. Set attribute values

Important:

- To set a value for an attribute to **<null>**, right-click the value field for the attribute. Select **Set to** and scroll down to select **<null>**.
- To add a child object, right-click the attribute of the appropriate type and choose **Add Elements**. Enter the number of elements you are adding for each attribute. In this sample, enter 2 when you add elements for the attribute **ArIbm_C_Cust_Interfaceobj** because there are two child objects of that type (two addresses), and enter 1 when you add elements for the attribute **ArIBM_C_Contactsobj** because there is one child object of that type (one contact).

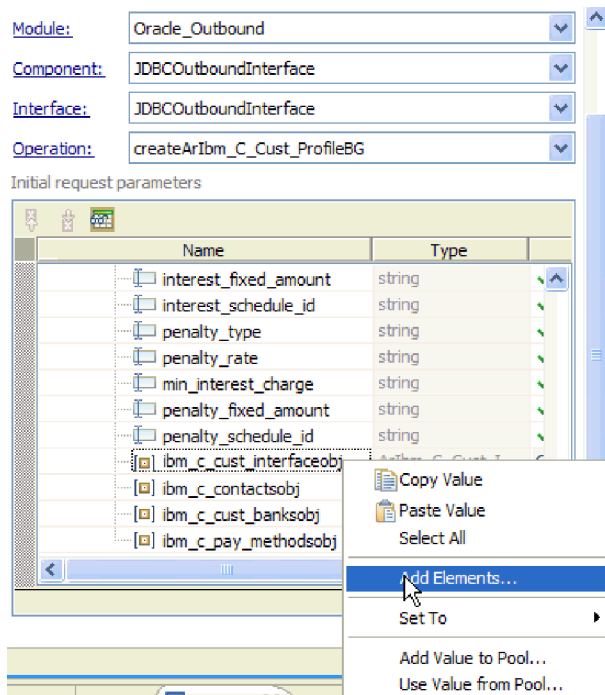


Figure 15. Use Add Elements to add a child object

Table 9. Values for the top-level object

| Attribute | Value | Notes |
|-----------------------------|---------------------|---|
| Insert_update_flag | I | |
| Orig_system_customer_ref | CustomerReference # | Use the value you recorded. The value must be unique. |
| Customer_profile_class_name | DEFAULT | |
| Credit_hold | N | |
| Last_updated_by | -1 | |
| Last_update_date | 2007-06-25 12:00:00 | Today's date and time. Does not have to be exact. |
| Creation_date | 2007-06-25 12:00:00 | Today's date and time. Does not have to be exact. |
| Created_by | -1 | |
| Org_id | 204 | |

The following table lists the values for the first element for the Address attribute.

Table 10. Values for the first element for the Address attribute

| Attribute | Value | Notes |
|--------------------------|--------------------|--|
| Orig_system_customer_ref | CustomerReference# | Use the value that you recorded. The value must be unique. |
| Site_use_code | BILL_TO | |

Table 10. Values for the first element for the Address attribute (continued)

| Attribute | Value | Notes |
|-------------------------|---------------------------|--|
| Orig_system_address_ref | <i>AddressReference#1</i> | Use the value that you recorded. The value must be unique. |
| Insert_update_flag | I | |
| Customer_name | <i>CustomerName</i> | Use the value that you recorded. The value must be unique. |
| Customer_number | <null> | |
| Customer_status | A | |
| Primary_site_use_flag | Y | |
| Location | <null> | |
| Address1 | Test Address 1 | You have the option of using your own address information. |
| Address2 | <null> | |
| Address3 | <null> | |
| Address4 | <null> | |
| City | San Mateo | |
| State | CA | |
| Province | <null> | |
| County | San Mateo | |
| Postal_code | 94010 | |
| country | US | |
| Cust_category_code | CUSTOMER | |
| Last_updated_by | -1 | |
| Last_update_date | 2007-06-25 12:00:00 | Today's date and time. Does not have to be exact. |
| Created_by | -1 | |
| Creation_date | 2007-06-25 12:00:00 | Today's date and time. Does not have to be exact. |
| Org_id | 204 | |
| Customer_name_phonetic | <i>CustomerName</i> | Use the value that you recorded. |

The following table lists the values for the second element for the Address attribute.

Table 11. Values for the second element for the Address attribute

| Attribute | Value | Notes |
|--------------------------|---------------------------|--|
| Orig_system_customer_ref | <i>CustomerReference#</i> | Use the value that you recorded. The value must be unique. |
| Site_use_code | SHIP_TO | |

Table 11. Values for the second element for the Address attribute (continued)

| Attribute | Value | Notes |
|-------------------------|---------------------------|--|
| Orig_system_address_ref | <i>AddressReference#2</i> | Use the value that you recorded. The value must be unique. |
| Insert_update_flag | I | |
| Customer_name | <i>CustomerName</i> | Use the value that you recorded. The value must be unique. |
| Customer_number | <null> | |
| Customer_status | A | |
| Primary_site_use_flag | Y | |
| Location | <null> | |
| Address1 | Test Address 2 | You have the option of using your own address information. |
| Address2 | <null> | |
| Address3 | <null> | |
| Address4 | <null> | |
| City | San Mateo | |
| State | CA | |
| Province | <null> | |
| County | San Mateo | |
| Postal_code | 94010 | |
| country | US | |
| Cust_category_code | CUSTOMER | |
| Last_updated_by | -1 | |
| Last_update_date | 2007-06-25 12:00:00 | Today's date and time. Does not have to be exact. |
| Created_by | -1 | |
| Creation_date | 2007-06-25 12:00:00 | Today's date and time. Does not have to be exact. |
| Org_id | 204 | |
| Customer_name_phonetic | <i>CustomerName</i> | Use the value that you recorded. |

The following table lists the values for an element for the Contacts attribute.

Table 12. Values for an element for the Contacts attribute

| Attribute | Value | Notes |
|---------------------------|----------------------------|--|
| Orig_system_contact_ref | <null> | |
| Orig_system_telephone_ref | <i>TelephoneReference#</i> | Use the value that you recorded. The value must be unique. |

Table 12. Values for an element for the Contacts attribute (continued)

| Attribute | Value | Notes |
|--------------------------|---------------------|--|
| Orig_system_customer_ref | CustomerReference# | Use the value that you recorded. The value must be unique. |
| Orig_system_address_ref | AddressReference#1 | Use the value that you recorded. The value must be unique. |
| Insert_update_flag | I | |
| Contact_first_name | <null> | |
| Contact_last_name | <null> | |
| Contact_title | <null> | |
| Contact_job_title | <null> | |
| Telephone | 5551212 | You have the option of using your own phone number. |
| Telephone_extension | 1234 | Use <null> if you don't have one. |
| Telephone_type | FAX | |
| Telephone_area_code | 650 | |
| Last_update_date | 2007-06-25 12:00:00 | Today's date and time. Does not have to be exact. |
| Last_updated_by | -1 | |
| Creation_date | 2007-06-25 12:00:00 | Today's date and time. Does not have to be exact. |
| Created_by | -1 | |
| Email_address | <null> | |
| Org_id | 204 | |

5. In the left pane below **Events**, click **Continue**. From the Deployment Location window, select the name of the server where your application will be deployed, and click **Finish**. The Starting the Integration Test Client window is displayed briefly.
6. Verify that the customer data has been imported properly
 - Choose one of the following two methods, either a or b, to verify the customer data.
 - a. Log on to Oracle E-Business Suite and switch to the Receivables Manager responsibility. Navigate to the Customers > Standard menu to open the Find Customers window. Query for the imported customer.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Write down the customer number displayed in the customer screen in Oracle.

Start the SQL*Plus tool, or a similar program for processing SQL commands.

Run this query: SQL> select cust_account_id from ar.hz_cust_accounts where account_number = *customer number*;

where *customer number* is the number you wrote down from the customer screen in Oracle.

Write down the cust_account_id value from this query. You will use this number when you test the Retrieve operation.

- b. Optionally, you can open a database editor, and connect to the Oracle database you are using for this sample.

Find the row of your inserted customer by looking at the creation_date attribute (it should be today's date or the date you inserted for the top-level object).

Write down the value for cust_account_id. You will need this number when you test the Retrieve operation.

What to do next

Next you will test the Retrieve operation.

Related reference

“Troubleshooting information for the sample applications” on page 90

If you encounter problems while running the tasks for these samples, they could be related to either running the Adapter for JDBC or to interacting with the Oracle database. Two known issues are described.

Testing the Retrieve operation

After you have tested the Create operation and verified that the imported customer is in the Oracle E-Business Suite base tables, test the Retrieve operation.

Before you begin

You must have the customer_account_id value from the SQL query you ran after verifying that the customer data had been imported properly when you tested the Create operation.

About this task

You test the assembled adapter application using the WebSphere Integration Developer integration test client.

Procedure

1. Select the module to test
In the Business Integration perspective of WebSphere Integration Developer, right-click the **Oracle_Outbound** module and select **Test > Test Module**.
2. Leave the default value for **verb**.
3. Set the test values
In the Oracle_Outbound_Test window, use the arrows to set the test values, which are provided in the following table:

Table 13. Test values for the Retrieve operation

| Item | Test value |
|-----------|---------------------------------|
| Module | Oracle_Outbound |
| Component | JDBCOutboundInterface |
| Interface | JDBCOutboundInterface |
| Operation | retrieveArIbm_R_Cust_AccountsBG |

Note: The test values for **Component** and **Interface** may include a number, for example, JDBCOutboundInterface1. WebSphere Integration Developer appends numbers so that each name in the workspace is unique.

4. Set the attribute value for the input business object
 - a. In the Events window, under **Initial request parameters**, click the **Value** column across from **cust_account_id** in the **Name** column. Set the value to match the *cust_account_id* value from the SQL query you ran in “Testing the Create operation.”
 - b. Do not change the default settings for the remaining values.
 - c. In the left pane, below **Events**, click **Continue**. From the Deployment Location window, select the name of the server where the object will be deployed, and click **Finish**.

The Starting the Integration Test Client window is displayed as the object is being processed.
5. Verify that the returned object matches the customer information that was imported into the Oracle interface tables.

Clearing the sample content

After you have tested the Create and Retrieve operations, clear the sample content to return the data to its original state. To do this, you must deactivate the customer in Oracle E-Business Suite, and then run SQL commands to delete the stored procedure.

Procedure

1. Use the Oracle E-Business Suite Client to log into the Receivables Manager responsibility. Navigate to the Customers > Standard menu and find the customer. Change the Status value for the customer to “Inactive.” Then save the customer.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

2. Use the SQL*Plus tool, or a similar program for processing SQL commands, to log into the database and run commands to clear artifacts from the database. Consult your database administrator if you need help performing this task.

Sample 2: Receiving inbound data from the database

In this sample, you use the event system within Oracle E-Business Suite to populate an event table with customer events, and you also use WebSphere Adapter for JDBC to poll for events and retrieve customer data. By performing this scenario, you can see how to set up information flows between Oracle E-Business Suite and WebSphere Process Server or WebSphere Enterprise Service Bus for inbound events.

Related concepts

“Inbound processing sample application” on page 7

Sample 2 uses the Oracle Business Event System to send inbound customer data from Oracle E-Business Suite to an event table where the Adapter for JDBC can read it. The Adapter for JDBC picks up the event from the event table and retrieves the business object from the Oracle database based on the information provided in the event. The Adapter for JDBC then processes the business object from the Oracle database. The content included for this sample is described in this section.

Preparing for inbound processing

Perform the setup steps needed to prepare your environment for inbound processing. One step inserts stored procedures into Oracle E-Business Suite, and another step increases the efficiency of database searches performed by the external service wizard. An optional step enables the Oracle database and the runtime environment to keep parallel data about inactive customers.

Modifying the delete operation

This task is optional. In this sample application, WebSphere Adapter for JDBC is set to pass a Delete operation to the runtime environment for customer objects that are identified as inactive by event data sent from Oracle. If you do not want the adapter to pass a Delete operation for an inactive customer, modify a SQL script to change the operation from Delete to Update.

About this task

This task is necessary only if you want to change the operation that the Adapter for JDBC performs on customer records that have become inactive in the Oracle database.

With Oracle E-Business Suite, you can make a customer inactive and modify data for an inactive customer, but you cannot delete customer data from the base tables. When the stored procedure used for inbound processing receives event data from Oracle that indicates a customer has become inactive, it sets the operation to delete for the customer object and then posts that information to the event table used by the Adapter for JDBC. The adapter uses that information to process the object and pass it to WebSphere Process Server or WebSphere Enterprise Service Bus. However, if you want to keep the customer object in the runtime environment rather than deleting it, edit the sample SQL package to change the operation that the adapter reads from the event table.

Procedure

1. Access the samples files, and open the `ibm_customer_event_pkg.pls` file in a text editor.
2. Remove the following statement from the script:

```
IF v_status <> 'A' THEN
  v_object_function := 'Delete';
END IF;
```
3. Save the script.
4. Run the script using the SQL* Plus tool or a similar tool for processing SQL statements.

Inserting artifacts into Oracle E-Business Suite

Run four SQL scripts to insert the artifacts into the Oracle database for performing the sample. These artifacts include tables, stored procedures, an event package, triggers, and sequences.

About this task

In inbound processing, the Business Event System in Oracle E-Business Suite is used to identify changes to the customer table and to route key information for changed customers to the Adapter for JDBC event table. The scripts in this task must be run to populate the Oracle database with the artifacts necessary for running this sample.

Procedure

Run the following four SQL scripts from the samples directory **in the listed order**.

- Ibm_websphere_event_table_create.sql
- Ibm_websphere_events_s.sql
- Ibm_customer_event_key_s.sql
- Ibm_customer_event_pkg.pls

For example, SQL>@C:\ibm\ibm_websphere_event_table_create.sql;

Results

These scripts place stored procedures and other artifacts into the Oracle Accounts Receivable (AR) schema.

Creating synonyms

Create synonyms to increase the efficiency of database searches performed with the external service wizard. You create synonyms by running a SQL script.

About this task

The size of the database and number of database artifacts in the Oracle database can lengthen the process of searching through large lists when running the external service wizard to generate objects. Creating the synonyms will help to make the search quicker. In addition, the synonyms help resolve an issue related to the Oracle database driver. The issue is documented in IBM technote 1218775.

Procedure

1. Open the SQL*Plus tool (or similar program for processing SQL statements).
2. Run the script `ibm_create_synonyms.sql`.

For example, enter: SQL> @C:\samplecontent\ibm_create_synonyms.sql;

Setup within Oracle E-Business Suite

This sample application uses the Oracle E-Business Suite Business Event System, a mechanism to process data based on events, subscriptions, and actions you define. Before you can use the Business Event System, you must complete some setup tasks within Oracle. These include creating a custom event, creating a subscription for the event, and creating a concurrent program for the event.

Creating a custom event

The Oracle E-Business Suite built-in Business Event System (BES) requires events, subscriptions, and a concurrent program to process data. When you create a custom event, you satisfy one of the requirements of the Business Event System mechanism.

About this task

This is the first setup step performed within Oracle E-Business Suite to enable the Business Event System to detect and pass changes in data to the Adapter for JDBC event table.

Procedure

1. Access your Oracle E-Business Suite application instance as the user SYSADMIN.

2. Use the responsibility for Web Applications Workflow Administrator.
3. Select the Business Events menu option, and navigate to the page where you can create a new business event.
4. Create a new business event, and enter or select all of the following information:

Table 14. Business event values

| Field | Value |
|--------------|-----------------------------|
| Name | ibm.apps.ar.customer.create |
| Display name | IBM Create Customer Event |
| Status | Enabled |
| Owner Name | JTF |
| Owner Tag | JTF |

5. Save the event definition.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Creating a subscription for the custom event

After you create a custom event, you must create a subscription for the Business Event System to use with the event.

Procedure

1. Access the Subscriptions menu.
2. Navigate to the page where you can create a new subscription.
3. Create the subscription, entering information from the following table:

Table 15. Subscription values

| Field | Value |
|--------------|----------------------------------|
| System | The name of your Oracle instance |
| Source Type | Local |
| Event Filter | ibm.apps.ar.customer.create |
| Phase | 10 |
| Status | Enabled |
| Rule Data | Message |
| Action Type | Custom |
| On Error | Stop and Rollback |

4. Navigate to the next section, where you can enter additional information about the subscription, and enter the information presented in the following table.

Table 16. Additional subscription values

| Field | Value |
|----------------------|---------------------------------|
| PL/SQL Rule Function | ibm_websphere_pkg.rule_function |
| Priority | Normal |
| Owner Name | JTF |
| Owner Tag | JTF |

5. Save the subscription.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Creating the concurrent program

The Oracle E-Business Suite Business Event System requires a concurrent program to process data based on events, subscriptions, and actions that you define. After you create a custom event and a subscription, you must create a concurrent program. To do this, create a program executable, define properties for it, and grant permissions to it.

Creating the concurrent program executable:

The concurrent program is required to define actions for the Business Event System. The first task you must perform to create the concurrent program is to create a program executable for it. Then, you define properties for it and grant permissions to it.

Before you begin

Make sure you are using the System Administrator responsibility.

Procedure

1. Select the Concurrent: Program > Executable menu option.
2. Enter or select the following information in the form:

Table 17. Concurrent program executable values

| Field | Value |
|---------------------|---|
| Executable | IBMCUSTOMEREVENT |
| Short Name | IBMCUSTOMEREVENT |
| Application | Receivables |
| Execution Method | PL/SQL Stored Procedure |
| Execution File Name | IBM_WEBSPPHERE_PKG.RAISE_CUSTOMER_EVENT |

3. Save the program executable definition.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Defining properties of the concurrent program:

After you have created the program executable for the concurrent program, you must define properties for the concurrent program. This is the second of three tasks you must complete to make the concurrent program ready for use by the Business Event System.

Procedure

1. Select the Concurrent: Program > Define menu option.

2. Enter the following information:

Table 18. Concurrent program definition values

| Field | Value |
|--------------|--------------------------|
| Program | IBM Raise Customer Event |
| Short Name | IBMCUSTOMEREVENT |
| Applications | Receivables |
| Description | IBM Raise Customer Event |
| Executable | IBMCUSTOMEREVENT |

3. Save the concurrent program properties definition.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Enabling the concurrent program:

After you have created a program executable and defined properties for the concurrent program, you must grant permissions to it that enable it to direct the actions of the Business Event System. This is the final procedure you must complete to make the concurrent program ready for use by the Business Event System.

Procedure

1. Select the Security > Responsibility > Request menu option.
2. Change the form into Enter Query mode.
3. In the group field, specify: Receivables All
4. Run the query.
5. Navigate to the Requests (details) section and click the New button to add a new request in the details.
6. In the name field, specify: IBM Raise Customer Event
7. Save your changes.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Results

The Business Event System is set up for processing customer events.

Creating the project

To begin the process of creating and deploying a module to communicate with an Oracle E-Business Suite application, you start the external service wizard in WebSphere Integration Developer. The wizard creates an Adapter for JDBC project, which is used to organize the files associated with the adapter.

Before you begin

Ensure that you have access to the documentation for the Adapter for JDBC. To view it, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

Procedure

1. Review the information in “Configuring the module for deployment” in the *WebSphere Adapters: Adapter for JDBC User Guide*.
2. Perform the task “Creating the project” in the *WebSphere Adapters: Adapter for JDBC User Guide*. Follow the steps in this section to set up the project in your workspace on WebSphere Integration Developer.

Configuring the module for inbound processing

To process data sent from Oracle E-Business Suite to WebSphere Process Server or WebSphere Enterprise Service Bus, you must create business objects. To do this, use the external service wizard in WebSphere Adapter for JDBC. First, you initialize the wizard by configuring values of the properties that enable the adapter to set up a communication channel to a specific database. Then, run a query to discover business objects. Finally, specify the values of the connection properties for the export file.

Setting connection properties for the external service wizard

Set the values of the connection properties for your database instance. These properties enable the external service wizard in the Adapter for JDBC to connect to the Oracle database for discovery and for creating the service description.

Before you begin

This task is performed within the external service wizard, which you already started for the tasks under “Creating the project.”

Procedure

1. In the Required Files and Libraries window, next to the top pane for JDBC driver JAR files, click **Add**. Browse to the location of the external JDBC driver, select the driver JAR file, and click **Open**.
The JDBC driver that the wizard needs to add to your project is the external JAR file that is provided with your database, which you can obtain from your database administrator or from the Web site for the database.
2. Click **Next**.
3. In the Processing Direction window, click **Inbound** and click **Next**.
4. In the Discovery Configuration window, in the left pane, select your database vendor, driver, and version.
5. Enter the information for your database that is shown in the following table.

The table lists the properties required for the sample and their descriptions. For details on these properties, see “Connection properties for the external service wizard” in the “Reference” section of *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html

Table 19. Connection properties for the external service wizard

| Property | Description |
|----------|---|
| Database | The database name. For Oracle databases, this is the system ID (SID). Database is used along with host name and port number to generate the Database URL that is used to connect to the database. |

Table 19. Connection properties for the external service wizard (continued)

| Property | Description |
|------------------------|--|
| Host name | The host name or IP address of the database server. Host name is used along with database name and port number to generate the Database URL that is used to connect to the database. |
| Port number | The port number, which, along with database name and host name, generates the Database URL that is used to connect to the database. |
| JDBC driver class name | The class name of the JDBC driver that is used to connect to the database. |
| User name | The database account you are using. It is used during the discovery process. Later in the wizard, you can specify a different name and password to use at run time. |
| Password | The password for the account you are using. |

Click Next.

Results

The wizard uses the connection properties to connect to the database.

Business object selection and linking

For this sample for inbound processing you select the same objects and create the same object hierarchy as is used to perform the Retrieve operation in Sample 1. When you select business objects and services, you import the objects and set the application-specific information, such as the foreign key relationship between the parent and child objects.

Operations for the business objects

For this sample, the adapter supports the Create, Update, and Delete operations for customer data that is modified in Oracle E-Business Suite. When the concurrent program is run, it compares the date of its last run to the created and modified dates in the customer table. The following table shows what operations are assigned under what conditions.

Table 20. How operations are assigned

| Operation | Condition |
|---|--|
| A Create operation is assigned to the customer entry | If the creation date of a customer is later than the last run date of the concurrent program |
| An Update operation is assigned to the customer entry | If the creation date is prior to the last run date of the concurrent program |
| A Delete operation is assigned to the customer entry | If the Status of the customer is not Active (having an A in the status column) |

Base tables for the Accounts Receivable schema

The following table lists the base tables for the Accounts Receivable (AR) schema, including the table name, suggested synonym, and description. The next section shows a diagram of the relationship of the business objects that you configure for inbound processing.

Table 21. Base tables for the AR schema

| Table name | Synonym | Description |
|---------------------------|---------------------------|---|
| AR.HZ_CUST_ACCOUNTS | ArIbm_R_Cust_Accounts | Customer number, type, and other customer information. |
| AR.HZ_CUSTOMER_PROFILES | ArIbm_R_Cust_Profiles | High-level profile information for the customer. |
| AR.HZ_PARTIES | ArIbm_R_Parties | Customer name and if it is a person, not a business, the person's details. |
| AR.HZ_PARTY_SITES | ArIbm_R_Party_Sites | Links between customer's address IDs and address location IDs. |
| AR.HZ_CUST_ACCT_SITES_ALL | ArIbm_R_Acct_Sites | Address IDs. |
| AR.HZ_CUST_SITE_USES_ALL | ArIbm_R_Site_Uses | Customer business address purposes (BILL_TO, SHIP_TO, etc.). |
| AR.HZ_LOCATIONS | ArIbm_R_Locations | Address details. |
| AR.HZ_CONTACT_POINTS | ArIbm_R_Contacts_Sites | Contact and phone details. The table is listed twice because both sites and customers (PARTY_SITES and PARTIES) use the same table for contact details. |
| AR.HZ_CONTACT_POINTS | ArIbm_R_Contacts_Customer | Contact and phone details. The table is listed twice because both sites and customers (PARTY_SITES and PARTIES) use the same table for contact details. |

Relationship of the business objects

For inbound processing to succeed, the links connecting the customer object to its child objects must match those in the Oracle database. The following diagram shows the relationship of the objects you must configure for inbound processing.

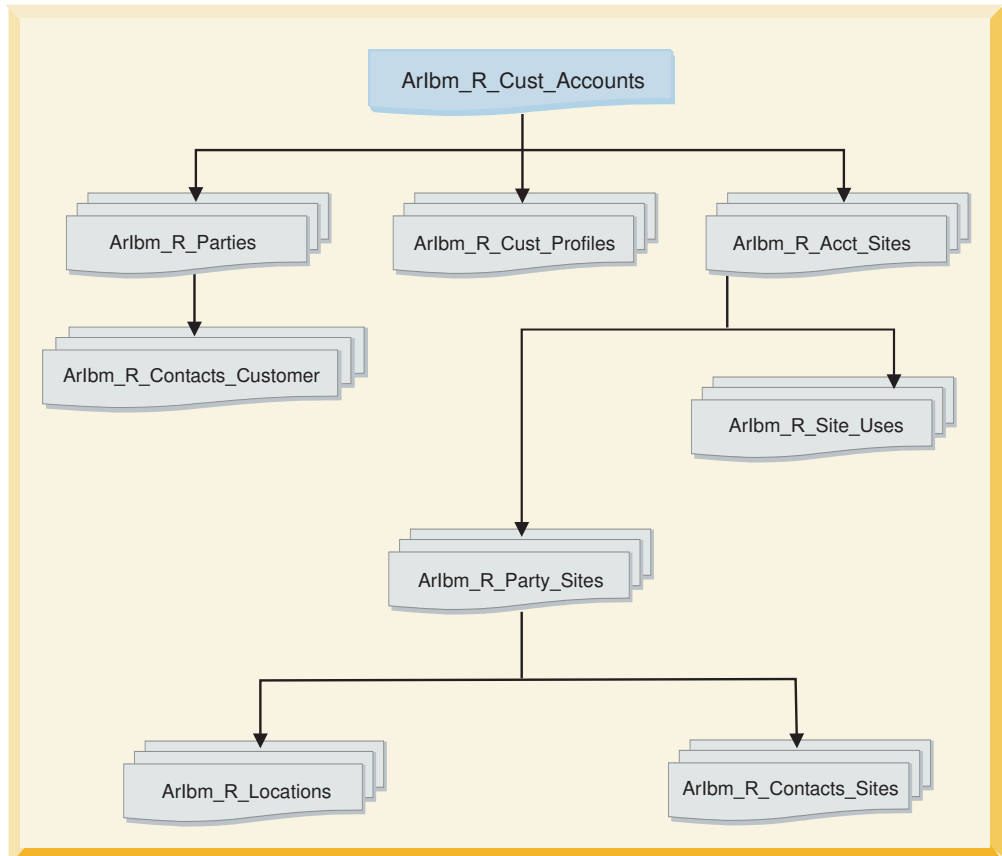


Figure 16. Relationship of objects for inbound processing

To link the objects, you import the child objects and set the foreign key relationship between the parent and child objects. For more information on the nature of business objects, refer to the “Business objects” section of the “Technical overview” in *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

Selecting business objects and services

After configuring connection properties, run a query for database objects. You can browse the metadata tree structure to understand the structure of the objects in the Oracle database, and select objects needed for the service description. Link the parent and child objects.

About this task

For this sample, you run a query on the accounts receivable (AR) schema and select synonyms that represent the data in the Oracle E-Business Suite base tables.

Procedure

1. Specify filter properties
 - a. In the Object Discovery and Selection window, click **Edit Query**.
 - b. In the Query Properties window, type AR in the **Schema name pattern** field to display the accounts receivable (AR) schema.
 - c. Select the check box **Prompt for additional configuration settings when adding business object**, and click **OK**.

Now whenever you select one of the discovered objects for import, you are prompted to enter application-specific information for the object.

2. Run the metadata query
 - a. Display objects discovered by the query.

Click **Run Query**. The AR schema and data elements of the schema are displayed.
 - b. Select the object for import

Expand the **AR** schema. Expand **Synonyms - Nicknames**. Highlight the synonym **IBM_R_CUST_ACCOUNTS**, and click > (the Add icon) to select this object to be imported.
 - c. Add business object application-specific information

In the Configuration Properties for **IBM_R_CUST_ACCOUNTS** window, click **Add** to select the primary key for the table related to **IBM_R_CUST_ACCOUNTS**. Select **CUST_ACCOUNT_ID** and click **OK**.
 - d. Select another object for import and link child object to parent

In the Object Discovery and Selection window, highlight the synonym **IBM_R_CUST_PROFILES**, and click > (the Add icon) to select this object to be imported.

In the Configuration Properties for **IBM_R_CUST_PROFILES** window, click **Add** to select the primary key. Select **APPLICATION_ID** and click **OK**.

Set the parent for the table related to **IBM_R_CUST_PROFILES** by selecting **IBM_R_CUST_ACCOUNTS**.

Under **Build a foreign key relationship by selecting a parent table column for each child column**, link the child to the parent's attributes by selecting a foreign key attribute and a foreign key value for that attribute.

For the foreign key attribute **CUST_ACCOUNT_ID**, select the foreign key attribute value **CUST_ACCOUNT_ID**.

Check **Parent object owns child object (cascade delete)** and click **OK**.

Configuration Properties for 'IBM_R_CUST_PROFILES'

Specify configuration properties for the selected object.

Select primary key for table IBM_R_CUST_PROFILES
 Select one or more columns:*

APPLICATION_ID Add...
Remove

Choose parent table from the list for the selected child
 Choose parent table : IBM_R_CUST_ACCOUNTS (AR) ▼
 Single cardinality

Build a foreign key relationship by selecting a parent table column for each child column

CUST_ACCOUNT_PROFILE_ID: NONE ▼
 LAST_UPDATED_BY: NONE ▼
 LAST_UPDATE_DATE: NONE ▼
 LAST_UPDATE_LOGIN: NONE ▼
 CREATED_BY: NONE ▼
 CREATION_DATE: NONE ▼
 CUST_ACCOUNT_ID: CUST_ACCOUNT_ID ▼
 STATUS: NONE ▼

Figure 17. Specify configuration properties

e. Select remaining synonyms

Repeat step 2d for each of the remaining synonyms **in the order the synonyms are presented in the following table**. Use the primary key, parent, foreign key attribute, and foreign key attribute value shown for each synonym in the table.

Table 22. Objects requiring foreign key values and child objects

| Synonym | Primary key | Parent | Foreign key attribute | Foreign key attribute value |
|---------------------------|---|-----------------------|-----------------------|-----------------------------|
| ArIbm_R_Parties | party_id | Aribm_R_Cust_Accounts | party_id | party_id |
| ArIbm_R_Acct_Sites | party_site_id and cust_acct_site_id | Aribm_R_Cust_Accounts | cust_account_id | cust_account_id |
| ArIbm_R_Party_Sites | location_id and party_site_id | Aribm_R_Acct_Sites | party_site_id | party_site_id |
| ArIbm_R_Site_Uses | application_id | Aribm_R_Acct_Sites | cust_acct_site_id | cust_acct_site_id |
| ArIbm_R_Locations | address_key | Aribm_R_Party_Sites | location_id | location_id |
| ArIbm_R_Contacts_Sites | application_id | Aribm_R_Party_Sites | owner_table_id | party_site_id |
| ArIbm_R_Contacts_Customer | application_id | Aribm_R_Parties | owner_table_id | party_id |

When you are finished, click **Next**.

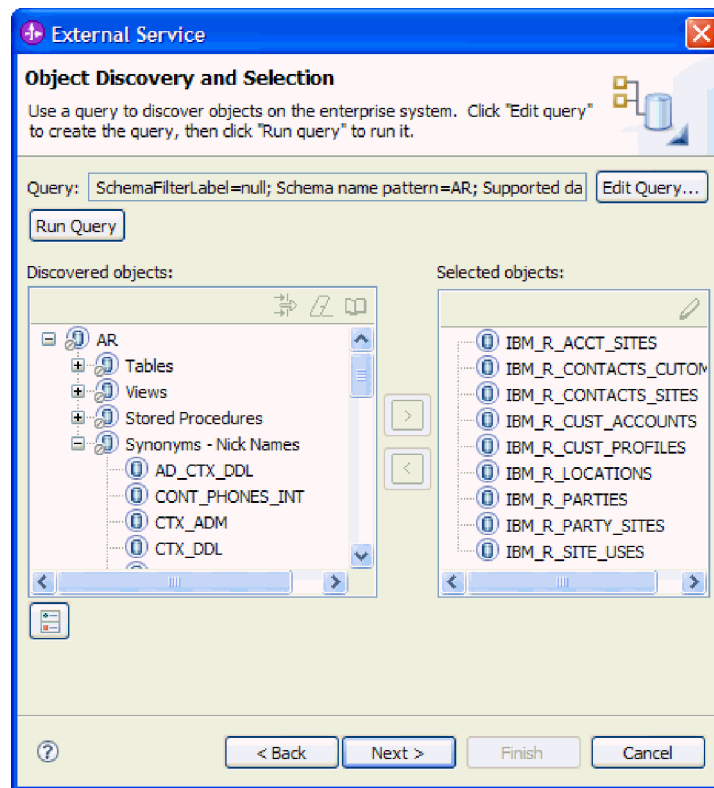


Figure 18. Select remaining synonyms

Configuring the selected objects

After you have selected and linked database objects, specify the operations and other properties that apply to the selected objects.

Procedure

1. Select operations

In the Configure Composite Properties window of the external service wizard, the Operations pane lists the operations that the adapter supports for the inbound service type. Do not change the list of operations.

The following operations are displayed:

- **Create**
- **Update**
- **Delete**

2. Leave the default values for these fields:

- **Business object namespace**
- **Folder**
- **Generate a business graph for each business object**

3. Click Next.

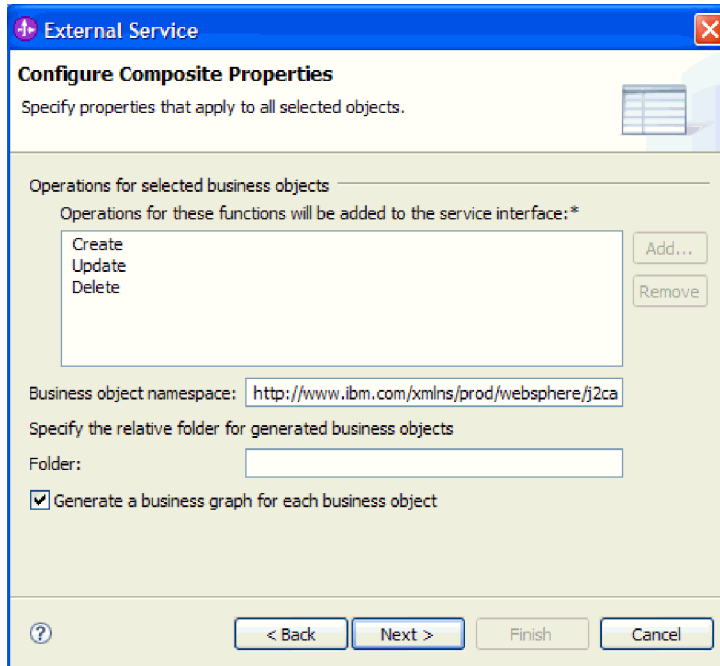


Figure 19. Configuring the selected objects

What to do next

Next, you will specify deployment information to use at run time and information for saving the service as a module.

Setting deployment properties and generating the service

Configure the activation specification and resource adapter properties that the adapter uses to set up a communication channel to a specific database for inbound processing. Then create a business integration module that contains all of the artifacts and property values.

About this task

Use the external service wizard to configure the properties.

Procedure

1. Set configuration property values
 - a. In the Service Generation and Deployment Configuration window, in the **Deploy connector project** field, ensure that this option is selected: **With module for use by single application**.
 - b. Click **Advanced** to see all of the properties under **Connection properties, Database system connection information**.
For inbound processing, the property fields are displayed for the activation specification and resource adapter properties.
 - c. Set the values of the required connection properties as shown in the following table.

The fields are initialized with the connection information you specified when you started the wizard. You can edit the existing values for Database vendor and User name, as required. You must type your Password.

Table 23. Database system connection information

| Properties | Values |
|-----------------|--|
| User name | The database account you are using |
| Password | The password for the account you are using |
| Database vendor | Oracle |

d. Click **Next**.

For information about these properties, refer to the “Reference” section of *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

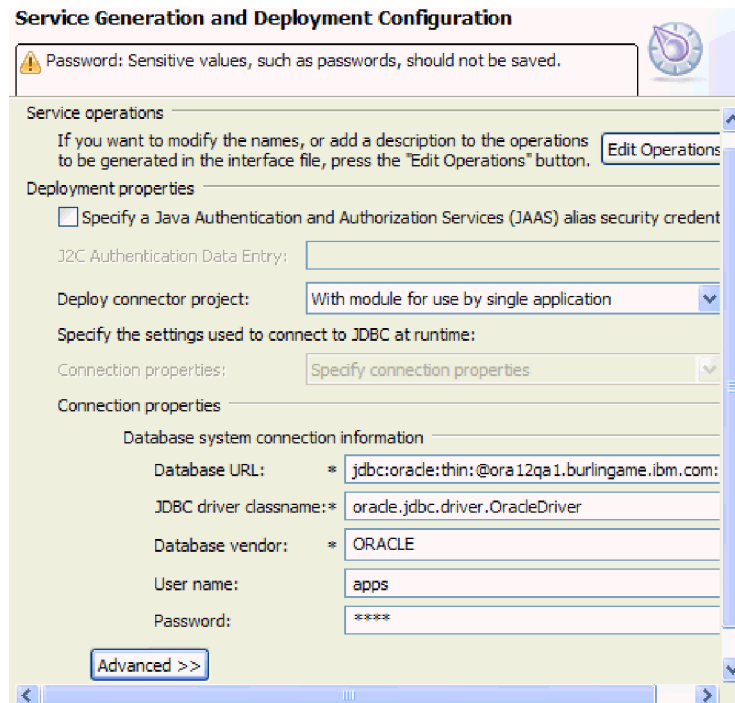


Figure 20. Set deployment properties

2. Create a module
 - a. In the Service Location Properties window, click **New**, next to the **Module** field.
 - b. In the Integration Project window, confirm that **Create a module project** is selected and click **Next**.
 - c. In the Module window, type `Oracle_Inbound` and click **Finish**.
 - d. In the Service Location Properties window, click **Finish**.

Results

This module will hold all of the generated artifacts.

Saving the module

You must save the business integration module for inbound processing that contains the artifacts and property values you created.

Procedure

1. In the Business Integration view, click the **Oracle_Inbound** module.
2. Select **File** → **Save**.

Note: If the option to **Save** is inactive (grey), then the module has already been saved.

Creating an inbound component

Because this sample is a standalone application, it is not wired to a processing export as a production application would be. Therefore, before you can deploy the project, you must create a component that will pick up inbound events, retrieve data, and display the retrieved data on the administrative console where you can review it.

Creating the component

Use the Assembly Editor within WebSphere Integration Developer to create a new component. You must do this so that the inbound sample application has an export that can perceive events, and retrieve and process data. This procedure is required because the sample application will be deployed to a test environment.

Procedure

1. Open the Assembly Diagram
From the Business Integration view within WebSphere Integration Developer, expand the **Oracle_Inbound** module. Double-click **Assembly Diagram**.
This opens the Assembly Diagram for the JDBCInboundInterface component.
2. On the left side of the Assembly Diagram, expand **Components** and click **Untyped Component**.

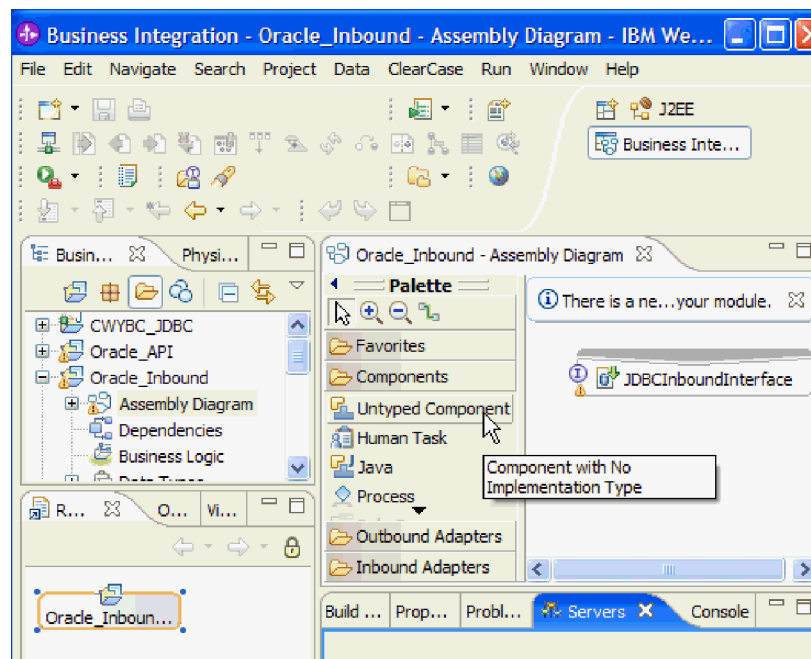


Figure 21. Create the component

3. To create the new Untyped component, click on the Assembly Diagram background.

- The new component is displayed in the Assembly Diagram as **Component1**.
4. Hover over the right side of the **JDBCInboundInterface** component until a yellow wire is displayed. Click on the wire and drag it to the left side of **Component1**.

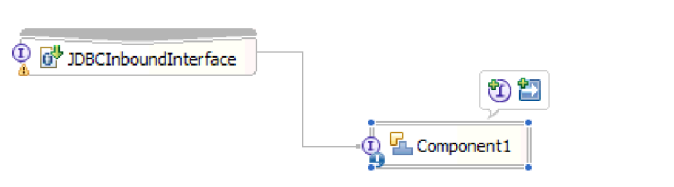


Figure 22. Add wire

5. In the **Add Wire** dialog box, click **OK**.

What to do next

Next, you will define the implementation of the new inbound component.

Defining the implementation

To enable the component to retrieve objects posted to the event store in the Adapter for JDBC and print them to the administrative console, add code that defines the behavior of the new inbound component.

Procedure

1. Right click **Component1** in the **Assembly Diagram** for **JDBCInboundInterface**.
2. Click **Generate Implementation** → **Java**.
3. In the **Generate Implementation** window, confirm that **default package** is selected, and click **OK**.
4. The **Component1Impl.java** window is displayed, where you can insert custom code.

At the top, add to the rest of the import statement:

```
import com.ibm.j2ca.base.AdapterBOUTil;
```

Replace the default sections of the implementation with code that directs the component to retrieve objects posted to the event store and to print them to the administrative console. Use the code in the next section in this user guide, titled “Java code for inbound component implementation.”

5. When you are finished adding the custom code, select **File** → **Save** to save the component code. Close the **Component1Impl.java** window.
6. Select **File** → **Save** to save the **Assembly Diagram**.

Java code for inbound component implementation

Custom Java code replaces default code in the inbound component. Three code blocks are provided that enable printing to the administrative console: code for printing a created object, an updated object, and a deleted object.

Insert the following code blocks in the appropriate empty methods in the inbound component, For example, for the first method block, locate:

```
public void createArIbm_R_Cust_AccountsBG(
    DataObject createArIbmRCustAccountsBGInput{}
```

then replace the empty method with the first code block.

After you have inserted all three code blocks, return to the previous section of this user guide to save the component code and the assembly diagram.

Code for printing a created object

```
public void createArIbm_R_Cust_AccountsBG(
    DataObject createArIbmRCustAccountsBGInput) {
    System.out.println("End point for createArIbm_R_Cust_AccountsBG ");
    int i = createArIbmRCustAccountsBGInput.getType().getProperties().size()-1;
    DataObject dataObj = createArIbmRCustAccountsBGInput.getDataObject(i);
    System.out.println("Data object to be traced: " + dataObj.getType().getName());
    try
    {
        String xmlString = AdapterBOUtil.serializeDataObject(dataObj);
        System.out.println(xmlString);
    }
    catch(Exception e)
    {
        System.out.println("Error in tracing the data object");
    }
}
```

Code for printing an updated object

```
public void updateArIbm_R_Cust_AccountsBG(
    DataObject updateArIbmRCustAccountsBGInput) {
    System.out.println("End point for updateArIbm_R_Cust_AccountsBG ");
    int i = updateArIbmRCustAccountsBGInput.getType().getProperties().size()-1;
    DataObject dataObj = updateArIbmRCustAccountsBGInput.getDataObject(i);
    System.out.println("Data object to be traced: " + dataObj.getType().getName());
    try
    {
        String xmlString = AdapterBOUtil.serializeDataObject(dataObj);
        System.out.println(xmlString);
    }
    catch(Exception e)
    {
        System.out.println("Error in tracing the data object");
    }
}
```

Code for printing a deleted object

```
public void deleteArIbm_R_Cust_AccountsBG(
    DataObject deleteArIbmRCustAccountsBGInput) {
    System.out.println("End point for deleteArIbm_R_Cust_AccountsBG ");
    int i = deleteArIbmRCustAccountsBGInput.getType().getProperties().size()-1;
    DataObject dataObj = deleteArIbmRCustAccountsBGInput.getDataObject(i);
    System.out.println("Data object to be traced: " + dataObj.getType().getName());
    try
    {
        String xmlString = AdapterBOUtil.serializeDataObject(dataObj);
        System.out.println(xmlString);
    }
    catch(Exception e)
    {
        System.out.println("Error in tracing the data object");
    }
}
```

Important:

Return to the previous section of the user guide, titled “Defining the implementation,” to perform the steps to save the component code and the assembly diagram.

Deploying the module for testing

To test the adapter application, install the module for inbound processing in the WebSphere Integration Developer integration test client.

Before you begin

The module that contains an Oracle database export file is produced by running the external service wizard.

Procedure

1. In WebSphere Integration Developer, click the **Servers** tab to open the Servers view.
2. If your server is not displayed in the Servers view, right-click the Servers view and select **New** → **Server**. Select your WebSphere Process Server or WebSphere Enterprise Service Bus.
Click **Next** and click **Finish**.
3. If your WebSphere Process Server or WebSphere Enterprise Service Bus is not running, right-click the name of it and select **Start**. The **Status** changes to **Started**.
4. Right-click the name of your WebSphere Process Server or WebSphere Enterprise Service Bus, and select **Add and remove projects**.
5. From the **Available Projects** list on the left, select the **Oracle_InboundApp** module, and click **Add**.

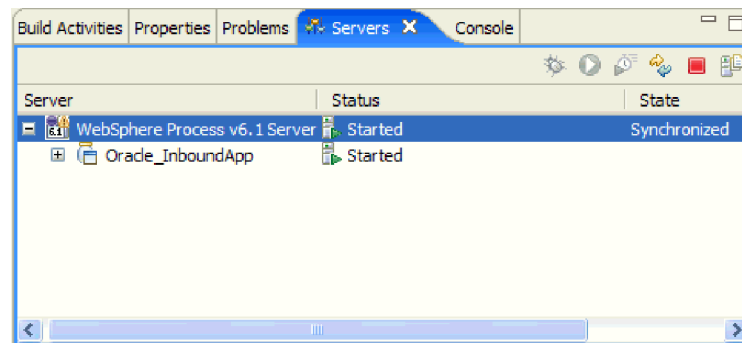


Figure 23. Servers view

6. Click **Finish**.

What to do next

Next, you can test the assembled application in the integration test client.

Testing the assembled adapter application

Test the generated service to verify that it produces the expected results. You complete three tasks to test this sample. In Oracle Receivables, you create a customer record. Then you run the concurrent program to raise the customer object event for the Create operation. Finally, in WebSphere Integration Developer, you use the test client to verify that the event has been processed.

Starting the test client for the module to retrieve the event

In WebSphere Integration Developer, start the test client to begin the process of testing the module to verify that the inbound event has been processed and received.

Procedure

1. Select the module to test
In WebSphere Integration Developer go to the Business Integration view of the Business Integration perspective.
2. Right-click the **Oracle_Inbound** module, and select **Test > Attach**.
The Oracle_Inbound_Test window is displayed.

What to do next

Next, you will set up a customer record to test the assembled application.

Creating a customer to test the assembled adapter application

Set up a customer record in Oracle Receivables to test the inbound processing scenario.

Before you begin

Start the test client before you create the customer record.

Procedure

1. Log in to your Oracle Applications instance as an Oracle E-Business Suite user.
2. Select the Receivables Manager responsibility.
If this responsibility is not available in the responsibilities list, you must add it with user-defined security.
3. Select the Customers->Standard menu option.
4. In the Customer window, click the Create button to create a new customer.
5. In the Create Organization window, enter or select all required information to create a new customer record.
 - Type Organization Name (customer name)
 - Select Country value if it is not defined
 - Type Address, City, and County
 - Select State value
 - Type Postal Code (fields in yellow background are required fields)
 - Set 'Bill TO' in the Purpose selection
6. Click Apply and Save to save the customer's information.

Note:

For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

What to do next

Next, you will run the concurrent program to process the customer event.

Running the concurrent program to test the sample

Run the concurrent program within Oracle to process the customer event and ensure that it works as you expect.

Before you begin

You must have created a customer record in Oracle Receivables before performing this task.

Procedure

1. Use the Home link to return to the Receivables Manager responsibility.
2. Click the Control: Requests menu option -> Run.
3. When the Submit a New Request dialog box is displayed, confirm that the button for Single Request is selected, and then click OK.
4. In the Submit Request window, select IBM Raise Customer Event from the Name field list and submit the request.
5. If an option to submit another request appears, click No.
6. Select View -> Requests from the main menu.
7. In the Find Requests window, click Find.
8. After the concurrent request completes with normal status, navigate to where you can view the process log.

The log screen shows the event details, including the customer ID and event key, for the customer that was created. For example:

```
Raising event for customer id: 1234  
Event Key: 100000
```

If the concurrent request does not complete with normal status, view the log file to see what error may have occurred. For example, a spelling mistake could have been made when the concurrent program was created. Resolve the error and rerun the concurrent program. For information about errors, refer to your documentation for Oracle Applications.

9. The test window displays the event in the WebSphere Integration Developer Console view. Verify that the returned object matches the customer information that you entered when you created the customer record.

Clearing the sample content

After you have tested the application, clear the sample content to return the data to its original state. You must deactivate the customer in Oracle E-Business Suite, and run SQL commands to delete the stored procedure.

Procedure

1. Deactivate the customer.

Use the Oracle E-Business Suite Client to log into the Receivables Manager responsibility. Go to the Customers > Standard menu and find the customer. Clear the check from the Active box on the Customer window, and save the customer.

For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

2. Use the SQL*Plus tool, or a similar program for processing SQL commands, to log into the database and run commands to clear artifacts from the database. Consult your database administrator if you need help performing this task.

Remove the following artifacts:

- IBM_CUSTOMER_EVENT_KEY_S
- IBM_WEBSPHERE_PKG
- IBM_WEBSPHERE_EVENTS

To remove tables, run the Drop command.

Sample 3: Calling an Oracle E-Business Suite API for outbound processing

The adapter can call directly any Oracle E-Business Suite API that uses simple data parameters. Follow the steps in this sample to call an API that creates a concurrent program executable for outbound processing. By performing this scenario, you can see how to structure simple data API calls between the adapter and an Oracle E-Business Suite database.

Related concepts

“Outbound processing sample applications” on page 4

Some outbound processing sample applications are provided with this solution. Sample 1 uses Oracle database interface tables, which is the standard outbound scenario for the Oracle E-Business Suite. Sample 3 uses API calls, while Sample 4 makes use of a workflow.

Creating the project

To begin the process of creating and deploying a module to communicate with an Oracle E-Business Suite application, you start the external service wizard in WebSphere Integration Developer. The wizard creates an Adapter for JDBC project, which is used to organize the files associated with the adapter.

Before you begin

Ensure that you have access to the documentation for the Adapter for JDBC. To view it, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

Procedure

1. Review the information in “Configuring the module for deployment” in the *WebSphere Adapters: Adapter for JDBC User Guide*.
2. Perform the task “Creating the project” in the *WebSphere Adapters: Adapter for JDBC User Guide*. Follow the steps in this section to set up the project in your workspace on WebSphere Integration Developer.

Configuring the module for outbound processing for API calls

To configure the module, create a business object that represents the stored procedure API call. Use the external service wizard in WebSphere Integration Developer for WebSphere Adapter for JDBC to perform the steps necessary to complete this task.

About this task

First, you configure the values of the properties that enable the adapter to set up a communication channel to a specific database. Then, run a query to discover business objects. Finally, you must specify operations and other properties that apply to the selected business object.

Setting connection properties for the external service wizard

Set the values of the connection properties for your database instance. These properties enable the external service wizard to connect to the Oracle database for discovery and for creating the service description.

Before you begin

This task is performed with the external service wizard, which you already started for the tasks under “Creating the project.”

Procedure

1. In the Required Files and Libraries window, next to the top pane for JDBC driver JAR files, click **Add**. Browse to the location of the external JDBC driver, select the driver JAR file, and click **Open**.

The JDBC driver that the wizard needs to add to the project is the external JAR file that is provided with your database, or that you can obtain from your database administrator or from the Web site for the database.

2. Click **Next**.
3. In the Processing Direction window, click **Outbound** and click **Next**.
4. In the Discover Configuration window, in the left pane, select your database vendor, driver, and version.
5. Enter the information for your database that is shown in the following table.

The table lists the properties required for the sample and their descriptions. For details on these properties, see “Connection properties for the external service wizard” in the “Reference” section of the *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html

Table 24. Connection properties for the external service wizard

| Property | Description |
|------------------------|---|
| Database | The database name. For Oracle databases, this is the system ID (SID). Database is used along with host name and port number to generate the Database URL that is used to connect to the database. |
| Host name | The host name or IP address of the database server. Host name is used along with database name and port number to generate the Database URL that is used to connect to the database. |
| Port number | The port number, which, along with database name and host name, generates the Database URL that is used to connect to the database. |
| JDBC driver class name | The class name of the JDBC driver that is used to connect to the database. |
| User name | The database account you are using. It is used during the discovery process. Later in the wizard, you can specify a different name and password to use at run time. |
| Password | The password for the account you are using. |

Click **Next**.

Results

The wizard uses the connection properties to connect to the database.

Selecting business objects and services

After setting connection properties, run a query to search for database objects. You can browse the metadata tree of discovered objects to understand the structure of the objects in the Oracle database, and select objects needed for the service description.

About this task

For this sample, you run a query on the applications (APPS) schema and select the stored procedure that the business object will represent.

Procedure

1. Specify filter properties

In the Object Discovery and Selection window, click **Edit Query**.

In the Query Properties window, type APPS in the **Schema name pattern** field to display the applications (APPS) schema.

The **Supported database object types** field lists the entries: tables, views, stored procedures, and synonyms - nicknames. Remove the nodes for **Tables**, **Views**, and **Synonyms - Nicknames**, and click **OK**.

2. Run the metadata query

- a. Display objects covered by the query

Click **Run Query**.

The APPS schema and data elements of the schema are displayed.

- b. Select the object for import

Expand the **APPS** schema. Click **Stored Procedures** but do not expand the node. Use a filter to help you locate and select the stored procedure.

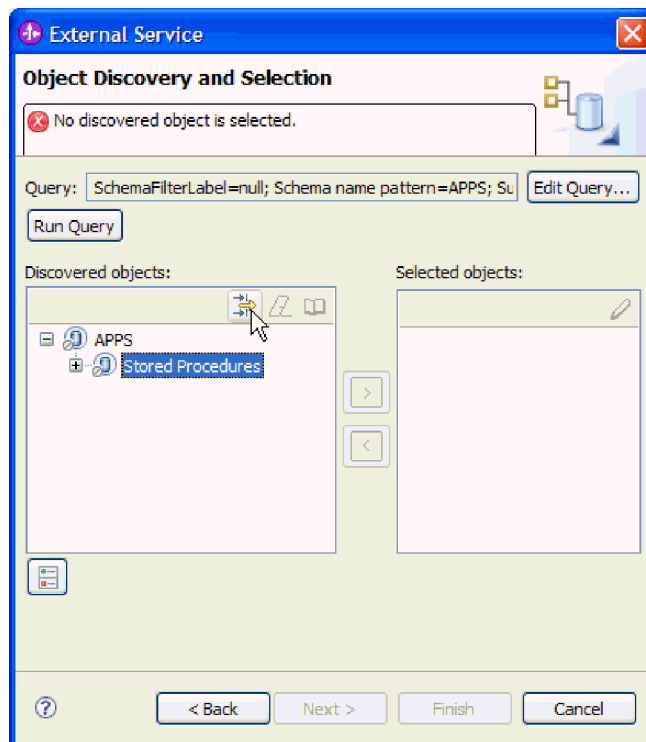


Figure 24. Use a filter to find the stored procedure

Click the Filter icon in the **Discovered objects** pane. In the Filter Properties for Stored Procedures window, in the **Object Name Filter** field, type EXECUTABLE and click **OK**.

Expand the **Stored Procedures** node. The adapter returns a list of stored procedures.

Note: Because of the high number of stored procedures in the schema, the list might not be populated immediately.

Select **EXECUTABLE(FND_PROGRAM)** from the list of stored procedures. Click > (the Add icon).

From the Configuration Properties for EXECUTABLE(FND_PROGRAM) window, click **OK**. The stored procedure is added to the **Selected objects** pane.

Click **Next**.

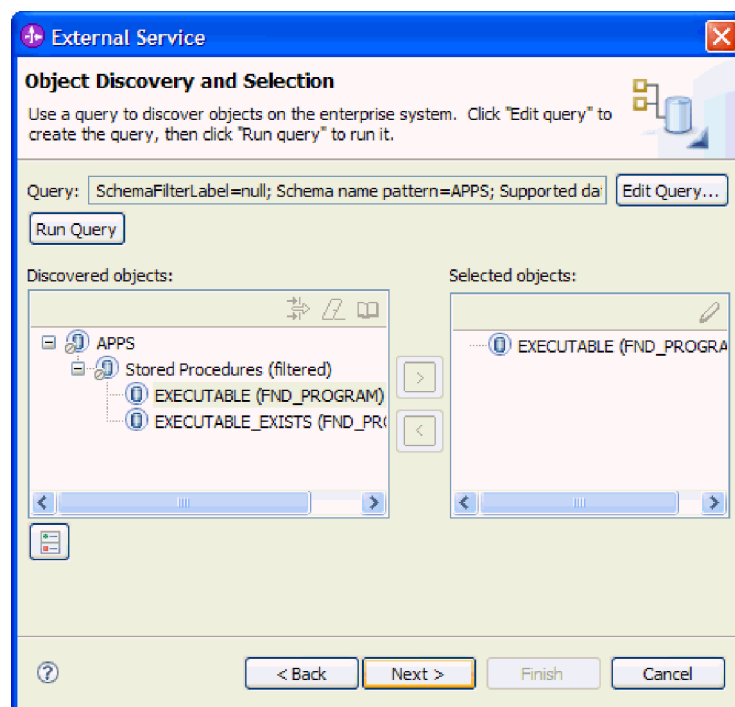


Figure 25. Select stored procedure

Configuring the selected objects

After you have selected database objects, you specify values for properties that apply to all selected objects.

Procedure

1. In the Configure Composite Properties window of the external service wizard, leave the default values for these fields:
 - **Maximum records for RetrieveAll operation**
 - **Business object namespace**
 - **Folder**
 - **Generate a business graph for each business object**
2. Click **Next**.

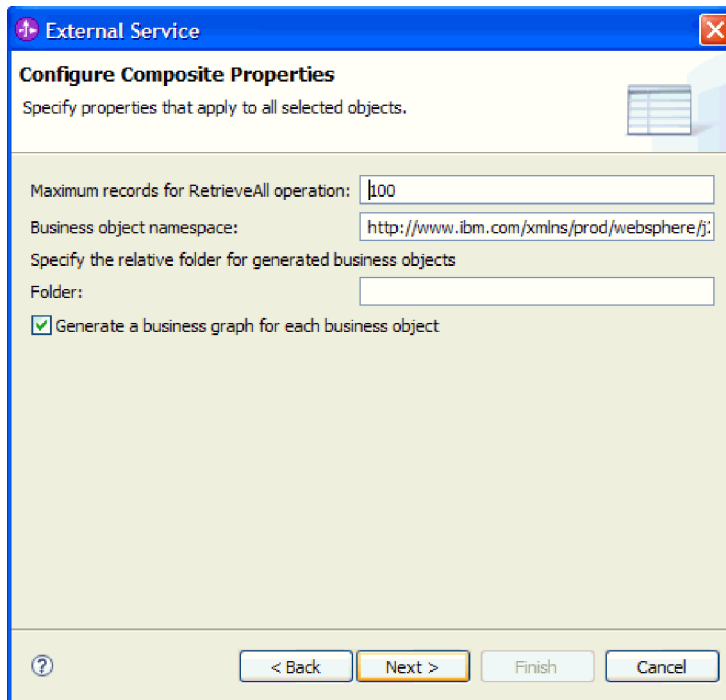


Figure 26. Specify composite properties

Setting deployment properties and generating the service

Use the external service wizard to configure the properties that the adapter uses to set up a communication channel to a specific database for outbound processing. Then create a business integration module where all of the artifacts and property values can be saved.

Procedure

1. Set configuration property values
 - a. In the Service Generation and Deployment Configuration window, in the **Deploy connector project** field, ensure that this option is selected: **With module for use by single application**.
 - b. Click **Advanced** to see all of the properties under **Connection properties, Database system connection information**.
For outbound processing, the property fields are displayed for the managed connection factory and resource adapter properties.
 - c. Set the values of the required connection properties as shown in the following table.

The fields are initialized with the connection information you specified when you started the wizard. You can edit the existing values for Database vendor and User name, as required. You must type your Password.

Table 25. Database system connection information

| Properties | Values |
|-----------------|--|
| User name | The database account you are using |
| Password | The password for the account you are using |
| Database vendor | Oracle |

d. Click **Next**.

For information about these properties, refer to the “Reference” section of *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

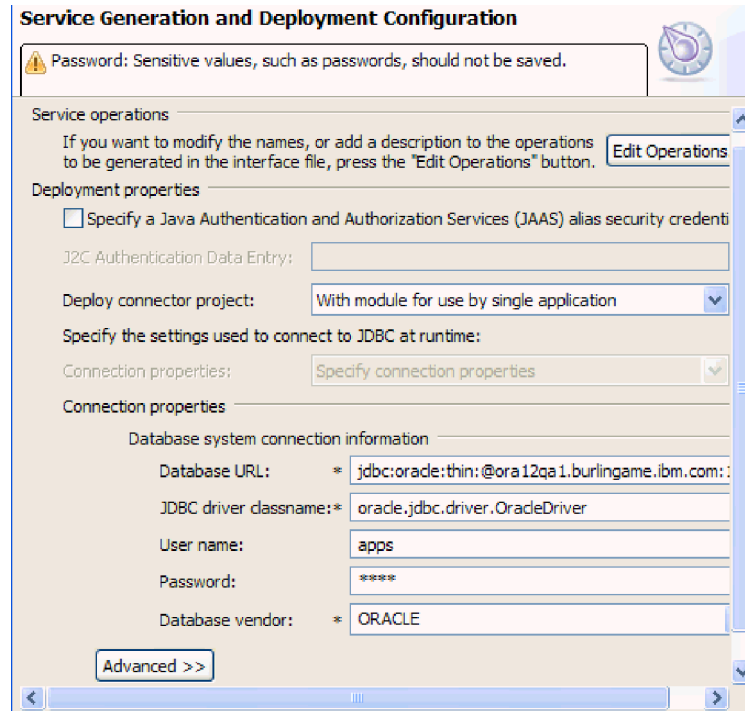


Figure 27. Set deployment properties

2. Create a module
 - a. In the Service Location Properties window, click **New** next to the **Module** field.
 - b. In the Integration Project window, confirm that **Create a module project** is selected and click **Next**.
 - c. In the Module window, type `Oracle_API` and click **Finish**.

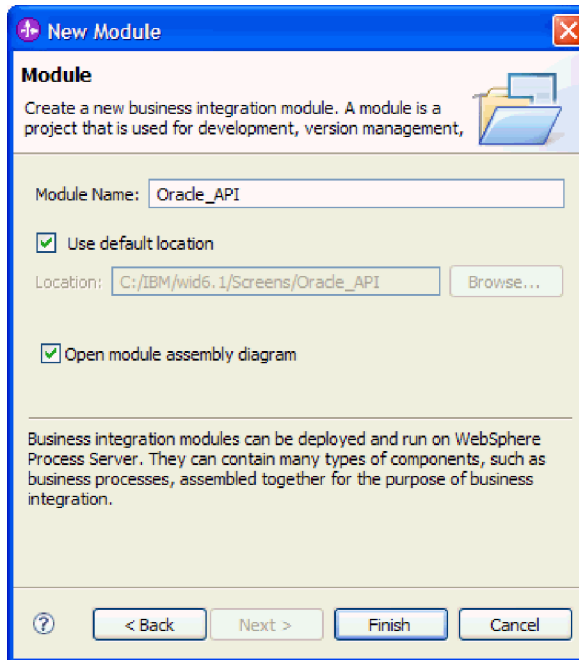


Figure 28. Create a module

- d. In the Service Location Properties window, click **Finish**.

Results

This module will hold all of the generated artifacts.

Saving the module

You must save the business integration module for outbound processing that contains the artifacts and property values you created.

Procedure

1. In the Business Integration view, click the **Oracle_API** module.
2. Select **File** → **Save**.

Note: If the **Save** option is inactive (grey), the module has already been saved.

Deploying the module for testing

Before you can test the adapter application, you must install the module for outbound processing in the WebSphere Integration Developer integration test client.

Before you begin

The module that contains an Oracle database import file is produced by running the external service wizard.

Procedure

1. In WebSphere Integration Developer, click the **Servers** tab to open the Servers view.

2. If your server is not displayed in the Servers view, right-click the Servers view and select **New** → **Server**. Select your WebSphere Process Server or WebSphere Enterprise Service Bus.
Click **Next** and click **Finish**.
3. If your WebSphere Process Server or WebSphere Enterprise Service Bus is not running, right-click the name of it and select **Start**. The **Status** changes to **Started**.
4. Right-click the name of your WebSphere Process Server or WebSphere Enterprise Service Bus, and select **Add and remove projects**.
5. From the **Available Projects** list on the left, select the **Oracle_APIApp** module, and click **Add**.
6. Click **Finish**.

Testing the assembled adapter application

Test the generated service to verify that it produces the expected results. Use the test client in WebSphere Integration Developer to set test values and run the service. Then in Oracle E-Business Suite, check to see that the service has run properly.

Testing in the WebSphere Integration Developer test client

Now that you have deployed your outbound processing module to the runtime environment, you can test the assembled application using the WebSphere Integration Developer integration test client. After testing in the test client, verify your results in Oracle E-Business Suite.

Procedure

1. Select the module to test
In WebSphere Integration Developer, go to the Business Integration view of the Business Integration perspective. Right-click the **Oracle_API** module, and select **Test** → **Test Module**.
2. Leave the default value for **verb**.
The value may be different from the one shown in the “Set attribute values” figure below.
3. Set the test values
In the Oracle_API_Test window, use the arrows to set the test values, as shown in the following table:

Table 26. Test values

| Item | Test value |
|---------------|---------------------------------------|
| Configuration | Default Module Test |
| Module | Oracle_API |
| Component | JDBCOutboundInterface |
| Interface | JDBCOutboundInterface |
| Operation | executeAppsFnd_ProgramU46ExecutableBG |

Note: The test values for **Component** and **Interface** may include a number, for example, JDBCOutboundInterface1. WebSphere Integration Developer appends numbers so that each name in the workspace is unique.

4. Set the following attribute values for the input business object:

Important: To set a value for an attribute to <null>, right-click the value field for the attribute. Select **Set to** and scroll down to select <null> .

Table 27. Attribute values for the input business object

| Attribute name | Value |
|---------------------|---|
| executable | IBMSAMPLENAME This value must be unique each time the sample is run. |
| application | AR |
| Short_name | IBMSAMPLENAME This value must be unique each time the sample is run. |
| Description | Concurrent Program Executable Definition |
| Execution_method | PL/SQL Stored Procedure |
| Execution_file_name | TEST_PACKAGE.TEST_PROCEDURE |
| Subroutine_name | <null> |
| Icon_name | <null> |
| Language_code | US |
| Execution_file_path | <null> |

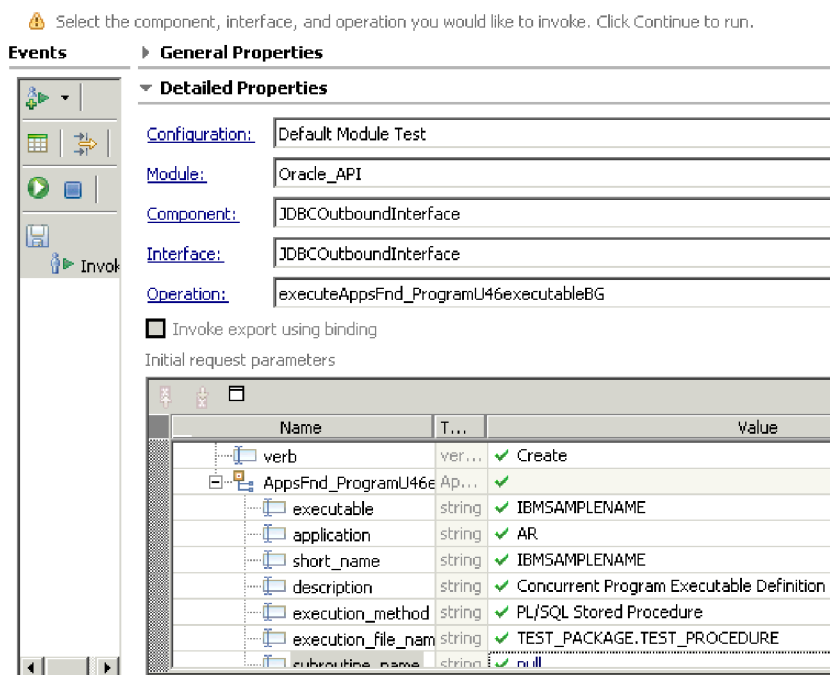


Figure 29. Set attribute values

5. In the left pane below **Events**, click **Continue**.
6. In the Deployment Location window, select the name of your WebSphere Process Server or WebSphere Enterprise Service Bus and click **Finish**. The Starting the Integration Client window is displayed briefly.

What to do next

Next, you can verify the results in Oracle E-Business Suite.

Verifying the test results in Oracle E-Business Suite

After running the test in the WebSphere Integration Developer integration test client, verify that the concurrent program has been created in Oracle E-Business Suite.

Procedure

1. Log in to Oracle E-Business Suite and select the System Administration responsibility.
2. Select the Concurrent: Program menu option.
3. From the Concurrent Program window, type IBMSAMPLENAME in the Short Name field.
4. Run the query.

The executable details are displayed. These details should match the input that was provided to create the concurrent program executable.

Clearing the sample content

After testing and verifying your test results, clear the sample content to return the data to its original state in Oracle E-Business Suite.

About this task

You need to remove the executable program. You can locate it just as you did in the previous section when you verified the test results.

Procedure

1. Log in to Oracle E-Business Suite, and select the System Administration responsibility.
2. Select the Concurrent: Program menu option.
3. From the Concurrent Program window, type IBMSAMPLENAME in the Short Name field.
4. Run the query.
5. After the query has run, click Edit and then click Delete.
6. Save your changes.

Sample 4: Importing a business object using the Business Event System and workflow

The Adapter for JDBC sends an outbound request to Oracle E-Business Suite to create customer data in Oracle's custom event table. From that event, the Oracle Business Event System triggers a workflow that creates the customer record in Oracle E-Business Suite. The workflow also handles errors during this process.

About this task

During outbound processing, the Adapter for JDBC uses a Create operation to insert customer data into the custom event table. The adapter also invokes a stored procedure, *AfterCreateSP*, that creates a custom event in the Business Event System. The Business Event System has a built-in subscription to this event. It builds a workflow that creates a customer record in Oracle E-Business Suite.

Related concepts

“Outbound processing sample applications” on page 4

Some outbound processing sample applications are provided with this solution. Sample 1 uses Oracle database interface tables, which is the standard outbound scenario for the Oracle E-Business Suite. Sample 3 uses API calls, while Sample 4 makes use of a workflow.

Preparing for outbound processing

Perform the setup steps needed to prepare your environment for outbound processing. One step inserts stored procedures into Oracle E-Business Suite, while another step creates a custom workflow for receiving the customer event.

Inserting artifacts into Oracle E-Business Suite

Run four SQL scripts to insert the artifacts into the Oracle database for performing the sample. These artifacts include tables, stored procedures, an event package, triggers, and sequences.

About this task

For this outbound processing sample, the Business Event System in Oracle E-Business Suite is used to create a business event that triggers a workflow. A custom event table is created in Oracle E-Business Suite, then the Adapter for JDBC is used to populate data in this custom event table. The Adapter for JDBC uses the AfterCreateSP stored procedure to create a custom event in the database, which triggers the workflow.

The scripts in this task must be run to populate the Oracle database with the artifacts necessary for running this sample.

Procedure

1. Open the SQL*Plus tool (or similar program for processing SQL statements). Log on to SQL*Plus using a database account that has been granted database administrator (DBA) rights. This sample uses the Oracle Applications database APPS schema.
2. Run the following four SQL scripts from the samples directory **in the listed order**.
 - Ibm_websphere_event_table_create.sql
 - Ibm_websphere_events_s.sql
 - Ibm_customer_event_key_s.sql
 - Ibm_customer_event_pkg.pls

For example, SQL>@C:\ibm\ibm_webshere_event_table_create.sql;

Results

These scripts place stored procedures and other artifacts into the Oracle Applications (APPS) schema.

Creating the workflow

A workflow must be created in Oracle E-Business Suite to receive the customer data from the outbound Create operation. The workflow creates the customer record in the database and does error handling.

Procedure

1. Open the Oracle Workflow Builder and click the File > Open menu option.

2. Provide the filename of the workflow scripts. Browse to locate the IMPCUST.wft file from the samples directory.
3. Click OK to load the workflow scripts into the workflow builder.
4. Click the menu option File > Save as to provide the information to save the workflow in the database.
 - a. Click Database and provide the following information, which you can obtain from your database administrator:

Table 28.

| Field | Value |
|----------|--|
| User | APPS |
| Password | The password for the user of the APPS database |
| Connect | The database connect string |

- b. Click OK to save the workflow in the APPS schema.

Setup within Oracle E-Business Suite

This sample application uses the Oracle E-Business Suite Business Event System, a mechanism to process data based on events, subscriptions, and actions you define. Before you can use the Business Event System, you must complete some setup tasks within Oracle. These include creating custom events and creating subscriptions for those events.

Creating a custom event

The Oracle E-Business Suite built-in Business Event System (BES) requires events and subscriptions. When you create a custom event, you satisfy one of the requirements of the Business Event System mechanism.

About this task

This setup step performed within Oracle E-Business Suite enables the Business Event System to detect an event.

Procedure

1. Access your Oracle E-Business Suite application instance as the user SYSADMIN.
2. Use the responsibility for Web Applications Workflow Administrator.
3. Select the Business Events menu option, and navigate to the page where you can create a new business event.
4. Create a new business event, and type in or select all of the following information:

Table 29. Business event values

| Field | Value |
|--------------|---|
| Name | ibm.apps.ar.customer.inbound |
| Display name | IBM Customer Inbound Event |
| Description | IBM Customer Inbound Event |
| Status | Enabled |
| Owner Name | An owner name recognized by Oracle E-Business Suite |
| Owner Tag | An owner tag recognized by Oracle E-Business Suite |

5. Save the event definition.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Creating a subscription for the custom event

After you create a custom event, you must create a subscription for the Business Event System to use with the event. During outbound processing, the Business Event System raises the event, which identifies it to the Oracle event manager. The Oracle event manager initiates subscriptions related to the event. A subscription identifies what action to take, such as launching a workflow and populating a table with information from the event.

Procedure

1. Access the Subscriptions menu.
2. Navigate to the page where you can create a new subscription.
3. Create the subscription, typing in information from the following table:

Table 30. Subscription values

| Field | Value |
|--------------|----------------------------------|
| System | The name of your Oracle instance |
| Source Type | Local |
| Event Filter | ibm.apps.ar.customer.inbound |
| Phase | 10 |
| Status | Enabled |
| Rule Data | Message |
| Action Type | Launch Workflow |
| On Error | Stop and Rollback |

4. Navigate to the next section, where you can add information about the subscription by typing in the information presented in the following table.

Table 31. Additional subscription values

| Field | Value |
|------------------|--|
| Workflow Type | IMPCUST |
| Workflow Process | IMPORT_CUSTOMER |
| Priority | High |
| Owner Name | <i>An owner name recognized by Oracle E-Business Suite (the same name you used for the custom event)</i> |
| Owner Tag | <i>An owner tag recognized by Oracle E-Business Suite (the same tag you used for the custom event)</i> |

5. Save the subscription.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Creating an error event

The Oracle E-Business Suite built-in Business Event System (BES) requires events and subscriptions to process data. In this task, you create an error event to enable the workflow to handle errors occurring during the creation of the customer record in the database.

About this task

This setup step is performed within Oracle E-Business Suite to enable the Business Event System to detect an error event.

Procedure

1. Access your Oracle E-Business Suite application instance as the user SYSADMIN.
2. Use the responsibility for Web Applications Workflow Administrator.
3. Select the Business Events menu option, and navigate to the page where you can create a new business event.
4. Create a new business event, and type in or select all of the following information:

Table 32. Business event values

| Field | Value |
|--------------|--|
| Name | ibm.apps.ar.customer.error |
| Display name | IBM Error Event For Customer Import |
| Description | IBM Error Event For Customer Import |
| Status | Enabled |
| Owner Name | <i>An owner name recognized by Oracle E-Business Suite</i> |
| Owner Tag | <i>An owner tag recognized by Oracle E-Business Suite</i> |

5. Save the event definition.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Creating a subscription for the error event

After you create an error event, you must create a subscription for the Business Event System to use with the event. The subscription tells what action to take for the error event.

Procedure

1. Access the Subscriptions menu.
2. Navigate to the page where you can create a new subscription.
3. Create the subscription, typing information from the following table:

Table 33. Subscription values

| Field | Value |
|--------------|----------------------------------|
| System | The name of your Oracle instance |
| Source Type | Local |
| Event Filter | ibm.apps.ar.customer.error |
| Phase | 10 |

Table 33. Subscription values (continued)

| Field | Value |
|-------------|-------------------|
| Status | Enabled |
| Rule Data | Message |
| Action Type | Custom |
| On Error | Stop and Rollback |

- Navigate to the next section, where you can add information about the subscription, and type in the information presented in the following table.

Table 34. Additional subscription values

| Field | Value |
|----------------------|--|
| PL/SQL Rule Function | IBM_WEBSPPHERE_PKG.RULE_FUNCTION |
| Out Agent | WF_JMS_OUT Note: This is optional. Enter the value only if the event information needs to be populated into the JMS queue. |
| Priority | Normal |
| Owner Name | An owner name recognized by Oracle E-Business Suite (the same name you used for the error event) |
| Owner Tag | An owner tag recognized by Oracle E-Business Suite (the same tag you used for the error event) |

- Save the subscription.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

Creating the project

To begin the process of creating and deploying a module to communicate with an Oracle E-Business Suite application, you start the external service wizard in WebSphere Integration Developer. The wizard creates an Adapter for JDBC project, which is used to organize the files associated with the adapter.

Before you begin

Ensure that you have access to the documentation for the Adapter for JDBC. To view it, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

Procedure

- Review the information in “Configuring the module for deployment” in the *WebSphere Adapters: Adapter for JDBC User Guide*.
- Perform the task “Creating the project” in the *WebSphere Adapters: Adapter for JDBC User Guide*. Follow the steps in this section to set up the project in your workspace on WebSphere Integration Developer.

Configuring the module for outbound processing

You must create the business objects necessary to process data into Oracle E-Business Suite. To accomplish this, use the external service wizard in WebSphere

Integration Developer for WebSphere Adapter for JDBC. First, you initialize the wizard by configuring values of the properties that enable the adapter to set up a communication channel to a specific database. Then, run a query to discover business objects. Finally, you must specify the operations and other properties for the selected business objects.

Setting connection properties for the external service wizard

Set values of the connection properties for your database instance. These properties enable the external service wizard in the Adapter for JDBC to connect to the Oracle database for discovery and for creating the service description.

Before you begin

This task is performed within the external service wizard, which you already started for the tasks under “Creating the project.”

About this task

First, the adapter needs a copy of certain files from the database to be able to communicate with it. These JAR files include the JDBC driver and any native system library files that are needed. You can obtain the necessary files for your database software and operating system from your database administrator or from the database software Web site.

Procedure

1. Add external software dependencies. In the Required Files and Libraries window, next to the top pane for JDBC driver JAR files, click **Add**. Browse to the location of the external JDBC driver, select the driver JAR file, and click **Open**.
2. Click **Next**.

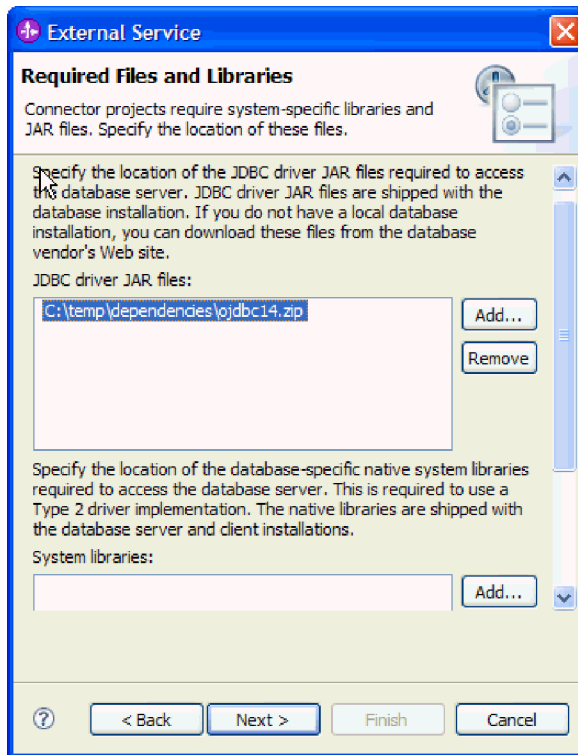


Figure 30. Add JDBC driver JAR files to project

3. In the Processing Direction window, click **Outbound** and click **Next**.
4. In the Discovery Configuration window, in the left pane, select your database vendor, driver, and version.
5. Enter the information for your database that is shown in the following table.
The table lists the properties required for the sample and their descriptions. For details on these properties, see "Connection properties for the external service wizard" in the "Reference" section of the *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html

Table 35. Connection properties for the external service wizard

| Property | Description |
|------------------------|--|
| Database | The database name. For Oracle databases this is the system ID (SID). Database is used along with host name and port number to generate the Database URL that is used to connect to the database. |
| Host name | The host name or IP address of the database server. Host name is used along with database name and port number to generate the Database URL that is used to connect to the database. |
| Port number | The port number, which, along with database name and host name, generates the Database URL that is used to connect to the database. |
| JDBC driver class name | The class name of the JDBC driver that is used to connect to the database. |

Table 35. Connection properties for the external service wizard (continued)

| Property | Description |
|-----------|---|
| User name | The database account you are using. It is used during the discovery process. Later in the wizard, you can specify a different name and password to use at run time. |
| Password | The password for the account you are using. |

Click Next.

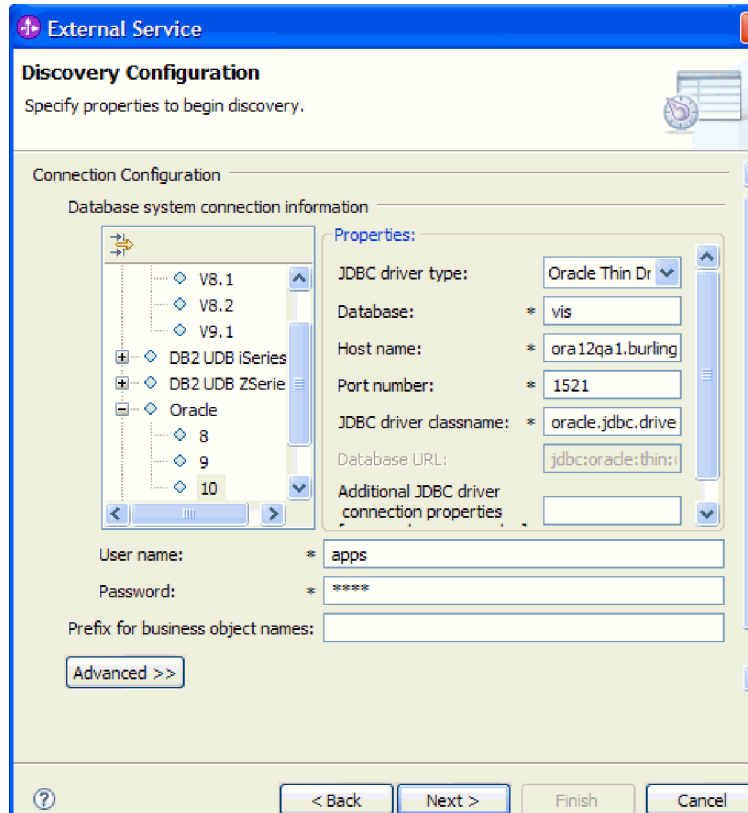


Figure 31. Add connection property values

Results

The wizard uses the connection properties to connect to the database.

Selecting business objects and services

After setting connection properties, run a query to search for database objects. Browse the metadata tree of discovered objects to understand the structure of the objects in the Oracle database and select objects that you want to include in the service description.

About this task

For this sample, you run a query on the accounts receivable (AR) schema, and select tables that represent data in the Oracle E-Business Suite interface tables.

Procedure

1. Specify filter properties
 - a. In the Object Discovery and Selection window, click **Edit Query**.
 - b. In the Query Properties window, type AR in the **Schema name pattern** field to display the accounts receivable (AR) schema.
 - c. Select the check box **Prompt for additional configuration settings when adding business object**, and click **OK**.

Now whenever you select one of the discovered objects for import, you are prompted to enter application-specific information for the object.

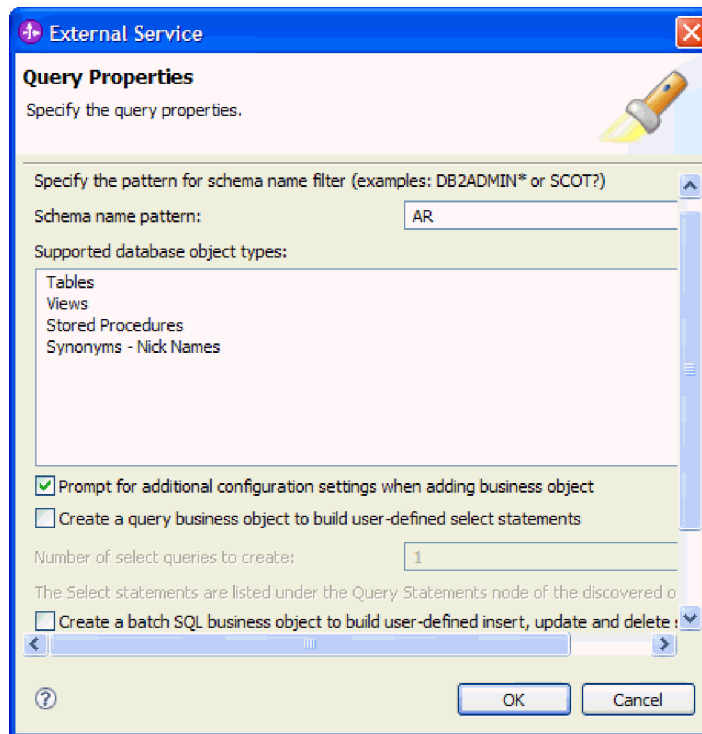


Figure 32. Specify query properties

2. Run the metadata query
 - a. Display objects discovered by the query
 - Click **Run Query**. The AR schema and data elements of the schema are displayed.
 - b. Select the object for import
 - Expand the **AR** schema. Expand **Tables**.
 - Highlight the table **RA_CUSTOMERS_INTERFACE_ALL**, and click > (the Add icon) to select this object to be imported.

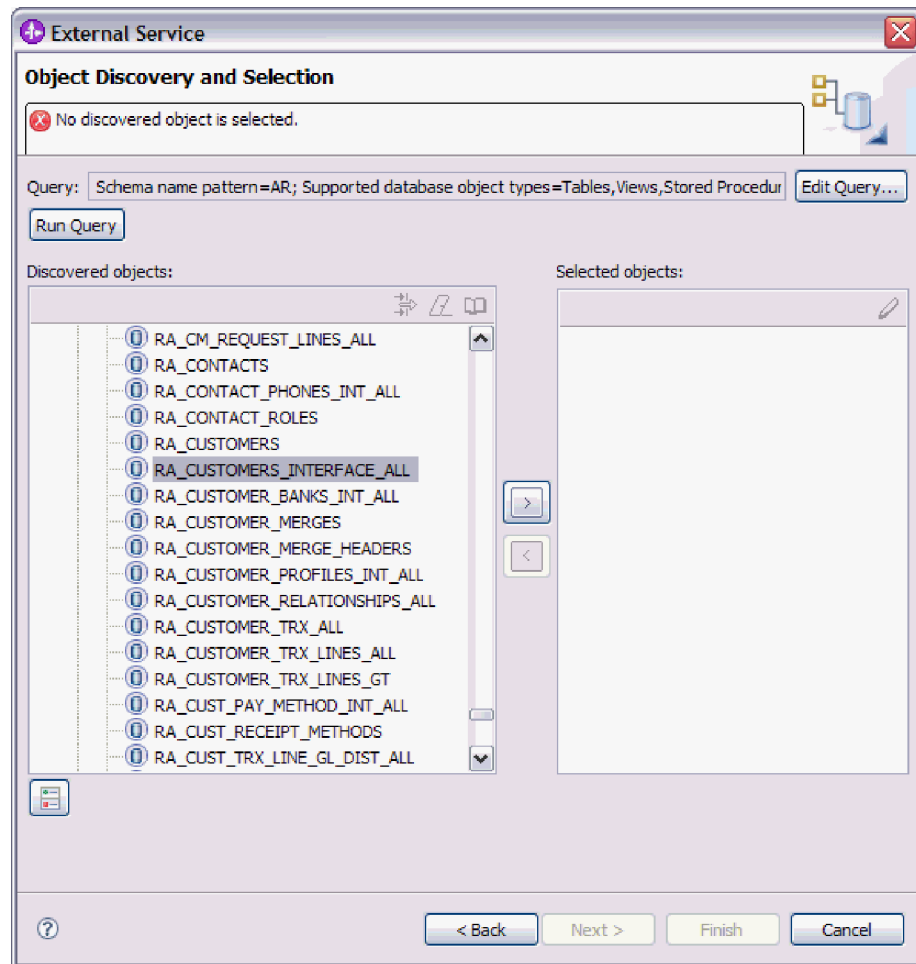


Figure 33. Select object for import

c. Add business object application-specific information

In the Configuration Properties for **RA_CUSTOMERS_INTERFACE_ALL** window, click **Add** to select the primary key for the table.

Select **REQUEST_ID** and click **OK**.

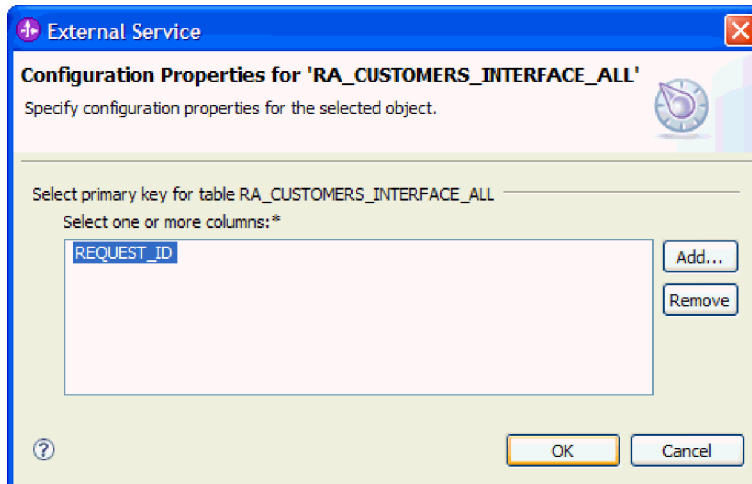


Figure 34. Add business object application-specific information

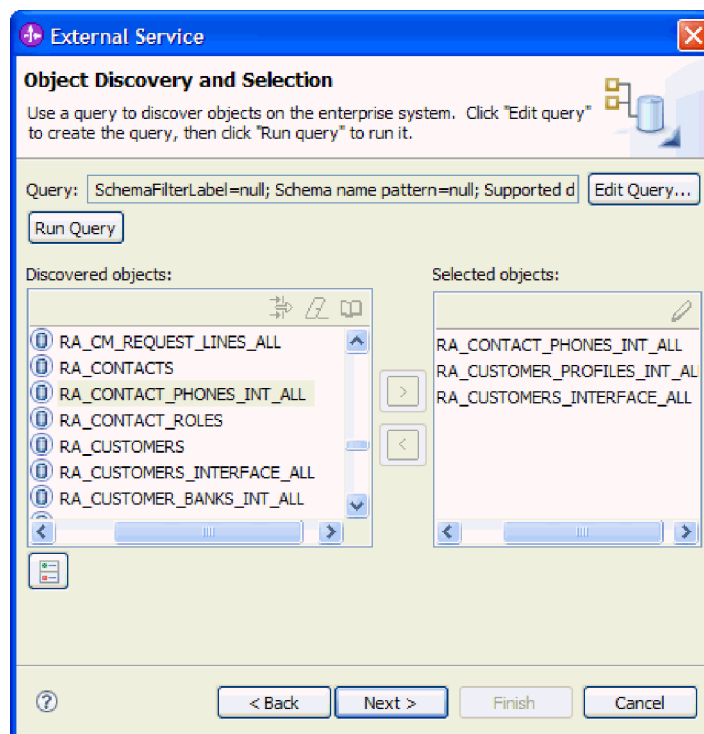
After a delay, the selected object is displayed in the Selected objects pane.

d. Select remaining tables

Repeat steps 2b and 2c for each of the remaining database tables listed in the following table. The primary key is specified for each table.

Table 36. Remaining tables to select for import and their primary key

| Table | Primary Key |
|---------------------------------|-------------|
| AR.RA_CUSTOMER_PROFILES_INT_ALL | REQUEST_ID |
| AR.RA_CONTACT_PHONES_INT_ALL | REQUEST_ID |



When you are finished, click **Next**.

For information on the object-level, operation, and attribute application-specific information, refer to “Business object information” in the “Reference” section of *IBM WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

Configuring the selected objects

After you have selected database objects and added their primary key, you can specify operations and other properties that apply to the selected business objects.

Procedure

1. Select operations

In the Configure Composite Properties window of the external service wizard, the Operations pane lists the operations that the adapter supports for the outbound service type. Remove the following operations by highlighting them and clicking **Remove**.

- **Update**
- **Delete**
- **RetrieveAll**
- **ApplyChanges**

The following operations remain:

- **Create**
- **Retrieve**

2. Leave the default values for these fields:

- **Maximum records for RetrieveAll operation**
- **Business object namespace**
- **Folder**
- **Generate a business graph for each business object**

3. Click Next.

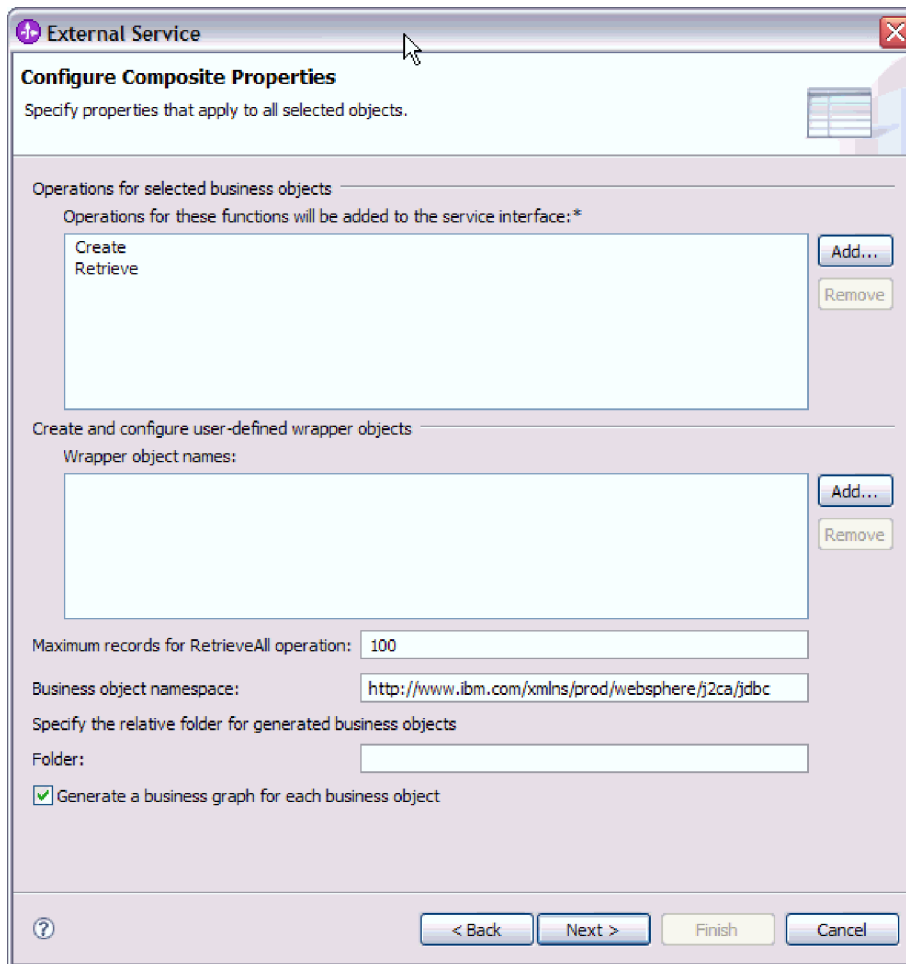


Figure 35. Select operations

What to do next

Next, you will specify deployment information to use at run time and information for saving the service as a module.

Setting deployment properties and generating the service

Use the external service wizard to configure the properties the adapter uses to set up a communication channel to a specific database for outbound processing. Then create a business integration module where all of the artifacts and property values can be saved.

Procedure

1. Set configuration property values
 - a. In the Service Generation and Deployment Configuration window, in the **Deploy connector project** field, ensure that this option is selected: **With module for use by single application**.
 - b. Click **Advanced** to see all of the properties under **Connection properties, Database system connection information**.
For outbound processing, the property fields are displayed for the managed connection factory and resource adapter properties.
 - c. Set the values of the required connection properties as shown in the following table.

The fields are initialized with the connection information you specified when you started the wizard. You can edit the existing values for Database vendor and User name, as required. You must type your Password.

Table 37. Database system connection information

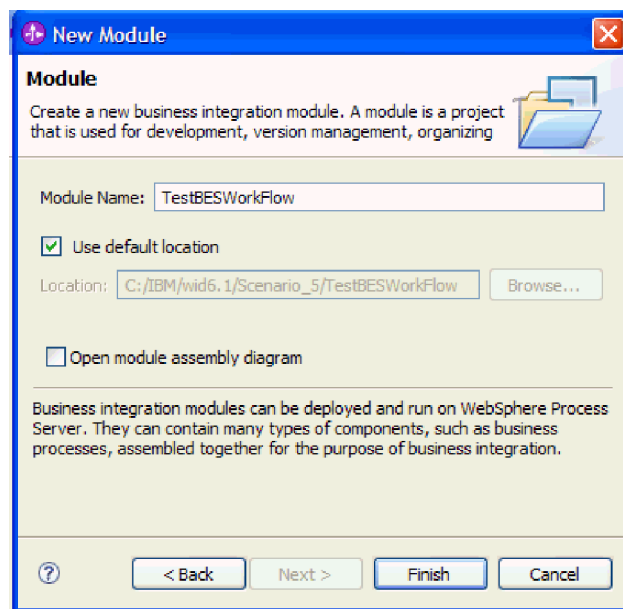
| Properties | Values |
|-----------------|--|
| User name | The database account you are using |
| Password | The password for the account you are using |
| Database vendor | Oracle |

d. Click **Next**.

For information about these properties, refer to the “Reference” section of *WebSphere Adapters: Adapter for JDBC User Guide*. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

2. Create a module

- a. In the Service Location Properties window, click **New** next to the **Module** field.
- b. In the Integration Project window, confirm that **Create a module project** is selected and click **Next**.
- c. In the Module window, type **TestBESWorkFlow**, and click **Finish**.



d. In the Service Location Properties window, click **Finish**.

Results

This module will hold all of the generated artifacts.

Saving the module

Save the business integration module for outbound processing that contains the artifacts and property values you created.

Procedure

1. In the Business Integration view, click the **TestBESWorkFlow** module.

2. Select **File > Save**.

Note: If the option to **Save** is inactive (grey), then the module has already been saved.

Deploying the module for testing

To test the adapter application, you need to install the module for outbound processing in the WebSphere Integration Developer integration test client.

Before you begin

The module that contains an Oracle database import file is produced by running the external service wizard.

Procedure

1. In WebSphere Integration Developer, click the **Servers** tab to open the Servers view.

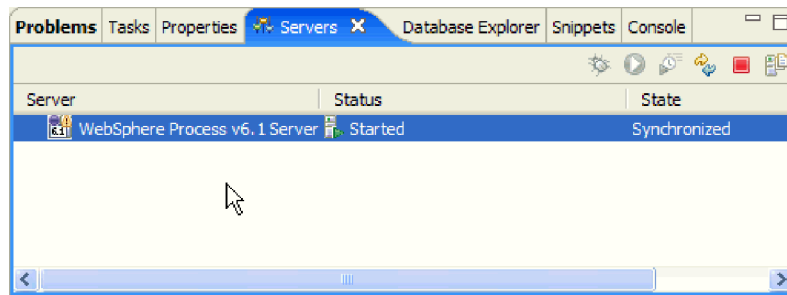


Figure 36. Servers view

2. If your server is not displayed in the Servers view, right-click the Servers view and select **New** → **Server**. Select your WebSphere Process Server or WebSphere Enterprise Service Bus.
Click **Next** and click **Finish**.
3. If your WebSphere Process Server or WebSphere Enterprise Service Bus is not running, right-click the name of it and select **Start**. The **Status** entry changes to **Started**.
4. Right-click the name of your WebSphere Process Server or WebSphere Enterprise Service Bus, and select **Add and remove projects**.
5. From the **Available Projects** list on the left, select the **TestBESWorkFlow** module and click **Add**. Click **Finish**.

What to do next

Next, you can test the assembled application in the integration test client.

Testing the Create operation

After you have deployed your outbound processing application to WebSphere Process Server or WebSphere Enterprise Service Bus, test the assembled application using the WebSphere Integration Developer integration test client.

About this task

To test the Create operation, you enter customer data into three database tables.

Procedure

1. Select the module to test
 In WebSphere Integration Developer, go to the Business Integration view of the Business Integration perspective. Right-click the **TestBESWorkFlow** module, and select **Test > Test Module**.
2. Leave the default value for **verb**.
 The value may be different from the one shown in the “Set attribute values” figure below.
3. Set the test values
 In the TestBESWorkFlow_Test window, use the arrows to set the test values shown in the following table. You set the test values three times, once for each Create operation listed.

Table 38. Test values for the Create operation

| Item | Test value |
|--|---|
| Configuration | Default Module Test |
| Module | TestBESWorkFlow |
| Component | JDBCOutboundInterface |
| Interface | JDBCOutboundInterface |
| Operation (select one operation at a time) | createArRa_Customers_Interface_AllBG createArRa_Customer_Profiles_Int_AllBG createArRa_Contact_Phones_Int_AllBG |

Note: The test values for **Component** and **Interface** may include a number, for example, JDBCOutboundInterface1. WebSphere Integration Developer appends numbers so that each name in the workspace is unique.

4. Set attribute values for the input business object
 Set the attribute values for the three tables you selected during object discovery with the external service wizard. The tables are
 - ArRa_Customers_Interface_AllBG.
 - ArRa_Customer_Profiles_Int_AllBG
 - ArRa_Contact_Phones_Int_AllBG

The following tables provide the values for each object and indicate any restrictions, such as whether the value has to be unique.

- a. Create and record unique ID values.
 Oracle requires ID values for outbound processing. These values uniquely identify the records inserted into Oracle E-Business Suite and are required for successful retrieval of the corresponding inserted data. The ID values are used in multiple objects, so you need to write down the values you will use to be able to enter them consistently for the different objects. For each of the following values, create unique string values.
 - CustomerReference#
 - AddressReference#1
 - TelephoneReference#
 - CustomerName

Note: For more information on required values and other allowed values for data in the Oracle interface tables, refer to your documentation for Oracle Applications, specifically the Oracle Receivables User Guide.

- b. In the Events window, under **Initial request parameters**, click the **Value** column across from the name of each attribute whose value you need to set, and enter the value from the following tables.

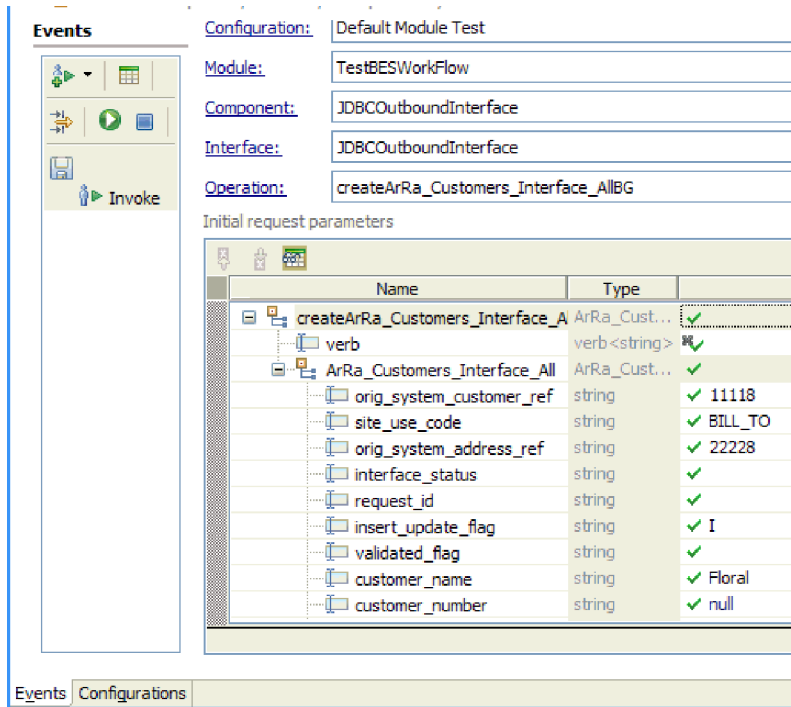


Figure 37. Set attribute values

Important:

To set a value for an attribute to **<null>**, right-click the value field for the attribute. Select **Set to** and scroll down to select **<null>**.

The following table lists the values for the ArRa_Customers_Interface_All table attributes.

Table 39. Values for the ArRa_Customers_Interface_All table attributes

| Attribute | Value | Notes |
|--------------------------|--------------------|--|
| Orig_system_customer_ref | CustomerReference# | Use the value that you recorded. The value must be unique. |
| Site_use_code | BILL_TO | |
| Orig_system_address_ref | AddressReference#1 | Use the value that you recorded. The value must be unique. |
| Insert_update_flag | I | |
| Customer_name | CustomerName | Use the value that you recorded. The value must be unique. |
| Customer_number | <null> | |

Table 39. Values for the ArRa_Customers_Interface_All table attributes (continued)

| Attribute | Value | Notes |
|------------------------|---------------------|---|
| Customer_status | A | |
| Primary_site_use_flag | Y | |
| Location | <null> | |
| Address1 | Test Address | You have the option of using your own address information. |
| Address2 | <null> | |
| Address3 | <null> | |
| Address4 | <null> | |
| City | San Mateo | |
| State | CA | |
| Province | <null> | |
| County | San Mateo | |
| Postal_code | 94401 | |
| country | US | |
| Cust_category_code | CUSTOMER | |
| Last_updated_by | -1 | |
| Last_update_date | 2007-01-08 12:00:00 | Today's date and time. Does not have to be exact. Use Date format yyyy-mm-dd hh:mm:ss |
| Created_by | -1 | |
| Creation_date | 2007-01-08 12:00:00 | Today's date and time. Does not have to be exact. Use Date format yyyy-mm-dd hh:mm:ss |
| Org_id | 204 | |
| Customer_name_phonetic | CustomerName | Use the value that you recorded. |

The following table lists the values for the ArRa_Customer_Profiles_Int_All table attributes.

Table 40. Values for the ArRa_Customer_Profiles_Int_All table attributes.

| Attribute | Value | Notes |
|-----------------------------|--------------------|--|
| Orig_system_customer_ref | CustomerReference# | Use the value that you recorded. The value must be unique. |
| Insert_update_flag | I | |
| Customer_profile_class_name | DEFAULT | |
| Credit_hold | N | |
| Last_updated_by | -1 | |

Table 40. Values for the ArRa_Customer_Profiles_Int_All table attributes. (continued)

| Attribute | Value | Notes |
|------------------|---------------------|---|
| Last_update_date | 2007-01-08 12:00:00 | Today's date and time. Does not have to be exact. Use Date format yyyy-mm-dd hh:mm:ss |
| Created_by | -1 | |
| Creation_date | 2007-01-08 12:00:00 | Today's date and time. Does not have to be exact. Use Date format yyyy-mm-dd hh:mm:ss |
| Org_id | 204 | |

The following table lists the values for the ArRa_Contact_Phones_Int_All table attributes.

Table 41. Values for the ArRa_Contact_Phones_Int_All table attributes

| Attribute | Value | Notes |
|---------------------------|----------------------------|--|
| Orig_system_contact_ref | <null> | |
| Orig_system_telephone_ref | <i>TelephoneReference#</i> | Use the value that you recorded. The value must be unique. |
| Orig_system_customer_ref | <i>CustomerReference#</i> | Use the value that you recorded. The value must be unique. |
| Orig_system_address_ref | <i>AddressReference#1</i> | Use the value that you recorded. The value must be unique. |
| Insert_update_flag | I | |
| Contact_first_name | <null> | |
| Contact_last_name | <null> | |
| Contact_title | <null> | Contact_title should exist in ar_lookups, where lookup_type=Contact_title |
| Contact_job_title | <null> | |
| Telephone | 4555555 | You have the option of using your own phone number. |
| Telephone_extension | 89555 | Use <null> if you don't have one. |
| Telephone_type | FAX | Telephone_type should exist in ar_lookups, where lookup_type=Phone_line_type |
| Telephone_area_code | 407 | |
| Last_update_date | 2007-01-08 12:00:00 | Use the system date and time. Use Date format yyyy-mm-dd hh:mm:ss |
| Last_updated_by | -1 | |

Table 41. Values for the ArRa_Contact_Phones_Int_All table attributes (continued)

| Attribute | Value | Notes |
|---------------|---------------------|---|
| Creation_date | 2007-01-08 12:00:00 | Use the system date and time. Use Date format yyyy-mm-dd hh:mm:ss |
| Created_by | -1 | |
| Email_address | <null> | |
| Org_id | 204 | |

5. In the left pane below **Events**, click **Continue**. From the Deployment Location window, select the name of the server where your application will be deployed, and click **Finish**. The Starting the Integration Test Client window is displayed briefly.
6. Start the SQL*Plus tool, or a similar program for processing SQL commands. Run the raise_inbound_event.sql script.
For example, SQL>@c:\ibm\raise_inbound_event.sql;
The script accepts the value: Customer Reference.
7. Verify the results of the workflow to import the customer data.
 - a. Log on to Oracle E-Business Suite as the System Administrator user and select the Workflow Administrator Web Applications responsibility.
 - b. Select the Administrator Workflow >Status Monitor menu option.
 - c. For the Type Internal Name field, type in IMPCUST and click Go.
 - d. At the screen showing the results of workflows, select the workflow process that you just ran and select the Status Diagram view.
 - e. A Java applet window displays the workflow diagram with the completion path. When the customer object is imported successfully, review the customer data in Oracle E-Business Suite by switching to the Receivables Manager responsibility and selecting the Customer > Standard menu option.
 - f. If the interface program ends with an error or cannot import the customer object because of data errors, the workflow process raises an error event and shows the diagram for an import error event. When the customer object fails in the interface, run the following SQL statement to review the error event record in the event table:

```
select * from ibm_websphere_events where object_key = <customer reference> and object_function = 'Error';
```

What to do next

Next, you return the data to its original state by clearing the sample content.

Clearing the sample content

After you have tested the workflow to import the customer object, clear the sample content to return the data to its original state. To do this, you must deactivate the customer in Oracle E-Business Suite, and then run SQL commands to delete the stored procedure.

Procedure

1. Use the Oracle E-Business Suite Client to log into the Receivables Manager responsibility. Navigate to the Customers > Standard menu and find the customer. Change the Status value for the customer to "Inactive." Then save the customer.

Note: For specific information about using the Oracle interface, refer to your documentation for Oracle Applications.

2. Use the SQL*Plus tool, or a similar program for processing SQL commands, to log into the database and run commands to clear artifacts from the database. Consult your database administrator if you need help performing this task.

Troubleshooting information for the sample applications

If you encounter problems while running the tasks for these samples, they could be related to either running the Adapter for JDBC or to interacting with the Oracle database. Two known issues are described.

Objects take a long time to generate

The time it takes to generate objects varies according to the computing power of the hardware system on which you are running the Oracle E-Business Suite.

During the Create operation, objects are not moved to base tables

During a Create operation, after the adapter sets the values in the interface tables, an Oracle function call is made using a stored procedure. The function call moves the data from the Oracle interface tables to the base tables. If a problem occurs when this function call is made, Oracle writes error codes into the column in the interface tables called `INTERFACE_STATUS`.

If your customer object fails to appear in the application after the Create operation is run, but no errors are reported by the Adapter for JDBC, then you should query the `INTERFACE_STATUS` column in each interface table to see if any errors have been reported. These errors could be due to required values not being set, or use of a reference number or customer name that is already in the Oracle application's base tables.

Note: For more information on the error codes for this column, refer to your documentation for Oracle Applications.

In a production environment, these errors can be detected in a number of ways. You can run a separate Adapter for JDBC instance to look for and report errors from these columns. Or you can create an Oracle workflow to check for errors and report them using the Oracle E-Business Suite reporting system.

For problems while running the Adapter for JDBC, refer to the section "Troubleshooting and support" in *WebSphere Adapters: Adapter for JDBC User Guide*. It describes how to configure logging and tracing, detect and resolve errors and faults, and use IBM Software Support resources. To view that documentation, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wsadapters.610.doc/doc/stbp_jdb_welcome.html.

Viewing the sample adapter artifacts

To view the sample artifacts for each sample, import into IBM WebSphere Integration Developer the reference files included with the adapter. Note that these artifacts are for reference only. They probably will not run in your Oracle environment. If you have not stepped through the samples, you can still use the reference files to view examples of correctly generated artifacts before you create your own.

Before you begin

Locate the reference files in the `referencefiles` subdirectory of the `samples` directory. There is a project interchange zip file for each sample. For instance, `Tutorial1.zip` is for sample 1.

Important: Do not modify or use the artifacts provided in the reference files. They are provided exclusively for viewing.

Reference files do not include third-party libraries. When imported into IBM WebSphere Integration Developer, the reference files might generate compilation error messages because dependent libraries are missing.

The artifacts in the reference files may not be compatible with the Oracle E-Business Suite you are using, depending on its version and configuration. The artifacts were generated with version 12 of Oracle E-Business Suite.

About this task

Import the reference files into WebSphere Integration Developer to view sample artifacts associated with each sample.

Procedure

1. In the Business Integration perspective of WebSphere Integration Developer, click **File** → **Import**.
2. In the Import window, select **Project Interchange** and click **Next**.
3. Select the project interchange file containing the tutorial artifacts you want to view.
4. Import all the projects in the project interchange file by clicking **Select All**.
5. Click **Finish**.

Results

A business integration module is created with the following artifacts:

- Service import and export definitions
- Business objects (service data objects)
- Interfaces.

Chapter 4. Reference information

Use the reference information for WebSphere Adapter for Oracle E-Business Suite to find related product information.

Related information

The following information centers, IBM Redbooks®, and Web pages contain related information for the WebSphere Adapter for Oracle E-Business Suite.

Samples and tutorials

The WebSphere Integration Developer online samples/tutorials gallery includes samples and tutorials to help you use WebSphere Adapters. You can access the online samples/tutorials gallery as follows:

- From the welcome page that opens when you start WebSphere Integration Developer. To see samples and tutorials for WebSphere Adapter for Oracle E-Business Suite, click **Retrieve**. Then browse the displayed categories to make your selections.
- At this location on the Web: <http://publib.boulder.ibm.com/bpcsamp/index.html>.

Information resources

- The WebSphere Business Process Management information resources Web page includes links to articles, Redbooks, documentation, and educational offerings to help you learn about WebSphere Adapters: <http://www14.software.ibm.com/webapp/wsbroker/redirect?version=pix&product=wps-dist&topic=bpmroadmaps>
- The WebSphere Adapters library page includes links to all versions of the documentation: <http://www.ibm.com/software/integration/wbiadapters/library/infocenter/>

Information about related products

- WebSphere Business Process Management, version 6.1.0, information center, which includes WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Integration Developer information: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp>
- WebSphere Adapters, version 6.0.2, information center: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsadapters602.doc/welcome_top_wsa602.html
- WebSphere Adapters, version 6.0, information center: http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/topic/com.ibm.wsadapters.doc/welcome_wsa.html
- WebSphere Business Integration Adapters information center: http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/index.jsp?topic=/com.ibm.wbi_adapters.doc/welcome_adapters.htm

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Support and assistance

- WebSphere Adapters technical support: <http://www.ibm.com/software/integration/wbiadapters/support/>
- WebSphere Adapters technotes: <http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>. In the **Product category** list, select the name of the adapter and click **Go**.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department 2Z4A/SOM1
294 Route 100
Somers, NY 10589-0100
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of

this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning:

Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

IBM, the IBM logo, developerWorks, Redbooks, ViaVoice, and WebSphere are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are registered trademark of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (<http://www.eclipse.org>).

Index

A

- accessibility
 - administrative console 8
 - external service wizard 8
 - IBM Accessibility Center 9
 - keyboard 8
 - shortcut keys 8
- actions 2
- Adapter for Oracle E-Business Suite
 - accessibility 8
 - standards compliance 8
- adapter technotes 94
- additional samples 18
- artifacts, sample 91

B

- base tables 3, 5
- before you begin 11
- Business Event System (BES) 2
- business object information 22

C

- clustered environment
 - deploying in 14
 - description 14
 - inbound processes 14
 - outbound processes 15
- compatibility matrix 1
- concurrent program 7
- Create operation 5
- creating the project 19, 45, 60, 74

D

- database user account 17
- deployment
 - options 11
- developerWorks 93
- developerWorks resources, WebSphere Adapters 93
- documentation, Adapter for JDBC 19, 74

E

- education, WebSphere Adapters 93
- embedded adapter
 - considerations for using 13
 - description 11
- enableHASupport property 14
- errors, detecting 90
- event 2
- external service wizard
 - accessibility 8

F

- features, new 1

- files
 - project interchange 91
 - quick start reference 91
- files for the samples 17

H

- hardware and software requirements 1
- hardware requirements 1
- high-availability environment
 - deploying in 14
 - description 14
 - inbound processes 14
 - outbound processes 15

I

- IBM WebSphere Adapter Toolkit 93
- inbound processing sample 3
- interface tables 3, 5
- Internet Protocol Version 6.0 (IPv6) 9
- IPv6 9

K

- keyboard 8

M

- matrix, compatibility 1

N

- new features 1
- non-record parameters 4

O

- objects, long time to generate 90
- outbound processing samples 3

P

- parameters
 - non-record 4
 - record 4
- prerequisites for using the adapter 11
- problem during Create operation 90
- problems, known 90
- project interchange files 91
- project, creating 19, 45, 60, 74

Q

- quick start reference files 91

R

- raise an event 2
- record parameters 4
- Redbooks, WebSphere Adapters 93
- reference files 91
- related information 93
- related products, information 93
- requirements, hardware and software 1

S

- Sample 1 3, 5, 18
- Sample 2 3, 7
- Sample 3 4, 5
- Sample 4 4, 6, 69
- sample artifacts 91
- sample files 17
- shortcut keys 8
- software requirements 1
- stand-alone adapter
 - considerations for using 13
 - description 11
- standards compliance 8
- subscriptions 2
- support
 - technical 94
- synonyms 41
 - creating 19
 - for quicker searches 19

T

- technical support 94
- technotes 1, 94
- technotes, WebSphere Adapters 93
- triggers 8
- troubleshooting information 90

U

- user account
 - changing responsibilities 17
 - rights 17
- user guide, Adapter for JDBC 19, 74

W

- WebSphere Adapters, version 6.0, information 93
- WebSphere Adapters, version 6.0.2, information 93
- WebSphere Application Server information 93
- WebSphere Business Integration Adapters information 93
- WebSphere Business Process Management, version 6.1.0,
information 93
- WebSphere Enterprise Service Bus
information 93
- WebSphere Extended Deployment 14
- WebSphere Integration Developer
information 93
- WebSphere Process Server
information 93
- workflow 6



Printed in USA