

IBM WebSphere Adapters
Version 7 Release 5 Fix Pack 1 (7.5.0.1)

*IBM WebSphere Adapter for Oracle
E-Business Suite User Guide
Version 7 Release 5 Fix Pack 1
(7.5.0.1)*



IBM WebSphere Adapters
Version 7 Release 5 Fix Pack 1 (7.5.0.1)

*IBM WebSphere Adapter for Oracle
E-Business Suite User Guide
Version 7 Release 5 Fix Pack 1
(7.5.0.1)*



Note

Before using this information and the product it supports, read the information in "Notices" on page 283.

November 2011

This edition applies to version 7, Release 5, Fix Pack 1 (7.5.0.1) 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 IBM Corporation 2006, 2011.

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

Contents

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

Hardware and software requirements	1
Technical overview of IBM WebSphere Adapter for Oracle E-Business Suite	2
Outbound processing	5
Inbound processing.	21
Business objects	25
Tables, views, and synonyms overview	38
Stored procedure overview	38
Stored procedure business object overview	38
Stored procedures used in place of or in addition to operations	41
Stored functions overview	46
Query business object overview	46
Business faults	47
XML gateway interface overview	48
Advanced Queue integration overview	49
Support for metadata discovery for the supported interfaces	50
The external service wizard	51

Chapter 2. Planning for adapter implementation 53

Before you begin	53
Security	53
Support for protecting sensitive user data in log and trace files	53
User authentication.	54
Deployment options	56
WebSphere Adapters in clustered environments	60
Support for prepared statement caching	61
Migrating to version 7.5 of WebSphere Adapter for Oracle E-Business Suite	62
Migration considerations	62
Performing the migration.	64
Upgrading but not migrating a project	66
Migrating WebSphere Business Integration applications	67
Migrating applications from WebSphere InterChange Server.	67
Migration considerations for WebSphere Business Integration adapters	68
Migrating application artifacts from WebSphere InterChange Server.	69
Migrating adapter-specific artifacts	70
Changes to the import, export, and WSDL files after migration	72

Chapter 3. Samples and tutorials . . . 75

Chapter 4. Configuring the module for deployment. 77

Road map for configuring the module	77
---	----

Creating the event store	79
Creating an authentication alias	80
Creating the project.	81
Adding external software dependencies	82
Setting connection properties for the external service wizard	83
Configuring the module for XML Gateway interface for web service	85
Configuring the module for XML Gateway interface for JMS.	90
Configuring the module for advanced queue integration.	95
Configuring the module for outbound processing	100
Discovering database objects	101
Selecting and configuring business objects.	105
Setting global properties for operations.	122
Setting deployment properties and generating the service	124
Completing the configuration	131
Configuring the module for inbound processing	133
Discovering database objects	133
Selecting and configuring business objects.	136
Setting global properties for operations.	145
Setting deployment properties and generating the service	146
Completing the configuration	156

Chapter 5. Modifying artifacts 159

Modifying service import	159
Modifying service export	161

Chapter 6. Changing interaction specification properties 163

Chapter 7. Deploying the module . . . 165

Deployment environments	165
Deploying the module for testing.	165
Adding external dependencies.	165
Generating and wiring a target component for testing inbound processing	167
Preparing to test outbound operations	168
Adding the module to the server.	168
Testing the module for outbound processing using the test client	169
Deploying the module for production	169
Adding external software dependencies on the server	170
Installing the RAR file (for modules using stand-alone adapters only)	170
Exporting the module as an EAR file	172
Installing the EAR file	172

Chapter 8. Administering the adapter module 175

Changing configuration properties for embedded adapters	175
Setting resource adapter properties for embedded adapters	175
Setting managed (J2C) connection factory properties for embedded adapters	177
Setting activation specification properties for embedded adapters	179
Changing configuration properties for stand-alone adapters	181
Setting resource adapter properties for stand-alone adapters	181
Setting managed (J2C) connection factory properties for stand-alone adapters	182
Setting activation specification properties for stand-alone adapters	183
Starting the application that uses the adapter.	185
Stopping the application that uses the adapter	185
Monitoring performance using Performance Monitoring Infrastructure	186
Configuring Performance Monitoring Infrastructure	186
Viewing performance statistics	188
Enabling tracing with the Common Event Infrastructure	189
Adding dependency libraries to the deployed resource adapter	190
Stand-alone deployment.	190
EAR deployment	191

Chapter 9. Troubleshooting and support 193

Log and Trace Analyzer	193
Configuring logging and tracing	193
Configuring logging properties	193
Changing the log and trace file names	195

First-failure data capture (FFDC) support	196
Tracing the XML Gateway Web Service status	197
Resolving namespace conflict in Web Service/JMS interface	197
Resolving connection information lost during EMD	197
Disabling end point applications of the passive adapter	197
Self-help resources.	198
Solutions to common problems	199

Chapter 10. Reference 207

Business object information.	207
Business object attributes	207
Attribute application-specific information	209
Business object-level application-specific information	217
Naming conventions	220
Fault business objects.	221
Configuration properties	223
Outbound configuration properties	223
Inbound configuration properties.	250
Globalization	276
Globalization and bidirectional transformation	276
Properties enabled for bidirectional data transformation	279
Adapter messages.	280
Related information	281
Video samples	281

Notices 283

Programming interface information	285
Trademarks and service marks	285

Index 287

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

With IBM® WebSphere® Adapter for Oracle E-Business Suite, you can create service-oriented integrated applications, which can interact and exchange information with Oracle E-Business Suite. By using the adapter, you can send requests as well as receive events from the underlying Oracle database, without special coding.

WebSphere Adapter for Oracle E-Business Suite enables a two-way communication between the application running on IBM Business Process Manager or WebSphere Enterprise Service Bus and the underlying Oracle database of Oracle E-Business Suite. Using the adapter, an application can send requests to read, create, modify, or delete data in the Oracle database, in many cases without writing any SQL code. To process requests received from an application, the adapter updates the Oracle database tables using SQL queries or stored procedures. An application can also receive events from the Oracle E-Business Suite. For example, it can be notified that specific objects in Oracle E-Business Suite are updated. To process events that result from changes to the Oracle E-Business Suite, the adapter delivers events to the application. Using the Oracle Business Event System and event notification, changes to the Oracle E-Business Suite can be automatically notified to other applications. By combining event processing by WebSphere Adapter for Oracle E-Business Suite and another adapter, updates can be automatically propagated to other enterprise applications such as Siebel, PeopleSoft, and SAP.

WebSphere Adapter for Oracle E-Business Suite provides a standard interface that integrates with the underlying Oracle database of the Oracle E-Business Suite applications; it supports the Oracle database server by using the Oracle Java Database Connectivity (JDBC) drivers. The Java Runtime Environment (JRE) version required by the Oracle JDBC driver must not be higher than the JRE version in the run time environment. The adapter uses business objects to exchange data between the application and the Oracle database, so the application does not need to use the JDBC application programming interface (API). *Business objects* are containers for application data that represent business functions or elements, such as an Oracle table or the result of an SQL query. The adapter understands the data format provided by the application, and can process the data, perform the operation, and send the results back in that format.

Hardware and software requirements

The hardware and software requirements for WebSphere Adapters are provided on the IBM Support website.

To view hardware and software requirements for WebSphere Adapters, see <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: http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family.

- Technotes for WebSphere Adapters provide workaround and additional information that are 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 IBM WebSphere Adapter for Oracle E-Business Suite

The adapter supports integration of databases that are accessible through the JDBC application programming interface (API) with applications running on IBM Business Process Manager or WebSphere Enterprise Service Bus. The adapter provides outbound and inbound processing under the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) and integrates with Service Component Architecture (SCA) components.

Outbound processing enables an application to access or modify data in a database. The adapter converts a request from the application to an outbound operation, which it runs to create, retrieve, update, or delete data in the database or to run a database program stored in the database. Processing these requests results in the creation, retrieval, update, or deletion of rows in the corresponding database tables. The adapter also enables you to run stored procedures or stored functions that are defined in the database, and to run user-defined SELECT, INSERT, UPDATE, and DELETE statements. You can use the adapter to integrate multiple applications with the same database.

Figure 1 on page 3 presents an overview of the flow of outbound processing.

An application running on IBM Business Process Manager or WebSphere Enterprise Service Bus invokes a service in an outbound module, which sends a request to the adapter to process one or more business objects. The adapter uses the JDBC API to connect to the database server, which accesses the tables and other objects in the database.

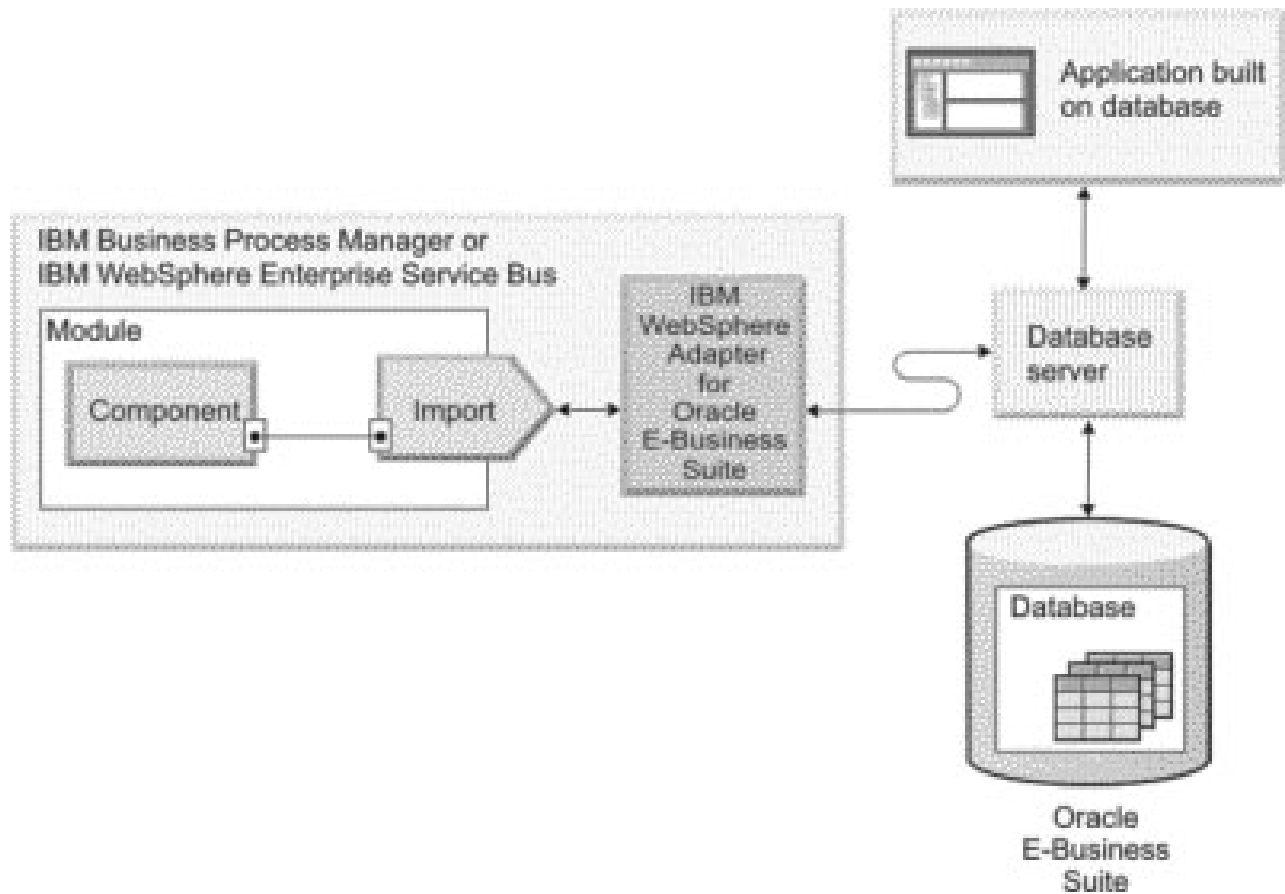


Figure 1. Processing outbound requests

Inbound processing enables an application to receive notification when objects in the database are changed. For example, an application can be notified when rows are created, updated, or deleted in selected database tables.

Figure 2 on page 4 presents an overview of the flow of inbound processing.

A database application changes tables in the database. The change causes a trigger, or another automated mechanism, to update the event store with information about the change. Periodically, the adapter polls the event store, retrieves, and processes events, and then delivers them to the export of a module that is part of an application that runs in IBM Business Process Manager or WebSphere Enterprise Service Bus.

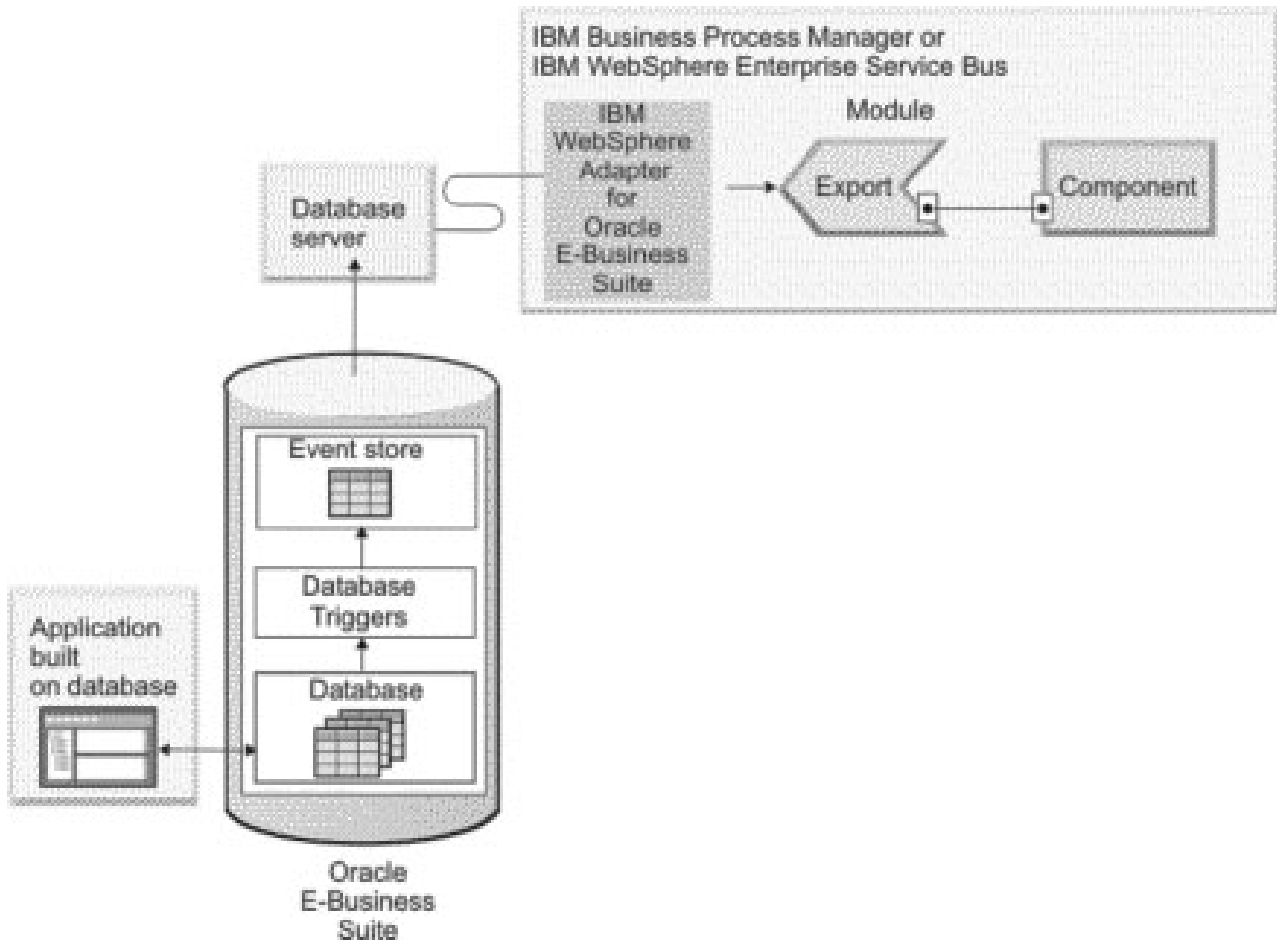


Figure 2. Processing inbound events

The adapter can process events in one of the following ways:

- Standard event processing, using an event store that is populated by the database application
- Custom event processing, using a user-defined database query

During *standard event processing*, when data is changed in the tables in the database, appropriate events are inserted into a database table called an event store, along with relevant information, such as key values. To capture the changed data in the database, you can place triggers on the respective tables, or use other methods such as Oracle Change Data Capture, which is provided for Oracle databases. To capture the changed data in Oracle E-Business Suite, configure the Oracle Business Event system and Concurrent Program. The adapter polls the event store and retrieves a batch of events. The events can be filtered by business object type and time stamp and connector ID. The adapter uses each event to construct a business graph or business object that contains the business objects changed by that event. The business object or business graph is then dispatched to the exports that are configured to receive the specific business object.

During *custom event processing*, the adapter runs a query that was specified by the user as a standard SQL statement, a stored procedure, or a stored function. Any of these actions returns a result set for data returned by the query. Each row of the result set corresponds to a row in the event store. The adapter constructs a

business object for each event and delivers it to the exports (also called endpoints) that are configured for (or have subscribed to) the specific business object.

For both standard and custom event processing, you can specify how often the adapter polls for events and how many events it retrieves each polling period.

Outbound processing

When an application component needs to query the existence of a record in the database or retrieve or modify data in the underlying Oracle database, the adapter acts as the connector between the application component and the database. The adapter provides a set of standard outbound operations, which process either the after-image or delta style business objects. The adapter also supports both local and XA (distributed) transactions for outbound processing.

The adapter business object model uses two styles of business objects for making updates: after-image and delta. An *after-image* business object is one that contains the complete state of the business object after the required changes have been made to it. A *delta* business object is one that contains only key values and the data to be changed. Delta business objects are used only in operations that update business objects.

Supported operations

Table 1 lists the outbound operations that are supported for each type of business object and indicates whether each supports after-image or delta style processing.

Table 1. Outbound operations supported by type of business objects

Business objects supported	Operation	After-image support	Delta support
Tables Views Synonyms - Nick Names	Create	Yes	Not applicable
	Update	Yes	Not applicable
	Delete	Yes	Not applicable
	Retrieve	Not applicable	Not applicable
	RetrieveAll	Not applicable	Not applicable
	Exists	Not applicable	Not applicable
	ApplyChanges	Yes	Yes
Stored procedures	Execute	Not applicable	Not applicable
Queries	RetrieveAll	Not applicable	Not applicable
XML Gateway	Invoke	Not applicable	Not applicable
	Publish	Not applicable	Not applicable
Advanced Queue	Enqueue	Not applicable	Not applicable

Transaction management

The adapter supports both local and XA (distributed) transactions for outbound processing. In the adapter, a transaction is an isolated interaction with the database. A transaction can consist of multiple operations on the database that are performed as an atomic unit. These operations are not affected by simultaneously occurring operations from other users of the database.

The adapter supports transactions only if the database server supports transactions. The types of transactions that are supported are local and XA transactions:

- A *local transaction* is one in which a component defines the start and end of the transaction with a single database. It uses a one-phase commit protocol. The transaction is managed and performed by the database.
- An *XA transaction* is one in which the transaction can span multiple heterogeneous databases. It uses a global or two-phase commit protocol. The transaction manager coordinates the transaction.

XA transactions

The adapter supports XA transactions for outbound processing. Choose one of these methods to configure the adapter for XA transactions:

- Specify a JNDI data source that supports XA transactions, using the `XADataSourceJNDIName` and `PoolDataSourceJNDIName` properties.
- Specify an XA data source, using the `XADataSourceName` property.

The `XADataSourceJNDIName` property represents a data source created within IBM Business Process Manager or WebSphere Enterprise Service Bus. If you define a JNDI data source that supports XA transactions on the server, and specify that data source when you configure the adapter, the adapter participates in XA transactions. Optionally, if you use an XA data source, the adapter participates in XA transactions.

Related reference

“Managed connection factory properties” on page 230

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Oracle E-Business Suite.

Outbound operations

Application components use operations to perform actions such as retrieving from a database. The adapter provides certain outbound operations. Details are provided on how the adapter processes business objects for each of the supported operations.

An operation can be performed using a standard SQL statement provided by the adapter or by a stored procedure that you define. You can run a stored procedure to perform the operation or to do custom processing before or after the operation. In each business object, you can configure how each operation is performed.

Related concepts

“Outbound processing” on page 5

When an application component needs to query the existence of a record in the database or retrieve or modify data in the underlying Oracle database, the adapter acts as the connector between the application component and the database. The adapter provides a set of standard outbound operations, which process either the after-image or delta style business objects. The adapter also supports both local and XA (distributed) transactions for outbound processing.

“Stored procedure overview” on page 38

A stored procedure can be a business object that your module runs with the `Execute` operation. It can run in place of the standard SQL for an operation on any business object, or it can perform additional actions before or after performing an operation.

Create operation:

The Create operation creates rows in database tables corresponding to the business object in the request. When given a hierarchical business object, the Create operation recursively traverses the business object, creating rows corresponding to each business object in the hierarchy.

To process the Create operation, the adapter performs the following actions:

1. Recursively inserts each single-cardinality child business object contained with ownership into the database. In other words, the adapter creates the child and all child business objects that the child and its children contain.

If the business object definition specifies that an attribute represents a child business object with single-cardinality and that attribute is empty, the adapter ignores the attribute. However, if the business object definition requires that the attribute represent a child, and it does not, the adapter returns an error and stops processing.

2. Retrieves and checks for the existence of each single-cardinality child business object contained without ownership. If the retrieval is unsuccessful, indicating that the child does not exist in the database, the adapter returns an error, and stops processing. If the Retrieve operation is successful, the adapter recursively updates the child business object. If the retrieve operation is successful, the adapter continues the process of creating the parent business object; the adapter does not update the child business object without ownership.

Note: For this approach to work correctly when the child business object exists in the database, primary-key attributes in child business objects must be cross-referenced correctly on Create operations. If the child business object does not exist in the application database, the primary-key attributes must not be set.

3. Inserts the top-level business object in the database by performing the following actions:
 - a. Sets each of the foreign-key values of the top-level business object to the primary key values of the corresponding child business object represented with single-cardinality. Because values in child business objects can be set by database sequences or counters or by the database itself during the creation of the child, this step ensures that the foreign-key values in the parent are correct before the adapter inserts the parent in the database.
 - b. Generates a new, unique ID value for each attribute that is set automatically by the database. The name of the database sequence or counter is stored in the attribute application-specific information. If an attribute has an associated database sequence or counter, the value generated by the adapter overwrites any value passed in by the application server.
 - c. Inserts the top-level business object into the database.

Note: The adapter treats an empty complex column as null value irrespective of setting it to null or unset.

4. Processes each of its multiple-cardinality child business objects as follows:
 - a. Sets the foreign-key values in each child to reference the value in the corresponding primary key attributes in the parent. Because the parent primary key values might have been generated during the creation of the parent, this ensures that the foreign-key values in each child are correct before the adapter inserts the child into the database.
 - b. Inserts each of the multiple-cardinality child business objects into the database.

Related reference

“Fault business objects” on page 221

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Retrieve operation:

The Retrieve operation extracts data from a database for a hierarchy of business objects.

To process the Retrieve operation, the adapter performs the following actions:

1. Removes all child business objects from the top-level business object it received. In other words, it makes a copy of the top-level business object without any children.
2. Retrieves the top-level business object from the database.
 - If the retrieval operation returns one row, the adapter continues processing.
 - If the retrieval operation returns no rows, indicating that the top-level business object does not exist in the database, the adapter returns the `RecordNotFoundException` error.
 - If the retrieval operation returns more than one row, the adapter returns the `MultipleMatchingRecordsException` error.

The Retrieve operation uses only the primary key. Other columns are ignored.

3. Recursively retrieves all multiple-cardinality child business objects.

Note: The adapter does not enforce uniqueness when populating an array of business objects. The database is responsible for ensuring uniqueness of the business objects. If the database returns duplicate child business objects, the adapter returns duplicate children.

4. Recursively retrieves each of the single-cardinality children, regardless of whether the child business object is contained with or without ownership.

Note: All single-cardinality child business objects are processed based on their occurrence in the business object and before the parent business object is processed.

Retrieving NULL data

The adapter can retrieve a record from a database table when the column value is NULL. For example, a Customer business object might have these columns: `custid`, `ccode`, `fname`, and `lname`, where `custid` and `ccode` form composite keys. Composite keys are primary keys that refer to more than one attribute and are used to define the uniqueness of the business object. You can retrieve a Customer record for which the value of `ccode` is NULL. The adapter generates a SELECT statement for the Retrieve operation as:

```
select custid, ccode, fname, lname from customer where custid=? and ccode is null
```

Related reference

“Fault business objects” on page 221

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

RetrieveAll operation:

The adapter uses the RetrieveAll operation to retrieve an array of business objects from the database. The adapter uses different processes depending on whether the RetrieveAll operation is for database table business objects or for user-specified SQL business objects.

For database table business objects

All of the key and non-key attributes populated in the incoming business object determine the selection criteria for the retrieval. The adapter may retrieve multiple rows for the top-level business object from the database, depending on the attributes selected. All values specified in the top-level business object are used. The settings in the child business object are ignored. If no attributes are populated in the incoming business object, all the rows are retrieved from the respective table in the database.

The name of a generated business object matches the name of the table in the database. For example, the Customer table in the database is represented as a business object named “Customer”.

To retrieve an array of business objects, the adapter performs the following actions:

1. Constructs a container business object for all of the retrieved rows. The name of the container business object is the name of the business object with the string “Container” appended to it.
2. If the module is configured to use business graphs, which are optional, then it constructs a top-level business graph for each of the retrieved rows. The name of the business graph is the business object name with the string “BG” appended to it.
3. Retrieves each of the business graphs in the container using the Retrieve operation.

The following figures show the structure of objects returned from a RetrieveAll operation, with and without business graphs.

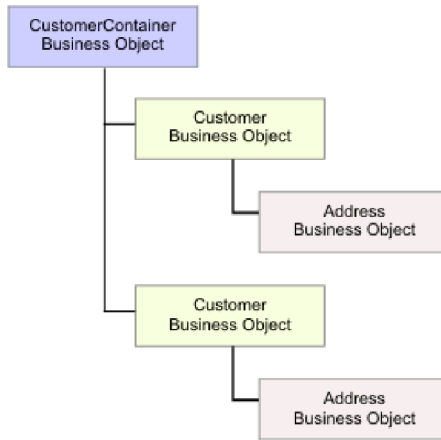


Figure 3. Structure of the business object returned in a RetrieveAll operation without optional business graphs

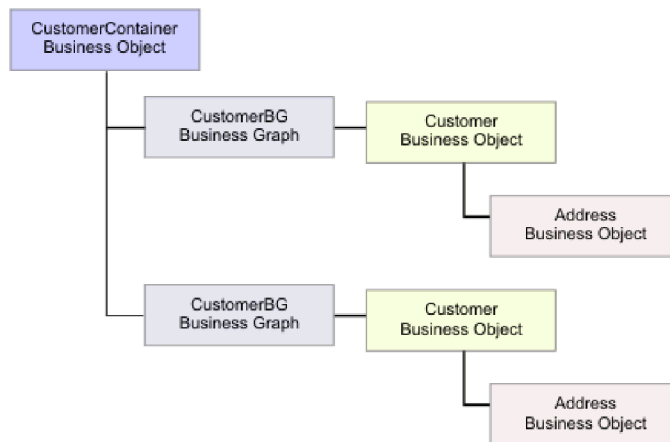


Figure 4. Structure of the business object returned in a RetrieveAll operation with optional business graphs

The following errors can result from a RetrieveAll operation:

- **RecordNotFoundException**– This exception is generated when one or more populated business objects in the input object does not exist in the enterprise information system and if the managed connection factory property for the `ErrorOnEmptyResultSet` property is set to `True`. To get an empty result set without the `RecordNotFoundException`, you can either modify the `.import` file and set the “Throw exception when no records are found (`ErrorOnEmptyResultSet`)” on page 235 property to `False`, or after you deploy the outbound application on IBM Business Process Manager or WebSphere Enterprise Service Bus, configure the MCF properties to change the value for the “Throw exception when no records are found (`ErrorOnEmptyResultSet`)” on page 235 property to `False`.
- **MatchesExceededLimitException** – The number of matching records in the database exceeds the value of the “Maximum records for RetrieveAll operation” on page 249 property that is defined in the interaction specification. The `MatchCount` attribute of the fault contains the actual number of matches that the adapter found in the database, so that you can either increase the limit or refine the search.

Note: If the “Maximum records for RetrieveAll operation” on page 249 property is set to a large number, problems can occur due to a lack of sufficient memory, depending upon the size and number of business objects returned.

- `EISSystemException` – One or more unrecoverable errors are reported by the database (the enterprise information system).

For query business objects

Business objects that are created for user-specified SELECT statements (query business objects) also support the RetrieveAll operation. The external service wizard generates the query business object by running the user-specified SQL SELECT statement and creating a hierarchy of query business objects.

If you are using optional business graphs, the hierarchy will look like that shown in Figure 5.

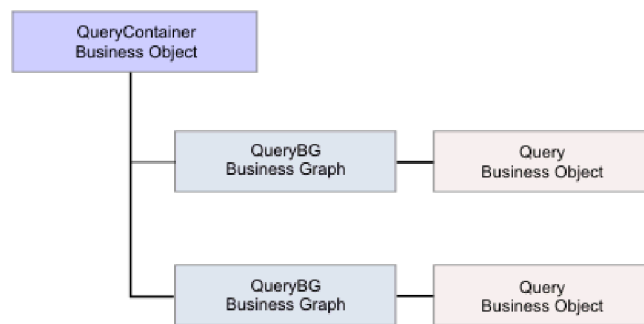


Figure 5. User-specified query business objects (with optional business graphs)

If you are not using optional business graphs, the hierarchy will look like that shown in Figure 6.

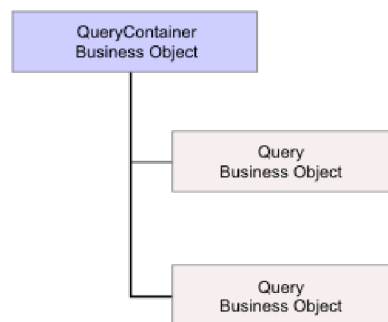


Figure 6. User-specified query business objects (without optional business graphs)

To process the query business object generated by the external service wizard for the user-specified SELECT statement, the adapter performs the following actions:

1. Obtains the SELECT SQL statement from the query business object.
2. Determines whether a dynamic WHERE clause is specified in the query business object.
 - If there is a dynamic WHERE clause, the adapter replaces the default WHERE clause in the SELECT statement with the dynamic one.
 - If there is no dynamic WHERE clause, the adapter replaces parameters in the SELECT statement with the corresponding values specified in the query business object.

3. Runs the SELECT statement.
4. Obtains the result set that is returned and populates the query business object values with the data returned from the database, creating a container business object with the structure shown in Figure 5 on page 11.
5. Retrieves the entire hierarchy (a *deep retrieve*) of each top-level query business object in the container, if any child business objects are defined for the query business objects.

Note: A query business object can be a top-level business object only. A query business object cannot have child query business objects.

Retrieving NULL objects

The adapter can retrieve records from a database table when the column value is NULL. For example, a Customer business object might have these columns: custid, ccode, fname, and lname, where ccode need not be a primary key. You can query all the Customer records for which the ccode column is NULL. The adapter generates a select query for the RetrieveAll operation as:

```
select custid, ccode, fname, lname from customer where custid=? and ccode is NULL
```

Related reference

“Fault business objects” on page 221

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Update operation:

In an Update operation, the source business object is compared to a business object that is retrieved from the database using the primary keys specified in the top-level, source business object.

When updating a hierarchical business object, the adapter performs the following actions:

1. Uses the primary key values of the source business object to retrieve the corresponding entity from the database. The retrieved business object is an accurate representation of the current state of the data in the database.

If the retrieval fails, indicating that the top-level business object does not exist in the database, the adapter returns the RecordNotFoundException exception, and the update fails.

If the retrieval succeeds, the adapter compares the retrieved business object to the source business object to determine which child business objects require changes in the database. The adapter does not, however, compare values in the source business object's simple attributes to those in the retrieved business object. The adapter updates the values of all non-key simple attributes.

If all the simple attributes in the top-level business object represent keys, the adapter cannot generate an update query for the top-level business object. In this case, the adapter logs a warning and continues.

2. Recursively updates all single-cardinality children of the top-level business object.

If ownership is true and the child is present in the source business object but not in the retrieved business object, the adapter recursively creates the child in the database.

The adapter handles single-cardinality children contained with ownership in one of the following ways:

- If the child is present in both the source and the retrieved business objects, instead of updating the existing child in the database, the adapter deletes the existing child and creates the child.
- If the child is present in the source business object but not in the retrieved business object, the adapter recursively creates the child in the database.
- If the child is present in the retrieved business object but not in the source business object, the adapter recursively deletes the child from the database.

For single-cardinality children contained without ownership, the adapter attempts to retrieve every child that is present in the source business object from the database. If it successfully retrieves the child, the adapter populates the child business object but does not update it, because the adapter never modifies single-cardinality children contained without ownership. If the retrieval fails, the adapter returns an `ObjectNotFound` exception.

3. Updates all simple attributes of the retrieved business object, except those whose corresponding attribute in the source business object is not specified. Because the business object being updated must be unique, the adapter verifies that only one row is processed as a result. If more than one row is returned, the adapter returns an error.
4. Processes each multiple-cardinality child of the retrieved business object in one of the following ways:
 - If the child exists in both the source and the retrieved business object arrays, the adapter recursively updates it in the database.
 - If the child exists in the source array but not in the array of the retrieved business object, the adapter recursively creates it in the database.
 - If the child exists in the array of the retrieved business object but not in the source array, the adapter recursively deletes it from the database unless the application-specific information for the attribute that represents the child in the parent has the `KeepRelationship` property set to `True`. In this case, the adapter does not delete the child from the database.

NULL data and the Update operation

The adapter can update a record from a database table when the column value is `NULL`. For example, a `Customer` business object might have these columns: `custid`, `ccode`, `fname`, and `lname`, where `custid` and `ccode` form composite keys. Composite keys are primary keys that refer to more than one attribute and are used to define the uniqueness of the business object. You can update a `Customer` record for which the value of `ccode` is `NULL`. The adapter would generate an update query for the Update operation as:

```
update customer set fname=?, lname=? where custid=? and ccode is null
```

Note: The adapter treats an empty complex column as null value irrespective of setting it to null or unset.

Related reference

“Fault business objects” on page 221

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

ApplyChanges operation:

The ApplyChanges operation provides both delta and after-image support for modifying or deleting a business object. The ApplyChanges operation is available only if you use business graphs.

If you set the verb property of the business graph to the name of an operation, such as create, update, or delete, the adapter performs after-image processing for the ApplyChanges operation. For example, if the verb is set to create, the adapter processes the ApplyChanges operation the same as the Create operation.

If you do not set the verb in the business graph, the adapter uses the ChangeSummary in the business graph to update the business object. In this mode, the ApplyChanges operation differs from the Update operation in the following ways:

- In the ApplyChanges operation, no Retrieve operation occurs before updating.
- No comparisons are made between the incoming business object and the business object in the database.
- All children are processed based on the operation set in the ChangeSummary for each child business object. If a child does not have an operation set in it, the adapter returns an error.

The adapter performs the following steps when updating a hierarchical business object from the ChangeSummary. It processes only the changes from the ChangeSummary.

1. It recursively processes all single-cardinality children of the parent object. If a child is marked required in the business object specification, it must be present in the inbound object. If it is not, the ApplyChanges operation fails, and the adapter returns an error.
2. It sets all foreign key values in the parent that reference attributes in single-cardinality children to their corresponding child values. This is necessary because single-cardinality children might have been added to the database during the previous steps, resulting in the generation of new sequence values.
3. It updates the current object being processed using an SQL UPDATE statement or a stored procedure. All simple attributes of the individual business object are updated. The adapter does not use property level changes to determine which attributes need to be added to the UPDATE statement; they are all updated. Because the object being updated should be unique, the adapter checks to ensure that only one row is processed as a result. An error is returned if more than one row is processed.
4. It sets all foreign key values in all cardinality N children of the current object that reference parent attributes to the corresponding parent values. Typically, these values are already cross-referenced during data mapping; however, this might not be the case for new children in cardinality N containers. This step ensures that the foreign key values in all cardinality N children are correct before those children are updated.
5. It updates all cardinality N containers of the current object.

When the child objects are processed, each child's operation is taken, and the appropriate operation is performed. The operations allowed on a child in ApplyChanges are Create, Delete, and Update:

- If a Create operation is found in the child, the child is created in the database if it is an ownership child. Non-ownership children are retrieved to validate their existence in the database.
- If a Delete operation is found in the child, that child is deleted.

- If an Update operation is found in the child, the child is updated in the database.

Related reference

“Fault business objects” on page 221

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Delete operation:

The Delete operation is performed by pruning the incoming business object and then retrieving the complete business object from the database. The Delete operation is then applied recursively on each business object in the hierarchy.

The Delete operation supports physical and logical deletes, depending on the StatusColumnName value in the application-specific information of the business object. If the StatusColumnName value is defined, the adapter performs a logical delete operation. If the StatusColumnName value is not defined, the adapter performs a physical delete operation.

Physical deletes

For physical deletes the adapter takes the following actions:

- It recursively deletes all multiple-cardinality child business objects.
- It deletes the top-level business object.
- It recursively deletes all single-cardinality child business objects contained with ownership.

Logical deletes

For logical deletes the adapter takes the following actions:

- It issues an update that sets the status attribute of the business object to the value specified by the business object-level application-specific information. The adapter ensures that only one database row is updated as a result, and it returns an error if this is not the case.
- It recursively logically deletes all single-cardinality children contained with ownership and all multiple-cardinality children. The adapter does not delete single-cardinality children contained without ownership.

NULL data and the Delete operation

The adapter can delete a record from a database table when the column value is null. For example, a Customer business object might have these columns: custid, ccode, fname, and lname, where custid and ccode form composite keys. Composite keys are primary keys that refer to more than one attribute and are used to define the uniqueness of the business object. You can delete a Customer record for which the value of ccode is null. The adapter generates a delete query for the Delete operation as:

```
delete from customer where custid=? and ccode is null
```

Related reference

“Fault business objects” on page 221

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Execute operation:

The Execute operation is used to run stored procedures and stored functions. The external service wizard generates the required stored procedure business object that corresponds to the stored procedure or stored function definition in the database. The adapter uses the Execute operation to process the stored procedure business object.

The following information provides a simple example of a stored procedure, the business object that is constructed from it, and the steps the adapter uses to process the stored procedure business object with an Execute operation.

A simple example of a stored procedure:

```
PROCEDURE testSP(IN int x,INOUT VARCHAR(10) msgSTR, OUT int status,  
                OUT struct outrec, OUT array retArr)
```

The procedure returns two result sets.

For this stored procedure, following is an example of the business object that is constructed:

```
BOLevel ASI  
  SPName=testSP  
  ResultSet=true  
  MaxNumberOfResultSets=2  
  ReturnValue = propName  
    Returned if the stored procedure is a function. function).  
    Will be property name corresponding to the child business  
    object if returned value is complex type(array/struct/resultset)  
  Defined only if it is a Function
```

Properties

```
x Type=IP  
msgStr Type=IO  
status Type=OP  
outrec Type OP - Child BO for outrec, ASI ChildBOType = struct  
retarr Type OP - n cardinality child BO for retArr, ASI ChildBOType = array  
childBOName1 - Child BO for 1st result set, ASI ChildBOType = resultset  
childBOName2 - Child BO for 2nd result set, ASI ChildBOType = resultset
```

To process this stored procedure business object with an Execute operation, the adapter:

1. Constructs the following stored procedure call: CALL testSP(x, msgStr, status, outrec, retArr).
2. Sets the input parameters x and msgStr on the callable statement.
3. Runs the callable statement.
4. Obtains the return value (if Function) and sets the value in the appropriate attribute if it is a scalar value, or in a child business object if it is a complex value (such as struct, array).
5. Obtains the first result set and creates the container for ResultSet1.
6. Obtains the second result set and creates the container for ResultSet2.

7. Obtains the output parameters msgStr and status, and sets the corresponding attributes on the business object.
8. Obtains the output parameter outrec and creates the child business object from the data returned in outrec. If outrec is a nested struct type, then the adapter recursively creates and stores data in the hierarchical child business object.
9. Obtains the output parameter retArr and creates a multiple cardinality child business object from the data returned in retArr. If retArr is a nested array type, then the adapter recursively creates and stores data in the hierarchical child business object.

Related reference

“Fault business objects” on page 221

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Exists operation:

The Exists operation determines whether the database contains records that match the attributes set in a business object.

You can use both key and non-key attributes in the selection criteria.

Note: When using the external service wizard to discover table objects in a database, you can select multiple tables and add those tables to the selected objects portion of the Find Objects in the Enterprise System screen. However, you cannot use the external service wizard to link or join the tables that you selected. If the goal of your business application requires the table business object to perform an Exists operation on joined tables, you need to join the tables in the database to create a view of the joined tables. After you have created a view of the joined tables, you can then perform discovery on the view. The Exists operation would be supported on this view.

To process the Exists operation and send the results based on the specified business object attributes, the adapter performs the following actions:

1. The adapter receives a table business object from the import. This business object can be flat (simple with no child business objects) or hierarchical (complex, containing one, or more child business objects).

If the business object is hierarchical, it is only for the top-level business object (the individual business object at the top of a hierarchical business object) for which the adapter builds the query.

Note: The input business object that supports the Exists operation varies depending on the business object type. In addition to being supported by the table business object, the Exists operation is also supported by the views business object and the synonyms and nicknames business object.

2. The adapter uses the table business object to generate an SQL SELECT statement that it sends to the server.

The SQL SELECT statement used is as follows:

```
select count(*) from TABLENAME where column1=? AND column2=?
```

Here is a sample SQL statement for our example:

```
select count(*) from CUSTOMER where fname='John' AND lname='Smith'
```

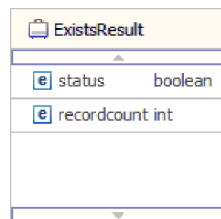
In this case, the SQL statement specifies *non-primary key* attributes *fname* and *lname*, with the assigned values of *John* and *Smith*.

The adapter includes the attribute information from the table business object in the *where* clause of the SQL query.

3. The database server runs the SQL query and sends the results back to the adapter.
4. The adapter obtains the results of the SQL query from the database server and sets the *recordcount* and *status* attributes on the **ExistsResults** business object. For example, if the Exists operation determined there are two records that match the attribute and value settings in the business object, the adapter sets *status=true* and *recordcount=2*.

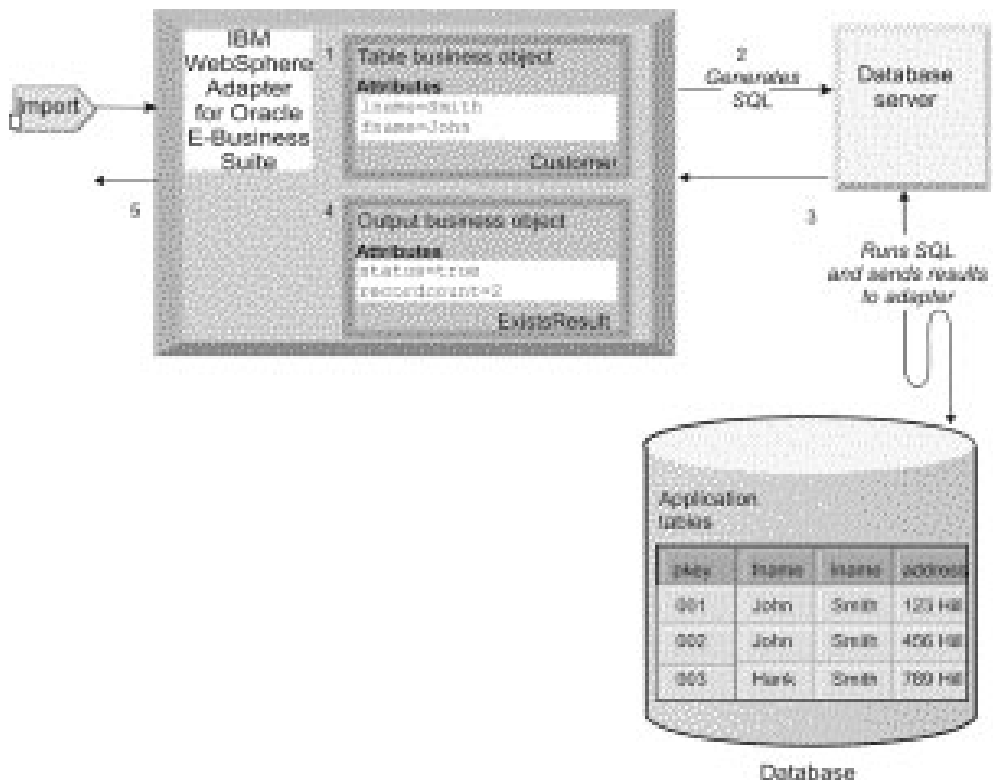
If a record with the specified attributes is not found, the *status* output parameter is **false** and the *recordcount* output parameter is **0**.

The following screen capture illustrates the structure of the ExistsResult business object returned from an Exists operation.



5. The adapter returns the ExistsResult business object to the caller.

The following illustration shows how the adapter processes a table business object with an Exists operation.



Related reference

“Fault business objects” on page 221

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Invoke operation:

The Invoke operation is used to invoke XML Gateway Web Service. The external service wizard generates the required XML Gateway Web Service business object that corresponds to the XML Gateway Web Service interface in Oracle E-Business Suite.

The following information provides a simple example of a XMLGateway Web Service: `http://hostname:port/webservices/AppsWSPProvider/oracle/apps/fnd/XMLGateway?wsdl`

```
<binding name="XMLGatewaySOAPBinding" type="tns:XMLGatewayPortType">
<Soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="ReceiveDocument">
<soap:operation soapAction="http://oraeb05.cn.ibm.com:port/webservices/AppsWSPProvider/oracle/apps/fnd/XMLGateway?wsdl" />
<input>
<soap:header message="tns:ReceiveDocument_Request" part="header" use="literal" />
<soap:body parts="body" use="literal" />
</input>
<output>
</output>
</operation>
</binding>
<service name="XMLGateway">
<documentation>Generic Web Service for receiving XML Gateway Transactions</documentation>
<port binding="tns:XMLGatewaySOAPBinding" name="XMLGatewayPort">
<soap:address location="http://oraeb05.cn.ibm.com:8000/webservices/AppsWSPProvider/oracle/apps/fnd/XMLGateway?wsdl" />
</port>
</service>
```

The following three child business objects are generated:

```
<xsd:element name="header" type="xgwxmlgateway_header:XGWXMLGateway_Header" nillable="true" minOccurs="1" maxOccurs="1"/>
<xsd:element name="payload" type="xgw_receivedocument:XGW_ReceiveDocument" nillable="true" minOccurs="1" maxOccurs="1"/>
<xsd:element name="response" type="xgwreceivedocument_response:XGWReceiveDocument_Response" nillable="true" minOccurs="1" maxOccurs="1"/>
```

The header contains the XMLGateway header properties, such as Message_Standard, Transaction_Type, and so on. The payload contains the integration data business object, such as Purchase Order, Invoice, and so on. The response contains the return information from Oracle E-Business Suite.

To process this stored procedure business object with an Invoke operation, the adapter:

1. Initializes the Web Service client with XMLGateway Web Service WSDL.
2. Sets the input parameters : header properties and payload integration data.
3. Invokes the operation:ReceiveDocument in XMLGateway Web Service WSDL.
4. Sends the data to Oracle E-Business Suite and receives the response information.

Publish operation:

The Publish operation is used to invoke XML Gateway JMS interface. The external service wizard generates the required XML Gateway JMS business object that

corresponds to the XML Gateway JMS interface in Oracle E-Business Suite. The adapter uses the Publish operation to process the XML Gateway JMS business object.

The following information provides a simple example of a XMLGateway JMS Service:

```
<EBSConnectionType>XML Gateway</EBSConnectionType>XML Gateway>  
<EBSHostName>ora220.cn.ibm.com</EBSHostName>  
<EBSPortNumber>1521</EBSPortNumber>  
<EBSQueueName>wf_jms_in</EBSQueueName>  
<EBSSchemaName>appls</EBSSchemaName>  
<EBSSystemID>VIS</EBSSystemID>  
<EBSTransport>JMS</EBSTransport>
```

The following 3 child business objects are generated:

```
<xsd:element name="header" type="xgwxmlgateway_header:XGWXMLGateway_Header" nillable="true" minOccurs="1" maxOccurs="1"/>  
<xsd:element name="payload" type="xgw_receivedocument:XGW_ReceiveDocument" nillable="true" minOccurs="1" maxOccurs="1"/>  
<xsd:element name="response" type="xgwreceivedocument_response:XGWReceiveDocument_Response" nillable="true" minOccurs="1" maxOccurs="1"/>
```

The header contains the XML Gateway header properties, such as Message_Standard, and Transaction_Type. The payload contains the integration data business object, such as Purchase Order, Invoice, and so on. The response contains the return information from Oracle E-Business Suite.

To process this stored procedure business object with Publish operation, the adapter:

1. Initializes the JMS client with XMLGateway JMS interface information such as hostname, port, queue name and so on.
2. Sets the input parameters : header properties and payload integration data.
3. Constructs the integration data into JMS Message with header and body.
4. Sends the data to Oracle E-Business Suite and receives the response information.

Enqueue operation:

The Enqueue operation is used to run a dynamic pl/sql script for enqueueing. The external service wizard generates the required advanced queue business object that corresponds to the advanced queue definition in the database. The adapter uses the Enqueue operation to process the advanced queue business object.

The following information provides a simple example of a advanced queue:

```
Schema:APPLSYS  
Queue:WF_JMS_IN
```

To process this stored procedure business object with an Enqueue operation, the adapter:

1. Constructs a dynamic call statement based on Oracle AQ PL/SQL API.
2. Sets the input parameters : queue header properties and queue payload integration data.
3. Runs the script by JDBC connection to enqueue the data into specific queue.
4. Receives the message id in queue.

Inbound processing

IBM WebSphere Adapter for Oracle E-Business Suite supports inbound event management with event delivery. Events are processed from an event store that is populated either by the database application or from the result of custom queries that you provide. You control how often the adapter polls for events and how many records are delivered to the export at one time.

The adapter polls for changes using one of these methods:

- Standard event processing, in which the adapter examines the event store for events that are stored there by the database application
- Custom event processing, in which the adapter runs user-defined queries, stored procedures, or stored functions

You can customize standard or custom event processing when you use the external service wizard to configure the adapter initially or at a later time by using the administrative console of the server to change the activation specification properties.

The database object that is the subject of the event is not retrieved until after a notification is delivered to the export. As a result, the detection and notification of any retrieval errors that occur is deferred until after notification of the export. This differs from the event processing in version 6.2.x of the adapter, where retrieval errors can be detected before the adapter notifies the export.

Related tasks

“Configuring the module for inbound processing” on page 133

To configure a module to use the adapter for inbound processing, use the external service wizard in IBM Integration Designer to find and select business objects and services from the Oracle database, and to generate business object definitions and related artifacts.

Related reference

“Activation specification properties” on page 256

Activation specification properties are properties that hold the inbound event processing configuration information for an export.

Standard event processing

In standard event processing, the adapter provides the SQL queries that poll for events and ensure that the event is delivered exactly one time.

When records are created, updated, or deleted in the tables in the database, an event program is run immediately. For Oracle database, the database triggers or tools such as Oracle Change Data Capture are run, and, for Oracle E-Business Suite, Oracle Business Event System and the concurrent programs are run. A trigger or other tool writes an event record into the *event store*, which is a persistent cache where event records are saved until a polling adapter can process them. The event store is implemented as a table in the same database as the user tables, which are the tables that contain the database objects accessed by the adapter.

You must define the triggers or set up other tools to report changes to the database tables about which you want to receive events.

The adapter offers assured once delivery, which guarantees that each event is delivered once and only once to the export. If you enable assured once delivery for the module, a transaction ID (XID) is set for each event in the event store. After an

event is obtained for processing, the XID value for that event is updated in the event store. The event is then delivered to its corresponding export, and then deleted from the event store. If the database connection is broken or the application is stopped before the event can be delivered, the event cannot be processed completely. In this case, the XID column indicates that the event must be reprocessed and sent to the export again. After the database connection is reestablished or the adapter starts again, the adapter checks for events in the event store that have a value in the XID column. The adapter processes these events first, and then polls the other events during the poll cycles.

The adapter can process all events or filter events by business object type, time stamp, or connector ID. The filter is set by use of the activation specification property `EventTypeFilter`, `FilterFutureEvents`, or `AdapterInstanceEventFilter`. The `EventFilterType` property has a comma-delimited list of business object types. Only the types specified in the property are processed. If no value is specified for the property, no filter is applied, and all the events are processed. If the activation specification property `FilterFutureEvents` is set to true, the adapter filters events by timestamp and connector ID. The adapter compares the system time in each poll cycle to the time stamp on each event. If an event is set to occur in the future, it will not be processed until that time. If the `AdapterInstanceEventFilter` activation specification property is set, only the connector ID specified in the `AdapterInstanceEventFilter` property are processed.

Custom event processing

In custom event processing, you provide the SQL queries or stored procedures that poll for events.

With custom event processing, you control which events are delivered to the export by providing a database query (the *custom event query*) for the adapter to run in place of the SQL query it uses to poll the event store in standard event processing. The custom event query must perform any necessary filtering. You specify that you want custom event processing by selecting an option in the wizard or by setting the `EventQueryType` activation specification property in the administrative console.

Custom event processing supports assured once delivery if you create the standard event store for storing XID values. The adapter stores the events returned by the custom event query in the event store and it updates the events with XID values. The adapter processes the events in the same way as for standard event processing. Do not create a custom query that queries the standard event store, because that table temporarily holds the events when the adapter is configured for assured once delivery. In addition, in this situation the event store must not have an automatic generation of event ID values, because the adapter populates the event ID value it retrieves from the custom query in the event store.

Note: When you are using custom event processing, set the property “Ensure once-only event delivery (`AssuredOnceDelivery`)” on page 265 to True.

Turn custom event processing on by selecting an advanced option in the wizard when you configure your module to use the adapter or by setting the `EventQueryType` activation specification property.

Custom event query

You specify the custom event query to run by providing a user-defined event query in an advanced option in the wizard or by setting the CustomEventQuery activation specification property. Specify one of the following types of programs:

- Standard SQL statements
- A stored procedure
- A stored function

Any of these programs takes an input parameter containing the poll quantity, an activation specification property that the adapter provides at run time. The program can accept other input parameters as well. These programs must return a result set that has the poll quantity number of records and contains the following columns in order: *event_id*, *object_key*, *object_name*, and *object_function*. The adapter generates the event object from the result set and processes the events.

Standard SQL statements

You can provide an SQL SELECT statement that selects the events to process. The query can have input parameters in addition to the input parameter for poll quantity.

Stored procedure

The custom query can be a stored procedure that accepts the poll quantity as input and returns an output parameter of type result set. Use the following syntax to specify a stored procedure:

```
call procedure_name (?, ?)
```

Where *procedure_name* is the name of the stored procedure to run. The first parameter represents the poll quantity and the second parameter represents the result set.

The stored procedure can accept other input parameters as well, which you provide in the call statement itself, for example:

```
call procedure_name (25, ?, ?)
```

Stored function

The custom query can be a stored function that accepts the poll quantity as input and returns a result set. Use the following syntax to specify a stored function:

```
? = call function_name (?)
```

Where *function_name* is the name of the stored function. The first parameter represents the result set, and the second parameter represents the poll quantity.

The stored function can accept other input parameters, which you provide in the call statement itself, for example:

```
? = call function_name (?, 'abc')
```

Custom update and delete queries

Custom event processing also allows you to provide custom update and delete queries, which are run after each event is processed. You typically use an *update query* to ensure that a database record does not get picked up for processing

during subsequent poll cycles. Use a *delete query* when database records need to be deleted after each event is processed. Both the update and delete queries are optional.

Update and delete queries are specified by the CustomUpdateQuery and CustomDeleteQuery activation specification properties. You can enter these queries as a standard SQL statement, a stored procedure, or a stored function. The syntax for the custom update or delete query is the same as the syntax for the custom query. Update and delete queries take an input parameter for the event ID. The adapter provides the value of event ID at run time. The queries can also have additional input parameters, which you provide in the query syntax itself, in the same way as described for the custom event query.

Event store

The event store is a persistent cache where event records are saved until the polling adapter can process them. The adapter uses the event store to track the inbound requests as they make their way through the system. Each time a database record is created, updated, or deleted, the adapter updates the status of the event in the event store. The status of each event is continually updated by the adapter for recovery purposes until the events are delivered to a configured export on the server.

The adapter polls the event records from the event store at regular intervals. In each poll call, a number of events are processed by the adapter. Events are processed in ascending order of priority and ascending order of the event time stamp. In each poll cycle, the adapter picks up all new events. For each new event, the adapter retrieves the value set in the object key field for the event and then loads the business object that corresponds to the value specified in the object name field. After the object is loaded, the adapter sets the primary key values of the business object based on the value specified in the object key field. After setting the keys, adapter performs a retrieval of the object based on the keys. The business object or optional business graph is created from the retrieved information and is published to the export.

If you have associated a stored procedure with the Retrieve operation of the business object, you can define the mapping between the input parameters of the stored procedure and the business object attributes (generally, primary keys). If such a mapping is defined, then the adapter sets the input parameters for the stored procedure, invokes the stored procedure, and populates the object based on the results obtained from the stored procedure.

For stored procedures and functions, if you defined a mapping between the input parameters of the stored procedure or function and the business object attributes (generally using primary keys) using the RetrieveSP application-specific information, then the adapter sets the input parameters on the stored procedure, invokes the stored procedure, and populates the business object based on the results obtained from the stored procedure.

When the object_function column has the value Delete, which indicates that the object was deleted, the object is not retrieved from the database. The keys are set on the data object and the business object or optional business graph is created and delivered to the export.

If an event is successfully posted, the entry is deleted from the event store. For failed events, the entries remain in the event store and the event_status column is set to -1.

The table format and content of the event store are described Table 2.

Table 2. Definition of the event store database table

Column name	Type	Description
XID	String	The unique transaction ID (XID) value for assured once delivery.
event_id	Number	The unique event ID, which is a primary key for the table. This can have the same value as the object_key.
object_key	String	A string that contains keys of the record in the event store that is retrieved. This column cannot be null. Specify the value as one or more <i>key=value</i> pairs, separated by the semicolon character (;). Alternatively, you can specify only the values for the primary keys separated by the semicolon (;) character. In this case, the values must be specified in the same order as primary keys are defined in the business object.
object_name	String	The name of the business object or business graph. The business object (or the business object within the business graph) can be a hierarchical business object. Each business object or business graph refers to a table or view. This column cannot be null.
object_function	String	The operation corresponding to the event (Delete, Create, Update, and so on). This column cannot be null.
event_priority	Number	Identifies the event priority. This value must be a positive integer. This column cannot be null.
event_time	Timestamp	Date and time when event was generated. The format is mm/dd/yyyy hh:mm:ss.
event_status	Number	The event status is initially set to a value for a new event and updated by the adapter as it processes the event. The status can have one of the following values: <ul style="list-style-type: none"> • 0: Identifies a new event. • 1: Identifies an event that has been delivered to an export. • -1: An error occurred while processing the event. This column cannot be null.
event_comment	String	Any comment associated with the event.
connector_ID	String	The unique identifier for the adapter instance that receives a specific event.

Business objects

A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. WebSphere Adapter for Oracle E-Business Suite uses business objects to represent tables and views in the database as well as the results of database queries, stored procedures,

and stored functions. Business objects can also create a hierarchy of objects from your database and group unrelated tables. Your component communicates with the adapter using business objects.

How the adapter uses business objects

An integrated application uses business objects to access a database. The adapter converts the business objects in outbound requests into JDBC API calls to access the database. For inbound events, the adapter converts the data in the events into business objects, which are returned to the application.

The adapter uses business objects to represent the following types of objects in a database:

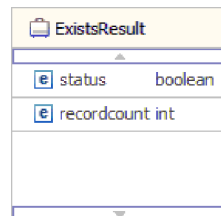
- Tables and views
- Synonyms and nicknames
- Stored procedures and stored functions

Query business objects do not represent database objects. Query business objects represent a user-defined SQL query to run against the database.

Note: Before using the business objects to represent the objects types mentioned earlier, ensure that the Java keywords are not used to define the names of tables, views, stored procedures, and stored functions parameters.

Adapters use some business objects for output. These business objects include:

- Container business object, which contains the output from a RetrieveAll operation.
- ExistsResult business object, which contains the output from an Exists operation.



How data is represented in business objects

For table or view business objects

Each column in the table or view is represented by a simple attribute of the table or view business object. A *simple attribute* is an attribute that represents a single value, such as a String, Integer, or Date. Other attributes represent a child business object or an array of child business objects.

Simple attributes within the same business object cannot be stored in different database tables; however, the following situations are possible:

- The database table can have more columns than the corresponding business object has simple attributes; that is, some columns in the database are not represented in the business object. Only those columns needed for the processing of your business object must be included in your design.
- The business object can have more simple attributes than the corresponding database table has columns; that is, some attributes in the business object are not

represented in the database. The attributes that do not have a representation in the database either have no application-specific information, are set with default values, or are parameters for stored procedures or stored functions.

- The business object can represent a view that spans multiple database tables. The adapter can use such a business object when processing events triggered by changes to the database, such as Create, Update, and Delete operations. When processing business object requests, however, the adapter can use such a business object only for Retrieve and RetrieveAll requests.

A table business object always has a primary key, even if the corresponding database table does not have a primary key. The adapter uses the column specified in the primary key attribute when it retrieves table business objects. The adapter supports tables that have composite, or multiple, primary keys. If a database table has one or more primary keys, the wizard sets the primary key property for those columns in the table business object. If the database table does not have a primary key, the external service wizard prompts you for primary key information when you configure that business object. Specify a column that contains unique data, such as a sequence or identity column.

If the table business object contains the Date and Timestamp data types, the format of these types can be customized in the **Application Info** section of the **Properties** view of the business object. For example, you can specify the date format in dd/MM/yy and timestamp in HH/mm/ss. If you want to customize the format of the Date and Timestamp data types, ensure that the data types are mapped to the default string data type in the **Configuration Properties** window.

Table and view business objects support the Create, Update, Delete, Retrieve, RetrieveAll, Exists, and ApplyChanges outbound operations. When running an Exists operation on a hierarchical table business object, only the top-level business object is queried.

Figure 7 on page 28 shows a table business object in the business object editor. The business object has an attribute for each of the columns in the database table. Because the table has no child business objects, all the attributes are simple attributes.

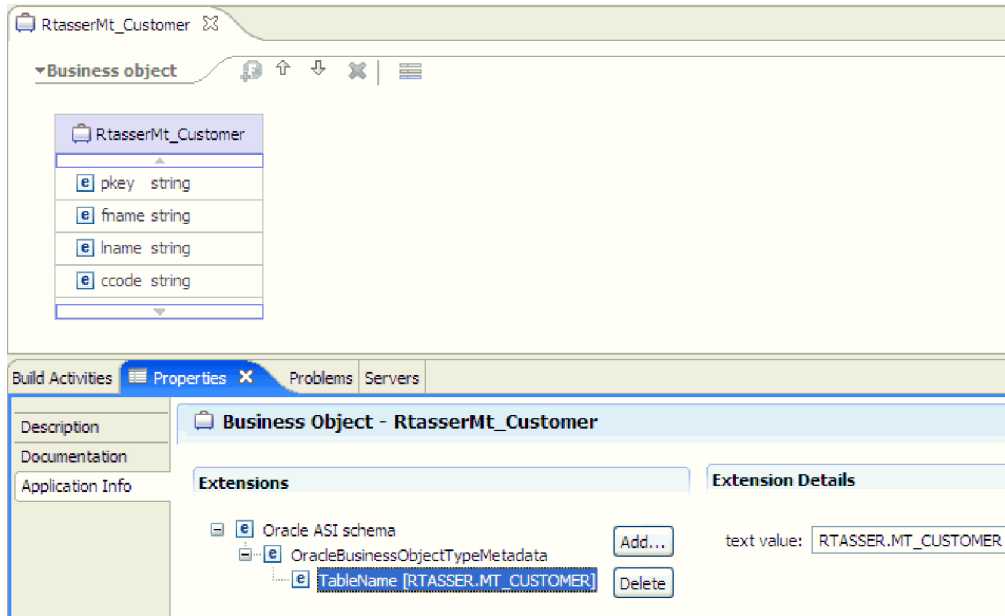


Figure 7. A table business object with no child.

Figure 8 shows a table business object that has one child table business object. The business object has simple attributes for each of the columns in the database table, plus a complex attribute pointing to a child business object.

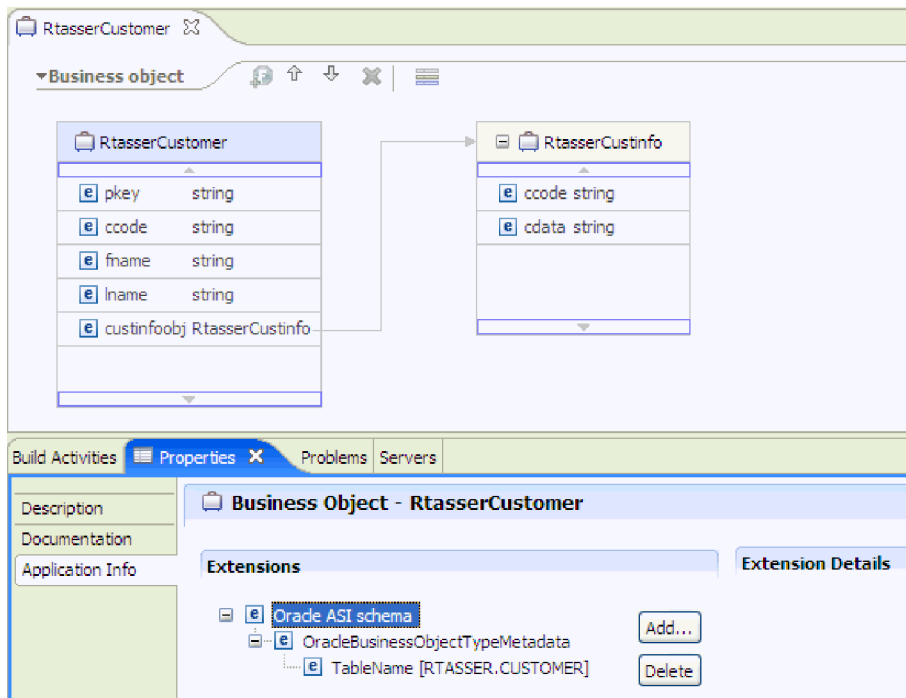


Figure 8. A table business object with one child business object.

For Oracle databases, the adapter supports complex data types such as array, table, structure, or nested structure in table business objects. The type name and the sub attribute details are automatically discovered and displayed for these types. The adapter processes these data types as child business objects of the table business

object.

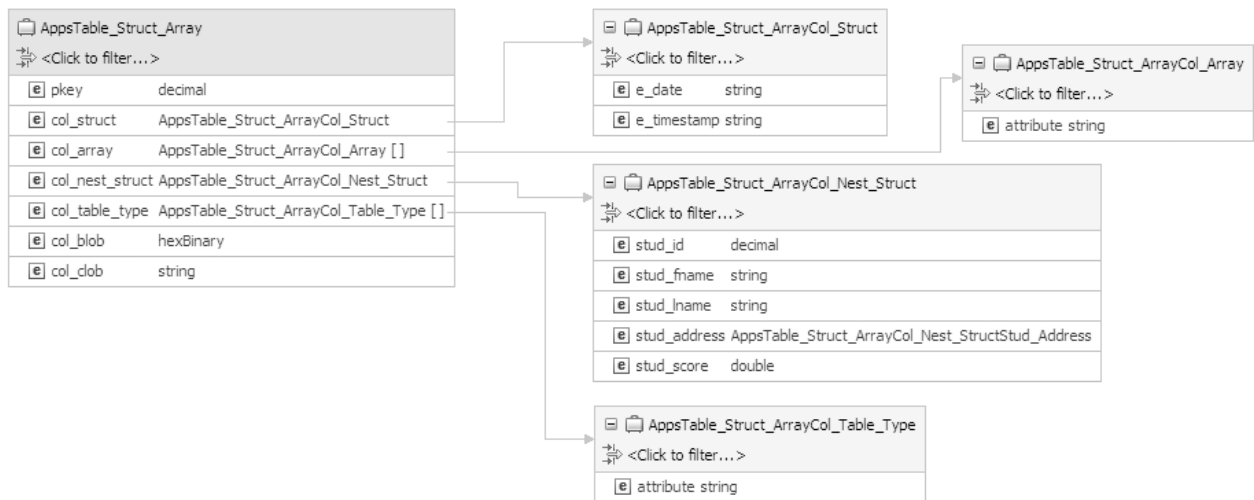


Figure 9. An Oracle table business object having complex data types as columns

For stored procedure and stored function business objects

In a business object for a stored procedure or stored function, all the input and output parameters for the stored procedure or stored function have corresponding attributes in the business object. If any of the input or output parameters is of a complex type, such as an array or structure, then the corresponding business object attribute is a child business object type with the child business object containing the attributes of the array or structure. If the stored procedure returns a result set, a child business object is created that contains the attributes of the returned result set.

The business object for stored procedures and stored functions supports the Execute outbound operation.

If the stored procedure or function business object contains the Date and Timestamp data types, the format of these types can be customized in the **Application Info** section of the **Properties** view of the business object. For example, you can specify the input or output parameter of date in dd/MM/yy and timestamp in HH/mm/ss format. If you want to customize the format of the Date and Timestamp data types, ensure that the data types are mapped to the default string data type in the **Configuration Properties** window.

The following Properties view shows business objects generated from a stored procedure that has one input type and two output types. One of the output parameters is of the Struct data type. The external service wizard generates a business object for the Struct type and adds it as a child object to the parent business object. For the attribute of type Struct in the parent business object, the ChildBOType application-specific information is set to Struct to indicate it is of type Struct.

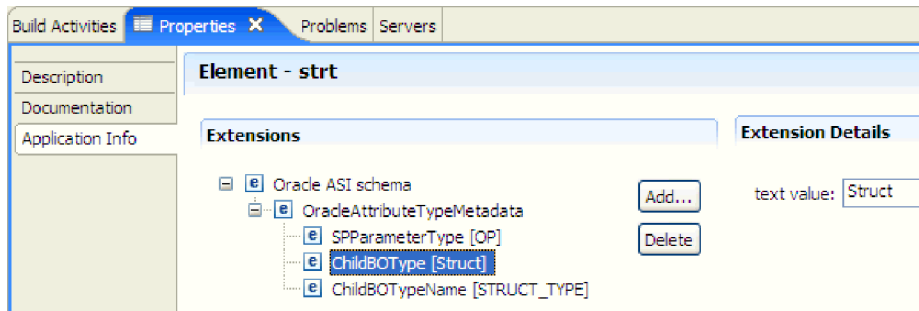


Figure 10. Child business object type with attribute of the structure data type

In the Properties view, the ChildBOTypeName application-specific information is set to the value of the user-defined Struct type in the database.

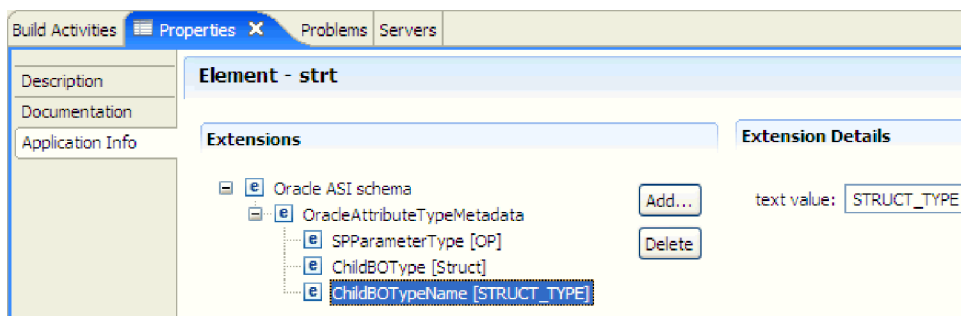


Figure 11. Child business object type name set to Struct type

For query business objects

A business object for a database query defines the SQL statement that performs the query and the parameters that the query requires. The query business object supports the RetrieveAll outbound operation.

As an example, assume a query business object to run the following SELECT statement:

```
select C.pkey, C.fname, A.city from customer C, address A
      WHERE (C.pkey = A.custid) AND (C.fname like ?)
```

The question mark (?) indicates an input parameter for the query. A query can have multiple parameters, each indicated in the SELECT statement by a question mark. Table 3 shows the attributes of the sample query business object. The query business object has simple attributes for each column to be extracted, a simple attribute for each parameter, and a “placeholder object” for the WHERE clause of the query, which holds the WHERE clause after parameter substitution.

Table 3. Attributes of a query business object

Business object attribute	Description
pkey	Corresponds to database column PKEY in the Customer table
fname	Corresponds to database column FNAME in the Customer table
city	Corresponds to database column CITY in the Address table

Table 3. Attributes of a query business object (continued)

Business object attribute	Description
parameter1	The parameter. There is one parameter for each ? (question mark) in the SELECT statement. In a SELECT statement with multiple parameters, subsequent parameters are named parameter2, parameter3, and so on.
jdbcwhereclause	A placeholder object for the WHERE clause

If the query business object contains the Date and Timestamp data types, the format of these types can be customized in the **Application Info** section of the **Properties** view of the business object. For example, you can specify the input or output parameter of date in dd/MM/yy and timestamp in HH/mm/ss format. If you want to customize the format of the Date and Timestamp data types, ensure that the data types are mapped to the default string data type in the **Configuration Properties** window.

The following figure shows the business object for the sample query in the business object editor.

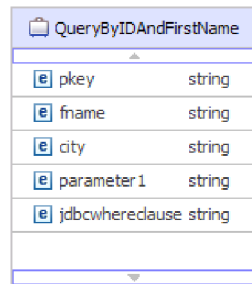


Figure 12. The attributes of a query business object

This figure shows the application-specific information for the query business object example. The SelectStatement application-specific information contains the SELECT statement.

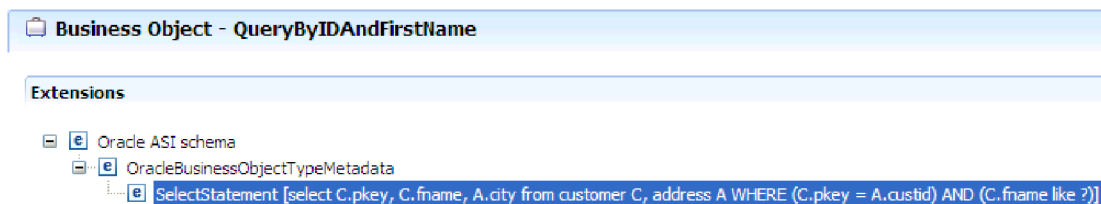


Figure 13. The SELECT statement is saved in the business object application-specific information

For Oracle databases, the adapter supports complex data types such as array, table, structure, or nested structure in the query result of the business object. The adapter does not support these complex types as parameters in query business objects.

Business graphs

You can optionally choose, during adapter configuration, to generate a business graph. In version 7.5, business graphs are required only in these situations:

- If you need to use the outbound ApplyChanges operation
- When adding business objects to a module created with a version of IBM Integration Designer earlier than version 6.1.0

If business graphs exist, they are processed, but the verb is ignored for all operations except ApplyChanges.

How business objects are created

You create business objects by using the external service wizard, launched from IBM Integration Designer. The wizard connects to the database, discovers database objects, and displays them to you. Select the database objects for which you want to create business objects. For example, specify which schemas you want to examine. In those schemas, select tables, views, stored procedures and functions, and synonyms and nicknames. In addition, you can create additional business objects. For example, you can create a business object to represent the results of user-defined SELECT, INSERT, UPDATE, or DELETE statements that are run against the database. The wizard helps you build a hierarchy of business objects, using parent-child relationships.

After you specify the business objects that you want and define the hierarchy of those objects, the wizard generates business objects to represent the objects that you have selected. It also generates other artifacts needed by the adapter.

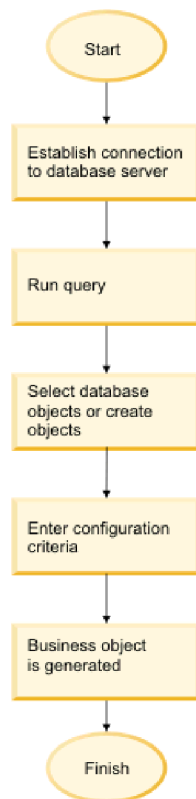


Figure 14. How business objects are created

In some instances, the wizard cannot completely configure a parent-child relationship. For these relationships, you use the business objects editor, launched from IBM Integration Designer, to modify or complete the definition of a business object hierarchy that was created by the wizard. For more information, see the

instructions for using the business object editor to modify business objects in the IBM Integration Designer information center at the following link:
<http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp>.

Related reference

“Business object information” on page 207

A business object is a structure that contains application-specific information (metadata) about how the adapter must process the business object as well as the operation to be performed on the business object. The name of the business object is generated by the external service wizard in accordance with the naming convention for the adapter.

Business object hierarchies

Define the relationships between database tables using parent-child relationships and data ownership in hierarchical business objects.

Business objects can either be flat or hierarchical. In a flat business object, all attributes are simple and represent one row in the database table. Hierarchies can contain related or unrelated business objects. Related business objects have parent-child relationships, with or without ownership.

The term *hierarchical* business object refers to a complete business object, including all the child business objects that it contains at any level. The term *individual* business object refers to one business object, independent of child business objects that it might contain or parent business objects that contain it. The individual business object can represent a view that spans multiple database tables. The term *top-level* business object refers to the individual business object at the top of the hierarchy, which does not itself have a parent business object.

A hierarchical business object has attributes that represent a child business object, an array of child business objects, or a combination of the two. In turn, each child business object can contain a child business object or an array of child business objects, and so on.

A *single-cardinality relationship* occurs when an attribute in a parent business object represents one child business object. In this case, the attribute is of the same type as the child business object. The adapter supports single-cardinality relationships, and single-cardinality relationships and data without ownership.

A *multiple-cardinality relationship* occurs when an attribute in the parent business object represents an array of child business objects. In this case, the attribute is of the same type as the child business objects.

Use the following types of relationships between business objects to define a hierarchy that represents your database tables:

- Single-cardinality relationships
- Single-cardinality relationships and data without ownership
- Multiple-cardinality relationships
- Child business objects with multiple parents

In each type of cardinality, the relationship between the parent and child business objects is described by the application-specific information of the key attributes in the business object storing the relationship.

Related tasks

“Selecting and configuring tables, views, and synonyms or nicknames for outbound processing” on page 106

Select and configure business objects for tables, views, and synonyms or nicknames to be used in your module. For inbound processing, these are the business objects that are delivered in events.

“Configuring the module for inbound processing” on page 133

To configure a module to use the adapter for inbound processing, use the external service wizard in IBM Integration Designer to find and select business objects and services from the Oracle database, and to generate business object definitions and related artifacts.

“Selecting and configuring tables, views, and synonyms or nicknames for inbound processing” on page 137

Select and configure business objects for tables, views, and synonyms or nicknames for use in your module. For inbound processing, the tables, views, and synonyms are the business objects that are delivered in events.

Related reference

“Business object attributes” on page 207

Business object attributes define the content of a business object and are built from the list of columns in the database object.

Single-cardinality relationships in business objects:

In a single-cardinality relationship, an attribute in a parent business object represents one child business object. In this case, the attribute is of the same type as the child business object. The adapter supports single-cardinality relationships, and single-cardinality relationships and data without ownership.

Single-cardinality relationships

Typically, a business object that contains a single-cardinality child business object has at least two attributes that represent the relationship. The type of one attribute is the same as the child's type. The other attribute is a simple attribute that contains the child's primary key as a foreign key in the parent. The parent has as many foreign key attributes as the child has primary key attributes.

Figure 15 illustrates a typical single-cardinality relationship. In the example, FKey in the ParentBOName object is the simple attribute that contains the child's primary key, and Child(1), also in the ParentBOName object, is the attribute that represents the child business object.

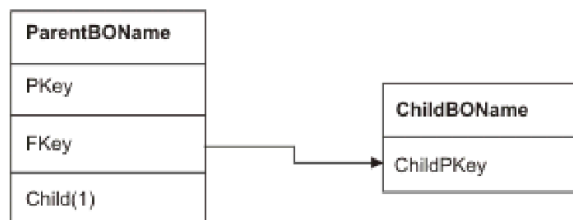


Figure 15. Typical single-cardinality relationship

Because the foreign keys that establish the relationship are stored in the parent, each parent can contain only one child business object of a given type.

A parent business object can have a single-cardinality child with ownership and a single-cardinality child without ownership. Lookup tables are used for relationships without ownership. Ownership is indicated by the value of the Ownership application-specific information.

Single-cardinality relationships and data without ownership

Typically, each parent business object owns the data within the child business object that it contains. For example, if each Customer business object contains one Address business object, when a new customer is created, a new row is inserted into both the Customer and Address tables. The new address is unique to the new customer. Likewise, when deleting a customer from the Customer table, the customer's address is also deleted from the Address table.

However, situations can occur in which multiple hierarchical business objects contain the same data, which none of them owns. For example, assume that the Address database table contains a reference to the StateProvince lookup table. Because the lookup table is rarely updated and is maintained independently of the address data, creating or modifying address data does not affect the state and province data in the lookup table. However, to be able to retrieve the StateProvince business object along with the Address business object, StateProvince must be a single-cardinality child of Address and the relationship must be defined without data ownership.

If your database design includes lookup tables, your business object design differs slightly from the database design. This is because the adapter retrieves data only for a table business object and its child table business objects. To use a lookup table, you need to create a single-cardinality parent-child relationship between the tables, without ownership. Although the StateProvince lookup table is not a child of the Address table in the database, the corresponding StateProvince business object is a single-cardinality child of the Address table business object because each address contains a single state or province. However, the Address business object does not "own" the StateProvince business object. Changes to an address do not result in a change to the list of states and provinces.

When the adapter receives a hierarchical business object with a Create, Delete, or Update request, the adapter does not create, delete, or update single-cardinality child business objects contained without ownership. The adapter performs only Retrieve operations on these business objects. If the adapter fails to retrieve such a single-cardinality business object, it returns an error and stops processing; it does not add or change values in the lookup table's business object.

Denormalized data and data without ownership

In addition to facilitating the use of static lookup tables, containment without ownership provides another capability: synchronizing normalized and denormalized data.

Synchronization of normalized to denormalized data: When the relationship is without ownership, you can create or change data when you synchronize from a normalized application to a denormalized one. For example, assume that a normalized source application stores data in two tables, A and B. Assume further

that the denormalized destination application stores all the data in one table such that each entity A redundantly stores B data.

In this example, to synchronize a change in table B data from the source application to the destination application, you must trigger a table A event whenever table B data changes. In addition, because table B data is stored redundantly in table A, you must send a business object for each row in table A that contains the changed data from table B.

Note: When making updates to denormalized tables, ensure that each record has a unique key so that multiple rows are not modified as a result of one update. If such a key does not exist, the adapter provides an error stating that multiple records have been updated.

Synchronization of denormalized to normalized data: When synchronizing data from a denormalized source application to a normalized destination application, the adapter does not create, delete, or update data contained without ownership in the normalized application.

When synchronizing data to a normalized application, the adapter ignores all single-cardinality children contained without ownership. To create, remove, or modify such child data, you must process the data manually.

Multiple-cardinality relationships:

In a multiple-cardinality relationship, an attribute in the parent business object represents an array of child business objects. The attribute is of the same type as the child business object. The foreign key that describes the relationship is stored in the child, except when an application stores a single-child entity. Then the parent-child relationship is stored in the parent.

Typically, a business object that contains an array of child business objects has only one attribute that represents the relationship, and this attribute is normally the primary key. The type of the attribute is an array of the same type as the child business objects. For a parent to contain more than one child, the foreign keys that establish the relationship are stored in the child.

Therefore, each child has at least one simple attribute that contains the parent's primary key as a foreign key. The child has as many foreign key attributes as the parent has primary key attributes.

Because the foreign keys that establish the relationship are stored in the child, each parent can have zero or more children.

Figure 16 on page 37 illustrates a multiple-cardinality relationship. In the example, ParentId in the three ChildBOName boxes is the simple attribute that contains the parent's primary key, and Child1 in the ParentBOName box is that attribute that represents the array of child business objects.

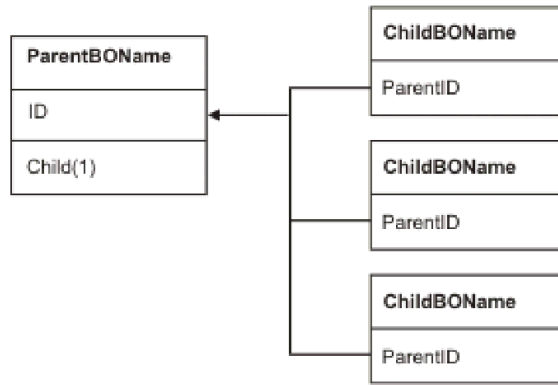


Figure 16. Multiple cardinality business object relationship with N>1

A multiple-cardinality relationship can be an N=1 relationship. Some applications store one child entity so that the parent-child relationship is stored in the child rather than in the parent. In other words, the child contains a foreign key whose value is identical to the value stored in the parent's primary key.

Applications use this type of relationship when child data does not exist independently of its parent and can be accessed only through its parent. Such child data requires that the parent and its primary key value exist before the child and its foreign key value can be created.

Figure 17 shows this type of relationship.

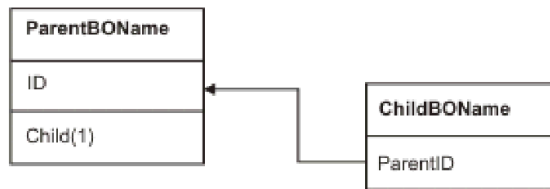


Figure 17. Multiple cardinality relationship with N=1

Database tables with multiple parent tables:

If a child table in the database has more than one parent table, you must manually configure the additional parent business objects using the assembly editor. The external service wizard configures only one parent.

The business object schema

The business object schema is built out of database objects that you select when you run the external service wizard. Each database object translates into a top-level business object.

The schema defines business objects names and application-specific information. The business objects and their attributes and application-specific information are represented in the schema as follows:

- The business object maps to a complex type definition.
- The application-specific information for the business object is contained in annotations at the complex type.
- The attributes of the business object map to element type definitions.
- The application-specific information for each property in the business object is contained in annotations for the element types.

The template for the application-specific properties for the business object and for the attributes is defined in the metadata schema for the adapter. The name of the schema file is `OracleEBSASI.xsd`. The schema file generated for the adapter has a reference to this template in its annotations.

Tables, views, and synonyms overview

The Oracle database provides the common database objects such as tables, views, synonyms, and so on.

Table is a common database schema object in a database. The table objects are used to store data in the database. Oracle provides a number of table types for entering data into the database such as object, normal, nested, clustered, index-organized, and so on. Data can be entered into the database table either directly or through the Oracle open interface tables. The Oracle open interface tables interface between two systems. These tables import data from one system, validates data, identifies errors if any, and transfers the valid data to the Oracle base tables appropriately.

View is a database object that is considered as a virtual table or a stored query. Views do not store the data physically in them; instead they derive the data from the base table on which they are created. All the operations that are performed on the tables can be performed on views too.

Synonym is an alias or an alternative name given for the database objects such as the tables, views, and other database objects.

Stored procedure overview

A stored procedure can be a business object that your module runs with the Execute operation. It can run in place of the standard SQL for an operation on any business object, or it can perform additional actions before or after performing an operation.

A stored procedure is a group of SQL statements that form a logical unit and perform a particular task. A stored procedure encapsulates a set of operations or queries for the adapter to run on an object in a database server. The adapter uses stored procedures in the following ways:

- By creating a stored procedure business object to run against your database
- By enhancing a business object's operations by replacing the SQL statement provided for a business object's operation or by performing actions before or after the operation runs

Stored procedure business object overview

You can create a stored procedure business object that corresponds to a stored procedure or stored function in the database. You can then use the Execute operation to run the stored procedure against the data in the database.

The external service wizard helps you to build stored procedure business objects that run a stored procedure or stored function. The wizard examines the stored procedure or stored function in the database to create the business object. A stored procedure business object has an attribute for each parameter.

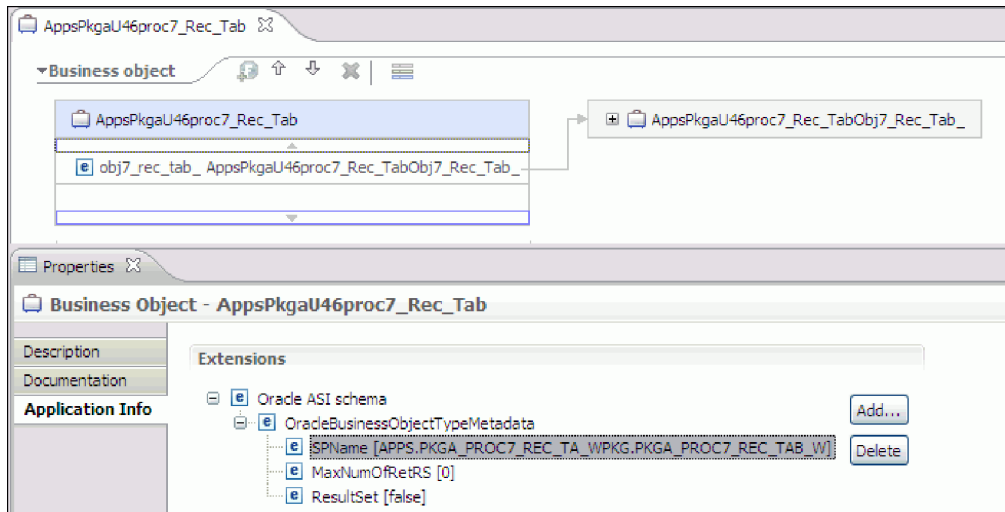
For validating the stored procedure attributes, a sample value parameter is provided with each attribute. The sample value parameter is provided for both simple and complex data type attributes. The wizard uses the sample values that you provide to validate the stored procedure before saving it. The adapter uses the result that the stored procedure returns to validate the parameters, to obtain the maximum number of result sets returned, and to use the metadata of these result sets to generate child business objects. The wizard generates the hierarchy for stored procedure business objects automatically if you validate the stored procedure business object.

For both simple and complex data type attributes, the adapter automatically discovers and displays the data type and type name for each attribute of the stored procedure. If the stored procedure has input/output parameters or returns value parameters that are of complex data types such as the Struct or Array, the data type and the corresponding user-defined type name are saved in the `SPComplexParameterTypeName` property.

If the stored procedure contains the Oracle PL/SQL data type such as Record, the adapter creates a wrapper package with a wrapper stored procedure which converts the Record data type to Object data type so that the Oracle E-Business Suite adapter can support the Oracle PL/SQL Record data type. The names of the wrapper packages and stored procedures created for this purpose comply with the Oracle database object naming conventions along with the appropriate suffixes to differentiate them from the Oracle database object names. The names of the wrapper package and wrapper stored procedure consists of both the original package and stored procedure names along with the appropriate suffixes such as “_WPKG” and “_W” (for example, `PKG_A_PROC7_REC_TAB_WPKG.PKG_A_PROC7_REC_TAB_W`, where “`PKG_A`” is the original package name, “`PROC7_REC_TAB`” is the original stored procedure name, “_WPKG” is the suffix for the package, and “_W” is the suffix for stored procedure). For each selected overloaded stored procedure and function which has PL/SQL Record type parameters, the adapter will create wrapper stored procedure in specific wrapper package with two different number tag (for example, “`XXXX_WPKG01` and `XXXX_WPKG02`”).

In the generated Wrapper for SP/SF with Record type parameter, if you select the boolean type parameter as TRUE or False during runtime, the adapter processes this datatype automatically.

WebSphere Adapter for Oracle E-Business Suite distinguishes the original SP/SF from the overloaded ones by a number tag that corresponds to an overload sequence in the Oracle database. These corresponding parameters for the selected SP/SF will be added based on the overload sequence.



If the stored procedure returns a result set, you need to set the number of result sets returned from this stored procedure in the `MaxNumberOfResultSets` property. This value represents the maximum number of result sets that are handled by the adapter run time.

During discovery and at run time, the WebSphere Adapter for Oracle E-Business Suite expects the returned result set from the stored procedure execution to contain columns with names. Some stored procedures return result set with unnamed columns. For example, a stored procedure with the SQL statements like the examples that follow return result set with unnamed columns:

```
SELECT COUNT(*) FROM EMPLOYEE;
SELECT 111,222,333 FROM CUSTOMER;
```

Oracle processes such SQL SELECT statements by assigning "dummy" names to the table columns in the returned result set- like `count(*)` or `d1`, `d2`, `d3` for the respective select statement examples shown above.

If the returned result set contains table columns with no names (because the database did not assign dummy names), the adapter creates dummy names for such columns.

Dummy column names, generated by either the database or by the adapter, are assigned to the attributes of the stored procedure business object.

The behavior (by the adapter or by the database) of assigning dummy names to unnamed table columns ensures that the stored procedure runs successfully during discovery and at run time.

For stored procedure business objects, the wizard supports nested Struct and Array objects, and can support any number of layers of nested hierarchy. The wizard can generate corresponding child business objects for all these nested Struct and Array objects.

Table 4. Complex data type properties for stored procedure business objects

Property name	Type	Description
SPComplexParameterType	String	Value can be one of: Array ResultSet Struct
SPComplexParameterTypeName	String	The name of the user-defined type. This property is required when the value of SPComplexParameterType is Struct or Array.
MaxNumberOfResultSets	Integer	The maximum number of returned result sets to be handled by the Adapter for Oracle E-Business Suite run time.

Stored procedures used in place of or in addition to operations

You can specify that the adapter use a stored procedure in the database in place of, before, or after the SQL statements that the adapter uses to perform an operation. Each business object can have a different set of stored procedures used with each operation.

The adapter can use simple SQL statements for Create, Update, Delete, Retrieve, or RetrieveAll operations. The column names used in the SQL statements are derived from an attribute application-specific information. The WHERE clause is constructed using key values specified in the business object. Each query spans one table only, unless posted to a view. However, you can replace or enhance the SQL statement provided by the adapter using stored procedures and stored functions.

WebSphere Adapter for Oracle E-Business Suite distinguishes the original SP/SF from the overloaded ones by a number tag that corresponds to an overload sequence in the Oracle database. These corresponding parameters for the selected SP/SF will be added based on the overload sequence.

The adapter can call a stored procedure or stored function in the following circumstances:

- Before processing a business object, to perform preparatory operational processes
- After processing a business object, to perform actions after the operation
- To perform a set of operations on a business object, instead of using a simple Create, Update, Delete, Retrieve, or RetrieveAll statement.

In a hierarchical business object, if you want the stored procedure to be performed for each business object in the hierarchy, you must separately associate a stored procedure with the top-level business object and each child business object or an array of business objects. If you associate a stored procedure with the top-level business object but do not associate it with each child business object, then the top-level business object is processed with the stored procedure, but the child business objects are processed using the standard SQL query.

Table 5 on page 42 lists the application-specific information elements for a stored procedure and describes their purpose and use. A complete description of each element is provided in the sections that follow the table. A screen showing the

stored procedure definition for a business object is shown in “View of business object with stored procedure definition” on page 45.

Table 5. Application-specific information for stored procedures in table and view business objects

Descriptive name	Element name	Purpose
Stored procedure type	StoredProcedureType	The stored procedure type defines the type of stored procedure to be used, and this determines when the stored procedure is called, for example, before processing a business object.
Stored procedure name	StoredProcedureName	The name of the stored procedure that is associated with the appropriate StoredProcedureType.
Result set	ResultSet	This value specifies whether the stored procedure returns a result set. If the result set is returned, a multiple-cardinality child for the current business object is created using the values returned in the result set rows.
Parameters	Parameters	Each Parameters element describes one parameter for a stored procedure or stored function.
Return value	ReturnValue	A value that indicates it is a function call, not a procedure call, because the value is returned by the stored procedure.

Stored procedure type

The stored procedure type defines the type of stored procedure to be used, and this determines when the stored procedure is called, for example, before processing a business object.

Table 6. Stored procedure type element characteristics

Required	Yes
Default	None
Possible values	Can be one of: <ul style="list-style-type: none"> • BeforeOperationSP • AfterOperationSP • OperationSP <i>Operation</i> specifies one of the operation names: Create, Update, Delete, Retrieve, or RetrieveAll.
Bidirectional transformation supported	No
Property type	String

Table 6. Stored procedure type element characteristics (continued)

Usage notes	<p>Stored procedure types associated with RetrieveAll apply to top-level business objects only.</p> <p>You can remove any selected application-specific information from the StoredProcedureType property. All the corresponding operation application-specific information property groups are also removed.</p>
Examples	<ul style="list-style-type: none"> • CreateSP: Performs the create operation • UpdateSP: Performs the update operation • BeforeCreateSP: Runs before creating a business object • AfterCreateSP: Runs after creating a business object • AfterDeleteSP: Runs after deleting a business object

Stored procedure name

The name of the stored procedure that is associated with the appropriate StoredProcedureType.

Table 7. Stored procedure name element characteristics

Required	Yes
Default	None
Bidirectional transformation supported	Yes
Property type	String

Result set

This value determines whether the stored procedure returns a result or not. If the result set is returned, a multiple-cardinality child for the current business object is created using the values returned in the result set rows.

Table 8. Result set element characteristics

Required	Yes
Default	None
Possible values	True False
Bidirectional transformation supported	No
Property type	Boolean
Usage notes	<p>If your stored procedure returns a result set, use the business object editor after finishing the external service wizard to verify that this attribute is set to true. The Oracle JDBC driver does not always return this value correctly.</p>

Parameters

There is one Parameters element for each parameter for a stored procedure or stored function. Each Parameters element defines the name and type of one parameter.

Table 9. Parameters element characteristics

Required	Yes
Default	None
Contents	Each Parameters element specifies the following information: <ul style="list-style-type: none">• PropertyName: Specifies the name of the business object attribute to pass as the parameter.• Type: Specifies the type of the parameter, one of the following values:<ul style="list-style-type: none">– IP for input only– OP for output only– IO for input and output– RS for result set
Bidirectional transformation supported	No
Property type	String
Usage notes	A result set can be returned only as an output parameter. In that case, one of the parameters must have the type RS, to indicate a result set.

Return value

A value that indicates it is a function call, not a procedure call, because a value is returned.

Table 10. Return value element characteristics

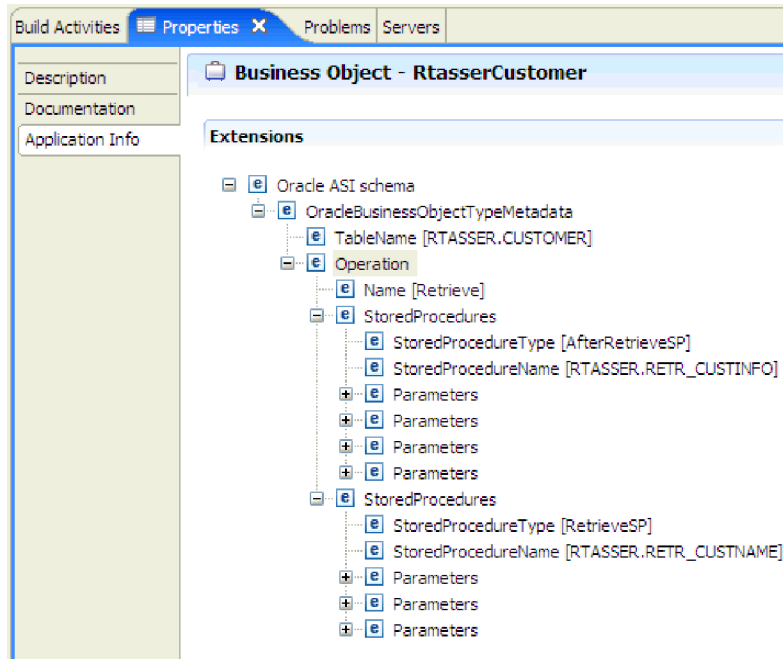
Required	No
Default	None
Possible values	Can be RS or the name of a business object attribute or child business object.
Bidirectional transformation supported	No
Property type	String

Table 10. Return value element characteristics (continued)

Usage notes	<p>If the returned value is RS, the returned value is a result set and is used to create the multiple-cardinality container corresponding to this business object. If the returned value is the name of an attribute, the value is assigned to that particular attribute in the business object. If the attribute is another child business object, the adapter returns an error.</p> <p>When you associate a stored procedure with a business object that is generated from a table or view, and if the stored procedure is a function, a value is returned from this stored procedure. One ReturnValue application-specific information value is added to the operation application-specific information. The existence of this application-specific information implies that it is a function call and not a procedure call, because a value is being returned by the function.</p> <p>If the value of this application-specific information is a business object attribute name, the returned value is assigned to that particular attribute in the business object.</p> <p>If the value of this application-specific information is another child business object, the adapter run time returns an error.</p> <p>In summary, if the returned value is of a simple data type, the wizard enables you to bind one business object attribute to it, and the value of this application-specific information is set to the name of that business object attribute. But if the returned value is a result set, the wizard sets the value of this application-specific information to RS.</p> <p>Note: A result set can be returned as an output parameter or as a returned value if it is a stored function. The type of the output parameter is set to RS to indicate that this parameter is used to return a result set.</p>
-------------	--

View of business object with stored procedure definition

The following Properties view screen shows the customer business object that has the associated stored procedure information for RetrieveSP and AfterRetrieveSP for the Retrieve operation. The adapter runs the RTASSER.RETR_CUSTNAME stored procedure in place of the standard SQL to retrieve a table business object. After the business object is retrieved, the adapter runs the RTASSER.RETR_CUSTINFO stored procedure.



Stored functions overview

The Oracle database supports stored functions in addition to stored procedures. Stored functions are like stored procedures except that they always return a value. The adapter supports them in a similar manner.

The adapter supports stored functions that a user creates with the CREATE FUNCTION statement. Although this type of function is sometimes called a *user-defined function* (UDF), that term more typically refers to a Java stored function or procedure, which the adapter does not support.

A function call has the following syntax:

```
? = call FunctionName parameter_list
```

Contrast this to a stored procedure call, which has the following syntax:

```
call SPName parameter_list
```

You specify the attribute that contains the returned value by using the ReturnValue business object application-specific information.

Query business object overview

Query business objects run a user-defined SELECT statement against the database and return the matching records in business objects.

The external service wizard helps you to build query business objects that run user-defined SELECT statements against the database. Specify the SELECT statement, using ? (the question mark) in place of any parameter that can be substituted in the SELECT statement. The wizard then provides an area where you can specify the data type of each parameter and provide a sample value. The sample value must match data in the database because wizard uses the SELECT statement results to create the query business object.

You must validate the configuration of the query in the wizard before saving it. When you validate, the wizard runs the SELECT statement using the sample values. After obtaining the result set, the wizard analyzes the metadata to obtain the column name and column type of all columns. For each column of the returned result set, the wizard generates one corresponding attribute in the query business object. For each parameter in the WHERE clause, the wizard generates one **jdbewhereclause** attribute in the query business object and sets this attribute's default value to be the WHERE clause. These attributes are used to generate one dynamic WHERE clause at run time to replace the default WHERE clause.

For example, assume that you have specified the following SELECT statement:

```
select * from customer where fname=? and age=?
```

The WHERE clause has two parameters, which are indicated by question marks (?). The first parameter has the data type **string**, to match the data type of the fname column. The second parameter has the data type **int**, matching the age column. If your database has a customer record where the fname column contains the string Mike and the age column contains the integer 27, you can specify those values as sample values when configuring the query business object. The wizard configures the business object to correspond to the returned result set.

You can also create query business objects to retrieve the interface status of the Oracle open interface tables. The interface status column in the interface table contains the error codes. These codes indicate if the data processed from the Oracle open interface table to the Oracle base table is successful or not. The error code can be used to troubleshoot the issue in case of failure during the data transfer.

Related tasks

“Discovering database objects” on page 101

After connecting to the database, run a query to search for database objects. Browse through the discovered objects navigation tree to understand the structure of objects in the Oracle database and use filters to display only the database objects you want to see. Define how many business objects you want to create for user-defined database queries.

“Selecting and configuring query business objects” on page 119

Select and configure query business objects for user-defined SELECT statements for use in your module.

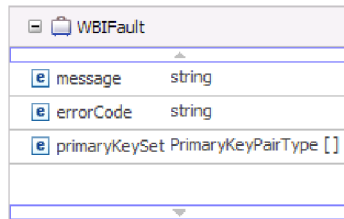
Business faults

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Although IBM Business Process Manager or WebSphere Enterprise Service Bus supports other types of faults, the adapter generates only business faults, which are simply called as *faults* in this documentation. Not all exceptions become faults. Faults are generated for errors that are actionable, that is, errors that can have a recovery action that does not require the termination of the application. For example, the adapter generates a fault when it receives a business object for outbound processing that does not contain the required data or when the adapter encounters certain errors during outbound processing.

Fault business objects

The external service wizard creates a business object for each fault that the adapter can generate. In addition, the wizard creates a WBIFault superset business object, which has information common to all faults, such as the message, errorCode, and primaryKeySet attributes as shown in Figure 18.



WBIFault	
message	string
errorCode	string
primaryKeySet	PrimaryKeyValuePair []

Figure 18. The structure of the WBIFault business object

Some faults contain the matchCount attribute, to provide additional information about the error. For others, WBIFault contains all the information needed to handle the fault.

Related reference

“Fault business objects” on page 221

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

XML gateway interface overview

XML Gateway is an interface type in Oracle E-Business Suite Integration Repository. It allows outbound integration with Oracle E-Business Suite; helps in transferring data to the application. The interface exposes the XML documents in the specified format that is widely used by the Oracle applications business objects and interfaces. It interacts with the Oracle interface tables and views, and it maps the underlying tables to XML format. The interface services contain a common standards-based approach for XML integration between Oracle E-Business Suite and other third-party applications (for example, Enterprise Application Integration and Web Service Integration).

IBM WebSphere Adapter for Oracle E-Business Suite supports the XML gateway interface, which integrates Oracle E-Business Suite with the other EIS by web service and JMS. When XML Gateway interface is selected, the adapter converts the payload from the Document Type Definitions (DTD) format you specify to the XSD format, and generates the request and response business objects.

The external service wizard helps you to create an XML-based integrated service document that can be used as a payload content to invoke the web service or JMS in runtime. After sending the business data to Oracle E-Business Suite by invoking the runtime method, you can use Oracle E-Business Suite's import request to import such business data. There are a few supporting terms used in this feature, which are described as follows:

Note: Oracle XML Gateway integration and Integration Repository metadata discovery is applicable for IBM Business Process Manager or WebSphere Enterprise Service Bus only.

- Payload is an XML document that contains the business information required for the business process in XML Gateway business integration.
- WSDL is a document written in XML. It contains the specifications of the web service, such as the location of the service and the operations or methods that the service uses (exposes).
- DTD defines the XML Gateway business data type for payload that contains a list of elements, attributes, entities, and notations. WebSphere Adapter for Oracle E-Business Suite will provide common translation from the DTDs into schema files. You should take care of the syntax gap between the DTD definition and schema files (XSD). The payload XML document, created based on the generated Schema files (XSD), contains the business data for business process in the XML Gateway interface.

Note: Currently, XML Gateway integration does not support iterative development.

Advanced Queue integration overview

Advanced Queue is an interface type in Oracle E-Business Suite. It allows outbound integration with Oracle E-Business Suite and helps in transferring data to the application. It also interacts with the Oracle advanced queue. The interface services contain a common standard-based approach for integration between Oracle E-Business Suite and other third-party applications (Example: Enterprise Application Integration).

IBM WebSphere Adapter for Oracle E-Business Suite supports the advanced queue feature, which integrates Oracle E-Business Suite with other EIS. When the advanced queue interface is selected, the adapter generates the advanced queue header and payload business objects. The external service wizard helps you to create the advanced queue header and payload business object to invoke outbound in the runtime.

Note: Oracle advanced queue integration is applicable for IBM Business Process Manager or WebSphere Enterprise Service Bus only.

WebSphere Adapter for Oracle E-Business Suite creates a dynamic SQL script, which uses some Oracle AQ PL/SQL API to enqueue or dequeue. After sending the business data to Oracle E-Business Suite by invoking the runtime method, you can use Oracle E-Business Suite's import request to import such business data.

There are few supporting terms used in this feature, which are described as follows:

- Queue Header contains the properties that are used in enqueue and dequeue operation. In the enterprise metadata discovery process, these properties are populated with the sample value based on the property type (Example: string, integer). The sample value with an incorrect property type will cause failure in the runtime. Example: Priority (int), Delay (int), and Correlation (string).
- Queue table Payload is an Oracle PL/SQL type that contains the business information required for the business process in advanced queue integration. The adapter will generate the BO for it after configuring the module for the advanced queue integration.
- Queue Name is the name of Oracle advanced queue, which will receive and send out the integration data.

Related tasks

“Configuring the module for advanced queue integration” on page 95
To configure the module for the advanced queue interface, use the external service wizard in Integration Designer.

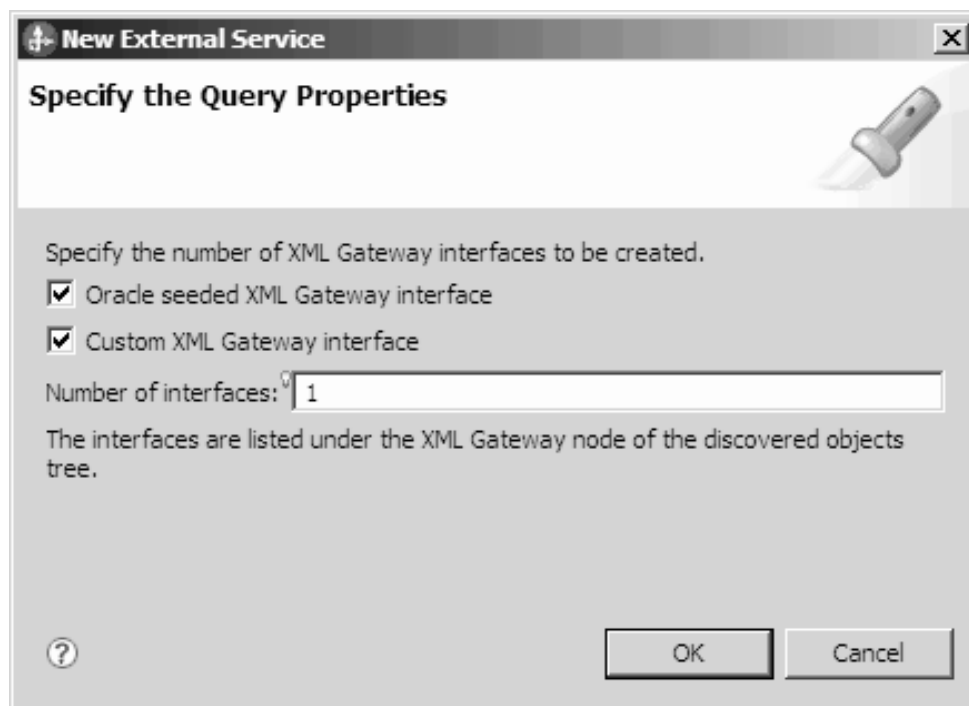
Support for metadata discovery for the supported interfaces

WebSphere Adapter for Oracle E-Business Suite provides a complete catalog of Oracle E-Business Suite business service interfaces, which can be invoked to get the structure and metadata of different business service interfaces.

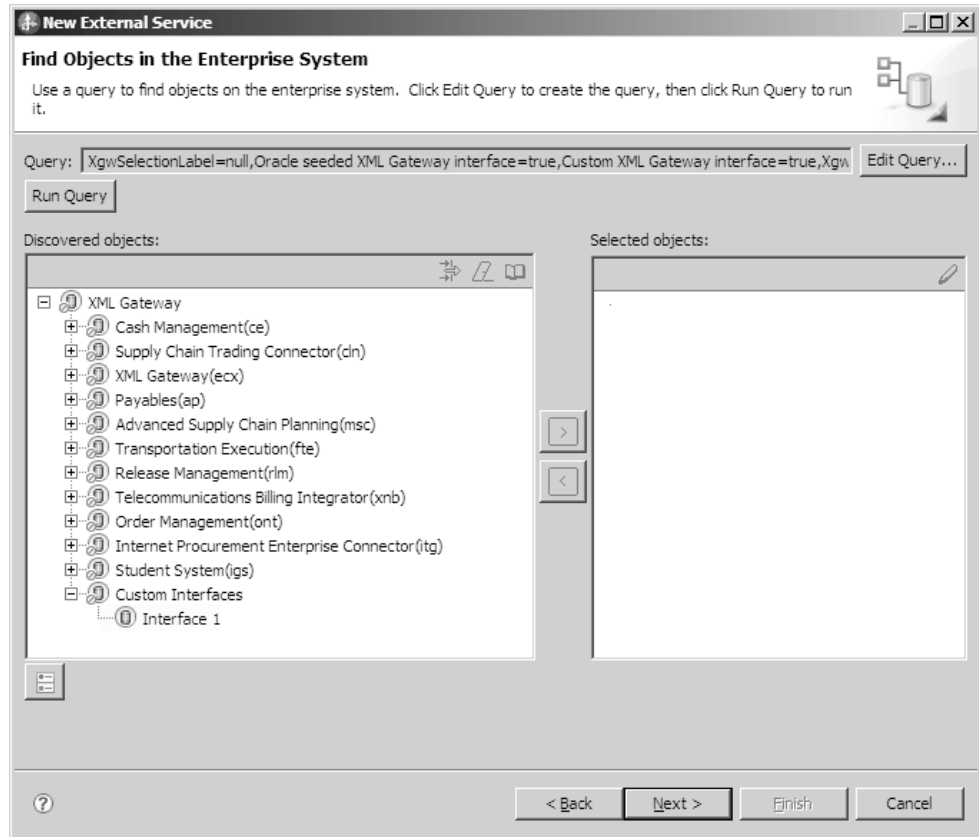
Oracle E-Business Suite exposes Integration Repository (IREP) as Web Service with a specific WSDL. Invoke Integration Repository (IREP) Web Service to receive the XML Gateway interface metadata. To get the structure and metadata of different business service interfaces:

Note: Oracle XML Gateway integration and Integration Repository metadata discovery is applicable for IBM Business Process Manager or WebSphere Enterprise Service Bus only.

- Select the Oracle E-Business Suite XML Gateway interface type, Web Service transport and specific EBS version and entering the required security credentials for connecting to the Oracle E-Business Suite.
- Select the Oracle seeded XML Gateway interface or custom XML Gateway interface or both in the Specify the Query Properties window.



- Click **OK** to save your changes to the query. In the Specify the Query Properties window, click **Run Query** to use the query to discover the objects and to create the interface. The result of running a typical query is shown in the following figure.



The Discovered objects pane lists the different business service interfaces supported by the adapter.

The external service wizard

Use the external service wizard in IBM Integration Designer to discover objects in the Oracle database of Oracle E-Business Suite, to generate query business objects, and to generate business objects from selected database objects. The wizard also generates the module and the service artifacts that enable the adapter to run as a Service Component Architecture (SCA) component.

Chapter 2. Planning for adapter implementation

Before using WebSphere Adapter for Oracle E-Business Suite, make sure that 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 configure and deploy the module, you must possess a thorough understanding of the business integration concepts, Java Database Connectivity (JDBC), the Oracle database knowledge, the Oracle E-Business Suite products in your environment, and the features and capabilities of IBM Integration Designer and IBM Business Process Manager or WebSphere Enterprise Service Bus.

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

- The business requirements of the solution you are building.
- Oracle E-Business Suite, JDBC, and the database products in your environment. This includes Oracle E-Business Suite integration interfaces, Oracle Business Event System, Oracle Workflow, Oracle data access issues, transactional models, and connections across heterogeneous relational databases, queues, and Web services.
- 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 such as the IBM Business Process Manager or WebSphere Enterprise Service Bus. You must 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 IBM Integration Designer. You must know how to use these tools to create modules, wire and test components, and complete other integration tasks.

Security

The adapter uses the J2C authentication data entry, or the authentication alias feature of Java 2 security to provide secure user name and password authentication. For more information about security features, see the documentation for IBM Business Process Manager or WebSphere Enterprise Service Bus.

Support for protecting sensitive user data in log and trace files

You can configure the adapter to prevent sensitive or confidential data, in the log and trace files, from being viewed by users without authorization.

Log and trace files for the adapter can contain data from your Oracle database, which might contain sensitive or confidential information. Sometimes these files

might be seen by individuals without authorization to view sensitive data. For example, a support specialist must use the log and trace files to troubleshoot a problem.

To protect the data in such situations, the adapter lets you specify whether you want to prevent confidential user data from displaying in the adapter log and trace files. You can select this option in the external service wizard or change the `HideConfidentialTrace` property. When this property is enabled, the adapter replaces the sensitive data with XXX's.

See “Managed connection factory properties” on page 230 for information about this optional property.

The following types of information are considered potentially sensitive data and are disguised:

- The contents of a business object
- The contents of the object key of the event record
- User name, Password, Environment, and Role
- The URL used to connect to the Oracle database

The following types of information are not considered user data and are not hidden:

- The contents of the event record that are not part of the event record object key, for example, the `XID`, event ID, business object name, and event status
- Business object schemas
- Transaction IDs
- Call sequences

User authentication

The adapter supports several methods for supplying the user name and password that are needed to connect to the Oracle database. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

To integrate an adapter into your application, a user name and password are needed at the following times:

- When the external service wizard connects to Oracle database to extract, or *discover*, information about the objects and services that you can access with the adapter.
- At run time on IBM Business Process Manager or WebSphere Enterprise Service Bus, when the adapter connects to Oracle database to process outbound requests and inbound events.

Authentication in the wizard

The external service wizard prompts for the connection information for the discovery process, and then reuses it as the default values of the adapter properties that specify the connection information used at run time. You can use a different user name and password while running the wizard than you use when the application is deployed to the server. You can even connect to a different Oracle database, although the schema name must be the same in both databases. For example, while developing and integrating an application that uses WebSphere

Adapter for Oracle E-Business Suite, you might not use the production database; using a test database with the same data format but fewer, simulated records lets you develop and integrate the application without impacting the performance of a production database and without encountering restrictions caused by the privacy requirements for customer data.

The wizard uses the user name and password that you specify for the discovery process only during the discovery process; they are not accessible after the wizard is completed.

Authentication at run time

At run time, the adapter needs to provide the user name and password to connect to the Oracle database. To connect without user intervention, the adapter must access a saved copy of the user information. In a server environment, there are several methods for saving user information. You can configure the adapter to get your user information, through any of the following methods:

- Adapter properties
- Data source
- J2C authentication alias

Saving the user name and password in adapter properties is a direct way to provide this information at run time. You provide this user name and password when you use the external service wizard to configure your module. Although directly specifying the user name and password seems the most straightforward method, it has important limitations. Adapter properties are not encrypted; the password is stored as clear text in fields that are accessible to others on the server. Also, when the password changes, you must update the password in all instances of the adapter that access that Oracle database. This includes the adapters embedded in application EAR files as well as adapters that are separately installed on the server.

Using a data source lets you use a connection already established for another application. For example, if multiple applications access the same database with the same user name and password, the applications can be deployed using the same data source. The user name and password can be known only to the first person who deploys an application to that data source or who defines a data source separately.

Using a J2C authentication data entry, or authentication alias, created with the Java Authentication and Authorization Service (JAAS) feature of Java 2 security is a robust, secure way to deploy applications. An administrator creates the authentication alias that is used by one or more applications that need to access a system. The user name and password must be known only to that administrator, who can change the password in a single place, when a change is required.

Related tasks

“Creating an authentication alias” on page 80

An authentication alias is a feature that encrypts the password used by the adapter to access the Oracle database. The adapter can use it to connect to the Oracle database instead of using a user ID and password stored in an adapter property.

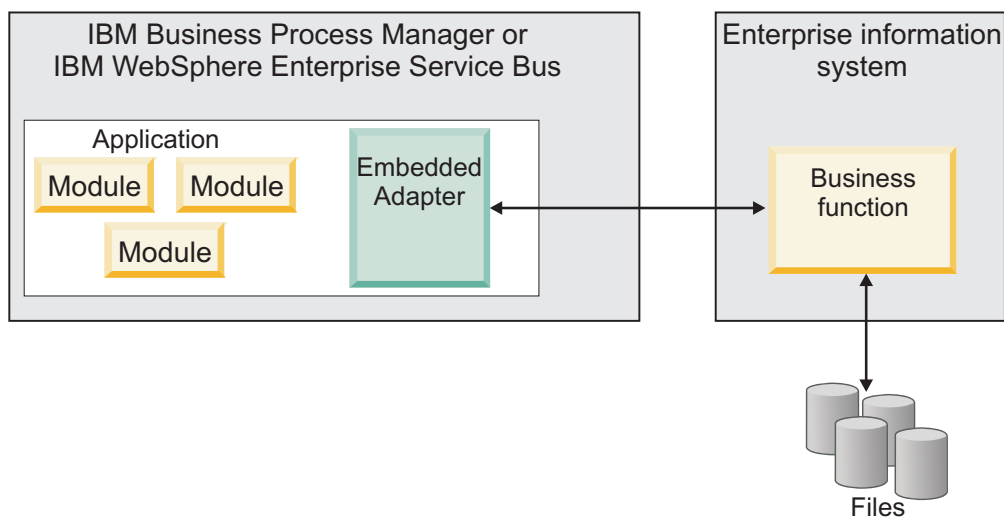
Deployment options

There are two ways to deploy the adapter. You can either embed it as part of the deployed application, or you can deploy it as a stand-alone RAR file. The requirements of your environment affect the type of deployment option you choose.

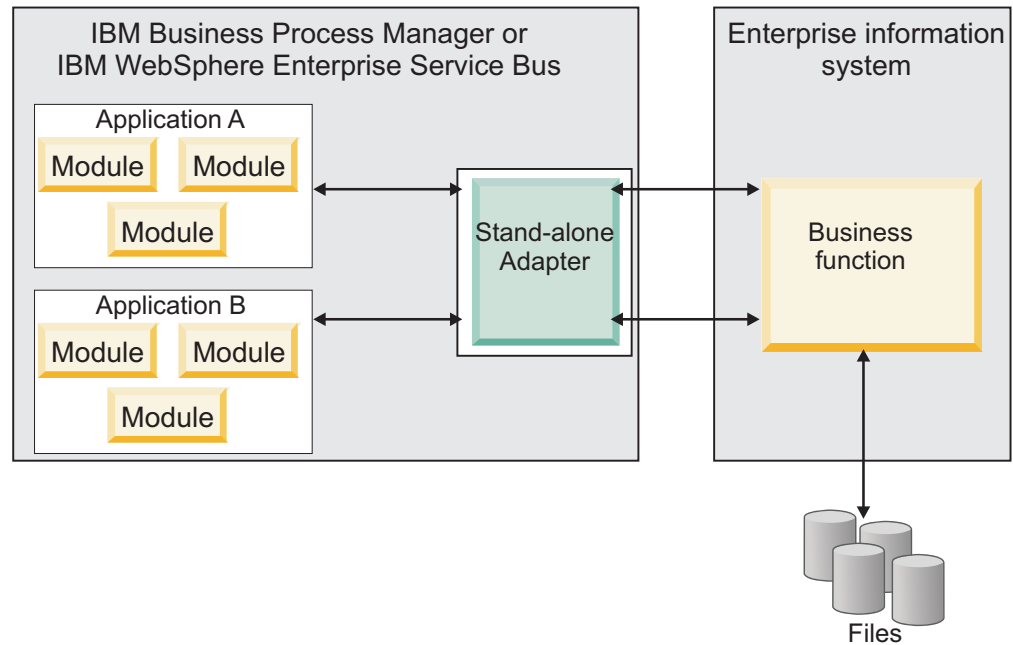
The following are the deployment options:

- **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 IBM Integration Designer, you can choose how to package the adapter [either bundled with the (EAR) file or as a stand-alone (RAR) file]. Your choice affects how the adapter is used in the run time 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 to run 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

Consider the following items if you plan to embed 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 run time environments. *Class loader isolation* means that 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

Consider the following items if you plan to use 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.
- Adapter Foundation Classes (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 class path in a stand-alone adapter, the one that is used is random; therefore, they all must be the latest version.

Note:

When you install multiple adapters with different versions of CWYBS_AdapterFoundation.jar, and if a lower version of the CWYBS_AdapterFoundation.jar is loaded during runtime, the adapter will return the ResourceAdapterInternalException error message, due to a version conflict. For example, when you install Oracle E-Business Suite adapter version 7.0.0.3 and WebSphere Adapter for Oracle E-Business Suite version 7.5, the following error message is displayed: IBM WebSphere Adapter for Oracle E-Business Suite has loaded file:/C:/IBM/WebSphere/ProcServer7/profiles/ProcSrv01/installedConnectors/CWYOE_OracleEBS.rar/CWYBS_AdapterFoundation.jar with version 7.0.0.3. However, the base level of this jar required is version 7.5. When you install multiple adapters with different CWYBS_AdapterFoundation.jar versions, the adapter returns the ResourceAdapterInternalException message, due to a version conflict. To overcome this conflict, you must migrate all adapters to the same version level. For further assistance, contact WebSphere Adapters Support for help.

Considerations while deploying WebSphere Adapter 7.5 with another version

There are occasions when you have to work with embedded adapters that do not need a client-server communication, standalone adapters that need a server connection, or a hybrid mix of adapter connections.

The following scenarios cover the different behaviors of AFC version conflict detection.

Deploying a standalone Adapter

1. Install WebSphere Adapter for Flat Files version 7.0.1.0 through the IBM Business Process Manager administrative console.
2. Install WebSphere Adapter for SAP Software version 7.5.0.0 through the administrative console.
3. Create ActivationSpec for an ALE passthrough inbound operation.
4. Create an application in IBM Integration Designer for a standalone ALE passthrough inbound operation.
5. Install and start the application through the administrative console.
6. Verify the error.

Note: An error message will be generated in the log/trace area of IBM Business Process Manager, to indicate an AFC version conflict.

Deploying an embedded Adapter

1. Import a build of WebSphere Adapter for FTP version 7.0.1.0, using a RAR file.
2. Create a FTP Inbound EMD operation.
3. Import a build of WebSphere Adapter for Oracle E-Business Suite version 7.5.0.0, using a RAR file.
4. Create an Oracle inbound EMD operation, in the same module where you have created the FTP Inbound EMD operation.
5. Deploy the module to IBM Business Process Manager.
6. Check the trace.

At step 5, the deployment will fail. At step 6, you will get an internal error message due to the AFC version conflict.

Note: To avoid a name conflict between the business object generated by the two adapters, you may need to generate the artifacts into different folders.

Deploying a combination of standalone and embedded Adapters

1. Install WebSphere Adapter for JDBC version 7.0.1.0 through the IBM Business Process Manager administrative console.
2. Create an ActivationSpec for a JDBC inbound operation.
3. Create an application in IBM Integration Designer for a JDBC inbound operation, for the standalone Adapter deployment.
4. Deploy the JDBC inbound application and trigger your inbound events.
5. Create an application in IBM Integration Designer for a WebSphere Adapter for SAP Software version 7.5.0.0 inbound embedded Adapter deployment.
6. Deploy an SAP inbound application, and trigger your inbound events.

Note: You can resolve the AFC version conflict by using different class loaders for the standalone and embedded deployments. With this approach, the migration process will handle different CWYBS_AdapterFoundation.jar files and will not conflict with each other. You can start both JDBC and SAP inbound applications successfully, and process Inbound events without exception.

For further assistance, visit http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family.

WebSphere Adapters in clustered environments

You can improve adapter performance and availability by deploying a module on a clustered server environment. Clusters are groups of servers that are managed together to balance workloads and to provide high availability and scalability.

The module you deployed is replicated across all servers in a cluster, regardless of whether you deploy the module using a stand-alone or an embedded adapter. The following IBM products support WebSphere Adapters in a clustered environment:

- IBM Business Process Manager or WebSphere Enterprise Service Bus
- WebSphere Application Server Network Deployment
- WebSphere Extended Deployment

When you set up a server cluster, you create a Deployment Manager profile. The HAManager, a subcomponent of the Deployment Manager, notifies the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) container to activate an adapter instance. For information about creating clustered environments, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/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 instance instead of a static Workload Manager. The dynamic Workload Manager instance 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 systems with different capacities and configurations to handle load variations evenly. For information about the benefits of WebSphere Extended Deployment, see <http://publib.boulder.ibm.com/infocenter/wxdinfo/v6r1m1/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 Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) 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.

Note: In the active-passive configuration mode of the adapters, the endpoint application of the passive adapter instance also listens to the events/messages even

if the `enableHASupport` property is set to `True`. This is because the `alwaysactivateAllMDBs` property in the JMS activation specification is set to `True`. To stop the endpoint application of the passive adapter instance from listening to the events, you must set the `alwaysactivateAllMDBs` property value to `False`. For more information, see “Disabling end point applications of the passive adapter” on page 197.

Note: In clustered environments, when the adapter works in a Active-Active configuration, it provides both high availability and load balancing support. This functionality is useful in production environments where high performance is needed.

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 about workload management, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html.

Support for prepared statement caching

WebSphere Adapter for Oracle E-Business Suite supports prepared statement caching by the server, which can reduce the time required to perform an outbound or inbound operation or a batch of operations.

The adapter uses *prepared statements*, which are Java objects containing an SQL QUERY statement that is compiled once, but can be run multiple times. The server caches prepared statements to optimize their handling. If you want to use prepared statement caching for the adapter, define a data source using the administrative console, and enable caching on the data source. Then configure the adapter to use the data source, using one of these methods:

- Using the external service wizard when you initially configure the adapter to use the JNDI name of the data source
- Using the administrative console of IBM Business Process Manager or WebSphere Enterprise Service Bus to set the `DatasourceJNDIName` property for inbound processing
- Using the administrative console of IBM Business Process Manager or WebSphere Enterprise Service Bus to set the `poolDataSourceJNDIName` or `XADatasourceJNDIName` property for outbound processing

Migrating to version 7.5 of WebSphere Adapter for Oracle E-Business Suite

When you migrate to WebSphere Adapter for Oracle E-Business Suite version 7.5, you automatically upgrade from the previous version of the adapter. WebSphere Adapter for Oracle E-Business Suite versions 6.0.2.x and 6.1.x do not have its own adapter but use the WebSphere Adapter for JDBC as the base. From version 6.2, WebSphere Adapter for Oracle E-Business Suite is an independent adapter and has no dependency on WebSphere Adapter for JDBC. If you migrate the adapter from version 6.0.2.x or 6.1.x, the underlying JDBC adapter is upgraded to WebSphere Adapter for Oracle E-Business Suite version 7.5. The project artifacts are also migrated to version 7.5 appropriately. Migrating the embedded applications belonging to an earlier version of the adapter, enables the applications to use the latest features and capabilities available in version 7.5.

Migration considerations

WebSphere Adapter for Oracle E-Business Suite version 7.5 may have some features and updates that might affect your existing adapter applications. Before migrating applications that use WebSphere Adapter for Oracle E-Business Suite, you must consider some factors that might affect your existing applications.

Compatibility with earlier versions

Since the version 6.1.x, and version 6.2.x, of the adapter were based on WebSphere Adapter for JDBC, the artifacts generated using those versions are not compatible with the latest adapter version. If you want your applications created using version 6.1.x, or version 6.2.x, of the adapter, to use the features and functions present in version 7.5, run the migration wizard to migrate the adapter artifacts too to the latest version.

Because the version 7.0 of WebSphere Adapter for Oracle E-Business Suite is not fully compatible with version 7.0, you have to modify some artifacts manually. For example, the 'resourceAdapter name' property in .import file must be changed from "XXXX.IBM WebSphere Adapter for Oracle" to "XXXX.IBM WebSphere Adapter for Oracle E-Business Suite." It is recommended to run the migration wizard for both adapter and the adapter artifacts to avoid manual changes being done to the artifacts.

The migration wizard replaces (upgrades) version 6.1.x, version 6.2.x, or version 7.0 of the adapter with version 7.5 and enables version 7.5 features and functionality for use with your applications.

If the artifacts within a module have inconsistent versions, the entire module is marked as unavailable for migration and cannot be selected. Version inconsistencies are recorded in the workspace log, as they indicate that a project might be corrupted.

The adapter migration wizard in IBM Integration Designer version 7.5 only supports the migration of adapters from version 6.1x, version 6.2x, and version 7.0 to version 7.5. It does not support the adapter migration from lower versions to any of the versions prior to version 7.5.

Deciding whether to upgrade or to upgrade and migrate

The default processing of the migration wizard is to upgrade the adapter and to migrate the application artifacts so that the applications can use features and functions available in version 7.5 of the adapter. When you choose to upgrade the adapter by selecting a project of version 7.0, the wizard automatically selects the associated artifacts for migration.

If you decide that you want to upgrade the adapter from 6.1.x, version 6.2.x and version 7.0 to version 7.5, but you do not want to migrate the adapter artifacts, you can do so by deselecting the adapter artifacts from the appropriate area of the migration wizard.

Running the migration wizard without selecting any adapter artifacts installs and upgrades your adapter. As the artifacts are not migrated, your applications cannot take advantage of the features and capabilities that exist in version 7.5 of the adapter.

Migrating multiple adapters referred within a project

When a module contains one or more connector projects and each of them refer to different adapters (for example, a module project that contains connector projects referring to Oracle and SAP adapters), the migration wizard identifies the artifacts belonging to each adapter and migrates the artifacts without disrupting the artifacts of the other adapters.

When you select the module project and launch the migration wizard:

- The **Source connector** field lists the connector projects with the selected module project.
- The **Dependent artifact projects** area lists only the selected module project.

If you select the connector project and launch the migration wizard:

- The **Source connector** field lists only the selected connector project.
- The **Dependent artifact projects** area lists all projects which reference the selected connector project, including the module project.

Run the migration wizard in a test environment

Because adapter migration might require you to change those applications that use version 7.5 of WebSphere Adapter for Oracle E-Business Suite, you must always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

The migration wizard is fully integrated with the development environment.

Deprecated features

A deprecated feature is one that is supported but no longer recommended and that might become obsolete. The following feature from earlier versions of WebSphere Adapter for Oracle E-Business Suite have been deprecated in version 7.5 and might require changes to your applications:

- Support for DataSourceJNDIName property

This property has been deprecated in version 7.0 and instead the two new properties, “XA DataSource JNDI name (XADataSourceJNDIName)” on page 244

and “Connection pool DataSource JNDI name (PoolDataSourceJNDIName)” on page 243, are provided as replacements. If an artifact from the earlier version is configured with the DataSourceJNDIName property, during the migration process, the new property “XA DataSource JNDI name (XADataSourceJNDIName)” on page 244 is set to the same value as the DataSourceJNDIName property. This deprecated property is still supported by the adapter at run time so that you can use the adapter version 7.5 smoothly, and it is displayed in the assembly editor. However, you cannot edit this property.

Performing the migration

You can migrate a project or EAR file to version 7.5 using the adapter migration wizard. When the tool is finished, the migration is complete and you can work in the project or deploy the module.

Before you begin

Review the information in *Migration considerations*.

About this task

To perform the migration in IBM Integration Designer, complete the following steps.

Note: After migration is complete, the module is no longer compatible with previous versions of the IBM Business Process Manager or WebSphere Enterprise Service Bus runtimes or IBM Integration Designer.

The following steps describe how to run the adapter migration wizard from the connector project menu while in the Java EE perspective in IBM Integration Designer.

Procedure

1. Import the PI (project interchange) file for an existing project into the workspace.

Note: Ensure that you do not modify the contents of the RAR or copy the adapter JAR file outside the connector project.

2. When projects are created in an earlier version of IBM Integration Designer, the Workspace Migration wizard starts automatically and selects the projects to migrate. Follow the wizard and complete the workspace migration. For more information, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.wid.imuc.doc/topics/tmigrscart.html>.
3. Change to the Java EE perspective.
4. Right-click the module and select **Migrate connector project**. For example, the adapter RAR module.

You can also launch the adapter migration wizard in the following ways:

- Right-click the project in the Java EE perspective and select **Migrate adapter artifacts**.
 - From the Problems view, right-click a migration-specific message and select **Quick Fix** to correct the problem.
5. In the Select Projects window, perform the following steps:

- a. The **Source connector** field displays the name of the connector project that you are migrating. If you are migrating a module project, this field lists all the connector projects in the module project. Select the source project from the list. For more information, see “Migrating multiple adapters referred within a project” on page 63.
 - b. The **Target connector** field displays the name of the connector to which you are migrating. If you are working with more than one adapter version, this list displays the names of all the compatible connectors. Select the connector you want to migrate.
 - c. The **Target version** field displays the version corresponding to the target connector that you selected in the previous step.
 - d. The **Dependent artifacts project** area lists the adapter artifacts that are migrated. If you are migrating a module project, this area lists only the selected module project. If you are migrating a connector project within the module project, this area lists all projects which reference the selected connector project, including the module project. By default, all the dependent artifact projects are selected. If you do not select a dependent artifact project, that project is not migrated. You can migrate any project that you have not selected at a later time. Previously migrated projects, projects with a current version, and projects that contain errors are unavailable for migration and are not selected. For more information, see “Upgrading but not migrating a project” on page 66.
 - e. Click **Next**. A warning window is displayed with the message, “Properties that are not supported in this version of the target adapter will be removed during the migration”.
 - f. Click **OK**.
6. In the Review Changes window, review the migration changes that occur in each of the artifacts that you are migrating. To view the details, expand each node by clicking the + sign.
 7. To complete the migration:
 - Click **Finish**.
 - If the files that need to be updated during migration are in read-only mode, you will be unable to click on the **Finish** button. To view these files, click **Next**. The Update Read-only files window displays the read-only files. To update these files and continue with the migration, click **Finish**. To exit the wizard without migrating the adapter, click **Cancel**.

Before running the migration process, the wizard performs a backup of all projects affected by the migration. The projects are backed up to a temporary folder within the workspace. If the migration fails for any reason, or if you decide to cancel the migration before it completes, the wizard deletes the modified projects and replaces them with the projects stored in the temporary folder.

Upon completing the migration successfully, all backed up projects are deleted.

8. If you are migrating an EAR file, optionally create a new EAR file with the migrated adapter and artifacts, and deploy it to IBM Business Process Manager or WebSphere Enterprise Service Bus. For more information about exporting and deploying an EAR file, see the topics devoted to it in this documentation.

Results

The project or EAR file is migrated to version 7.5. You do not need to run the external service wizard after exiting the adapter migration wizard.

Upgrading but not migrating a project

You can upgrade the adapter from an earlier version, to version 7.5 while choosing not to migrate the adapter project artifacts.

About this task

Running the migration wizard without selecting any adapter artifacts installs and upgrades your adapter. As the artifacts are not migrated, your applications cannot take advantage of the features and capabilities that exist in version 7.5 of the adapter.

Procedure

1. Import the PI (project interchange) file into the workspace.
2. When projects are created in an earlier version of IBM Integration Designer, the Workspace Migration wizard starts automatically and selects the projects to migrate. Follow the wizard and complete the workspace migration. For more information, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.wid.imuc.doc/topics/tmigrcart.html>.
3. In the Java EE perspective, right-click the project name and click **Migrate connector project**. The **Adapter Migration** wizard is displayed.
4. In the Select Projects window, clear the dependent artifact projects, and click **Next**. A warning window is displayed with the message, "The properties that are not supported in the version of the target adapter will be removed during the migration."
5. Click **OK**.
6. In the Review Changes window, review the migration changes that occur during updating the project. To view the details, expand each node by clicking the + sign.
7. To complete the migration:
 - Click **Finish**.
 - If the files that need to be updated during migration are in read-only mode, you will be unable to click on the **Finish** button. To view these files, click **Next**. The Update Read-only files window displays the read-only files. To update these files and continue with the migration, click **Finish**. To exit the wizard without migrating the adapter, click **Cancel**.

Results

The project can now be used with WebSphere Adapter for Oracle E-Business Suite, version 7.5.

Note: After upgrading the adapter, you have to modify some artifacts manually depending on the versions of the adapters involved in migration. For example, if you are upgrading the adapter from version 7.0 to version 7.0, the 'resourceAdapter name' property in .import file must be changed from "XXXX.IBM WebSphere Adapter for Oracle" to "XXXX.IBM WebSphere Adapter for Oracle E-Business Suite." It is recommended to run the migration wizard for both adapter and the adapter artifacts to avoid manual changes being done to some artifacts after the migration.

Migrating WebSphere Business Integration applications

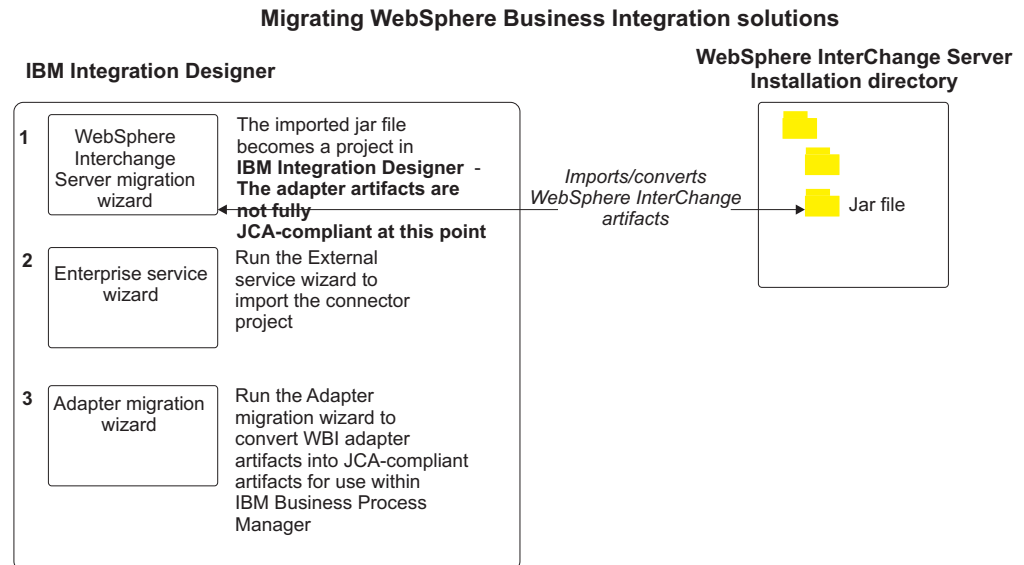
You need to migrate the WebSphere Business Integration applications so that they become compatible with Version 7.5 of your adapter.

About this task

Migrating WebSphere Business Integration applications for use with Version 7.5 of your WebSphere adapter is a multistep process. First, the artifacts from WebSphere InterChange Server are migrated and converted. A project is then created for the artifacts in IBM Integration Designer. In the remaining steps, the adapter-specific artifacts are migrated and converted into the JCA-compliant format supported by Version 7.5 of the adapter.

Example

The following diagram shows the wizards that you use to migrate WebSphere Business Integration solutions from WebSphere InterChange Server, so that these applications can be used with Version 7.5 of your adapter.



Migrating applications from WebSphere InterChange Server

To use Version 7.5 of WebSphere Adapter for Oracle E-Business Suite with applications from WebSphere InterChange Server, you need to migrate the application artifacts and convert them so that they can be deployed and run on IBM Business Process Manager or WebSphere Enterprise Service Bus. Understanding this task at a high level helps you perform the steps that are needed to accomplish the task.

The following figure illustrates the flow of the migration task. The steps that follow the figure describe this task at a high level only. See the topics following this roadmap for the details on how to perform each of these steps.

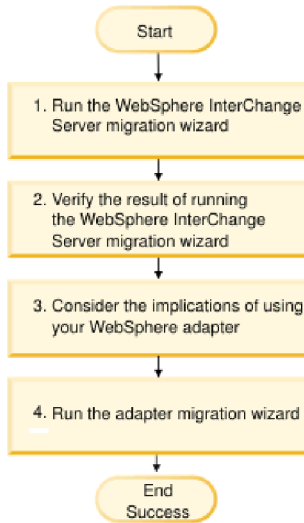


Figure 19. Roadmap for migrating applications from WebSphere InterChange Server

Migrating applications from WebSphere InterChange Server

This task consists of the following steps:

1. Run the WebSphere InterChange Server migration wizard.
The WebSphere InterChange Server migration wizard moves the application artifacts into IBM Integration Designer. The migrated adapter artifacts are not fully JCA-compliant at the completion of this task.
2. Verify that the WebSphere InterChange Server migration is successful.
Review all messages from the Migration results window and take action if required.
3. Consider the implications of using Version 7.5 of WebSphere Adapter for Oracle E-Business Suite.
In addition to considerations for migrating WebSphere InterChange Server applications, you need to consider how Version 7.5 of WebSphere Adapter for Oracle E-Business Suite works with the migrated applications. Some of the adapter operations supported by WebSphere InterChange Server applications might be supported and implemented differently with Version 7.5 of the adapter.
4. Run the adapter migration wizard.
Run the adapter migration wizard to update adapter-specific artifacts such as the schemas and service definition files (.import,.export, and .wsdl files) for use with Version 7.5 of the adapter.

Migration considerations for WebSphere Business Integration adapters

By migrating to WebSphere Adapter for Oracle E-Business Suite Version 7.5, you have an adapter that is compliant with the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) and designed specifically for service-oriented architecture.

Application artifacts

Before running the adapter migration wizard, use the WebSphere InterChange Server migration wizard to generate the application artifacts for the WebSphere

Business Integration adapter, including the business objects, maps, and collaborations. Then you can run the adapter migration wizard to update the adapter-specific artifacts such as the schemas and service definition files (.import,.export, and .wsdl) so that they are suitably converted into a format that is compliant with JCA.

Run the migration wizard in a test environment first

Because migrating from a WebSphere Business Integration adapter to WebSphere Adapter for Oracle E-Business Suite might require changes to the applications that use Version 7.5 of WebSphere Adapter for Oracle E-Business Suite, always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

Migrating application artifacts from WebSphere InterChange Server

To migrate the application artifacts into IBM Integration Designer, run the WebSphere InterChange Server migration wizard. The wizard imports and converts most of the artifacts into a format that is compatible with IBM Business Process Manager or WebSphere Enterprise Service Bus.

Before you begin

Launch the WebSphere InterChange Server migration wizard from within IBM Integration Designer to migrate the application artifacts from WebSphere InterChange Server format into artifacts that are compatible with IBM Business Process Manager or WebSphere Enterprise Service Bus.

For information about how to prepare to migrate artifacts from WebSphere InterChange Server and for detailed instructions on performing the migration and verifying that the migration was successful, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.wid.imuc.doc/topics/twics.html>.

About this task

Running WebSphere InterChange Server migration wizard might not fully convert adapter-specific artifacts (such as service descriptors, service definitions, and business objects) into IBM Business Process Manager or WebSphere Enterprise Service Bus compatible artifacts. To complete the migration of adapter-specific artifacts, run the adapter migration wizard after you have successfully run the WebSphere InterChange Server migration wizard.

Note: While you run the WebSphere InterChange Server migration wizard, ensure that you set each connector in the repository to the same adapter version.

Results

The project and application artifacts are migrated and converted into IBM Business Process Manager compatible artifacts.

What to do next

Run the adapter migration wizard to migrate the adapter-specific artifacts.

Migrating adapter-specific artifacts

After a project is created for the artifacts in IBM Integration Designer, you can migrate the project using the adapter migration wizard. The adapter migration wizard updates adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl) for use with version 7.5 of the adapter. When you finish running the adapter migration wizard, the migration is complete and you can work in the project or deploy the module.

Before you begin

Before running the adapter migration wizard, you should do the following steps:

- Review the information in “Migration considerations” on page 62.
- Run the WebSphere InterChange Server migration wizard to migrate the project and convert data objects for use with IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

After migration is complete, the module will work only with Version 7.5 of your adapter.

To perform the migration in IBM Integration Designer, complete the following steps.

Procedure

1. Import the PI (project interchange) file for an existing project into the workspace.
2. When projects are created in an earlier version of IBM Integration Designer, the Workspace Migration wizard starts automatically and selects the projects to migrate. Follow the wizard and complete the workspace migration. For more information, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.wid.imuc.doc/topics/tmigrscart.html>.
3. Change to the Java EE perspective.
4. Right-click the connector project and select **Migrate connector project**.
You can also launch the adapter migration wizard by using the right-click option and selecting the module project in the Java EE perspective and selecting **Migrate adapter artifacts**.

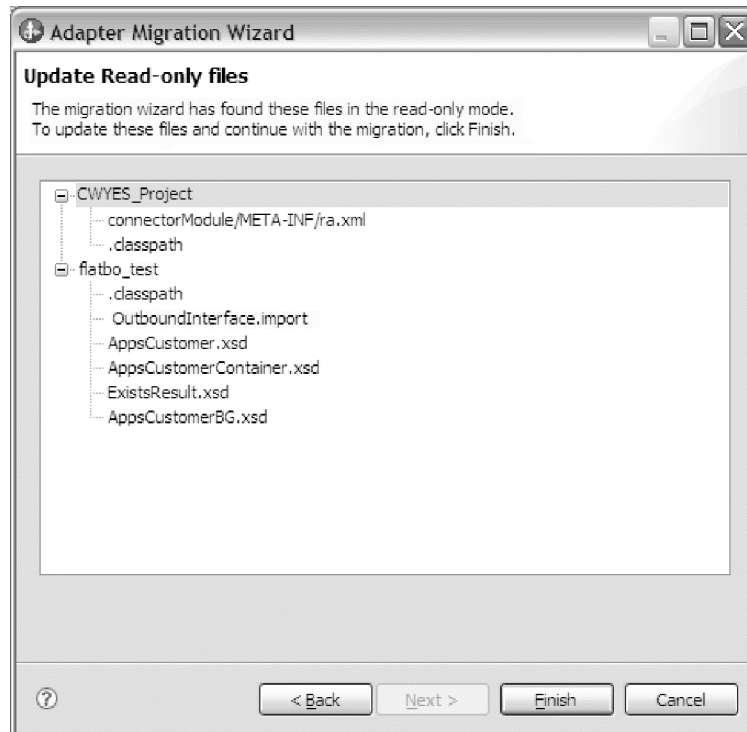
Note:

If the adapter type (for example, CICS/IMS adapter) is not supported by the migration wizard, the **Migrate connector project** and **Migrate adapter artifacts** menus are not available for selection. If the adapter project is of the latest version and the module projects referencing this adapter project are also of the latest version, these menus are disabled.

When you launch the migration wizard from the connector project while in the Java EE perspective, by default all the dependent artifact projects are selected. If you do not select a dependent artifact project, that project is not migrated.

5. In the Select Projects window, perform the following steps:
 - a. The **Source connector** field displays the name of the connector project that you are migrating. Select the source project from the list.

- b. The **Target connector** field displays the name of the connector to which you are migrating. If you are working with more than one adapter version, this list displays the names of all the compatible connectors. Select the connector to which you want to migrate.
 - c. The **Target version** field displays the version corresponding to the target connector you selected in the previous step.
 - d. The **Dependent artifacts project** area lists the adapter artifacts that are migrated.
 - e. Review the tasks and warnings presented on the welcome page, and click **Next**. A warning window is displayed with the message, “The properties that are not supported in the version of the target adapter are removed during the migration.”
 - f. Click **OK**.
6. In the Review Changes window, review the migration changes that occur in each of the artifacts that you are migrating. To view the details, expand each node by clicking the + sign.
 7. To complete the migration:
 - Click **Finish**.
 - If the files that need to be updated during migration are in read-only mode, you will be unable to click on the **Finish** button. To view these files, click **Next**. The Update Read-only files window displays the read-only files. To update these files and continue with the migration, click **Finish**. To exit the wizard without migrating the adapter, click **Cancel**.



Before performing the migration process, the wizard backs up all projects affected by the migration. The projects are backed up to a temporary folder within the workspace. If the migration fails for any reason, or if you decide to cancel the migration before it completes, the wizard deletes the modified projects and replaces them with the projects stored in the temporary folder.

8. Select **Project > Clean**, to refresh and rebuild the workspace for the changes to take effect.
9. If you are migrating an EAR file, create a new EAR file with the migrated adapter and artifacts, and deploy it to IBM Business Process Manager or WebSphere Enterprise Service Bus. For information about exporting and deploying an EAR file, see “Deploying the module for production” on page 169.

Results

The project is migrated to Version 7.5. You do not need to run the external service wizard after exiting the adapter migration wizard.

Changes to the import, export, and WSDL files after migration

When the WebSphere InterChange Server migration wizard moves the application artifacts into IBM Integration Designer, changes made are reflected in the service definition files: the import, export and WSDL files.

The migrated adapter artifacts are not fully JCA-compliant at the completion of this task. You can complete the migration of the adapter-specific artifacts (such as service descriptors, service definitions, and business objects) to a JCA compatible format by running the adapter migration wizard.

Changes to the import file

During migration, the affected module artifacts are migrated to an import file. The existing JMS Binding property is changed to the EIS Binding property in the import file. The other property details added in the import file include information about the data binding configuration, changes to the connection information in the Managed Connection Factory properties, and several new method bindings.

Changes to the export file

During migration, the affected module artifacts are migrated to an export file. The existing JMS Binding property is changed to the EIS Binding property in the export file. The other property details added in the export file include information about the data binding configuration, changes to the connection information in the Activation Specification properties, and several new method bindings.

Changes to the WSDL file after migration

During migration, the affected module artifacts are migrated to corresponding WSDL files that include adapter specific service description WSDL artifacts. The service description files become JCA compatible. The WSDL files will have an input and output type for each operation. Both the inbound and outbound operations work on their specific input types to produce corresponding output types after the operations are performed.

Note:

- When you migrate multiple top level inbound business objects in the project, only the first top-level business object inbound feature works correctly. For the other top level inbound business object to work correctly, you must manually modify the "emit + [verb name] + after image + [business object name]" method in the Input_Processing.java and Input_Async_Processing.java class to call the correct destination services.

- The WebSphere Business Integration Adapter for Oracle Applications properties that are either not valid or not supported by WebSphere Adapter for Oracle E-Business Suite are removed from the migrated artifacts.

Chapter 3. Samples and tutorials

To help you use WebSphere Adapters, samples and tutorials are available from the Business Process Management Samples and Tutorials website.

You can access the samples and tutorials in either of the following ways:

- From the welcome page of IBM Integration Designer, click **Go to Samples and Tutorials**. In the Samples and Tutorials pane, under More samples, click **Retrieve**. Browse the displayed categories to make your selection.
- From the Business Process Management Samples and Tutorials website:
<http://publib.boulder.ibm.com/bpcsamp/index.html>.

Chapter 4. Configuring the module for deployment

To configure the adapter so that it can be deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus, use IBM Integration Designer to create a module, which is exported as an EAR file when you deploy the adapter. You then specify the business objects you want to discover and the system on which you want to discover them.

Road map for configuring the module

Before you can use WebSphere Adapter for Oracle E-Business Suite in a runtime environment, you must configure the module. Understanding this task at a high level helps you perform the steps that are needed to accomplish the task.

You configure the module for the adapter to use by using IBM Integration Designer. The following figure illustrates the flow of the configuration task, and the steps that follow the figure describe this task at a high level only. See the topics following this road map for the details on how to perform each of these steps.

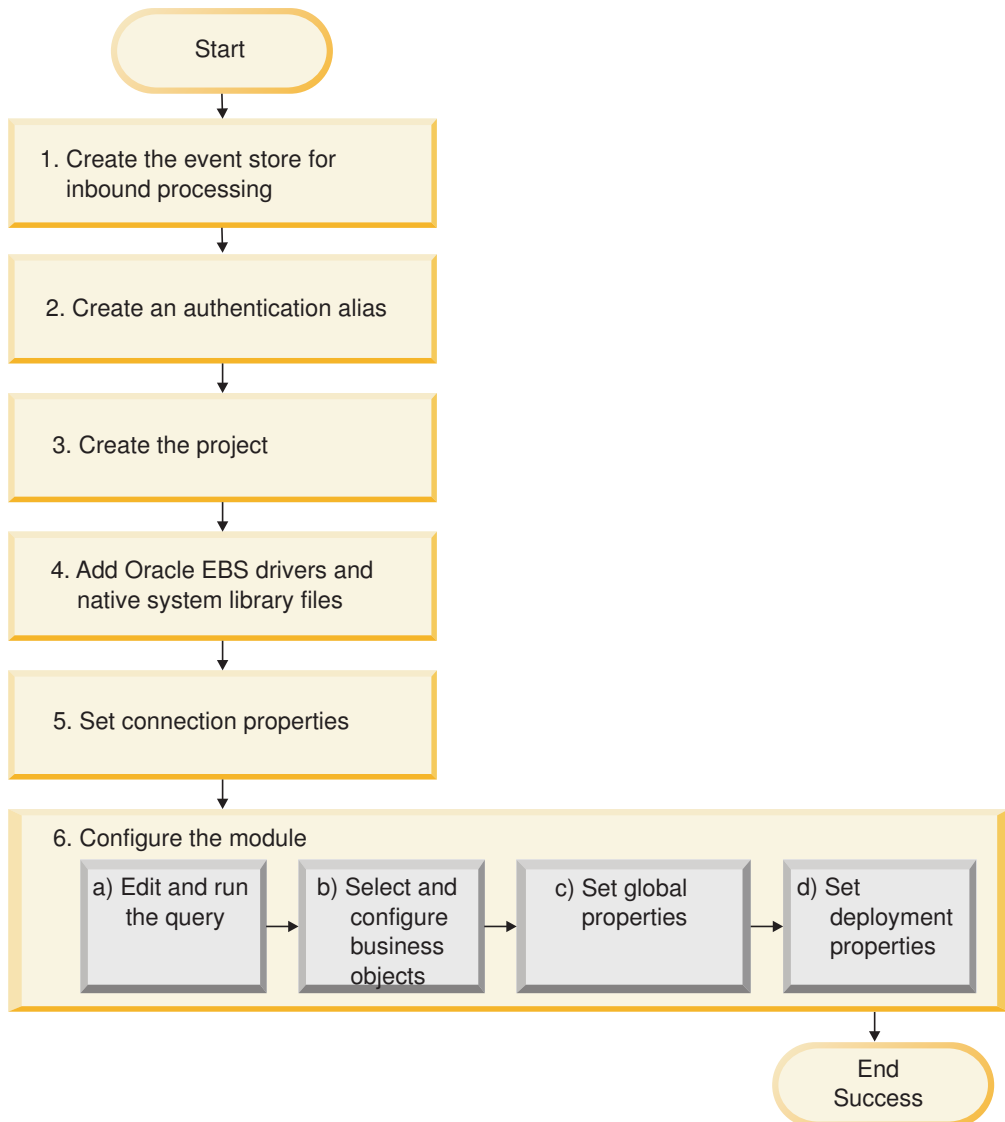


Figure 20. Road map for configuring the module

Configuring the module for deployment environment

This task consists of the following step:

1. Create the event store for inbound processing.
2. Create an authentication alias to access the Oracle database with an encrypted password. This step is optional, depending on your policy for handling passwords and IDs. You perform this step using the server.
3. Create the project. First, start the external service wizard in IBM Integration Designer to begin the process of creating and deploying a module. The wizard creates a project that is used to organize the files associated with the module.
4. Add the JDBC drivers and native system library files required by WebSphere Adapter for Oracle E-Business Suite to the project. These dependencies are also required when you export the module as an EAR file, and deploy the EAR file to the server.
5. Set connection properties that the external service wizard needs to connect to the Oracle database for discovery of objects and services.

6. Configure the module for inbound processing or outbound processing by using the external service wizard to find and select business objects and services from the Oracle database, and to generate business object definitions and related artifacts.
 - a. Edit and run the query that discovers database objects that you can access.
 - b. Select and configure business objects for inbound or outbound processing.
 - c. Set global properties for operations.
 - d. Set deployment properties that the adapter uses to connect to the Oracle database at run time. Then, generate the service by using the external service wizard to save the new module, which contains the business object or objects you configured, the import file or export file, and the service interface.

Creating the event store

You need to create the event store in the Oracle database before the adapter can process inbound events. You can set triggers on user tables as needed to populate the event store. For Oracle E-Business Suite, you need to configure the Oracle Business Event System and Concurrent Program to capture the event.

About this task

Perform this task only if you need inbound processing of events. Create the event store in the database that contains the tables for which events are reported.

Procedure

1. Create the event store. Sample scripts are provided to create the event store for the Oracle database, as follows:
 - `ibm_websphere_event_table_create.sql`These files are located in the `IID_installation_dir/ResourceAdapters/OracleEBS_version/scripts` directory, where `IID_installation_dir` is the installation directory for IBM Integration Designer, and `version` identifies the version of the adapter, for example, 7.5.
2. For Oracle database, set up triggers on user tables so that the changes to the user tables can automatically generate events that is stored in the event store. If using triggers is not an option, then you can populate the event store by using custom SQL code or stored procedures that are not invoked through triggers. Instead, the triggers can be invoked through a batch program that runs periodically or that you run manually. For Oracle E-Business Suite, it is recommended that you configure the Oracle Business Event System and Concurrent Programs to define the events and automatically capture the events. The Concurrent Programs can be scheduled to run periodically or run manually when required.

Results

The event store is available for event processing.

Related concepts

“Event store” on page 24

The event store is a persistent cache where event records are saved until the polling adapter can process them. The adapter uses the event store to track the inbound requests as they make their way through the system. Each time a database record is created, updated, or deleted, the adapter updates the status of the event in the event store. The status of each event is continually updated by the adapter for recovery purposes until the events are delivered to a configured export on the server.

Creating an authentication alias

An authentication alias is a feature that encrypts the password used by the adapter to access the Oracle database. The adapter can use it to connect to the Oracle database instead of using a user ID and password stored in an adapter property.

Before you begin

To create an authentication alias, you must have access to the administrative console of IBM Business Process Manager or WebSphere Enterprise Service Bus. You must also know the user name and password to connect to the Oracle database.

The following procedure shows you how to gain access to the administrative console through IBM Integration Designer. If you are using the administrative console directly (without going through IBM Integration Designer, log on to the administrative console and skip to step 2.

About this task

Using an authentication alias eliminates the need to store the password in clear text in an adapter configuration property, where it might be visible to others.

To create an authentication alias, use the following procedure.

Procedure

1. Start the administrative console.
 - To start the administrative console through IBM Integration Designer, perform the following steps:
 - a. In the Business Integration perspective of Integration Designer, click the **Servers** tab.
 - b. If the server does not show a status of **Started**, right-click the name of the server (for example, IBM Business Process Manager) and click **Start**. Wait for the status of the server to be displayed as Started.
 - c. Right-click the name of the server and click **Run administrative console**.
 - d. Log on to the administrative console. If your administrative console requires a user ID and password, type the ID and password and click **Log in**. If the user ID and password are not required, click **Log in**.
2. In the administrative console, click **Security > Global security**.
3. Under **Java Authentication and Authorization Service**, click **J2C authentication data**.
4. Create an authentication alias:
 - a. In the list of J2C authentication aliases that is displayed, click **New**.

- b. In the General Properties area, type the name of the authentication alias in the **Alias** field.
- c. Type the User ID and password that are required to establish a connection to the Oracle database.
- d. Optional: Type a description of the alias.
- e. Click **OK**.

The newly created alias is displayed.

The full name of the alias contains the node name and the authentication alias name you specified. For example, if you create an alias on the node `widNode` with the name `ProductionServerAlias`, then the full name is `widNode/ProductionServerAlias`. This full name is the one you use in subsequent configuration windows.

- f. Click **Save**.

Results

You have created an authentication alias, which you specify when you configure the adapter properties later in the wizard.

Related concepts

“User authentication” on page 54

The adapter supports several methods for supplying the user name and password that are needed to connect to the Oracle database. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

Creating the project

To begin the process of creating and deploying a module, you start the external service wizard in IBM Integration Designer. The wizard creates a connector project, which is used to organize the files associated with the module.

Before you begin

Ensure that you have gathered the information you need to establish a connection to the Oracle database. For example, you need the name or IP address of the Oracle database and the user ID and password to access it.

About this task

If you have an existing project, you can use it instead of creating a new one. Select it before you start the wizard.

Procedure

1. To start the external service wizard, go to the Business Integration perspective of IBM Integration Designer, and then click **File > New > External Service**. Click **Next**.
2. In the New External Service window, expand the **Adapters** folder, select **Oracle E-Business Suite**, and then click **Next**.
3. In the Select an Adapter window, expand the Oracle node. Select **IBM WebSphere Adapter for Oracle E-Business Suite (IBM : *version*)**, where *version* is the version of the adapter you want to use, and then click **Next**.
4. In the Import a RAR File window, accept the default project name in the **Connector project** field or type a different name.

5. In the **Target runtime** field, select the type of server where you want to deploy the module. The wizard creates the artifacts that are appropriate to that server.
6. Click **Next**. The Locate the Required Files and Libraries window is displayed.

Results

A new connector project is created, which contains the adapter RAR file. The project is listed in the Business Integration perspective .

What to do next

Continue working in the external service wizard. The next step is to add database-specific files to the project.

Adding external software dependencies

The external service wizard needs a copy of certain files from the Oracle E-Business Suite to be able to communicate with it. Specify the location of the JAR files that contain the JDBC driver for Oracle and any native system library files that are needed.

Before you begin

You must run the external service wizard in IBM Integration Designer to perform this task.

About this task

In addition to performing this task when you configure the module, you also need to deploy files on IBM Business Process Manager or WebSphere Enterprise Service Bus.

Procedure

1. Obtain the Oracle JDBC driver-specific files or native libraries for your database software and operating system from your database administrator or from the database software website. The following table lists the Oracle JDBC driver files needed for the Oracle database software.

Table 11. Oracle JDBC driver files for Oracle database software

Database software	Driver	JDBC driver files	Native System Libraries
Oracle	Thin driver	ojdbc6.jar or ojdbc14.jar depending on the JRE version	None

Note: The Java Runtime Environment (JRE) version required by the Oracle JDBC driver must not be higher than the JRE version in the runtime environment. For example, if the JRE version is 1.5 in the runtime environment, then the Oracle JDBC Driver must be "ojdbc14.jar" which supports both Java Development Kit (JDK) 1.4 and JDK 1.5; if the JRE version is 1.6 in the runtime environment, then the Oracle JDBC Driver must be "ojdbc6.jar" which supports JDK version 1.6.

2. Obtain the Oracle JMS driver-specific files or native libraries for your database software and operating system from your database administrator or from the database software website. The following table lists the Oracle JMS files needed for Oracle database software.

Table 12. Oracle JMS driver files for Oracle database software

Database software	Driver	JMS driver files	Native System Libraries
Oracle	JDBC Thin	ojdbc6.jar or ojdbc14.jar depending on the JRE version	None
Oracle	JMS	apaxi*.jar	None
Oracle	JMS	jmscommon*.jar	None

Note: The jdbc driver and jms driver should be compatible with Java Runtime Environment (JRE) version and Oracle Database version.

3. In the Locate the Required Files and Libraries window, specify the location of the Oracle JDBC driver-specific files that are required by the adapter.
 - a. In **Oracle JDBC driver JAR files**, click **Add** and select the Oracle JDBC driver files or Oracle JMS driver files.
 - b. If you use the Oracle JDBC type 2 driver, click **Add** in **System libraries** to add the native system libraries that are required to access the database server. If you use Oracle JDBC type 4 driver, leave this field empty.
4. Click **Next**. The wizard displays the Select the Processing Direction window.

Results

The wizard has the files it needs to communicate with the database server.

What to do next

Continue working in the external service wizard. The next step is to provide the information that the wizard needs to communicate with the Oracle database.

Setting connection properties for the external service wizard

For the external service wizard to connect to the database instance to discover database objects, connection properties must be specified.

Before you begin

Before you can configure the connection properties, you must have started the external service wizard.

About this task

The external service wizard requires these properties to connect to the database for discovery and for creating the service description. For more information about the properties, see “Connection properties for the wizard” on page 225.

Procedure

1. In the Select the Processing Direction window, select **Outbound** to pass data from your service import to the adapter or **Inbound** to pass data from the adapter to your service export, and then click **Next**. The Specify the Discovery Properties window is displayed.
2. In the Specify the Discovery Properties window, specify the connection properties for the wizard to connect to the Oracle database.
 - a. In the list of database software, select your product and version. The **Properties** area displays fields where you specify database-specific connection properties.
 - b. In the **JDBC driver type** field, select the type of JDBC driver you want to use.
 - c. In the **System ID** field, specify the database name.
 - d. In the **Host name** field, specify the host name or IP address of the database server. If you specify the IP address in IPv6 format, enclose the address within square brackets ([]).
 - e. In the **Port number** field, specify the port number for connecting to the database. If you select a named driver in the **JDBC driver type** field, the wizard provides a default value for the **Port number** field. If you select the driver **Other**, the port number is not enabled.
 - f. If you select a named driver in the **JDBC driver type** field, the wizard provides a default value for the **JDBC driver class name** field and builds the value for the **Database URL** field from other connection fields. If you select the driver **Other**, you must specify the driver class name and database URL (although part of the database URL might be filled in for you).
 - g. In the **Additional JDBC driver connection properties** field, specify additional properties to be set when connecting to the database. Specify one or more *name:value* pairs, separated by the semicolon character (;). For example:

```
loginTimeout:20;readOnly:true;  
securityMechanism:USER_ONLY_SECURITY
```

The connection information is used only for the discovery process. Later in the wizard, you can specify different connection information to be used at run time.

3. In the **User name** and **Password** fields, type the user name and password to use to connect to the database from the wizard. This user name is used only during the discovery process and is not saved. Later in the wizard, you can specify a different user name and password or a different authentication method to be used at run time.
4. In the **Prefix for business object names** field, type a string to be placed at the beginning of business object names. This prefix field does not require a value, and can be left blank.
5. Optional: To set additional advanced properties, click **Advanced**.
6. Optional: To enable bidirectional support for the adapter at run time:
 - a. In the **Bidi properties** area, select **Bidi transformation**.
 - b. Set the ordering schema, text direction, symmetric swapping, character shaping, and numeric shaping properties to control how bidirectional transformation is performed.
7. Optional: If you want to turn on AUTOCOMMIT for the database, select the **Set auto commit on database connection** check box.

8. To change the location of the log files or the amount of information included in the logs, select the **Change logging properties for wizard** check box, and then provide the following information:

- In the **Log file output location** field, specify the location of the log file for the wizard.
- In the **Logging level** field, specify the severity of errors that you want logged.

This log information is for the wizard only; at run time, the adapter writes the error messages and the trace information into the standard log and trace files for the server.

9. Click **Next**.

If the wizard generates the `com.ibm.adapter.framework.BaseException` exception, the adapter cannot connect to the database server. The message contains additional information about a possible cause for the problem. In addition, you can check the logs, which are located in the directory specified in the **Log file output location** field. Make sure that the connection information is correct.

Results

The external service wizard connects to the database and displays the Find Objects in the Enterprise System window.

What to do next

Continue working in the wizard. The next step is to examine the database to locate the objects for which you want the wizard to create business objects.

Related concepts

“Globalization and bidirectional transformation” on page 276

The adapter is globalized to support single and multibyte character sets and deliver message text in the specified language. The adapter also performs bidirectional transformation, which refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.

Related reference

“Connection properties for the wizard” on page 225

External service wizard connection properties are used to establish a connection between the external service wizard, a tool that is used to create business objects, and the database.

Configuring the module for XML Gateway interface for web service

To configure the module for XML Gateway interface, use the external service wizard in Integration Designer. You can configure this interface only for the outbound integration. You can select the number of interfaces you would want to create. For each interface you create, you can select the required elements as per your business requirement.

Before you begin

You must place the required DTD files for the XML Gateway interface in a local directory. Also, you need to know the root DTD file and the root element for the selected XML Gateway interface.

About this task

This task is to create an outbound integration application using XML Gateway interface.

Procedure

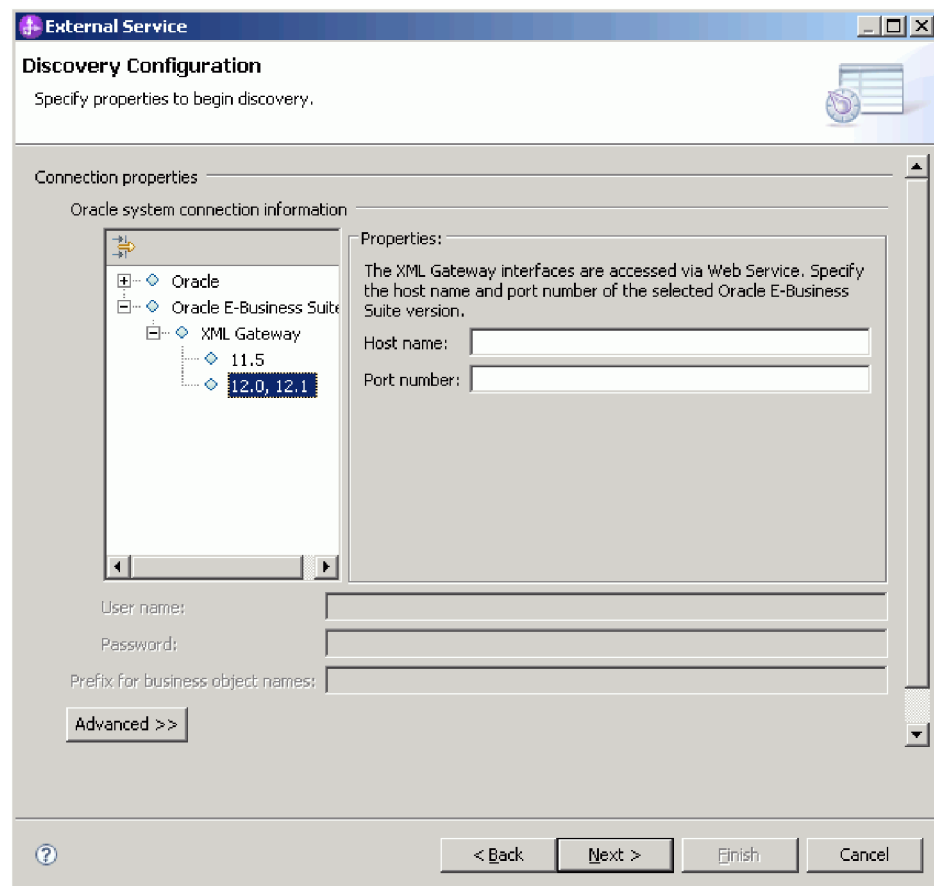
1. Create a project. For information about how to create a project, see [Creating the Project](#).
2. In the Locate the Required Files and Libraries window, click **Next**.

Note: The JDBC driver files must be provided only for connecting to the Oracle database server and for the JDBC-based Oracle interfaces.

3. In the Select the Processing Direction window, select **Outbound**. Click **Next**.

Note: The XML Gateway interface configuration can be performed only for the Outbound processing.

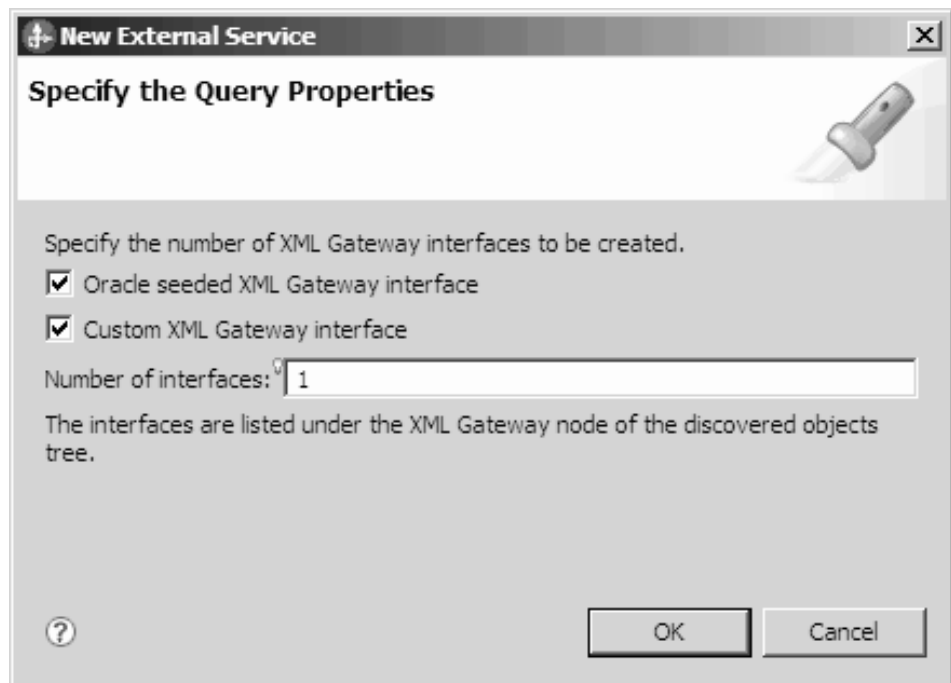
4. In the Specify the Discovery Properties window, select **Oracle E-Business Suite > XML Gateway > Web Service**.



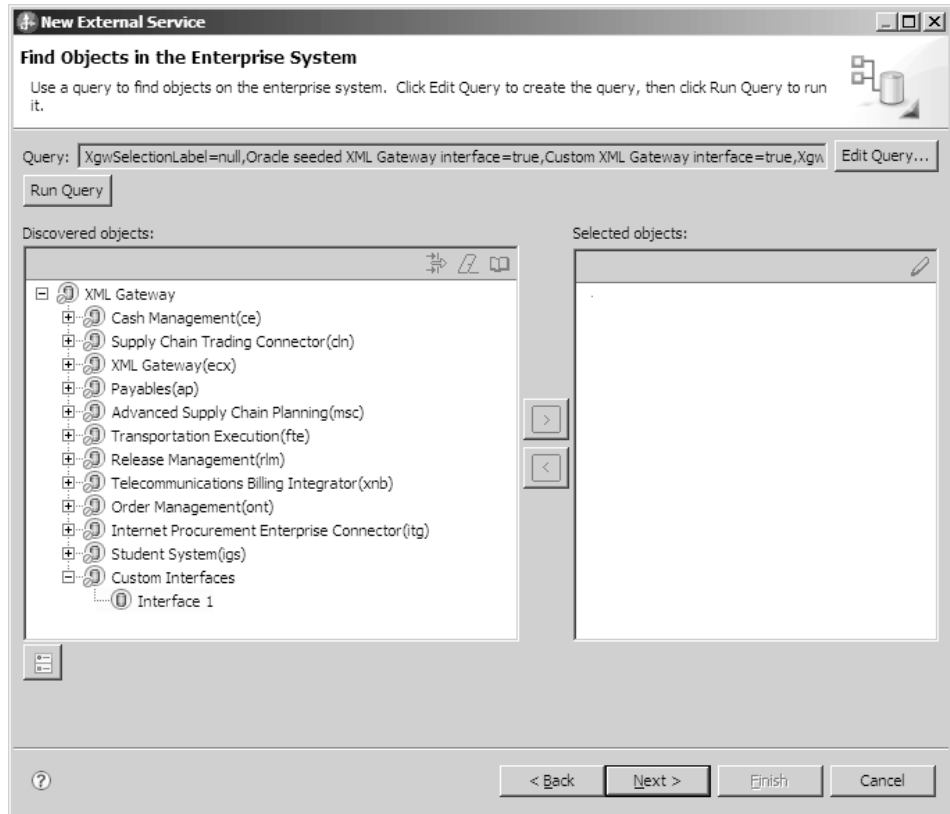
Note: You can select either the **Oracle** database node or the **XML Gateway** node and not both, in one EMD process. When you select the **XML Gateway** node, the **Bidi** and **Advanced** properties are not available for selection.

5. Select the Oracle E-Business Suite version and specify the connection information of the selected XML Gateway interface.
 - a. In the **Host name** field, enter the host name or the IP address of the system on which Oracle E-Business Suite is installed.

- b. In the **Port number** field, specify the port number for connecting to the Oracle E-Business Suite.
 - c. The **WSDL URL** field displays the URL of the XML Gateway interface. It is generated by the adapter based on the version, host name and port number that you specify.
 - d. In the **User name** and **Password** fields, type the user name and password to use to connect to the Oracle E-Business Suite.
6. Click **Next**.
 7. In the Find Objects in the Enterprise System window, click **Edit Query**.
 8. In the Specify the Query Properties window, you can work with Oracle seeded XML Gateway interface or custom XML Gateway interface or both. To create a custom XML interface, select the **Custom XML Gateway interface** check box and then type the number of interfaces that you want to create.

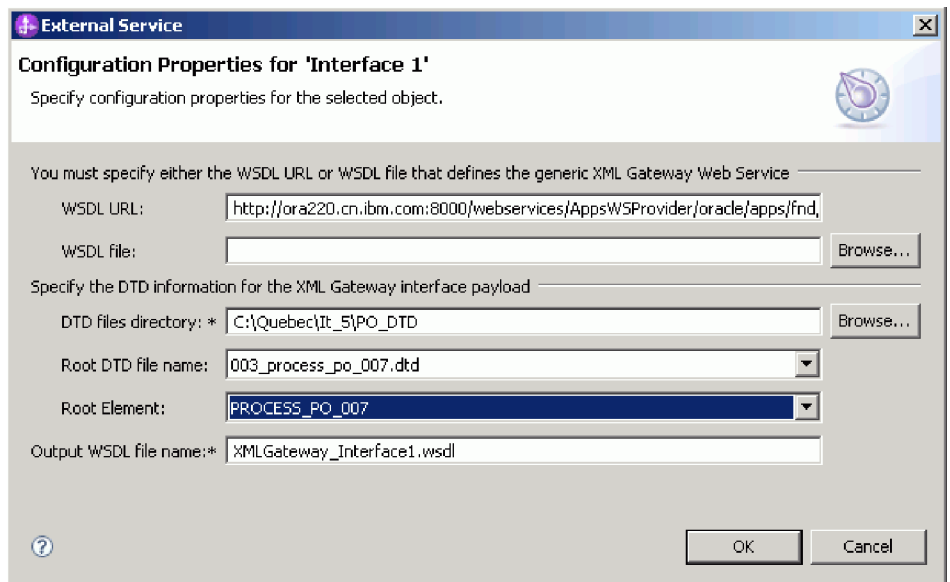


9. Click **OK** to save your changes to the query. In the Specify the Query Properties window, click **Run Query** to use the query to discover the objects. The result of running a typical query is shown in the following figure.



The **Discovered objects** pane lists the different business service interfaces supported by the adapter.

10. Select one or more interfaces from the **Discovered objects** list, and click the > (Add) button to add the interfaces to the **Selected objects** list.
11. In the Specify the Configuration Properties for 'object' window, specify the DTD information.



- a. In the **DTD files directory** field, type the path where the DTD files are located or click **Browse** to search for the location of the files.

- b. From the **Root DTD file name** list, select the root DTD file name from the list.
 - c. From the **Root Element** list, select a corresponding root element.
12. Click **OK**.
 13. In the Find Objects in the Enterprise System window, click **Next**.
 14. In the Specify Composite Properties window, click **Next**.
 15. In the Specify the Service Generation and Deployment Properties window, select **Using an existing JAAS alias(recommended)** or **Using security properties from the managed connection factory**.

New External Service

Specify the Service Generation and Deployment Properties
Specify properties for generating the service and running it on the server.

Service Operations
To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations. Edit Operations...

Deployment Properties
How do you want to specify the security credentials?

Using an existing JAAS alias (recommended)
A Java Authentication and Authorization Services (JAAS) alias is the preferred method.
J2C authentication data entry:

Using security properties from the managed connection factory
The properties will be stored as plain text; no encryption is used.

User name: *
Password: *

Other
Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.

The quality of service that is used to join the transaction provides a higher degree of data integrity, especially when a failure occurs. To participate in a global transaction, a predefined XA DataSource or XA database connection information must be specified in the connection properties. [More ...](#)

Join the global transaction

Deploy connector project:

Specify the settings used to connect to Oracle E-Business Suite at run time:
Connection settings:

Connection Properties
Database system connection information
The deployment properties are required to access the XML Gateway interface of Oracle E-Business Suite.

Oracle E-Business Suite Version: 12.1
Oracle E-Business Suite Connection Type: XML Gateway
Oracle E-Business Suite Transport: Web Service
WSDL URL: *

? < Back Next > Finish Cancel

16. Since there is no transaction support for XML Gateway Web Services invoking, clear the **Join global transaction** check box, and click **Next**.
17. In the Specify the Location Properties window, select or create a module name, and then click **Finish**.

What to do next

You can test or deploy your module.

Configuring the module for XML Gateway interface for JMS

To configure the module for XML Gateway interface for JMS, use the external service wizard in Integration Designer. You can configure this interface only for the outbound integration. You can select the number of interfaces you would want to create. For each interface you create, you can select the required elements as per your business requirement.

Before you begin

You must place the required DTD files for the XML Gateway interface in a local directory. Also, you need to know the root DTD file and the root element for the selected XML Gateway interface.

About this task

This task is to create an outbound integration application using XML Gateway interface.

Procedure

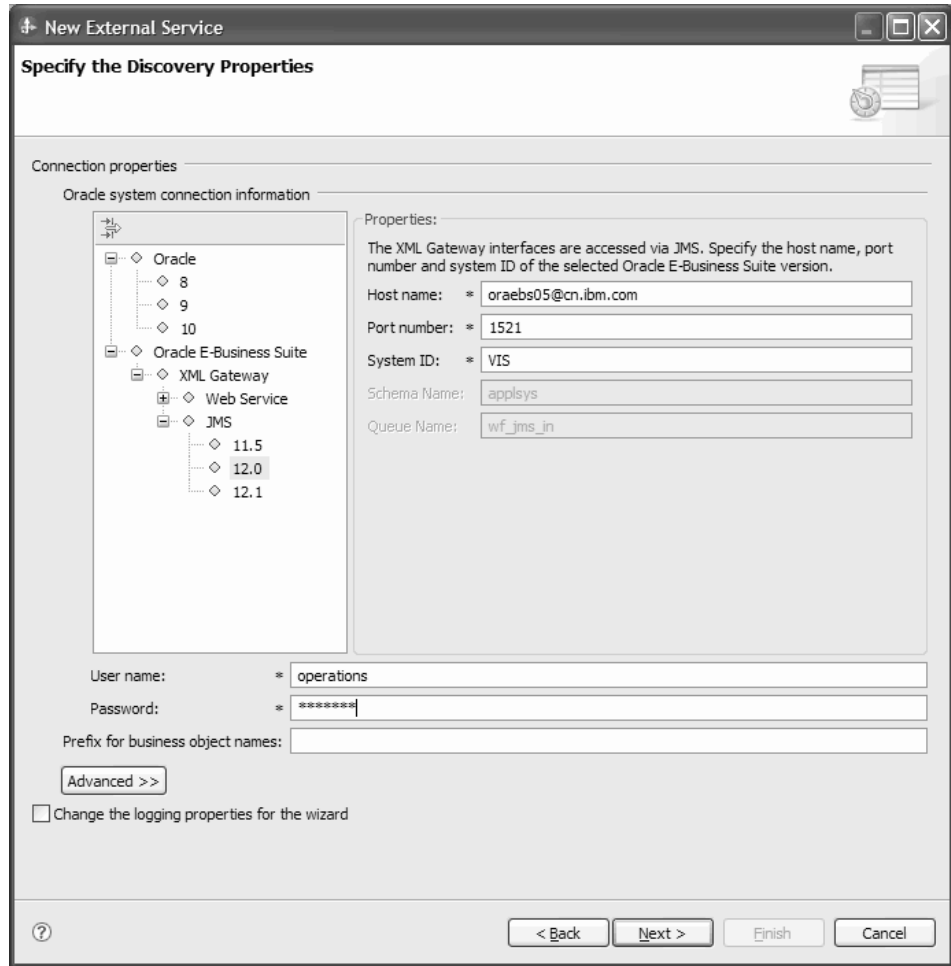
1. Create a project. For information about how to create a project, see “Creating the project” on page 81.
2. In the Locate the Required Files and Libraries window, select the jar files used in JMS runtime. Click **Next**.

Note: The JDBC driver files must be provided only for connecting to the Oracle database server and for the JDBC-based Oracle interfaces.

3. In the Select the Processing Direction window, select **Outbound**. Click **Next**.

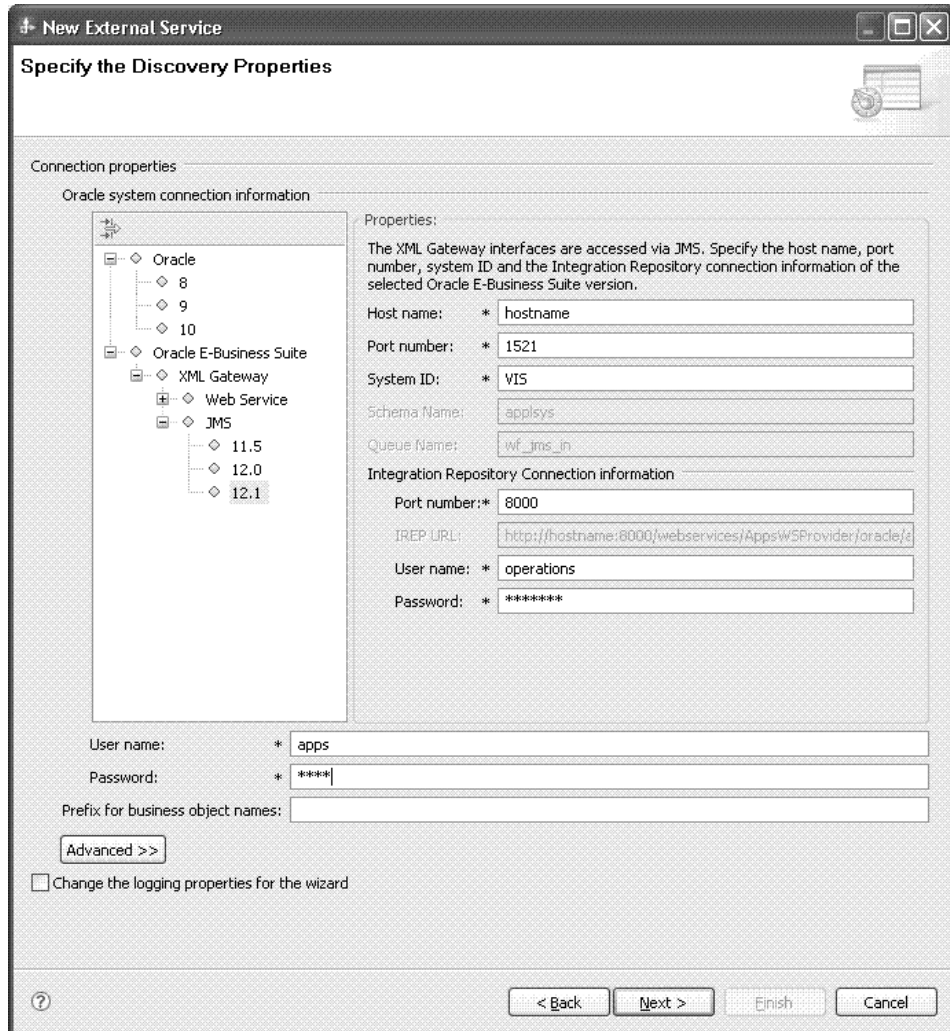
Note: The XML Gateway interface configuration can be performed only for the Outbound processing.

4. In the Specify the Discovery Properties window, select **Oracle E-Business Suite > XML Gateway > JMS**.

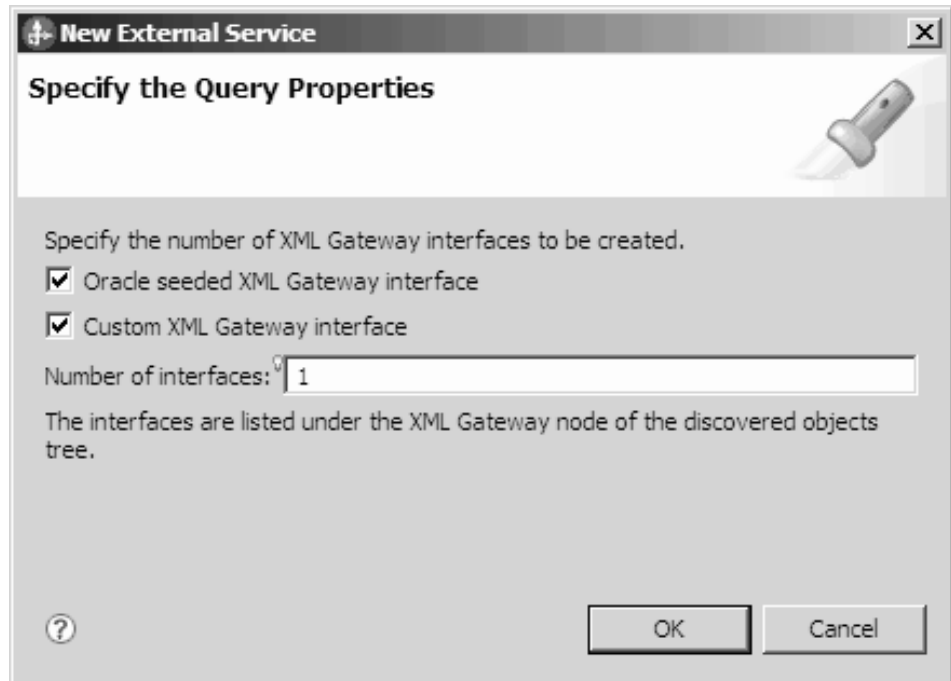


Note: You can select either the **Oracle** database node or the **XML Gateway** node but not both, in one EMD process. When you select the **XML Gateway** node, the **Bidi** and **Advanced** properties are not available for selection.

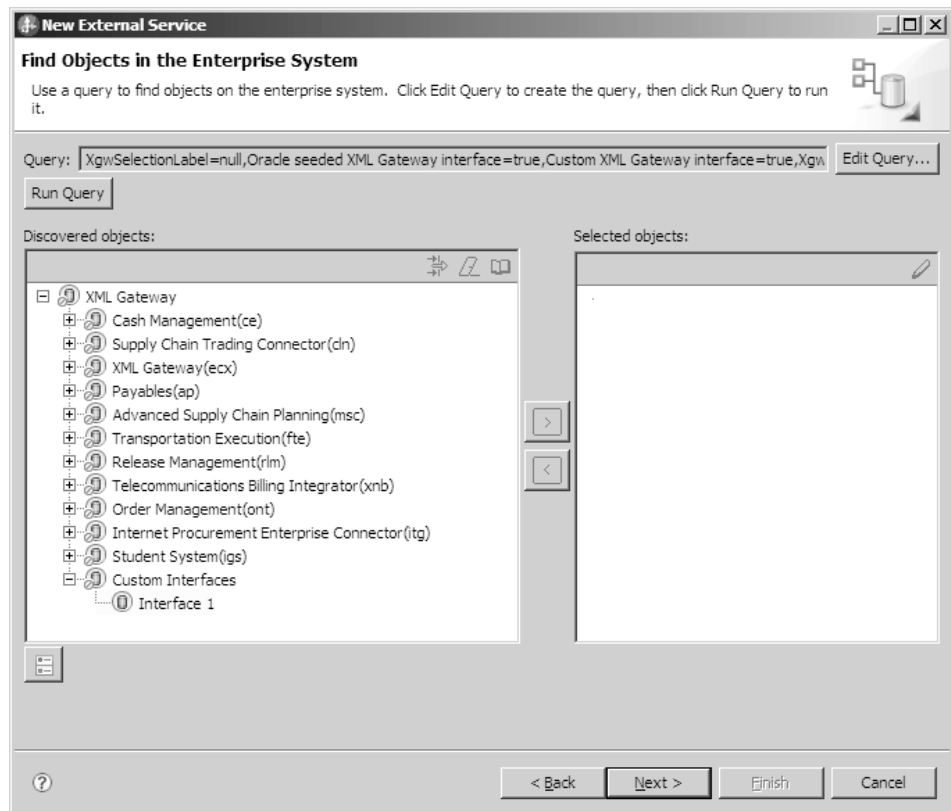
5. Select the Oracle E-Business Suite version, and specify the connection information of the selected XML Gateway interface.
 - a. In the **Host name** field, enter the host name or IP address of the system on which Oracle E-Business Suite is installed.
 - b. In the **Port number** field, specify the port number to connect to the Oracle E-Business Suite.
 - c. In the **System ID** field, specify the system ID (SID) to connect to the Oracle E-Business Suite.
 - d. In the **User name** and **Password** fields, type the user name and password to connect to the Oracle E-Business Suite.
6. For Oracle E-Business Suite V12.1, specify the Integration Repository (IREP) connection information to retrieve the interface metadata.



- a. In the **Port number** field, specify the port number for connecting to the Integration Repository.
 - b. In the **User name** and **Password** fields, type the user name and password to connect to the Integration Repository.
7. Click **Next**.
 8. In the Find Objects in the Enterprise System window, click **Edit Query**.
 9. In the Specify the Query Properties window, you can work with Oracle seeded XML Gateway interface or custom XML Gateway interface or both. To create a custom XML interface, select the **Custom XML Gateway interface** check box, and then type the number of interfaces that you want to create.

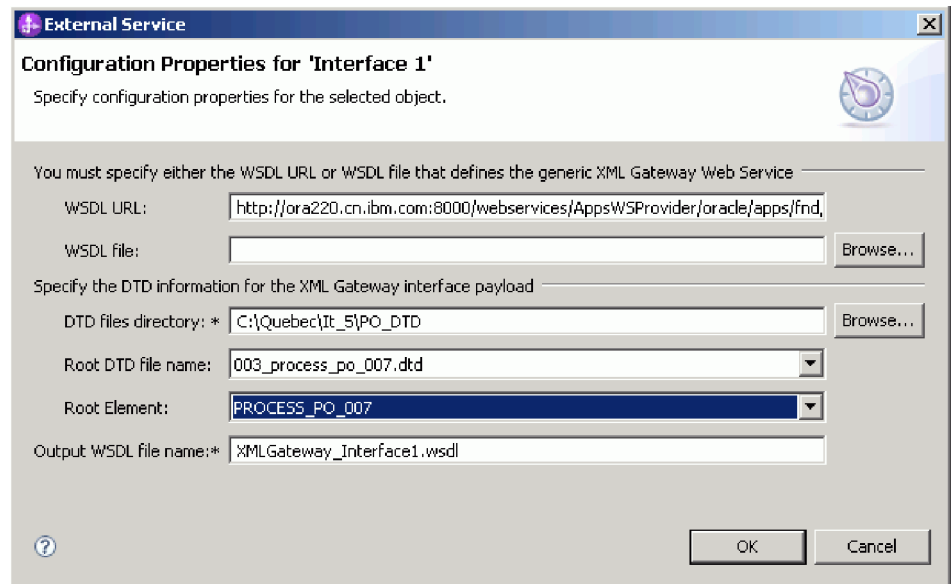


10. Click **OK**, to save your changes to the query. In the Specify the Query Properties window, click **Run Query** to use the query to discover the objects. The result of running a typical query is shown in the following figure.

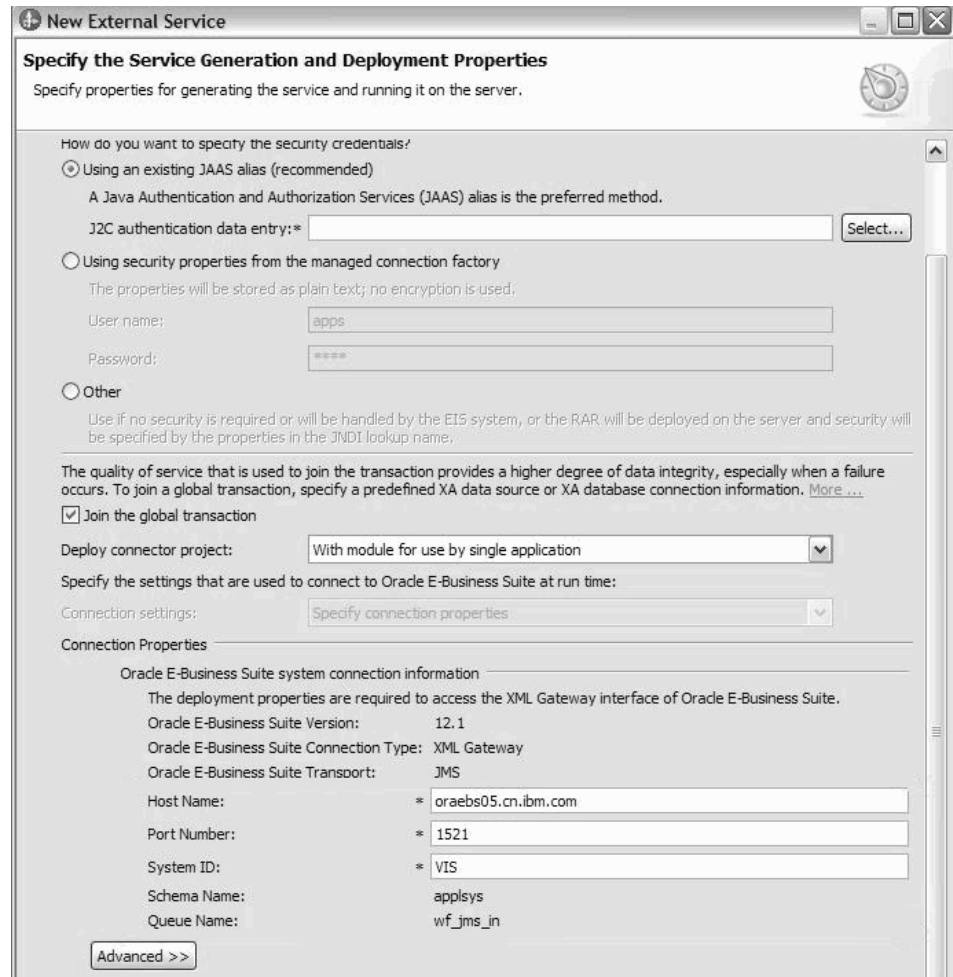


The **Discovered objects** pane lists the different business service interfaces supported by the adapter.

11. Select one or more interfaces from the **Discovered objects** list, and click the > (Add) button to add the interfaces to the **Selected objects** list.
12. In the Specify the Configuration Properties for 'object' window, specify the DTD information.



- a. In the **DTD files directory** field, type the path where the DTD files are located or click **Browse** to search for the location of the files.
- b. From the **Root DTD file name** list, select the root DTD file name from the list.
- c. From the **Root Element** list, select a corresponding root element.
13. Click **OK**.
14. In the Find Objects in the Enterprise System window, click **Next**.
15. In the Specify Composite Properties window, click **Next**.
16. In the Specify the Service Generation and Deployment Properties window, select **Using an existing JAAS alias(recommended)** or **Using security properties from the managed connection factory**.



17. Clear the **Join global transaction** check box, as XML Gateway JMS Services does not support any transaction, and click **Next**.
18. In the Specify the Location Properties window, select or create a module name, and then click **Finish**.

What to do next

You can test or deploy your module.

Configuring the module for advanced queue integration

To configure the module for the advanced queue interface, use the external service wizard in Integration Designer.

Before you begin

You must place the required jar files for the advanced queue interface in a local directory.

About this task

This task is to create an outbound integration application using advanced queue interface.

Procedure

1. Create a project. For information about how to create a project, see “Creating the project” on page 81.
2. In the Select the Processing Direction window, select **Outbound**. Click **Next**.

Note: The advanced queue interface can be executed only for outbound processing.

3. In the Specify the Discovery Properties window, select **Oracle E-Business Suite > Advanced Queue**.

The screenshot shows the 'Specify the Discovery Properties' window in the 'New External Service' wizard. The tree view on the left shows the following structure:

- Oracle
 - 8
 - 9
 - 10
- Oracle E-Business Suite
 - XML Gateway
 - Advanced Queue
 - 11.5
 - 12.0
 - 12.1

The 'Properties' section on the right contains the following fields:

- JDBC driver type: Oracle Thin Driver
- System ID: * VIS
- Host name: * oraeb05@cn.ibm.com
- Port number: * 1521
- JDBC driver class name: * oracle.jdbc.driver.OracleDriver
- Database URL: jdbc:oracle:thin:@oraeb05@cn.ibm.com:1521:VIS

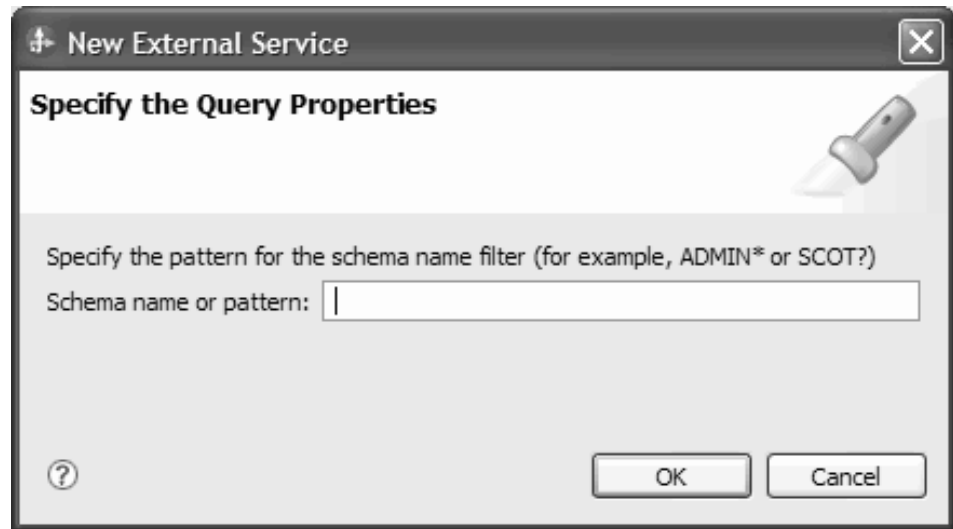
Below the properties section, there are fields for:

- User name: * apps
- Password: * ****
- Prefix for business object names: (empty)

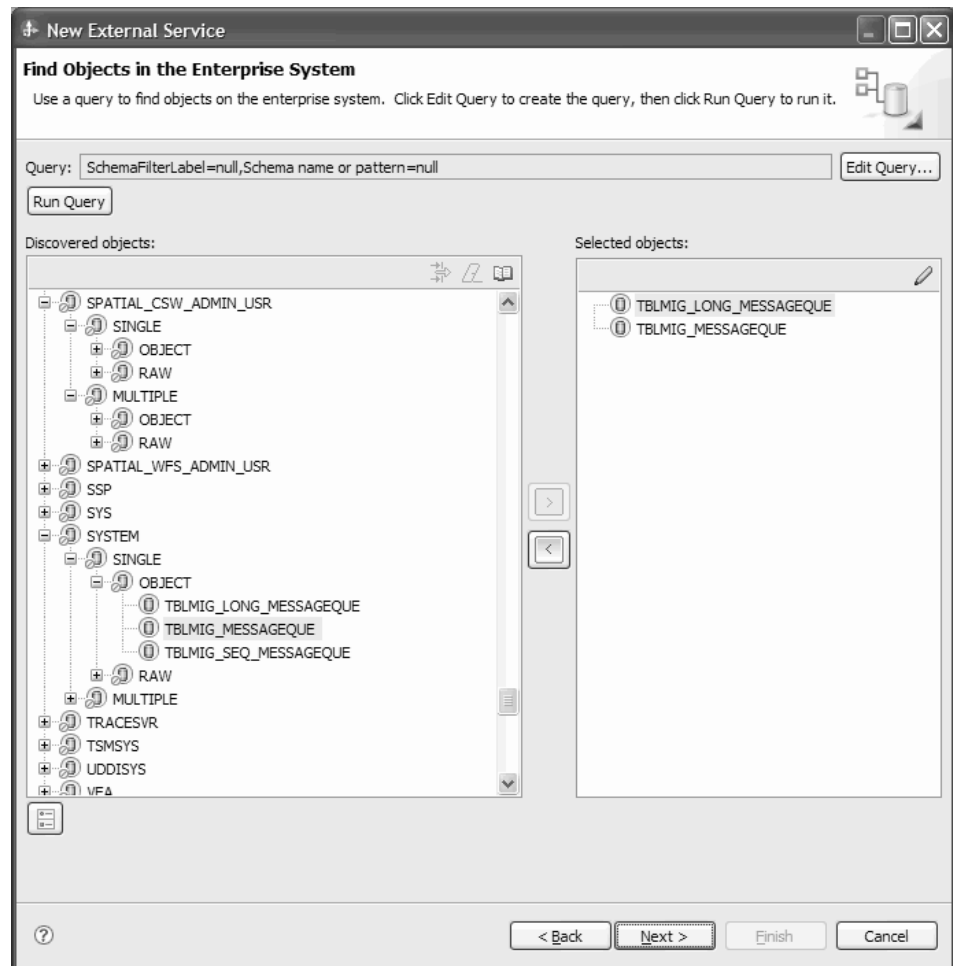
At the bottom, there is an 'Advanced >>' button and a checkbox labeled 'Change the logging properties for the wizard' which is unchecked. Navigation buttons '< Back', 'Next >', 'Finish', and 'Cancel' are located at the bottom right.

4. Select the Oracle E-Business Suite version, and specify the connection information for the advanced queue interface.
 - a. In the **System ID** field, specify the system ID (SID) to connect to the Oracle E-Business Suite.
 - b. In the **Host name** field, enter the host name or IP address of the system on which Oracle E-Business Suite is installed.
 - c. In the **Port number** field, specify the port number to connect to the Oracle database.
 - d. In the **User name** and **Password** fields, type the user name and password to connect to the Oracle database.
5. Click **Next**.
6. In the Find Objects in the Enterprise System window, click **Edit Query**.

7. In the Specify the Query Properties window, specify the schema name or pattern you want to work with.



8. Click **OK** to save your changes to the query. In the Find Objects in the Enterprise System window, click **Run Query** to use the query to discover the objects. The result of running a typical query is shown in the following figure:



The **Discovered objects** pane lists the different business service interfaces supported by the adapter.

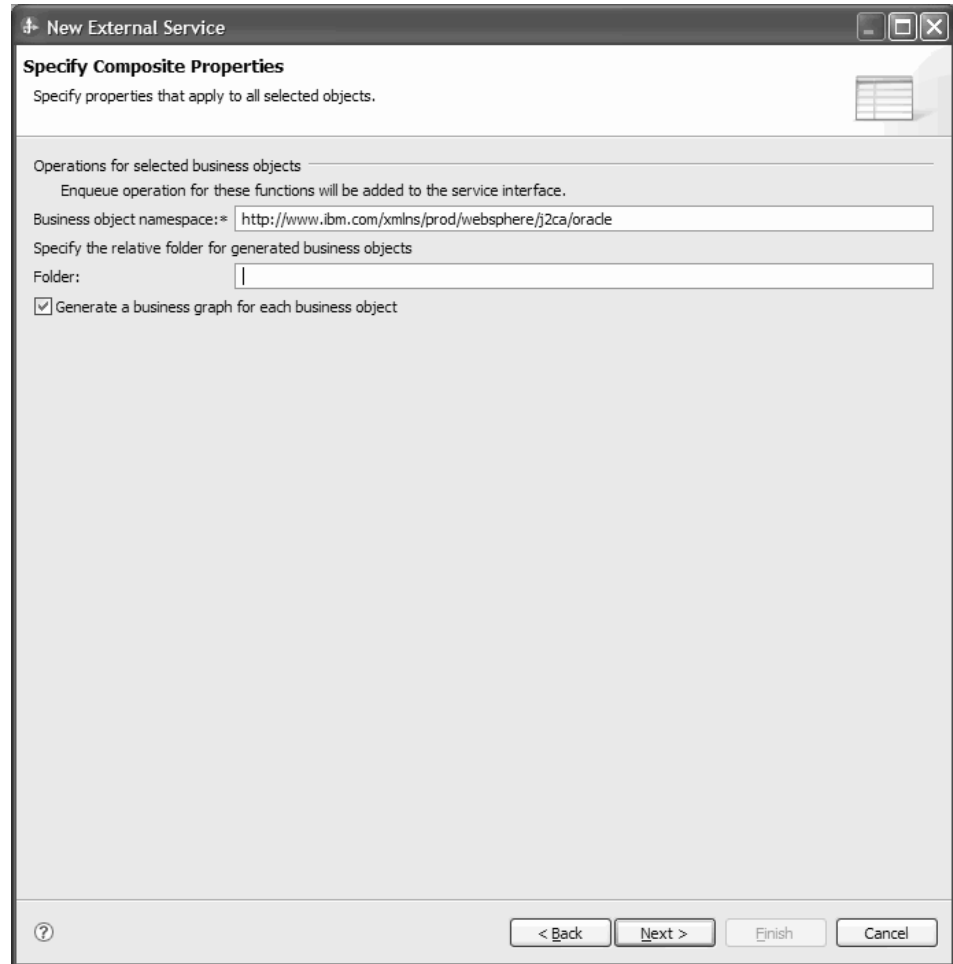
9. Select one or more interfaces from the **Discovered objects** list, and click the > (Add) button to add the interfaces to the **Selected objects** list. Configure the properties for the Advanced queue header. The business objects for the advanced queue payload will be generated with other artifacts. Click **OK**.

The screenshot shows a dialog box titled "New External Service" with a subtitle "Specify the Configuration Properties for 'TBLMIG_LONG_MESSAGEQUE'". The dialog is used to configure an advanced queue. It contains the following fields and values:

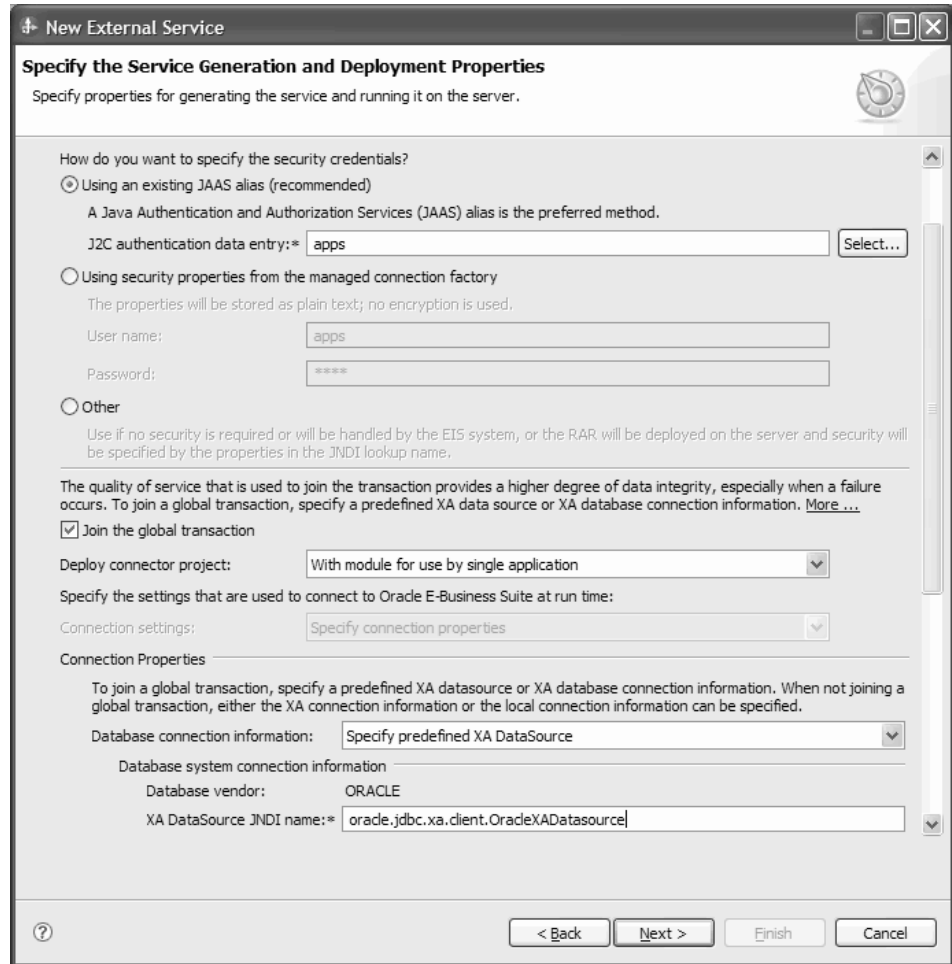
- Business object:** Advanced queue name: SYSTEM.TBLMIG_LONG_MESSAGEQUE
- Queue property:** AQHeader
- AQHeader:**
 - Data type: STRUCT
 - Type name: SYSTEM_TBLMIG_LONG_MESSAGEQUE_Header
- Attributes:**
 - Priority:** Data type: int; Value:
 - Delay:** Data type: int; Value (second):
 - Expiration:** Data type: int; Value (second):
 - Correlation:** Data type: string; Value:
 - RecipientList:** Data type: string; Value:
 - ExceptionQueue:** Data type: string; Value:
- ReturnMessageID:** Data type: string

At the bottom of the dialog, there is a help icon (question mark), an "OK" button, and a "Cancel" button.

10. In the Specify Composite Properties window, specify the business object information for the advanced queue interface.



11. Click Next.
12. In the Specify the Service Generation and Deployment Properties window, select **Using an existing JAAS alias (recommended)** or **Using security properties from the managed connection factory**.



- In the Specify the Location Properties window, select or create a module name, and then click **Finish**.

What to do next

You can test or deploy your module.

Related information

“Advanced Queue integration overview” on page 49

Advanced Queue is an interface type in Oracle E-Business Suite. It allows outbound integration with Oracle E-Business Suite and helps in transferring data to the application. It also interacts with the Oracle advanced queue. The interface services contain a common standard-based approach for integration between Oracle E-Business Suite and other third-party applications (Example: Enterprise Application Integration).

Configuring the module for outbound processing

To configure a module to use the adapter for outbound processing, use the external service wizard in IBM Integration Designer to find and select business objects and services from the Oracle database, and to generate the business object definitions and related artifacts.

Related concepts

“Outbound processing” on page 5

When an application component needs to query the existence of a record in the database or retrieve or modify data in the underlying Oracle database, the adapter acts as the connector between the application component and the database. The adapter provides a set of standard outbound operations, which process either the after-image or delta style business objects. The adapter also supports both local and XA (distributed) transactions for outbound processing.

Discovering database objects

After connecting to the database, run a query to search for database objects. Browse through the discovered objects navigation tree to understand the structure of objects in the Oracle database and use filters to display only the database objects you want to see. Define how many business objects you want to create for user-defined database queries.

Before you begin

You must understand the data requirements of the program that needs to access the database. For example, you need the following information about the database:

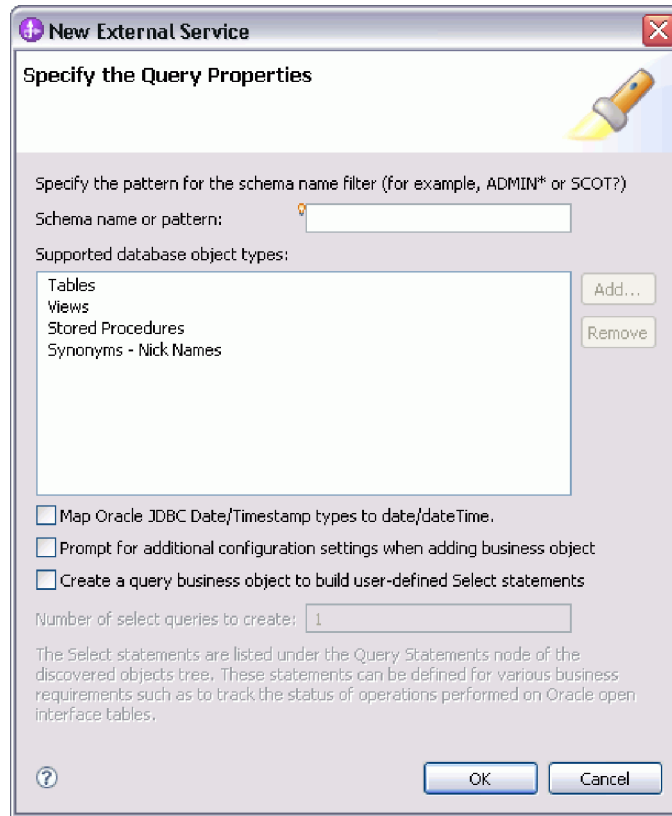
- Which schemas your module needs to access
- What type of database objects you need to access in those schemas
- Which tables, views, synonyms or nicknames, or stored procedures or stored functions you need to access
- How many query business objects you need to create, including parameter values and sample database values for the parameters

About this task

This task starts in the Find Objects in the Enterprise System window of the external service wizard.

Procedure

1. In the Find Objects in the Enterprise System window, click **Edit Query**. The Specify the Query Properties window is displayed.



Use the Specify the Query Properties window to perform the following tasks:

- Reduce the search time by searching a subset of database schemas
 - Omit one or more types of database objects from the search
 - Make the wizard prompt you for application-specific information that cannot be automatically determined based on information in the database
 - Specify the number of query business objects you want to create
 - Map the Oracle data types Date and Timestamp to date and dateTime
2. To limit the number of database schemas that are retrieved, type the name of the schema or a name pattern in **Schema name or pattern**. Use the question mark or underscore (? or _) character to match a single character and the asterisk or percentage sign (* or %) to match multiple characters. Only schemas that start with that string or match that pattern are displayed when you run the query. If you do not specify a schema name pattern, all schemas in the database are displayed. Using a filter can speed up the discovery process if your database contains many schemas.
 3. To omit one or more types of objects from the search, select the types of objects that you want to omit (tables, views, stored procedures and stored functions, and synonyms or nicknames) in **Supported database object types**, and then click **Remove**. If you change your mind, click **Add** to add the object type back. If you need to access only specific types of database objects and speed the discovery process, omit the ones you do not need.
 4. The table, stored procedure, and stored function objects with the date and timestamp data types are mapped to the string data type by default. If you want to map these objects to the actual data types that are supported by the JDBC driver such as the date and datetime data types, select the **Map Oracle JDBC Date/Timestamp types to date/dateTime** check box.

Note: The default data type mapping differs based on the different Oracle JDBC driver versions, for example, the JDBC driver version ojdbc6.jar maps the Date to dateTime data type instead of mapping it to Date. In such cases, the appropriate data type must be manually selected in the Specify the Configuration Properties for 'object' window. If you want to customize the format of the Date and Timestamp data types in the **Application Info** section of the **Properties** view, clear the **Map Oracle JDBC Date/Timestamp types to date/dateTime** check box. Also, ensure that the data types are mapped to the default string data type in the Specify the Configuration Properties for 'object' window.

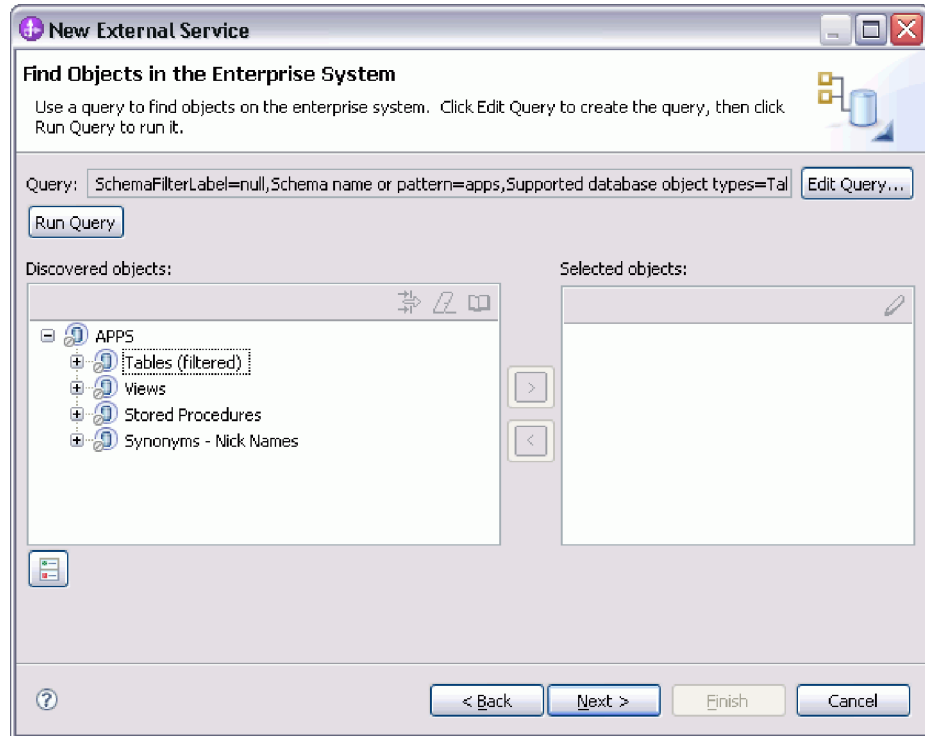
5. Select the **Prompt for additional configuration settings when adding business object** check box. Then, when you add a database object to the list of business objects to create, the wizard automatically prompts you for all user-configurable application-specific information for the object. For example, if you select this option, the wizard guides you through the process of building a simple parent-child hierarchy of business objects. If you need a hierarchy, that a table business object has two attributes referring to attributes in two different tables (that is, it has two parent business objects), complete the configuration in the IBM Integration Designer assembly editor.

Important: If you do not select this option, the wizard prompts only for required information. You must complete the configuration of the business objects using the assembly editor.

6. To create business objects to run the user-defined database queries, select **Create a query business object to build user-defined Select statements** and then type the number of query business objects you want to create. In the **Query Properties** window, specify only the number of query business objects you want to create; the wizard prompts for the name and other details about the business objects at a later time.

You can create user-defined queries for various business requirements such as to track the interface status of the Oracle open interface table. The interface status ensures if the data transferred from the interface tables to the base tables is successful. The error code stored in the interface status column helps in troubleshooting any failure in data transfer.

7. Click **OK** to save your changes to the database query.
8. In the Find Objects in the Enterprise System window, click **Run Query** to use the query to discover database objects and to create a template for query business objects. The result of running a typical query is shown in the following figure.



The **Discovered objects** pane lists the database objects that were discovered.

9. In the **Discovered objects** list, click + (the plus sign) to expand a schema node and then expand **Tables**, **Views**, **Stored Procedures**, and **Synonyms - Nicknames** nodes beneath it, to see the database objects discovered by the wizard, or search a subset of each object type by using filtering.

To limit the number of objects displayed for a particular object type, highlight a node but do not expand it. Click the **Filtering** icon. Use the question mark or underscore (? or _) character to match a single character, and the asterisk or percentage sign (* or %) to match multiple characters. Only object types, such as table or view, which start with that string or match that pattern are displayed under the node.

10. Click + (the plus sign) to expand the node for **Query Statements** to display the template for query business objects.

Results

The wizard displays the database objects you can access using the adapter and business object template for query business objects.

What to do next

Continue working in the external service wizard. The next step is to select the objects you want to use in your module, configure each business object, and create hierarchies of business objects.

Related concepts

“RetrieveAll operation” on page 9

The adapter uses the RetrieveAll operation to retrieve an array of business objects from the database. The adapter uses different processes depending on whether the RetrieveAll operation is for database table business objects or for user-specified SQL business objects.

“Query business object overview” on page 46

Query business objects run a user-defined SELECT statement against the database and return the matching records in business objects.

Selecting and configuring business objects

Using the list of database objects discovered by the external service wizard, and the query object templates you specified, continue using the wizard to select the database objects that you need to access in your module. Then provide the configuration information for your new business objects.

About this task

The Find Objects in the Enterprise System window allows you to select and configure objects in any order, with the exception that you must select and configure a parent table before you can select and configure its child tables. Aside from this restriction, you have the flexibility to add objects individually or to add them several at a time. You can mix objects from the various nodes of the **Discovered objects** list. For example, you can select several table objects, a stored procedure object, and a Query statement, and add and configure them at the same time.

The high-level flow of selecting and configuring business objects is as follows:

1. Select one or more objects in the **Discovered objects** list of the Find Objects in the Enterprise System window.
2. Click the > (Add) button.
3. The wizard opens the Specify the Configuration Properties for 'object' window.
 - If you select a single object, a single Specify the Configuration Properties for 'object' window is displayed.

Complete that window, specifying the user-configurable attributes and other information that the wizard cannot discover by examining the database, and click **OK** to save the configuration.

- If you select multiple objects, the Specify the Configuration Properties for 'object' window displays with one page for each object selected.

Click the name of each object in turn. The window displays the same information you would see had you selected this object individually.

Important: Do not click **OK** until you complete the configuration pages for all the objects. The wizard does not close the notebook until you provide all the required fields, but you can close the window before providing optional fields. If you do not configure the optional fields in the wizard, you must use the business object editor to configure them after exiting the wizard.

4. The wizard adds the configured object to the **Selected objects** list.

As long as you do not exit from the wizard, you can work iteratively to select and configure the business objects you need in your module. However, before you start the wizard to add objects to an existing module, understand the requirements of the program that uses the business objects carefully. The wizard overwrites the

existing business objects in the same path.

Selecting and configuring tables, views, and synonyms or nicknames for outbound processing

Select and configure business objects for tables, views, and synonyms or nicknames to be used in your module. For inbound processing, these are the business objects that are delivered in events.

Before you begin


To perform this task, you need to understand the structure of the data in the database and know what database objects the module needs to access. Specifically, you need to know the following information:

- The structure of the tables, views, and synonyms or nicknames, including the columns you need, and column attributes such as data type
- The relationships between the tables, including the cardinality and ownership of parent-child relationships

About this task

This task is performed through the external service wizard. Start in the Find Objects in the Enterprise System window and then work in a Specify the Configuration Properties for 'object' window that is specific to the business object you are configuring.

Procedure

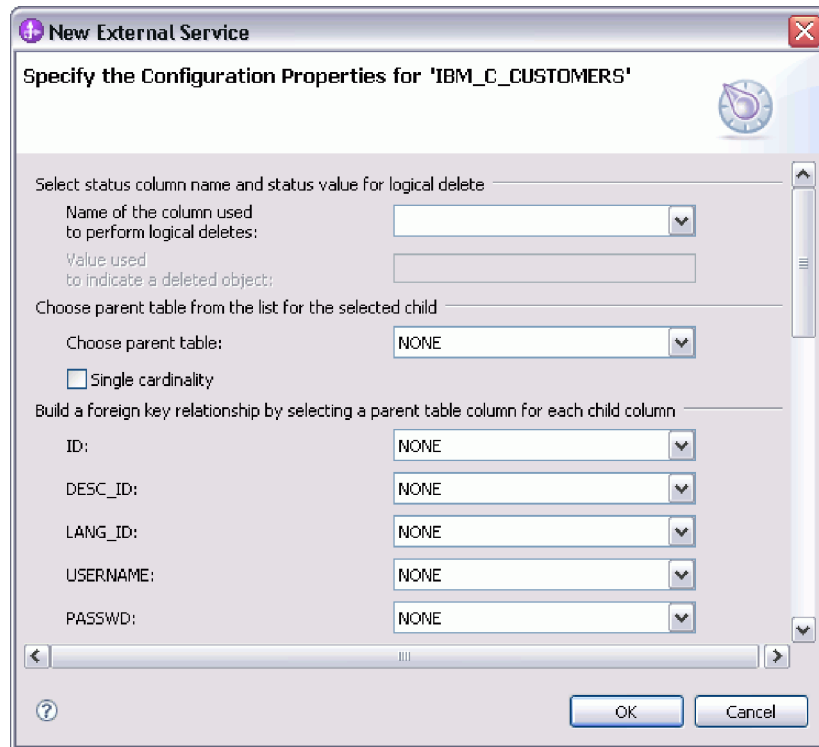
1. In the **Discovered objects** list of the Find Objects in the Enterprise System window, select one or more tables, views, or synonyms from the **Discovered objects** list, and click the > (Add) button to add the object or objects to the **Selected objects** list. Alternatively, you can also filter the tables, views, or synonyms by specifying a valid name or pattern for at least one of the filter fields in the Filter Properties window.
 - a. Click the object you want to filter, and then click the  (Create or edit filter.) button, located at the top of the **Discovered objects** pane.
 - b. In the Filter Properties window, type a name or pattern in the **Object name or pattern** field. Use the question mark or underscore (? or _) to match a single character, and the asterisk or percentage (* or %) to match multiple characters. The name is not case sensitive.
 - c. Click **OK**. The object that matches the given filter condition is displayed.
 - d. Select one or more objects from the discovered list, and click the > (Add) button to add the object to the **Selected objects** list.



The following two figures show a typical Specify the Configuration Properties for 'object' window for a table, view, synonym, or nickname business object. The first figure shows a typical window for the first table or group of tables that are selected.



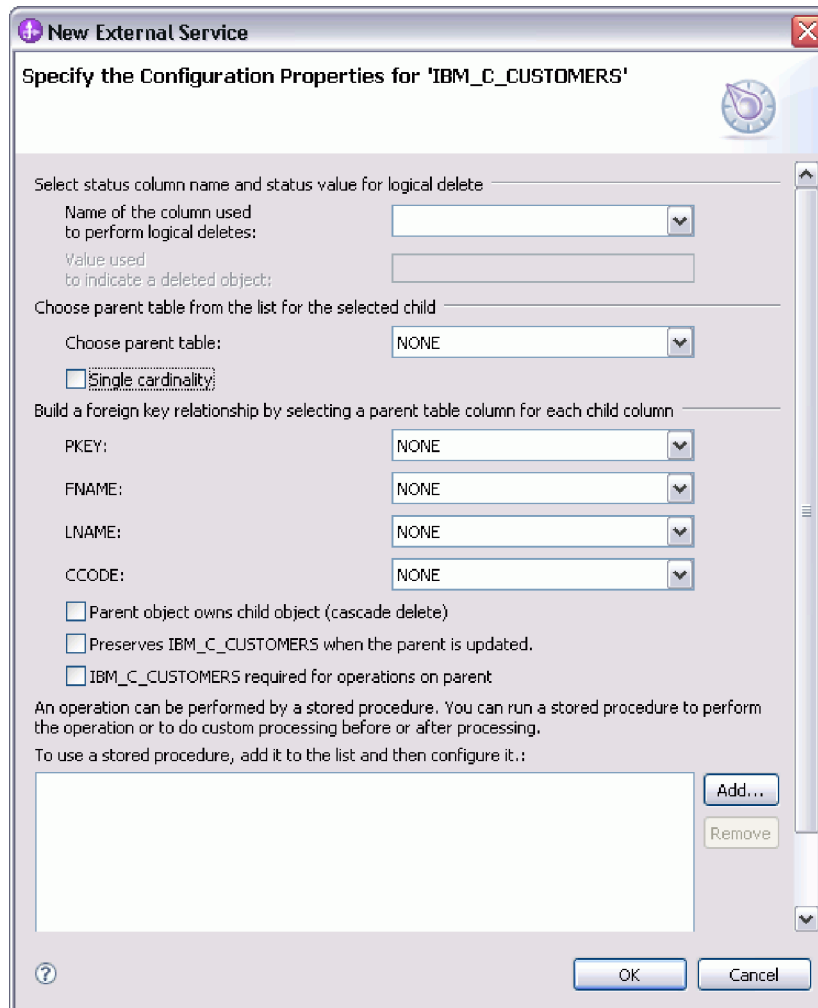
The following figure shows a typical window for subsequent tables you select. After you select and configure at least one table, the Specify the Configuration Properties for 'object' window for subsequent tables displays an area where you can optionally define a parent-child hierarchy between tables.



As you configure the object, choices that require advanced configuration might present additional fields in this window, causing the window to scroll. Ensure that you examine all the fields on the window before clicking **OK**.

2. If the table has a column that is used to indicate logical deletes:
 - a. Select the column name in the **Name of the column used to perform logical deletes** field.
 - b. In the **Value used to indicate a deleted object** field, type the value that indicates that a row is logically deleted. You can get this value from your database administrator.
3. If the **Select primary key for table *table_name*** area is displayed, click **Add**, select the column to be used as the primary key for the table business object, and then click **OK**. If the table has a composite key, you can select multiple columns. The **Select primary key for table *table_name*** area is displayed only when the database table does not have a column designated as the primary key. Each table business object must have a primary key, even if the associated database table does not have a key. If the primary key is defined in the database, this section of the window is not displayed.
4. Optional: Define a parent-child relationship between business objects.
To build a parent-child hierarchy, configure the parent table first, and return to the Find Objects in the Enterprise System window to select and configure the child tables.

Configure the parent-child relationship using the area of the Specify the Configuration Properties for 'object' window shown in the following figure. These fields are not displayed for the first table you configure.



- a. In the **Choose parent table** field, select the name of the parent table you are configuring. If you do not see the parent table in the list, the parent table has not yet been configured. Go back and configure the parent object before configuring the child objects.
- b. Specify the cardinality of the relationship:
 - If the table has a single-cardinality relationship with the parent table, select the **Single cardinality** check box. In a single cardinality relationship, a parent can have only one child business object of this type. A single-cardinality relationship can be used with ownership to represent a true child or without ownership to represent lookup tables or other peer objects in a database.
 - If the table has a multiple-cardinality relationship, do not select the **Single cardinality** check box. In a multiple-cardinality relationship, a parent can have an array of child business objects of this type.
- c. Build the foreign key relationship between the parent and child by specifying for each child column whether it is a foreign key in the parent table.
 - If the child column is not a foreign key, select NONE.
 - If a child column is a foreign key, select the column in the parent table that corresponds to the child column.

Note: The wizard can configure only a single parent table. If the child table has multiple parent tables, you must use the business object editor to configure the remaining parent tables after exiting the wizard.

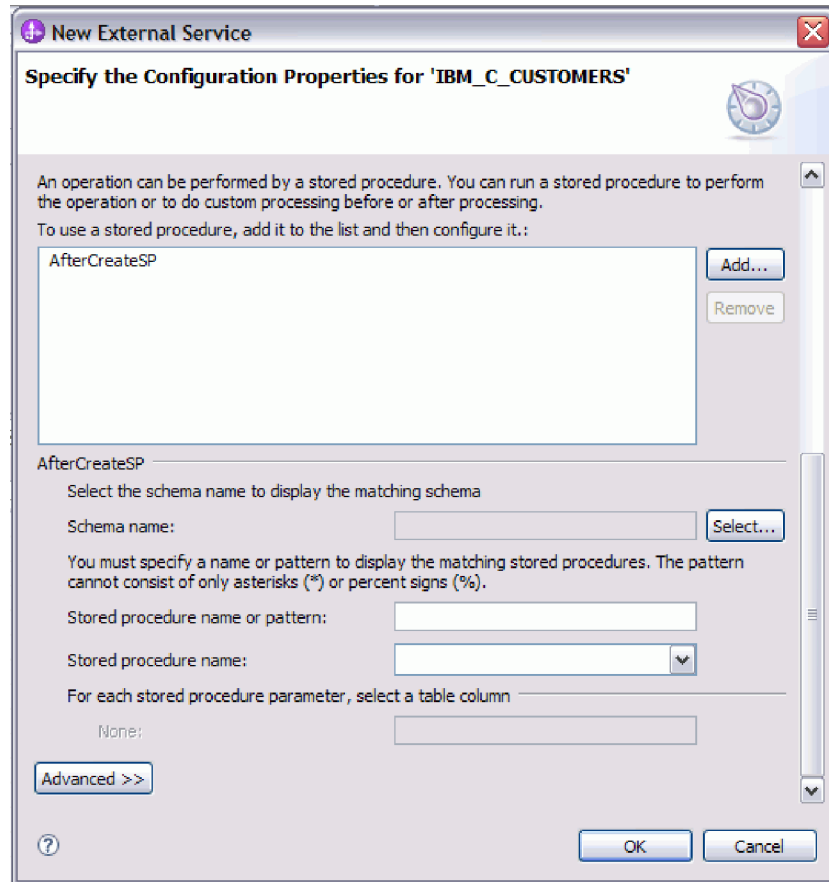
- d. If the parent object owns the child object, then the child objects in the database are deleted when the parent is deleted. To indicate that this child is owned by its parent, select the **Parent object owns child object (cascade delete)** check box. Otherwise, clear this option to prevent child objects, such as lookup tables, from being deleted when their parent is deleted.
 - e. If you do not want the child objects to be deleted as part of an Update operation, select the **Preserves *child_table_name* when the parent is updated** check box.

When a parent table is updated, the adapter compares the child business objects present in the input with the child business objects returned from the database. By default, the adapter deletes any child objects returned from the database that are not present in the input business object.
 - f. By default, you can perform operations on parent business objects without specifying the child business objects. If you want to ensure that a parent business object specifies its child business objects when the parent is submitted for a change, select the ***Child_table_name* required for operations on parent** check box.
5. An operation can be performed using either a standard SQL statement generated by the adapter or using stored procedures or stored functions from the database. If you want to use stored procedures or stored functions:

- a. Click **Add**.
- b. In the Add window, select the type of the stored procedure you want to run. For each operation, you can select a stored procedure that performs the operation, as well as stored procedures that run before or after the operation. For example, for the Create operation, you can specify any of these stored procedures: CreateSP, BeforeCreateSP, and AfterCreateSP.

Note: If you configure the table with RetrieveAllSP, ensure that at least one parameter of the stored procedure is a Cursor and the ResultSet ASI for the stored procedure is set to true to avoid the "No resultset found associated with the stored procedure" exception being generated at run time.

- c. Click **OK**. The Specify the Configuration Properties for 'object' window now shows the stored procedure types you selected and expands to display an area where you configure each one. It might be necessary to scroll down to see the new areas.



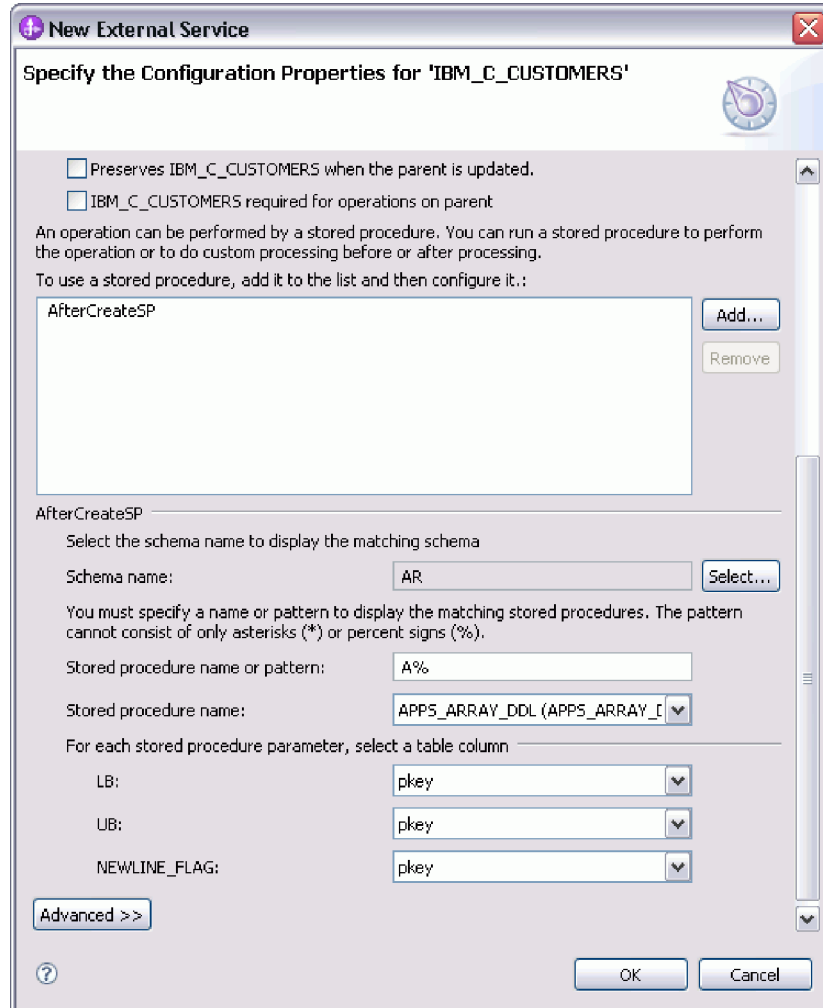
Note: In a hierarchical business object, if you want the stored procedure to be performed for each business object in the hierarchy, you must separately associate a stored procedure with the top-level business object and each child business object or an array of business objects. If you associate a stored procedure with the top-level business object but do not associate it with each child business object, then the top-level business object is processed with the stored procedure, but the child business objects are processed using the standard SQL query.

6. For each stored procedure type that you selected, specify the name of the stored procedure in the database and then configure the business object.
 - a. In the **Schema name** field, select the name of the schema that contains the stored procedure.
 - 1) Click **Select**.
 - 2) In the Select Value window, select the name of the schema you want to work with.
 - 3) Click **OK**.
 - b. Specify the name of the stored procedure or stored function.
 - 1) In the **Stored procedure name or pattern** field, either type the name of the stored procedure or stored function, or type a name pattern. Use the question mark or underscore (? or _) to match a single character, and the asterisk or percentage sign (* or %) to match multiple characters.
 - 2) In the **Stored procedure name** field, select the name of the procedure you want. If the stored procedure list contains many items, the **Select**

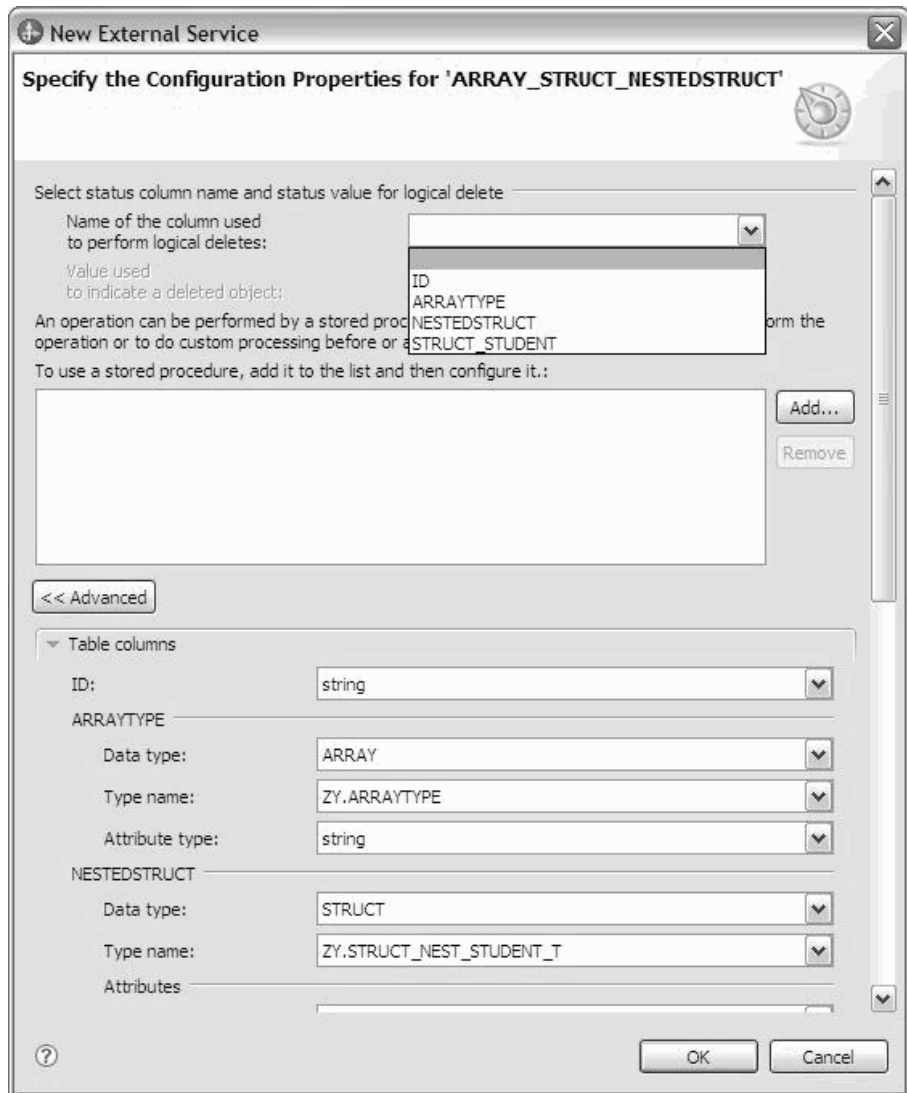
button is displayed next to the **Stored procedure name** field. Click **Select** to open the Select window and select the name of the stored procedure or stored function.

The Specify the Configuration Properties for 'object' window expands to provide an area where you configure the stored procedure. The wizard automatically generates the list of parameters by examining the stored procedure in the database.

- c. For each parameter in the stored procedure (on the left), select the table column (on the right) to pass to the stored procedure in that parameter. The following figure shows a portion of the window after a stored procedure has been configured.




7. To specify the data type mapping for each column in the table:
 - a. Click **Advanced**.
 - b. Expand **Table columns**. For each column in the table, the default data type mapping is displayed. For Oracle databases, if the table contains any complex data type such as an array, structure, nested structure or table, the type name and the child attribute details are also automatically discovered and displayed. The following figure displays the type name and child attribute details of an Oracle table containing complex data types.



c. Review the mapping and make changes if required.

Note: If the primary key in a table is of the date or timestamp type, then the object_key in the event_table must be in the 'yyyy-mm-dd hh-mm-ss' format.

8. When all the fields in the window are completed, click **OK** to save the configuration of the business object. The table, view, synonym, and nickname business objects you defined are now listed in the Find Objects in the Enterprise System window.
9. To change the configuration of an object from the **Selected objects** list, select the object name and then click the  (Edit) icon.
10. When you have selected and configured all business objects that you need, click **Next** to set global properties and configure wrapper business objects.

What to do next

Continue working in the Find Objects in the Enterprise System window to select and configure other types of business objects.

Related concepts

“Business object hierarchies” on page 33

Define the relationships between database tables using parent-child relationships and data ownership in hierarchical business objects.

“Business objects” on page 25

A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. WebSphere Adapter for Oracle E-Business Suite uses business objects to represent tables and views in the database as well as the results of database queries, stored procedures, and stored functions. Business objects can also create a hierarchy of objects from your database and group unrelated tables. Your component communicates with the adapter using business objects.

“Database tables with multiple parent tables” on page 37

If a child table in the database has more than one parent table, you must manually configure the additional parent business objects using the assembly editor. The external service wizard configures only one parent.

Related reference

“Business object information” on page 207

A business object is a structure that contains application-specific information (metadata) about how the adapter must process the business object as well as the operation to be performed on the business object. The name of the business object is generated by the external service wizard in accordance with the naming convention for the adapter.

Selecting and configuring stored procedures and stored functions

To select and configure business objects that correspond to stored procedures and stored functions in the database, you filter the database objects, and specify the configuration properties for the database object.

Before you begin


To select and configure business objects for stored procedures or stored functions, you need to understand the structure of the data in the database and know what objects the module needs to access. In particular, you need to know the parameters passed to the stored procedures or stored functions that your module needs to access.

About this task

This task is performed through the external service wizard. Start in the Find Objects in the Enterprise System window and then work in a Specify the Configuration Properties for 'object' window that is specific to the business object you are configuring.

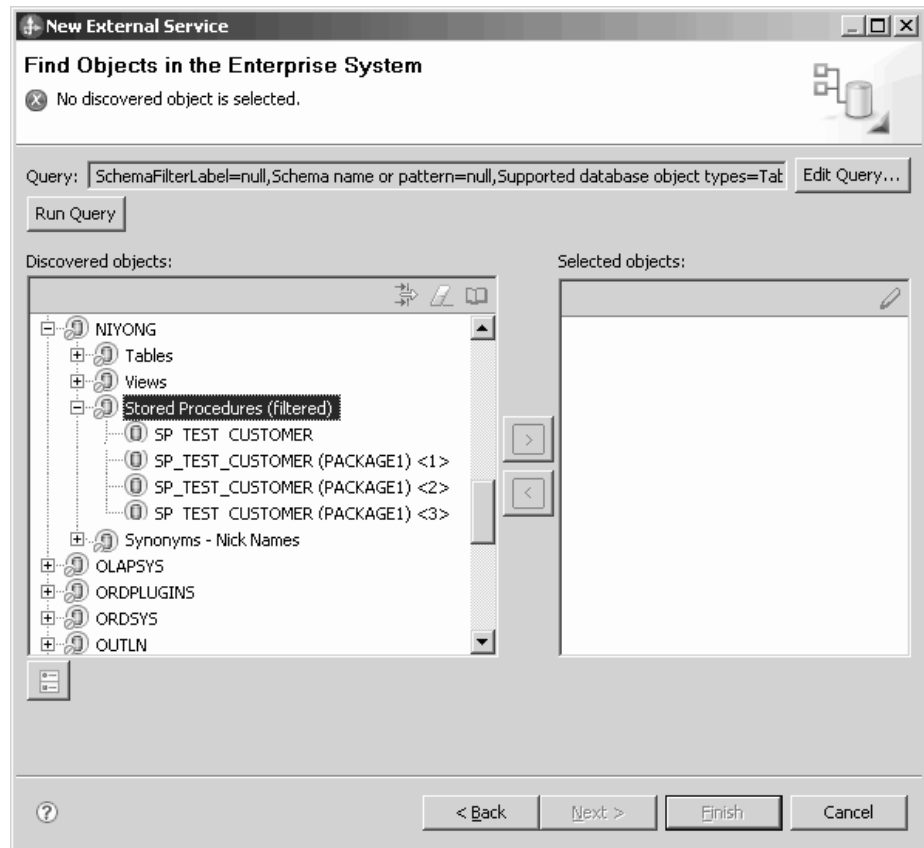
Procedure

1. In the **Discovered objects** list of the Find Objects in the Enterprise System window, expand the node for the schema that contains the stored procedure or stored function you want to work with.
2. Filter the stored procedures by specifying a valid name or pattern for at least one of the filter fields in the Filter Properties window.

- a. Click **Stored Procedures** and then click the  (Create or edit filter) button, located at the top of the **Discovered objects** pane.

- b. In the Filter Properties window, type a name or pattern in the **Object name or pattern** field. Use the question mark or underscore (? or _) to match a single character and the asterisk or percentage (* or %) to match multiple characters. The name is case-sensitive.
- c. In the **Catalog name or pattern** field, type the name or a pattern. Use the question mark or underscore (? or _) to match a single character and the asterisk or percentage (* or %) to match multiple characters.
- d. Click **OK**. The Stored Procedures node displays all the stored procedures that match the given filter condition.

WebSphere Adapter for Oracle E-Business Suite distinguishes the original SP/SF from the overloaded ones by a number tag that corresponds to an overload sequence in the Oracle database. These corresponding parameters for the selected SP/SF will be added based on the overload sequence.



3. Select one or more objects from the **Stored Procedures** list, and click the > (Add) button to add the object to the **Selected objects** list.

Stored procedures that are defined in the PL/SQL packages are displayed in the format *SPName(PackageName)*. For example, if the EMP_MGMT package contains the CREATE_DEPT stored procedure, the stored procedure is displayed in the list as CREATE_DEPT(EMP_MGMT). The Specify the Configuration Properties for 'object' window lists the attributes of the stored procedure business object, which include the names and data types of the parameters of the stored procedure, and information about the result sets that are returned. If the stored procedure contains any complex data type (such as Array, Object, and Table), the type name and the sub attribute details are also automatically discovered and displayed. If the stored procedure contains the Oracle PL/SQL Record data type, the adapter creates a wrapper package with a wrapper stored procedure which converts the Record data type to Object data

type so that the Oracle E-Business Suite adapter can support the PL/SQL Record data type.

4. If the stored procedure returns any result set, make sure that the value for the **The maximum number of ResultSets returned from the stored procedure** field reflects the maximum number expected. The wizard creates the required number of result set business objects to hold the results.
5. Configure each parameter:
 - a. The **Data type** field displays the data type of the parameter.
 - b. In the **Sample Value** field, type a valid value.

Note: If the stored procedure or function contains a VArray or an Object attribute with a Date data type, the sample value must be given in a specific format to successfully validate the stored procedure or function. The Date data type can also be mapped to the String data type. Therefore, the format for the sample value depends on the data type to which the Date is mapped. The following table provides the format to be followed for each Date data type mapping.

Table 13. Date data type formats

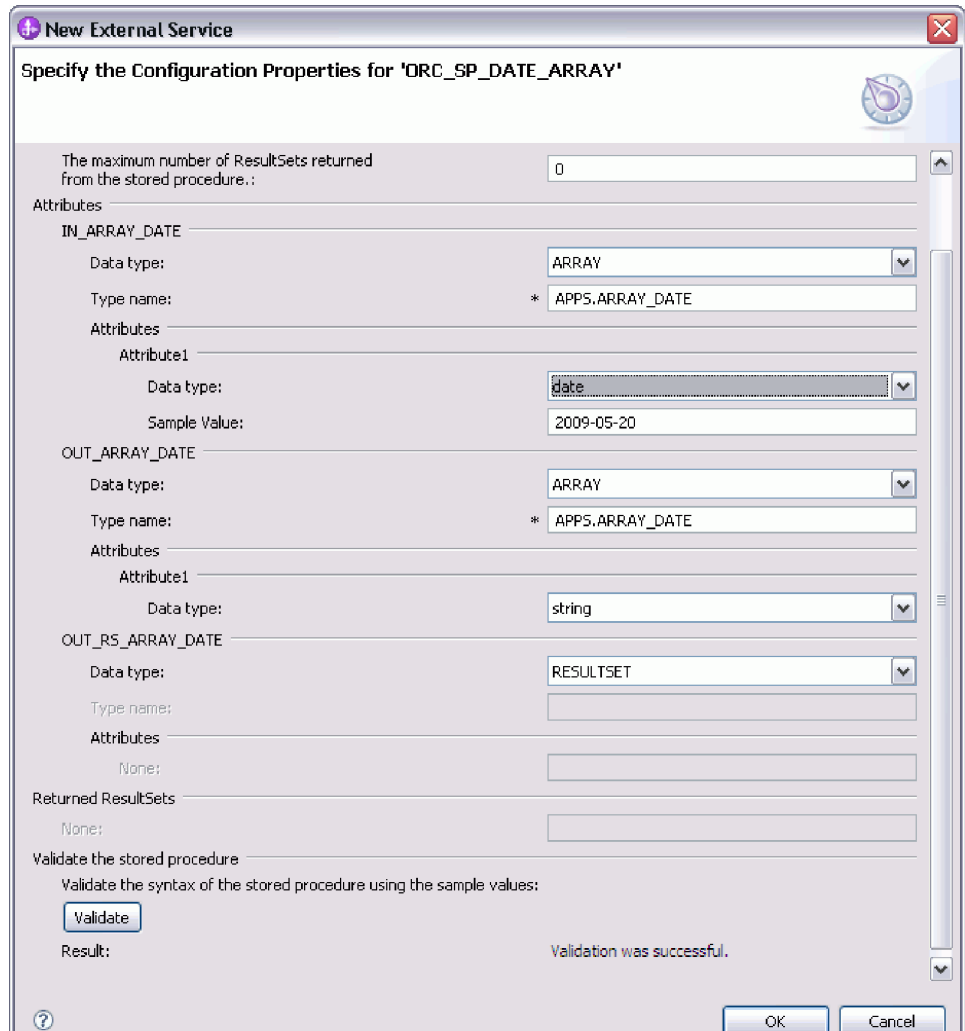
Data type mapping	SP/SF with Date data type	VArray/Object in SP/SF with Date data type
Date to String	Format: Day-Month-Year (Example: '01-JAN-0001')	Format: Year-Month-Day Hour:Minute:Second.[Millisecond] (Example: '0001-01-01 01:00:00.000000000').
Date to Date	Format:Year-Month-Day (Example: '0001-01-01')	Format:Year-Month-Day (Example '0001-01-01')

- The Year parameter can be specified from a minimum of one digit to a maximum of four digits in all the formats.
 - The Month parameter can be specified from a minimum of one digit to a maximum of two digits in all the formats, except in Date to String mapping of the SP/SF with the Date data type where the Month parameter is specified in three alphabetic characters as specified in the example in the previous table.
 - The Day, Hour, Minute, and Second parameters can be specified from a minimum of one digit to a maximum of two digits in all the formats.
 - The Milliseconds, which is an optional parameter, can be specified from a minimum of one digit to a maximum of nine digits in all the formats.
6. To validate the syntax of the stored procedure using the sample values, click **Validate**. The result of the validation is displayed in the **Result** area.

Note: Ensure that the number of result sets is correct after you validate the syntax of the stored procedure because the Oracle driver does not always return the expected result set information. If the number is not correct after validation, set the correct number, and then click **OK** to save and close the window. After you close the wizard, you might verify the setting in the MaxNumOfRetRS application-specific parameter for the business object.


If the **Result** area displays the Validation failed message, there is a problem in the information you provided. Use the error message from the database server, which follows Validation failed message, to correct the definition. Ensure that the data type of the parameters and the sample data are correct. The .log file in the .metadata folder of your workspace contains additional information about the problem.

The following window is displayed after a stored procedure has been validated.



When you see the message **Validation was successful**, click **OK** to save the definition of the stored procedure business object.

Important: If the stored procedure or stored function returns a result set, do not click **OK** until the validation succeeds. The wizard uses the results returned during validation to create business objects to hold the result. If the stored procedure validation is not successful, the adapter does not return the result set at run time.

7. To change the configuration of an object from the **Selected objects** list, select the object name and then click the  (Edit) icon.

Results

The business objects you configured for stored procedures and stored functions are listed in the Find Objects in the Enterprise System window.

What to do next

In the Find Objects in the Enterprise System window, continue to select and configure other types of business objects. When you are finished, click **Next** to set the global properties.

Related concepts

“Business objects” on page 25

A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. WebSphere Adapter for Oracle E-Business Suite uses business objects to represent tables and views in the database as well as the results of database queries, stored procedures, and stored functions. Business objects can also create a hierarchy of objects from your database and group unrelated tables. Your component communicates with the adapter using business objects.

“Database tables with multiple parent tables” on page 37

If a child table in the database has more than one parent table, you must manually configure the additional parent business objects using the assembly editor. The external service wizard configures only one parent.

Related reference

“Business object information” on page 207

A business object is a structure that contains application-specific information (metadata) about how the adapter must process the business object as well as the operation to be performed on the business object. The name of the business object is generated by the external service wizard in accordance with the naming convention for the adapter.

Selecting and configuring query business objects

Select and configure query business objects for user-defined SELECT statements for use in your module.

Before you begin

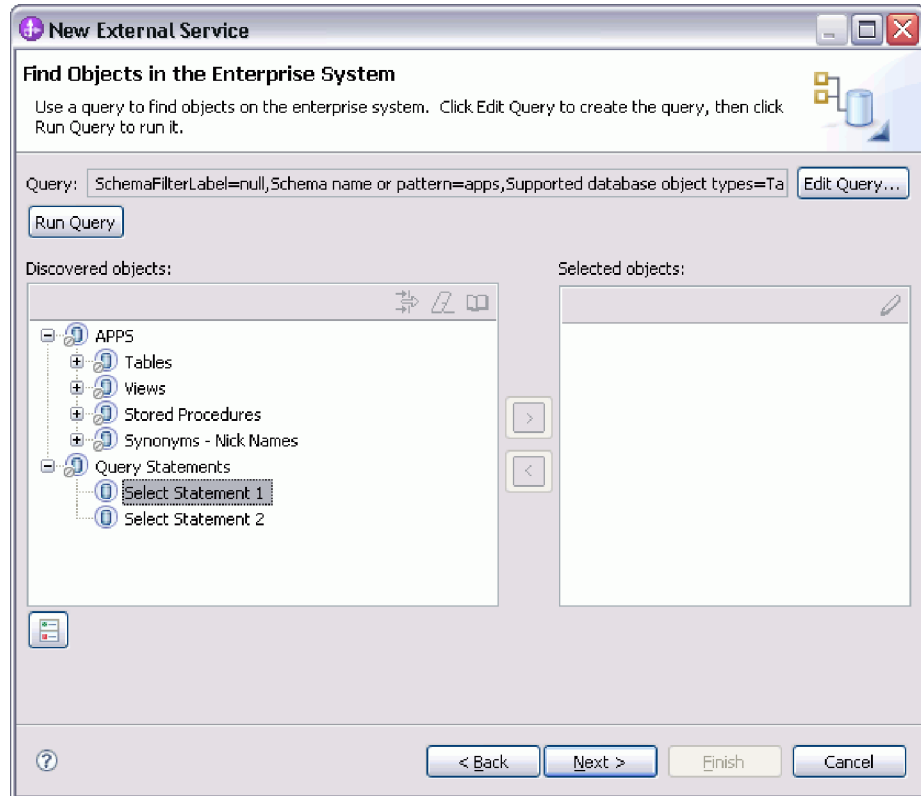
To configure query business objects, you must know the structure of the data in your database, including the tables and views. You need to know the name and data type of the columns that your module needs to access. You must also be able to write the SQL SELECT statements.

About this task

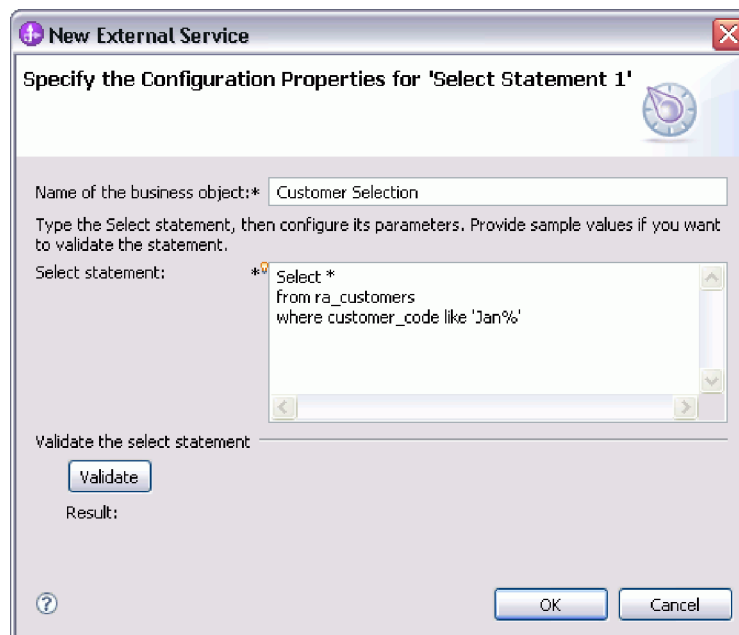
This task is performed through the external service wizard. Start in the Find Objects in the Enterprise System window and then work in a Specify the Configuration Properties for 'object' window that is specific to the business object you are configuring.

Procedure

1. In the **Discovered objects** list of the Find Objects in the Enterprise System window, expand the **Query Statements** node. This node contains an object template, named **Select Statement *n***, for each query business object you requested in the Specify the Query Properties window. For example, if you specified a count of two query business objects in that window, the **Discovered objects** list contains two object templates, as illustrated in the following figure.



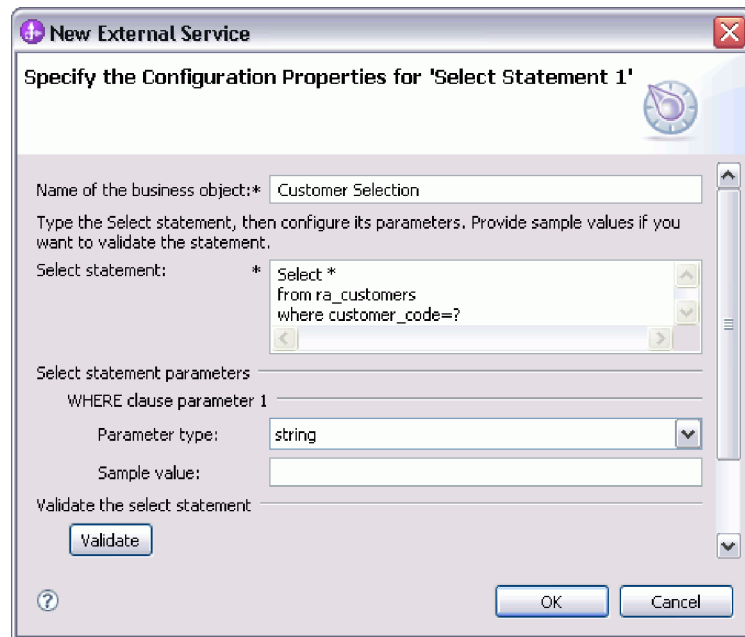
2. Select one or more of the object templates and click the > (Add) button to add the objects to the **Selected objects** list. The following figure displays the Specify the Configuration Properties for 'object' window that opens when you click > (Add) for a query business object.



3. In **Name of the business object**, type a name for the business object. The name can contain spaces and national language characters.

4. In **Select statement**, type the SELECT statement you want to run. Indicate each parameter with a question mark (?). The following sample SELECT statements illustrate the flexibility of the query business object:
 - select * from customer where ccode=?
 - select * from customer where id=? and age=?
 - select * from customer where lname like ?
 - select C.pkey, C.fname, A.city from customer C, address A WHERE (C.pkey = A.custid) AND (C.fname like ?)

As you type each ?, the window expands to display an area where you define the WHERE clause for that parameter. The following figure displays the Specify the Configuration Properties for 'object' window for a query business object that has a single parameter.



5. In **Where clause parameter *n***, provide information about each parameter in the SELECT statement.
 - a. In **Parameter type**, select the data type of the parameter. For Oracle databases, the adapter does not support the complex types such as array, table, structure, or nested structure as parameters in the query business objects.
 - b. In **Sample value**, type a sample value for the parameter.

For example, for a parameter corresponding to a column containing the family name of the customer, you might select string as the data type and provide a sample value of Smith.
6. Click the **Validate** button to validate the syntax of the select statement using the sample values. **Result** displays the result of the validation.

If **Result** displays Validation failed, there is a problem in the information you provided. Use the error message from the database server, which follows Validation failed, to correct the definition. Check the syntax of the SELECT statement, the data type of the parameters, and the sample data.
7. To specify the data type mapping for each column in the result set returned by the select statement:
 - a. Click **Advanced**.

- b. Expand **Result set returned by the Select statement**. For each column in the result set, the default data type mapping is displayed. For Oracle databases, if the query result contains any complex data type, such as an array, structure, nested structure or table, the type name and the child attribute details are also automatically discovered and displayed.
 - c. Review the mapping and make changes if required.
8. Click **OK** to save the definition of the query business object.

Results

The query business objects you defined are now listed in the Find Objects in the Enterprise System window.

What to do next

In the Find Objects in the Enterprise System window, continue to select and configure other types of business objects. When you are finished, click **Next** to set the global properties.

Related concepts

“Query business object overview” on page 46

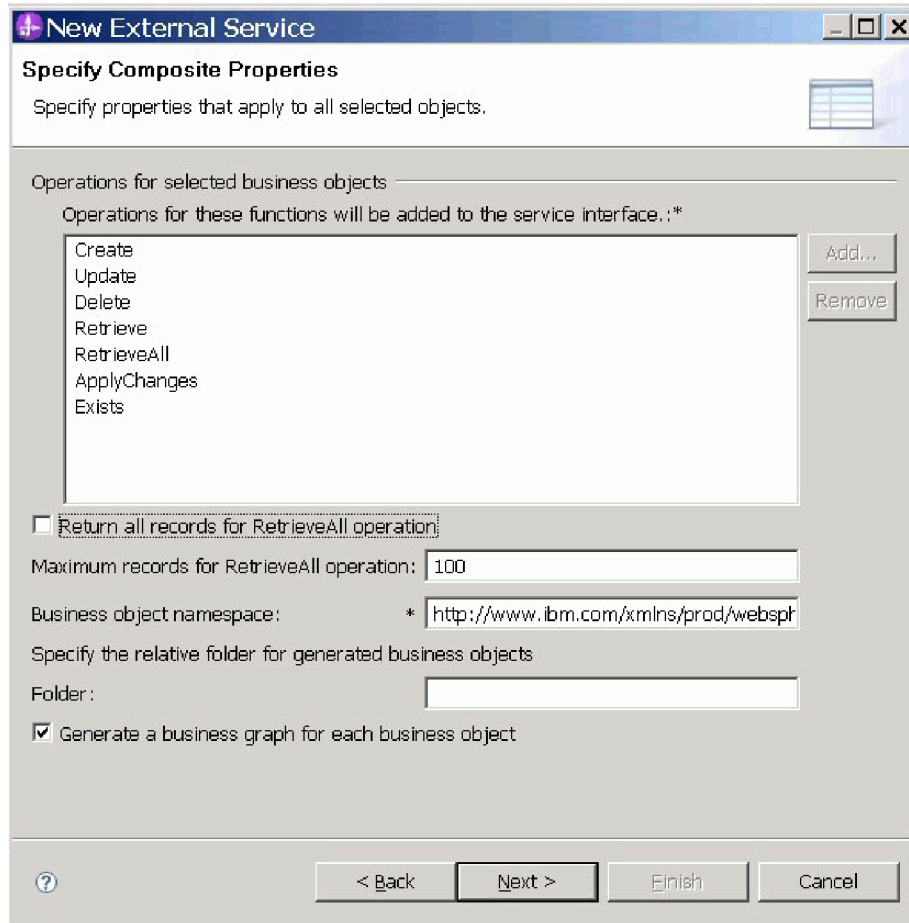
Query business objects run a user-defined SELECT statement against the database and return the matching records in business objects.

Setting global properties for operations

After you select database objects in the external service wizard, you need to specify properties that apply to all business objects.

Procedure

1. When the **Selected objects** list in the Find Objects in the Enterprise System window contains all the business objects you want to use in your application, click **Next**.
2. In the Specify Composite Properties window, review the list of operations.
This window lists all the operations that the adapter supports for the outbound services for all business objects that you selected on the previous window. Not all operations are supported by each business object. For example, query business objects support only the RetrieveAll operation. Stored procedure business objects support only the Execute operation.



3. To remove an operation that you do not need, select the operation name and click **Remove**. If you change your mind, click **Add** and restore a removed operation.
4. Specify how you want to retrieve the records.
 - If you want the RetrieveAll operation to return all records matching the query, select the **Return all records for RetrieveAll operation** check box or enter -1 in the **Maximum records for RetrieveAll operation** field.
 - If you want to specify the maximum number of records, the RetrieveAll operation must return, enter a value in the **Maximum records for RetrieveAll operation** field. The default value is 100. For more information about this property, see “Maximum records for RetrieveAll operation” on page 249.

Note: The “Maximum records for RetrieveAll operation” on page 249 property applies only if you are using the RetrieveAll operation. This property field is disabled if you remove the RetrieveAll operation in step 3 or if you select the **Return all records for RetrieveAll operation** check box.

5. In **Business object namespace**, accept the default namespace or type the full name of another namespace.
The namespace is prefixed to the business object name to keep the business object schemas logically separated.
6. Optionally, in **Folder**, type the relative path to the folder where the generated business objects are to be stored.

Note: If you are creating multiple adapter artifacts within a module, ensure that you specify different business object folders for each adapter within the module. For example, if you are creating artifacts for Oracle, JDBC, SAP, and JDE within a module, you need to create different relative folders for each of these adapters. If you do not specify different relative folders, the existing artifacts are overwritten when you generate new artifacts.

7. If you want a business graph to be created for each business object, click **Generate a business graph for each business object**. Business graphs are needed only in these situations:
 - If you need to use the ApplyChanges operation
 - When adding business objects to a module created with a version of IBM Integration Designer earlier than version 7.0.

Note: You must select this option if you are adding business objects to a module that was created with an earlier version of IBM Integration Designer. Otherwise, you must rewire your interface.

8. Click Next.

Results

You have provided information that applies to all business objects in the module.

What to do next

Continue working in the wizard. The next step is to specify deployment information to use at run time and information for saving the service as a module.

Setting deployment properties and generating the service

After you select and configure the business objects for your module, use the external service wizard to configure properties that the adapter uses to connect to a specific database. The wizard creates a new business integration module where all the artifacts and property values are saved.

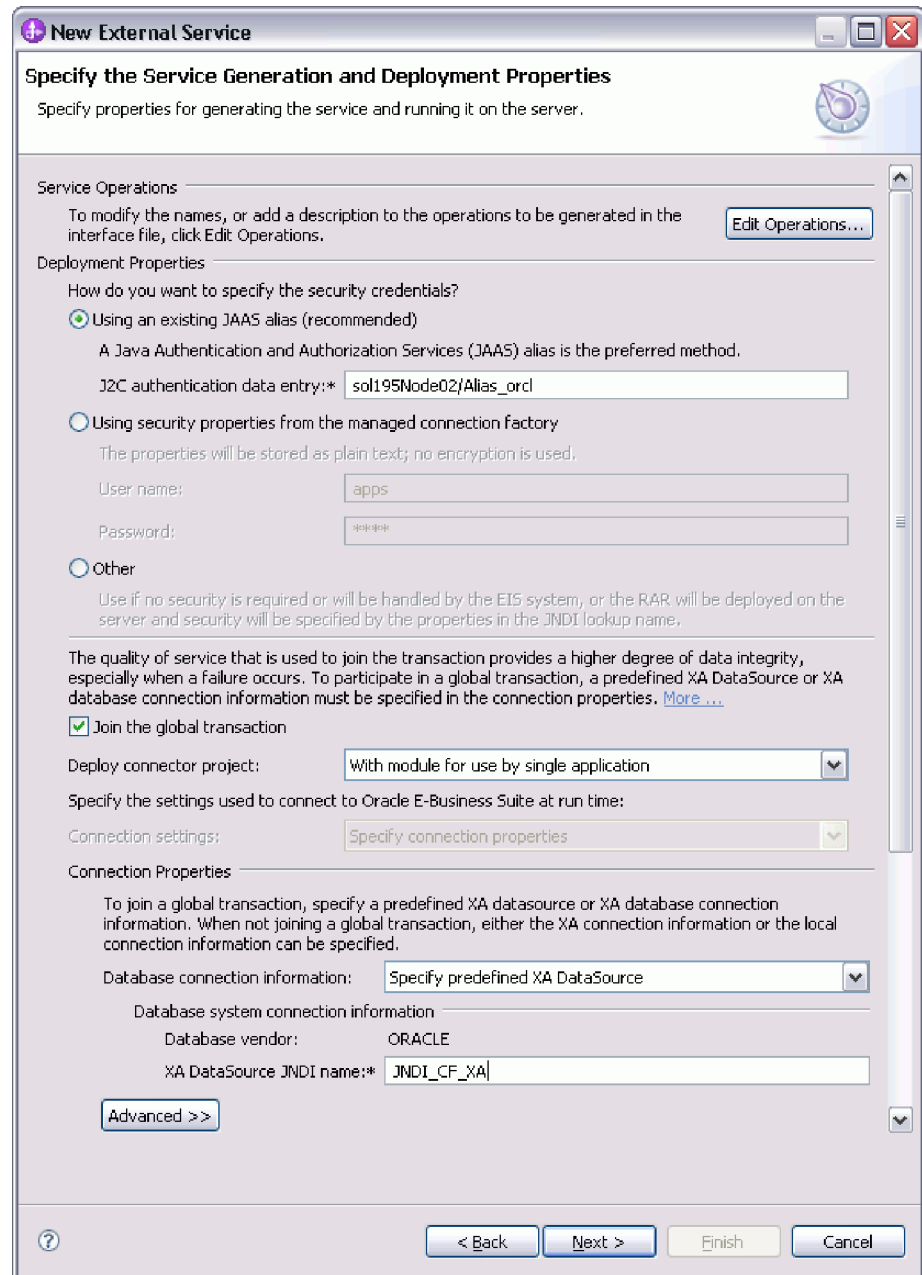
About this task

This task is performed through the Specify the Service Generation and Deployment Properties and Specify the Location Properties windows of the external service wizard.

The connection properties in this task are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change the values here. For example, to use the Oracle JDBC driver at run time instead of the Oracle Thin driver, set the driver information here.

Procedure

1. In the Specify the Service Generation and Deployment Properties window, click **Edit Operations** to review the names of the operations or add a description for the operations for the business objects you are creating.



2. In the **Deployment Properties** area, specify how you want the adapter to get the user name and password at run time.
 - To use a J2C authentication alias, click **Using an existing JAAS alias (recommended)** and enter the name of the alias in the **J2C authentication data entry** field. You can specify an existing authentication alias or create one at any time before deploying the module. The name is case-sensitive and includes the node name.
 - To use the managed connection properties, click **Using security properties from the managed connection factory** and enter values in the **User name** and **Password** fields.
 - To administer the user name and password from another mechanism, click **Other**.

Note: If you use local connection information to establish the database connection, a security credential is required. You can select either **Using an existing JAAS alias (recommended)** or **Using security properties from the managed connection factory** as the security mechanism. If you use an existing data source on the server, the security credential is not required. In this case, you can select **Other** as the security mechanism. Also, if you set the **J2C authentication data entry** field or set the **User name** and **Password** fields, they override the user name and password in the data source.

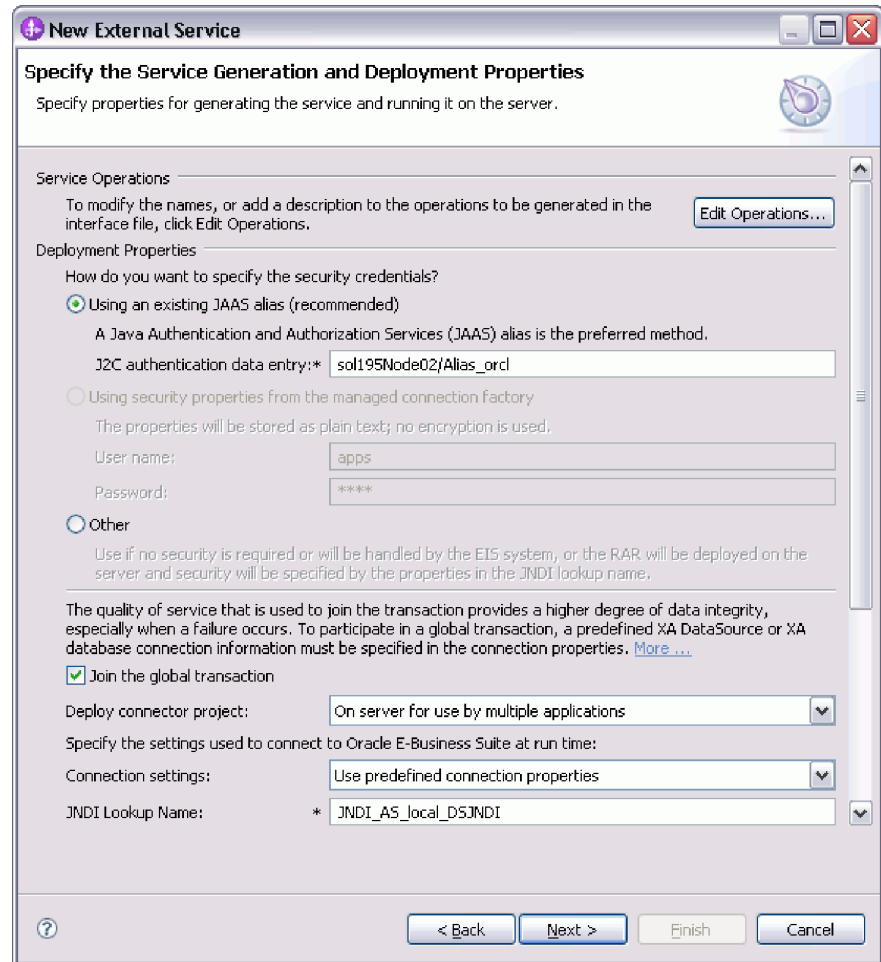
3. By default, the adapter is configured to join the global transaction. The adapter uses the XA connection for global transaction. You can configure a XA connection by specifying either a predefined XA data source or XA database connection information. For a local transaction, clear the **Join the global transaction** check box. You can configure a local transaction by specifying either a predefined connection pool data source, local database connection information, predefined XA data source, or XA data source connection information.

Note: If you select the **Join the global transaction** check box, the **Database connection information** list displays only the XA related database connection options. If an artifact from an earlier version contains only the local connection properties and the **Join the global transaction** check box was selected, after migration, the **Database connection information** list in the assembly editor displays the option for local database connection.

Note: If you do not specify a valid connection information for global transaction, the adapter generates the ResourceException.

4. In the **Deploy connector project** field, specify whether to include the adapter files in the module. Select one of the following values:
 - **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.
5. If you selected **On server for use by multiple adapters** in the previous step, specify the connection properties used at run time.
 - If you manually created and configured a managed connection factory or activation specification on the server or if you have already deployed an application that connects to the same database using the same managed connection factory or activation specification properties, you can reuse the managed connection factory or activation specification by specifying the name of its Java Naming and Directory Interface (JNDI) data source:
 - a. From the **Connection settings** list, select **Use predefined connection properties**.
 - b. In the **JNDI Lookup Name** field, type the name of the JNDI data source for an existing managed connection factory or activation specification.

The following figure displays the typical settings for reusing a managed connection factory or activation specification for a stand-alone deployment of the adapter.



- c. Click **Next** to complete this task.
 - If this is the first application that connects to the database with a specific user name and password, or if you want to administer the user name and password separately from other applications, select **Specify connection properties**.
6. In the **Connection Properties** area, specify how you want the adapter to establish database connection at run time. You can establish connection with the adapter at run time in one of the following ways:
 - To use a predefined XA data source on the server (for XA connection):
 - a. Select the **Join the global transaction** check box.
 - b. From the **Database connection information** list, select **Specify predefined XA DataSource**.
 - c. In the **Database system connection information** area, enter the value in the **XA DataSource JNDI name** field. This value must be set to a JNDI data source that supports XA transactions that is created on the IBM Business Process Manager or WebSphere Enterprise Service Bus. For more information about this property, see “XA DataSource JNDI name (XADataSourceJNDIName)” on page 244.
 - To specify the connection information to be saved in the adapter properties (for XA connection):

- a. Select the **Join the global transaction** check box.
 - b. From the **Database connection information** list, select **Specify XA database connection information**.
 - c. In the **Database system connection information** area, enter values in the **XA DataSource name** and **Database URL** fields. For more information about these properties, see “Database URL (DatabaseURL)” on page 233 and “XA DataSource name (XADataSourceName)” on page 239.
- To use a predefined connection pool data source (for local connection):
 - a. Clear the **Join the global transaction** check box.
 - b. From the **Database connection information** list, select **Specify predefined connection pool DataSource**.
 - c. In the **Database system connection information** area, enter the name of the existing JNDI data source in the **Connection pool DataSource JNDI name** field. For more information about this property, see “Connection pool DataSource JNDI name (PoolDataSourceJNDIName)” on page 243.
 - To specify connection information to be saved in the adapter properties (for local connection):
 - a. Clear the **Join the global transaction** check box.
 - b. From the **Database connection information** list, select **Specify local database connection information**.
 - c. In the **Database system connection information** area, enter values in the **Database URL** and **JDBC driver class name** fields. For more information about these properties, see “Database URL (DatabaseURL)” on page 233 and “JDBC driver class (JDBCDriverClass)” on page 235.
7. Review and, if necessary, change the values of the required connection properties. The fields are initialized with the connection information you specified when you started the wizard. You can change the values to specify a different user name and password at run time. You can also connect to an alternate Oracle database, although the schema names must be the same in both databases. The format of the connection properties is database-specific. For more information about the properties, see “Managed connection factory properties” on page 230.
 8. Optional: Click **Advanced** to specify the advanced properties. Expand each of the advanced sections to review the properties.
 - **Additional connection configuration**

Additional connection configuration

Set auto commit on database connection

Additional JDBC driver connection properties [name:value;name:value]:

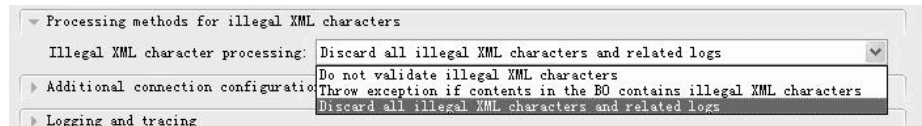
SQL query to verify the connection:

Query timeout (seconds):

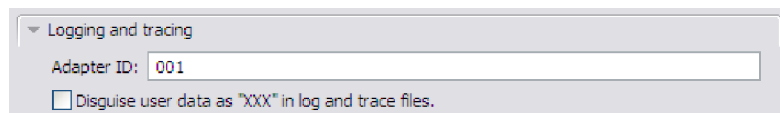
Return business object even when the stored procedure result set is empty:

- a. If you want to turn on AUTOCOMMIT for the database, select **Set auto commit on database connection**. For more information about the property, see “Auto commit (AutoCommit)” on page 233.
- b. Set **Additional JDBC driver connection properties**. For more information about the property, see “Additional JDBC driver connection properties [name:value;name:value] (JDBCDriverConnectionProperties)” on page 233.

- c. Set **SQL query to verify the connection**. For more information about the property, see “SQL query to verify the connection (PingQuery)” on page 238.
 - d. In **Query timeout**, type the length of time, in seconds, that the adapter must wait for a response to a database query. For more information about the property, see “Query timeout (seconds) (QueryTimeOut)” on page 237.
 - e. Set **Return business object even when the stored procedure result set is empty**. For more information about the property, see “Return business object even when the stored procedure result set is empty (ReturnDummyBOForSP)” on page 237.
- **Processing methods for illegal XML characters**
 - a. From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:
 - **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
 - **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
 - **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

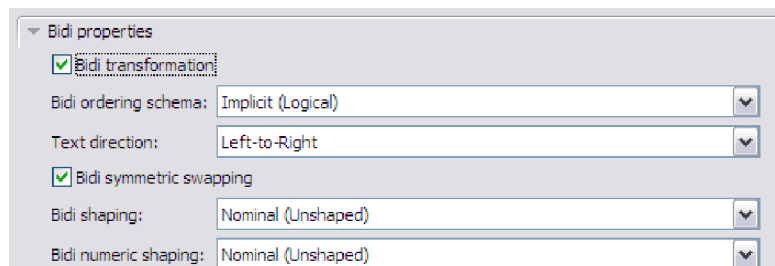


- **Logging and tracing**

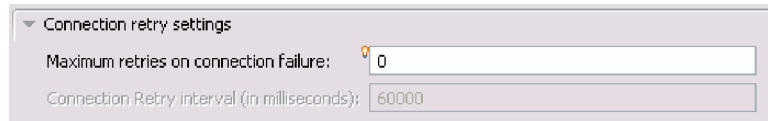


- If you have multiple instances of the adapter, set Adapter ID to a value that is unique for this instance. For more information about the property, see “Adapter ID (AdapterID)” on page 232.
- If you want to mask the potentially sensitive user information in the log and trace files, select the **Disguise user data as “XXX” in log and trace files** field. For more information about the property, see “Disguise user data as “XXX” in log and trace files (HideConfidentialTrace) ” on page 234.

- **Bidi properties**



- To enable bidirectional support for the adapter at run time, select the **Bidi transformation** check box.
 - Set the properties, ordering schema, text direction, symmetric swapping, character shaping, and numeric shaping to control how bidirectional transformation is performed.
- **Connection retry settings**



- To specify the number of times the adapter can attempt to reconnect to Oracle E-Business Suite in case of connection failure, set **Maximum retries on connection failure** to a value greater than or equal to zero. For more information, see “Maximum retries on connection failure (connectionRetryLimit)” on page 239.
 - To specify the time interval between retries if connection fails, set **Connection Retry interval (in milliseconds)** to a value in milliseconds. This property is enabled only when the connectionRetryLimit property has a value greater than zero. For more information, see “Connection Retry interval (in milliseconds) (connectionRetryInterval)” on page 240.
9. Click **Next**. The Specify the Location Properties window is displayed.
 10. In the Specify the Location Properties window, specify the name of the module you want to create. This can be a new or existing module.
 - If the module name that you want appears in the **Module** list, select its name.

Important: If the module contains an interface or business object with the same name as any you are now configuring, the original interface or business object in the module is replaced by the new version.
 - Otherwise, create a module:
 - a. Click **New**.
 - b. In the Select a Business Integration Project Type window, select **Create a module project** and click **Next**.
 - c. In the Create a Module window, type a name for the module. For example, OracleOutboundModule.
 - d. If you want the service description files (the .import and .wsdl files) to be located in the default folder in the module, leave **Use default location** selected. If you want to specify a different folder in the module, clear the option and then click **Browse** to specify a different folder in **Location**.
 - e. If you want the module to be automatically opened in the assembly diagram in IBM Integration Designer when the wizard closes, select **Open module assembly diagram**. Otherwise, clear this option.
 - f. Click **Finish** to create the module.
 11. Specify the namespace you want to use for your business artifacts.
 - If you want the business objects in the module to use the default derived namespace, leave **Use default namespace** selected.
 - To specify a different namespace, clear the option and type a different value in **Namespace**.
 12. Optional: Specify the folder within the new module where the service description must be saved. In **Folder**, type the folder name or browse to an

existing folder. If you do not specify a folder name, the artifacts (the import, XSD, and WSDL files) are stored in the root folder of the module, that is, the folder with the module name.

13. In the **Name** field, accept the default import name or type a different name.
14. Optional: If you want to save the business objects in a library where they can be used by other modules, select **Save business objects to a library** and specify the location of the library in the **Library** field.
15. Optional: In the **Description** field, type a descriptive comment about the module.
16. When you are finished setting properties, click **Finish**.
17. If the Model Changed window is displayed, click **Yes**.

Results

The wizard exits. The module is created in the project and artifacts are generated.

What to do next

In some instances, you might need to use the assembly editor to complete the configuration. Then you can test or deploy your module.

Related concepts

“User authentication” on page 54

The adapter supports several methods for supplying the user name and password that are needed to connect to the Oracle database. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

Completing the configuration

In some situations, manual configuration steps are needed to complete the configuration of your business objects.

About this task

Perform this task when you need to customize the artifacts generated by the wizard. You might do this in the following situations:

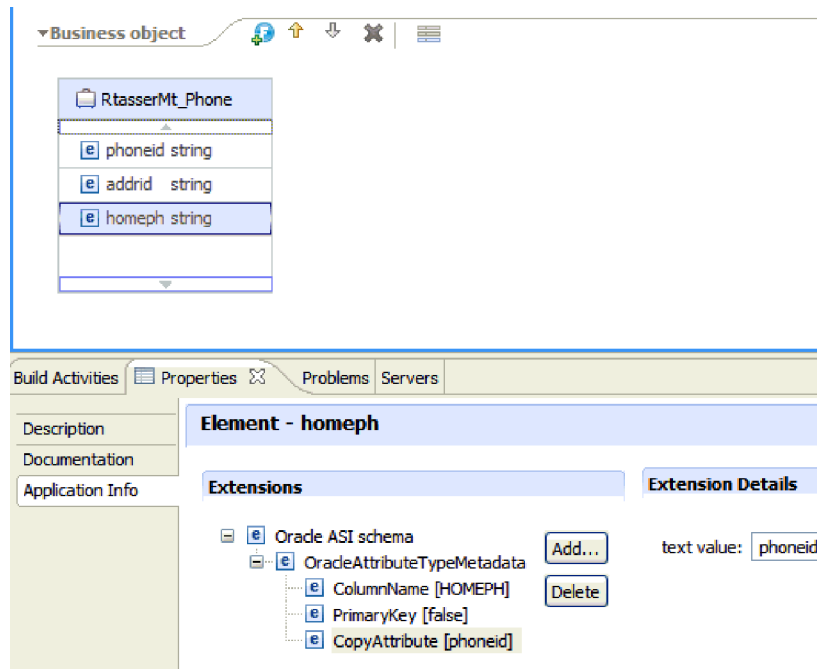
- To set the CopyAttribute parameter for a column so that its value is set to the same value as another column.
- To add or remove attributes from a business object. For example, you can simplify your business object design by removing the simple attribute corresponding to any database column that you do not need to reference.
- To configure additional parents for a table business object that has multiple parents. The wizard configures only one parent for a table business object.

This topic provides detailed instructions for setting the CopyAttribute parameter to a table business object. Other changes to business object structures can be accomplished using similar techniques.

The CopyAttribute parameter is contained in the properties of the attribute for the column that you want to populate with values and application-specific information from another column. For example, if you want the contact column of a new row in the table to contain the same value as the e-mail column, set the CopyAttribute parameter of the contact attribute to e-mail. You can use the assembly editor in IBM Integration Designer to set the value.

Procedure

1. In the Business Integration perspective of IBM Integration Designer, expand the module name, expand **Data Type**, and then locate the table business object. The business object name is the name of the database schema plus the name of the database table. An optional namespace might be included at the beginning of the name.
2. Right-click the business object name and select **Open**. The assembly editor displays the business object, which has a field for each column.
3. In the assembly editor, select the column you want to set to match another column.
4. In the Properties view, select **Application Info**. If the Properties view is not visible, right-click the column name and click **Show in Properties**.
5. Expand **Oracle ASI schema**, and then expand **OracleAttributeTypeMetadata**.
6. Right-click **OracleAttributeTypeMetadata** and then select **New > oracleasi:CopyAttribute**.
7. Select the **CopyAttribute** property.
8. In the Extension Details area, set the text value to the name of the column that contains the information to copy. The column can be in the current business object or its parent business object. To copy from a column in the current business object, set the value to the column name, for example, phoneid. To copy from a column in the parent business object, prefix the column name with two periods (..), for example, ..phone. The following figure shows the assembly editor with the CopyAttribute property set to a column in the current table.



Results

The business object is configured to use the CopyAttribute property to set the business object attribute and properties for a database column on information in another column.

What to do next

You can now test and deploy the module.

Configuring the module for inbound processing

To configure a module to use the adapter for inbound processing, use the external service wizard in IBM Integration Designer to find and select business objects and services from the Oracle database, and to generate business object definitions and related artifacts.

Related concepts

“Business object hierarchies” on page 33

Define the relationships between database tables using parent-child relationships and data ownership in hierarchical business objects.

“Inbound processing” on page 21

IBM WebSphere Adapter for Oracle E-Business Suite supports inbound event management with event delivery. Events are processed from an event store that is populated either by the database application or from the result of custom queries that you provide. You control how often the adapter polls for events and how many records are delivered to the export at one time.

Discovering database objects

After configuring the connection properties, run a query to search for database objects. Browse through the discovered objects navigation tree to understand the structure of objects in the Oracle database and use filters to display only the database objects you want to see.

Before you begin

You must know the data requirement of the program that needs to access the database. For example, you need the following information about your database:

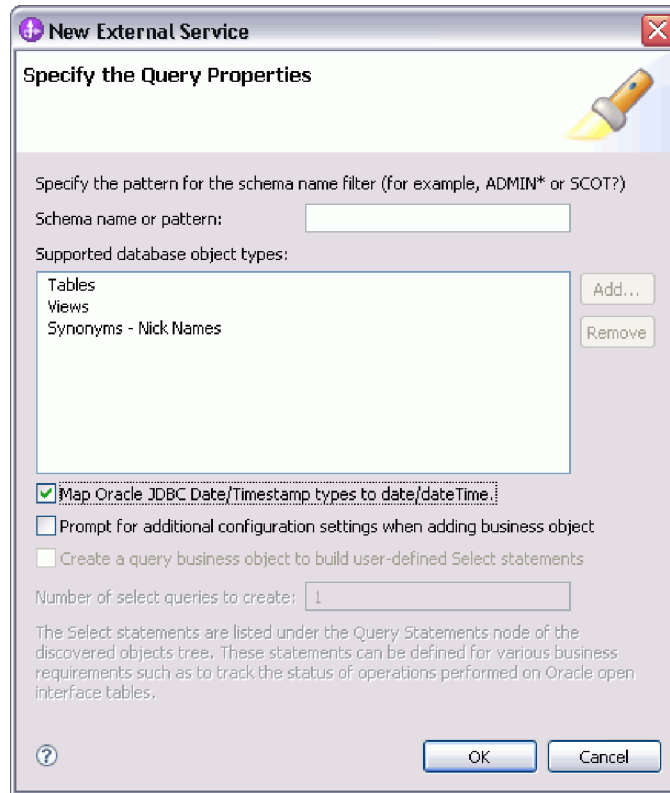
- Which schemas your module needs to access
- What type of database objects you need to access in those schemas

About this task

This task starts in the Find Objects in the Enterprise System window of the external service wizard.

Procedure

1. In the Find Objects in the Enterprise System window, click **Edit Query**. The Specify the Query Properties window is displayed.



Note: The **Create a query business object to build user-defined select statements** is available only for outbound processing.

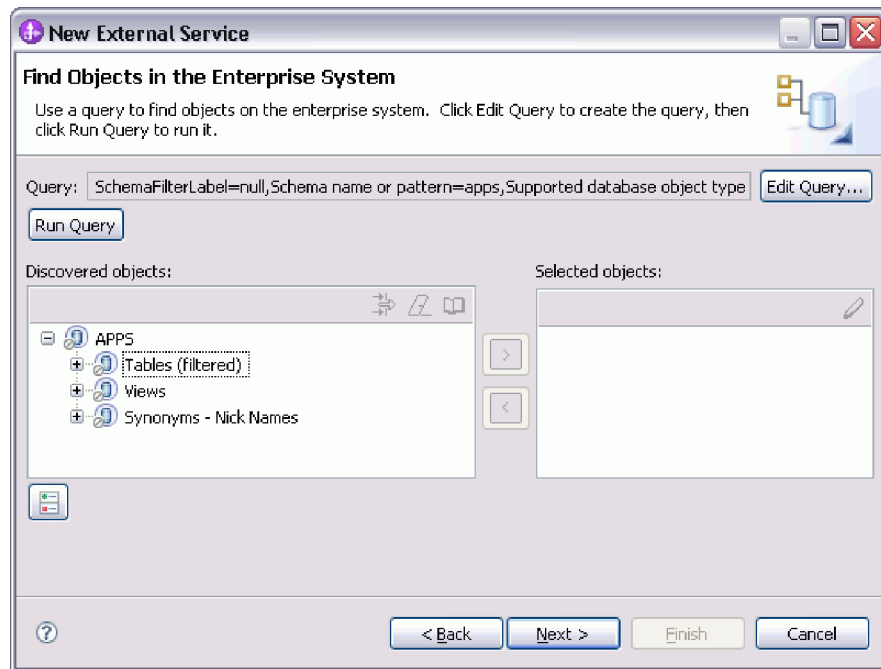
Use the Specify the Query Properties window to perform the following tasks:

- Reduce the search time by searching a subset of database schemas
 - Omit one or more types of database objects from the search
 - Make the wizard prompt you for application-specific information that cannot be automatically determined based on information in the database
 - Map the Oracle data types Date and Timestamp to date and dateTime
2. To limit the number of database schemas that are retrieved, type the name of the schema or a name pattern in **Schema name or pattern**. Use the question mark or underscore (? or _) character to match a single character and the asterisk or percentage sign (* or %) to match multiple characters. Only schemas that start with that string or match that pattern are displayed when you run the query. If you do not specify a schema name pattern, all schemas in the database are displayed. Using a filter can speed up the discovery process if your database contains many schemas.
 3. To omit one or more types of objects from the search, select the types of objects that you want to omit (tables, views, and synonyms or nicknames) in **Supported database object types**, and then click **Remove**. If you change your mind, click **Add** to add the object type back. If your database contains object types that you do not need to access, omitting them can speed up the discovery process.
 4. Select the **Prompt for additional configuration settings when adding business object** check box. Then, when you add a database object to the list of business objects to create, the wizard automatically prompts you for all user-configurable application-specific information for the object. For example, if you select this option, the wizard guides you through the process of building a

simple parent-child hierarchy of business objects. If you need a hierarchy, that a table business object has two attributes referring to attributes in two different tables (that is, it has two parent business objects), complete the configuration in the IBM Integration Designer assembly editor.

Important: If you do not select this option, the wizard prompts only for required information. You must complete the configuration of the business objects using the assembly editor.

5. Click **OK** to save your changes to the query.
6. In the Find Objects in the Enterprise System window, click Run Query to use the query to discover database objects. The result of running a typical query is shown in the following figure.



The **Discovered objects** pane lists the objects that were discovered. The tables, views, and synonyms/nicknames are sorted by schema name.

7. In the **Discovered objects** list, click + (the plus sign) to expand a schema node and the **Tables**, **Views**, and **Synonyms - Nicknames** nodes to see the database objects discovered by the wizard.

Results

The wizard has discovered the database objects you can access using the adapter.

What to do next

Continue working in the external service wizard. The next step is to select the objects you want to use in your module, configure each business object, and create hierarchies of business objects.

Related concepts

“RetrieveAll operation” on page 9

The adapter uses the RetrieveAll operation to retrieve an array of business objects from the database. The adapter uses different processes depending on whether the RetrieveAll operation is for database table business objects or for user-specified SQL business objects.

“Query business object overview” on page 46

Query business objects run a user-defined SELECT statement against the database and return the matching records in business objects.

Selecting and configuring business objects

Using the list of database objects discovered by the external service wizard, and the query object templates you specified, continue using the wizard to select the database objects that you need to access in your module. Then provide the configuration information for your new business objects.

About this task

The Find Objects in the Enterprise System window allows you to select and configure objects in any order, with the single exception that you must select and configure a parent table before you can select and configure its child tables. Apart from this restriction, you have the flexibility to add objects individually or to add them several at a time. You can mix objects from the various nodes of the **Discovered objects** list. For example, you can select several tables and view objects and a stored procedure object, and add them at the same time.

The high-level flow of selecting and configuring business objects is as follows:

1. Select one or more objects in the **Discovered objects** list of the Find Objects in the Enterprise System window.
2. Click the > (Add) button.
3. The wizard opens the Specify the Configuration Properties for 'object' window.
 - If you select a single object, a single Specify the Configuration Properties for 'object' window is displayed.

Complete that window, specifying the user-configurable attributes and other information that the wizard cannot discover by examining the database, and click **OK** to save the configuration.

- If you select multiple objects, the Specify the Configuration Properties for 'object' window displays with one page for each object selected.

Click the name of each object in turn. The window displays the same information you would see had you selected this object individually.

Important: Do not click **OK** until you complete the configuration pages for all the objects. The wizard does not close the notebook until you provide all the required fields, but you can close the window before providing optional fields. If you do not configure the optional fields in the wizard, you must use the business object editor to configure them after exiting the wizard.

4. The wizard adds the configured object to the **Selected objects** list.

As long as you do not exit from the wizard, you can work iteratively to select and configure the business objects you need in your module. However, before you start the wizard to add objects to an existing module, understand the requirements of the program that uses the business objects carefully. The wizard overwrites the existing business objects in the same path.

Selecting and configuring tables, views, and synonyms or nicknames for inbound processing

Select and configure business objects for tables, views, and synonyms or nicknames for use in your module. For inbound processing, the tables, views, and synonyms are the business objects that are delivered in events.

Before you begin


To perform this task, you need to understand the structure of the data in the database and know what database objects the module needs to access. Specifically, you need to know the following information:

- The structure of the tables, views, and synonyms or nicknames, including the columns you need, and column attributes such as data type
- The relationships between the tables, including the cardinality and ownership of parent-child relationships

About this task

This task is performed through the external service wizard. Start in the Find Objects in the Enterprise System window and then work in a Specify the Configuration Properties for 'object' window that is specific to the business object you are configuring.

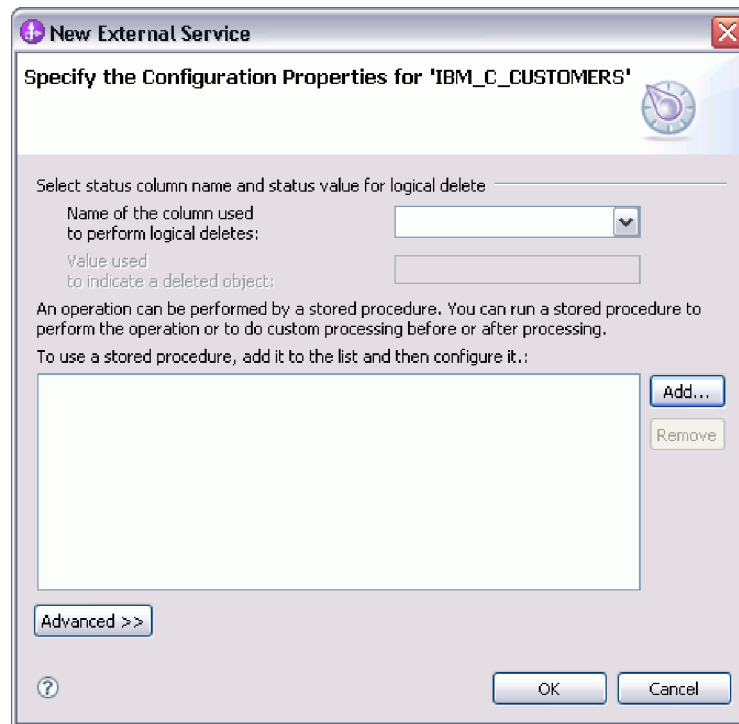
Procedure

1. In the **Discovered objects** list of the Find Objects in the Enterprise System window, select one or more tables, views, or synonyms. Click the > (Add) button to add the object or objects to the **Selected objects** list. Alternatively, you can also filter the tables, views, or synonyms by specifying a valid name or pattern for at least one of the filter fields in the **Filter Properties** window.
 - a. Click the object you want to filter, and then click the  (Create or edit filter.) button, located at the top of the **Discovered objects** pane.
 - b. In the Filter Properties window, type a name or pattern in the **Object name or pattern** field. Use the question mark or underscore (? or _) to match a single character and the asterisk or percentage (* or %) to match multiple characters. The name is not case sensitive.
 - c. Click **OK**. The object that matches the given filter condition is displayed.
 - d. Select one or more objects from the discovered list, and click the > (Add) button to add the object to the **Selected objects** list.

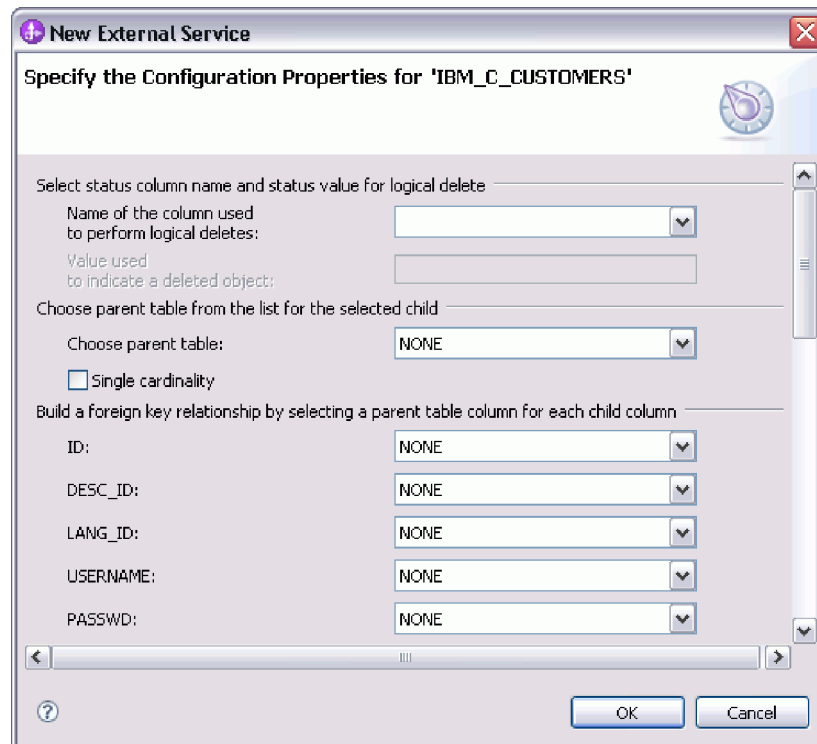


The following two figures show a typical Specify the Configuration Properties for 'object' window for a table, view, synonym, or nickname business object.

The first figure shows a typical window for the first table or group of tables selected.



The second figure shows a typical window for subsequent tables you select. After you select and configure at least one table, the Specify the Configuration Properties for 'object' window for subsequent tables displays an area where you can optionally define a parent-child hierarchy between tables.

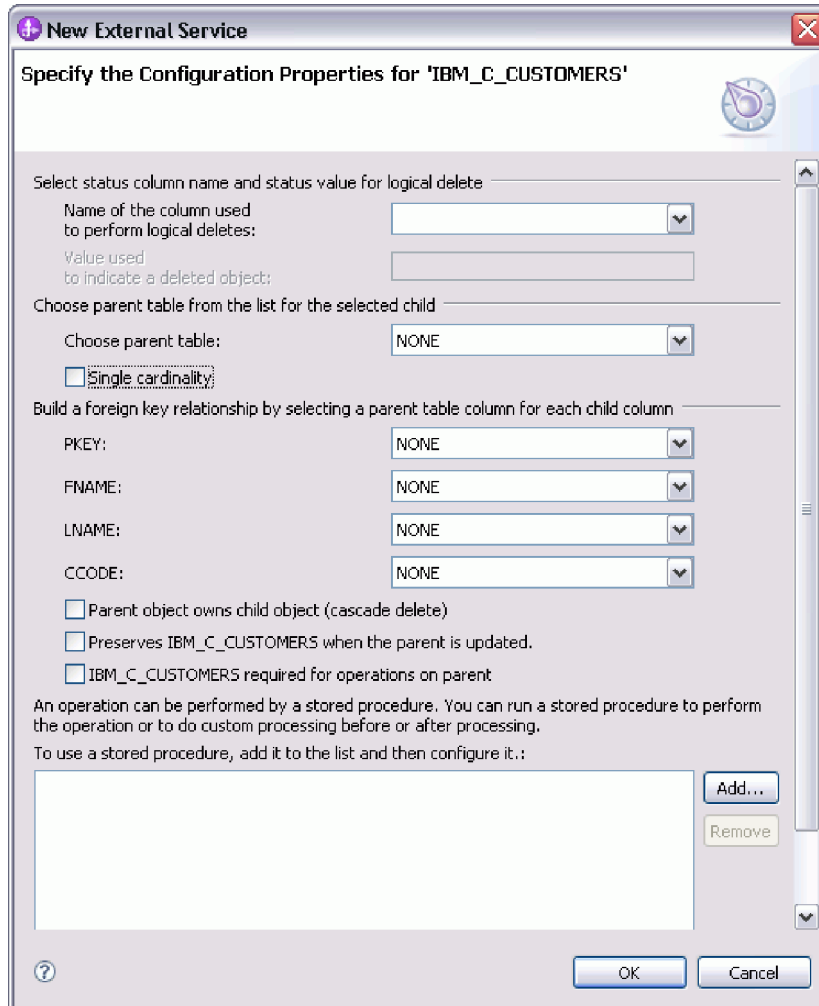


Note: As you configure the object, choices that require advanced configuration might present additional fields in this window, causing the window to scroll. Be sure that you examine all fields on the window before clicking **OK**.

2. If the table has a column that is used to indicate logical deletes:
 - a. Select the column name in the **Name of the column used to perform logical deletes** field.
 - b. In the **Value used to indicate a deleted object** field, type the value that indicates that a row is logically deleted. You can get this value from your database administrator.
3. If the **Select primary key for table** *table_name* area is displayed, click **Add**, select the column to be used as the primary key for the table business object, and then click **OK**. If the table has a composite key, you can select multiple columns. The **Select primary key for table** *table_name* area is displayed only when the database table does not have a column designated as the primary key. Each table business object must have a primary key, even if the associated database table does not have a key. If the primary key is defined in the database, this section of the window is not displayed.
4. Optional: Define a parent-child relationship between business objects.

To build a parent-child hierarchy, configure the parent table first, and return to the Find Objects in the Enterprise System window to select and configure the child tables.

Configure the parent-child relationship using the area of the Specify the Configuration Properties for 'object' window shown in the following figure. These fields are not displayed for the first table you configure.



- a. In the **Choose parent table** field, select the name of the parent table you are configuring. If you do not see the parent table in the list, the parent table has not yet been configured. Go back and configure the parent object before configuring the child objects.
- b. Specify the cardinality of the relationship:
 - If the table has a single-cardinality relationship with the parent table, select the **Single cardinality** check box. In a single cardinality relationship, a parent can have only one child business object of this type. A single-cardinality relationship can be used with ownership to represent a true child or without ownership to represent lookup tables or other peer objects in a database.
 - If the table has a multiple-cardinality relationship, do not select the **Single cardinality** check box. In a multiple-cardinality relationship, a parent can have an array of child business objects of this type.
- c. Build the foreign key relationship between the parent and child by specifying for each child column whether it is a foreign key in the parent table.
 - If the child column is not a foreign key, select NONE.
 - If a child column is a foreign key, select the column in the parent table that corresponds to the child column.

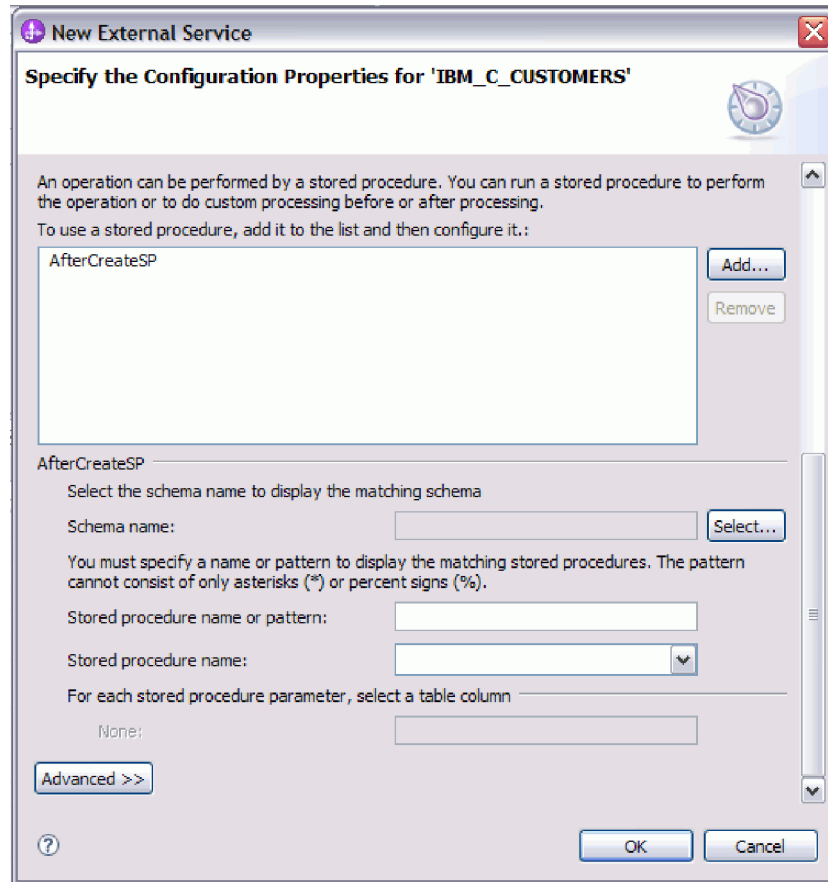
Note: The wizard can configure only a single parent table. If the child table has multiple parent tables, you must use the business object editor to configure the remaining parent tables after exiting the wizard.

- d. If the parent object owns the child object, then the child objects in the database are deleted when the parent is deleted. To indicate that this child is owned by its parent, select the **Parent object owns child object (cascade delete)** check box. Otherwise, clear this option to prevent child objects, such as lookup tables, from being deleted when their parent is deleted.
 - e. If you do not want the child objects to be deleted as part of an Update operation, select the **Preserves *child_table_name* when the parent is updated** check box.
- When a parent table is updated, the adapter compares the child business objects present in the input with the child business objects returned from the database. By default, the adapter deletes any child objects returned from the database that are not present in the input business object.
- f. By default, you can perform operations on parent business objects without specifying the child business objects. If you want to ensure that a parent business object specifies its child business objects when the parent is submitted for a change, select the ***Child_table_name* required for operations on parent** check box.
5. An operation can be performed using either a standard SQL statement generated by the adapter or using stored procedures or stored functions from the database. If you want to use stored procedures or stored functions:

- a. Click **Add**.
- b. In the Add window, select the type of the stored procedure you want to run. For each operation, you can select a stored procedure that performs the operation, as well as stored procedures that run before or after the operation. For example, for the Create operation, you can specify any of these stored procedures: CreateSP, BeforeCreateSP, and AfterCreateSP.

Note: If you configure the table with RetrieveAllSP, ensure that at least one parameter of the stored procedure is a Cursor and the ResultSet ASI for the stored procedure is set to true to avoid the "No resultset found associated with the stored procedure" exception being generated at run time.

- c. Click **OK**. The Specify the Configuration Properties for 'object' window now shows the stored procedure types you selected and expands to display an area where you configure each one. It might be necessary to scroll down to see the new areas.



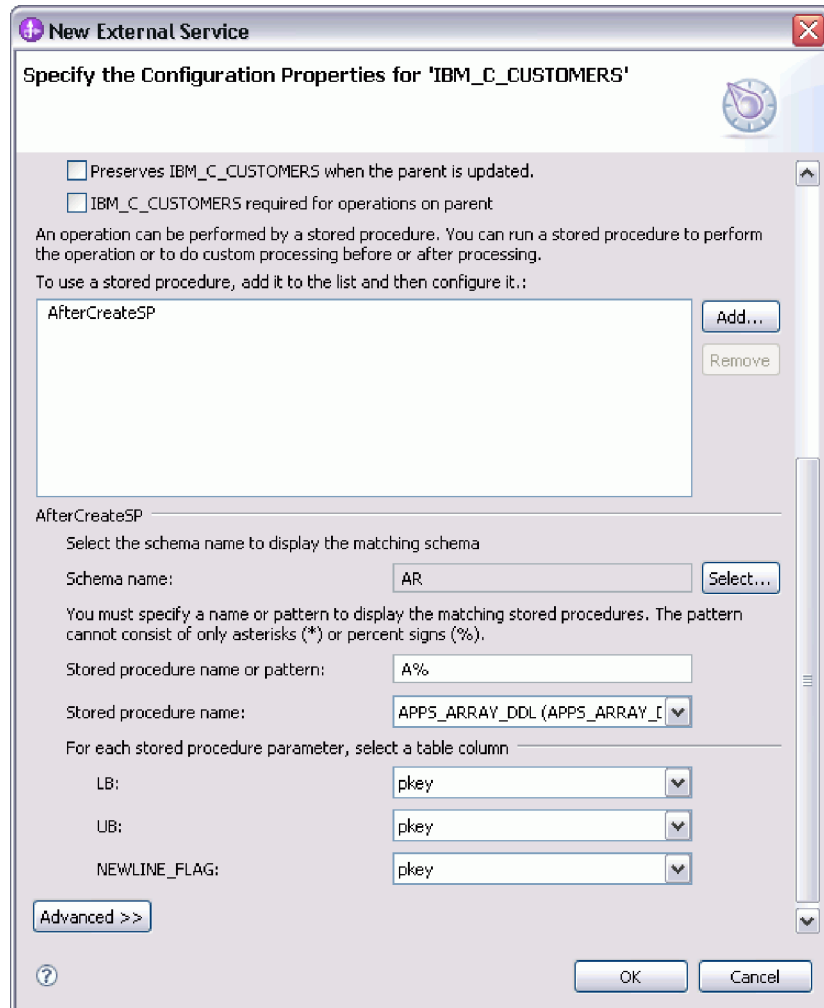
Note: In a hierarchical business object, if you want the stored procedure to be performed for each business object in the hierarchy, you must separately associate a stored procedure with the top-level business object and each child business object or an array of business objects. If you associate a stored procedure with the top-level business object but do not associate it with each child business object, then the top-level business object is processed with the stored procedure, but the child business objects are processed using the standard SQL query.

6. For each stored procedure type that you selected, specify the name of the stored procedure in the database and then configure the business object.
 - a. In the **Schema name** field, select the name of the schema that contains the stored procedure.
 - 1) Click **Select**.
 - 2) In the Select Value window, select the name of the schema you want to work with.
 - 3) Click **OK**.
 - b. Specify the name of the stored procedure or stored function.
 - 1) In the **Stored procedure name or pattern** field, either type the name of the stored procedure or stored function, or type a name pattern. Use the question mark or underscore (? or _) to match a single character, and the asterisk or percentage sign (* or %) to match multiple characters.
 - 2) In the **Stored procedure name** field, select the name of the procedure you want. If the stored procedure list contains many items, the **Select**

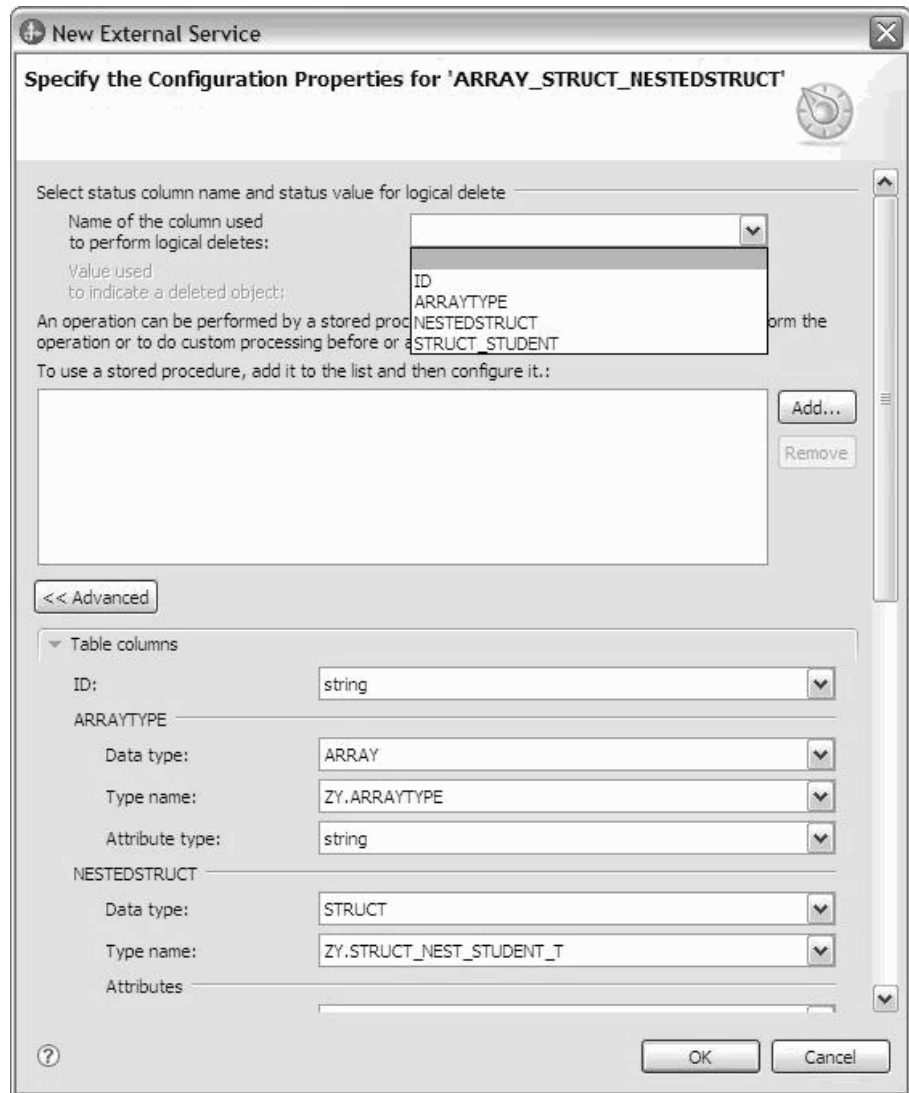
button is displayed next to the **Stored procedure name** field. Click **Select** to open the Select window and select the name of the stored procedure or stored function.

The Specify the Configuration Properties for 'object' window expands to provide an area where you configure the stored procedure. The wizard automatically generates the list of parameters by examining the stored procedure in the database.

- c. For each parameter in the stored procedure (on the left), select the table column (on the right) to pass to the stored procedure in that parameter. The following figure shows a portion of the window after a stored procedure has been configured.




7. To specify the data type mapping for each column in the table:
 - a. Click **Advanced**.
 - b. Expand **Table columns**. For each column in the table, the default data type mapping is displayed. For Oracle databases, if the table contains any complex data type, such as an array, structure, nested structure or table, the type name and the sub attribute details are also automatically discovered and displayed. The following figure displays the type name and sub attribute details of an Oracle table containing complex data types.



c. Review the mapping and change them if required.

Note: If the primary key in a table is of the date or timestamp type, then the object_key in the event_table must be in the 'yyyy-mm-dd hh-mm-ss' format.

8. When all the fields in the window are completed, click **OK** to save the configuration of the business object. The table, view, synonym, and nickname business objects you defined are now listed in the Find Objects in the Enterprise System window.
9. To change the configuration of an object from the **Selected objects** list, select the object name and then click the  (Edit) icon.
10. When you have selected and configured all business objects that you need, click **Next** to set global properties.

What to do next

Continue working in the Find Objects in the Enterprise System window to select and configure other types of business objects.

Related concepts

“Business object hierarchies” on page 33

Define the relationships between database tables using parent-child relationships and data ownership in hierarchical business objects.

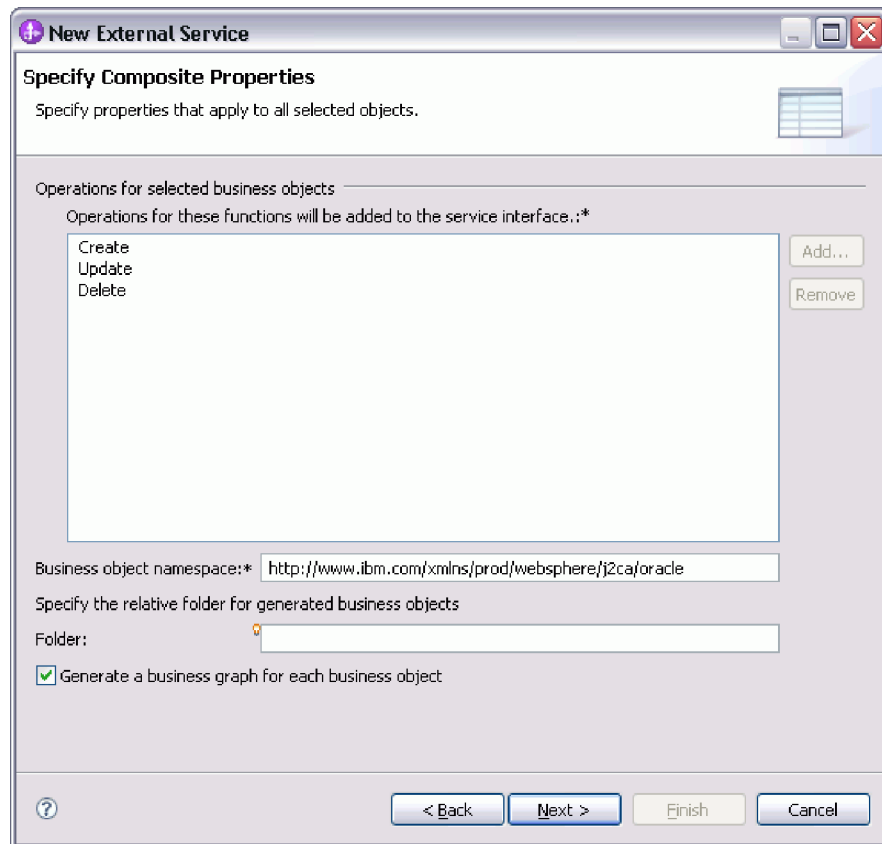
Setting global properties for operations

After you have selected database objects in the external service wizard, you need to specify properties that apply to all business objects.

Procedure

1. When the **Selected objects** list in the Find Objects in the Enterprise System window contains all the business objects you want to use in your application, click **Next**.
2. In the Specify Composite Properties window, review the list of operations. This list contains the operations that the adapter supports for the inbound services. To add to the list of operation includes operations for all business objects you selected on the previous window.

The specified operations are set for all business objects that are generated.



3. To remove an operation that you do not need, select the operation name and click **Remove**. If you change your mind, click **Add** and restore a removed operation.
4. In **Business object namespace**, accept the default namespace or type the full name of another namespace.

The namespace is prefixed to the business object name to keep the business object schemas logically separated. For more information about this property, see “Business object namespace (BusinessObjectNameSpace)” on page 260.

5. Optionally, in **Folder**, type the relative path to the folder where the generated business objects are to be stored.

Note: If you are creating multiple adapter artifacts within a module, ensure that you specify different business object folders for each adapter within the module. For example, if you are creating artifacts for Oracle, JDBC, SAP, and JDE within a module, you need to create different relative folders for each of these adapters. If you do not specify different relative folders, the existing artifacts are overwritten when you generate new artifacts.

6. If you want a business graph to be created for each business object, click **Generate a business graph for each business object**. Business graphs are needed only when adding business objects to a module created with a version of IBM Integration Designer earlier than version 7.0.

Note: You must select this option if you are adding business objects to a module that was created with an earlier version of IBM Integration Designer. Otherwise, you must rewire your interface.

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

Results

You have provided information that applies to all business objects in the module.

What to do next

Continue working in the wizard. The next step is to specify deployment information to use at run time and information for saving the service as a module.

Setting deployment properties and generating the service

After you select and configure the business objects for your module, use the external service wizard to configure properties that the adapter uses to connect to a specific database. The wizard creates a new business integration module where all the artifacts and property values are saved.

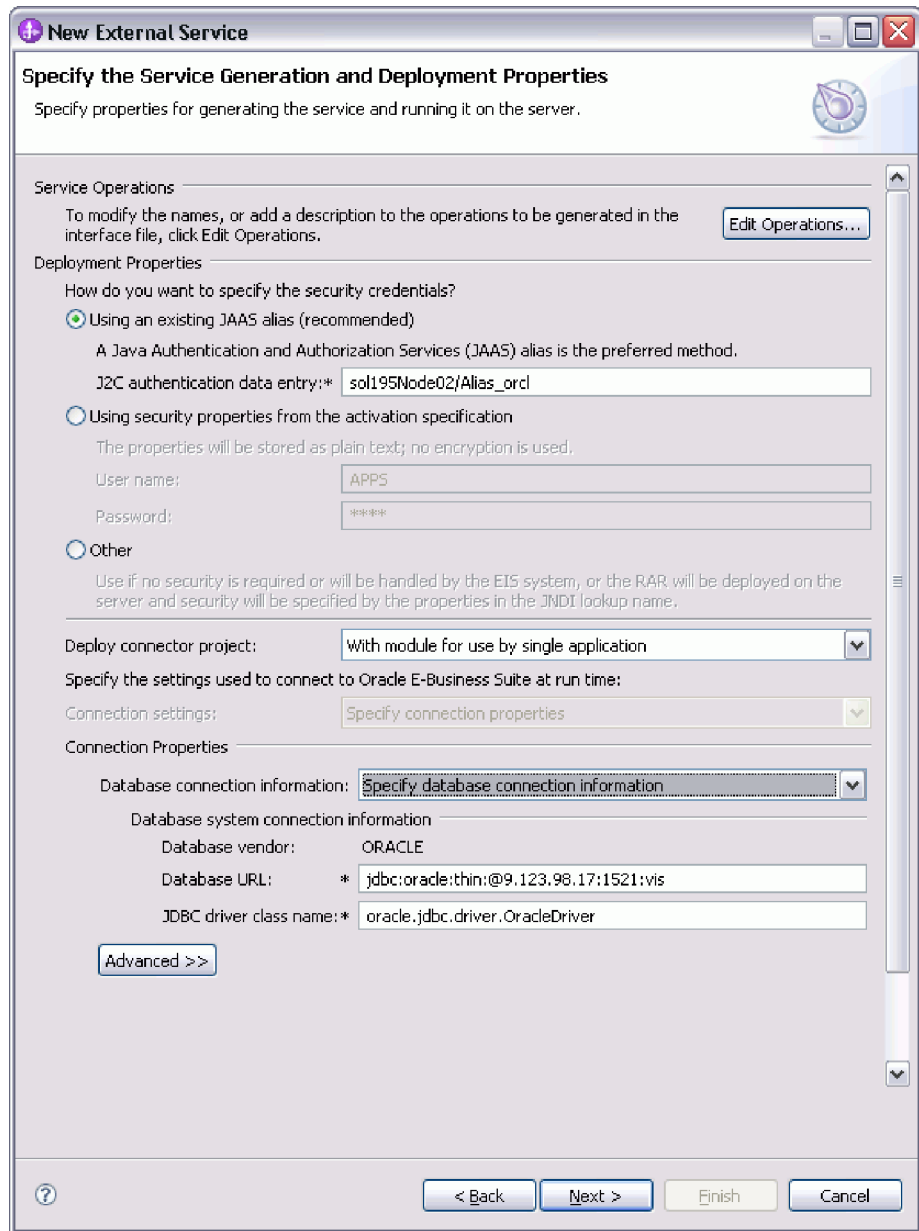
About this task

This task is performed through the Specify the Service Generation and Deployment Properties and Specify the Location Properties windows of the external service wizard.

The connection properties in this task are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change the values here. For example, to use the Oracle JDBC driver at run time instead of the Oracle Thin driver, set the driver information here.

Procedure

1. In the Specify the Service Generation and Deployment Properties window, click **Edit Operations** to review the names of operations or add a description for the operations for the business objects you are creating.



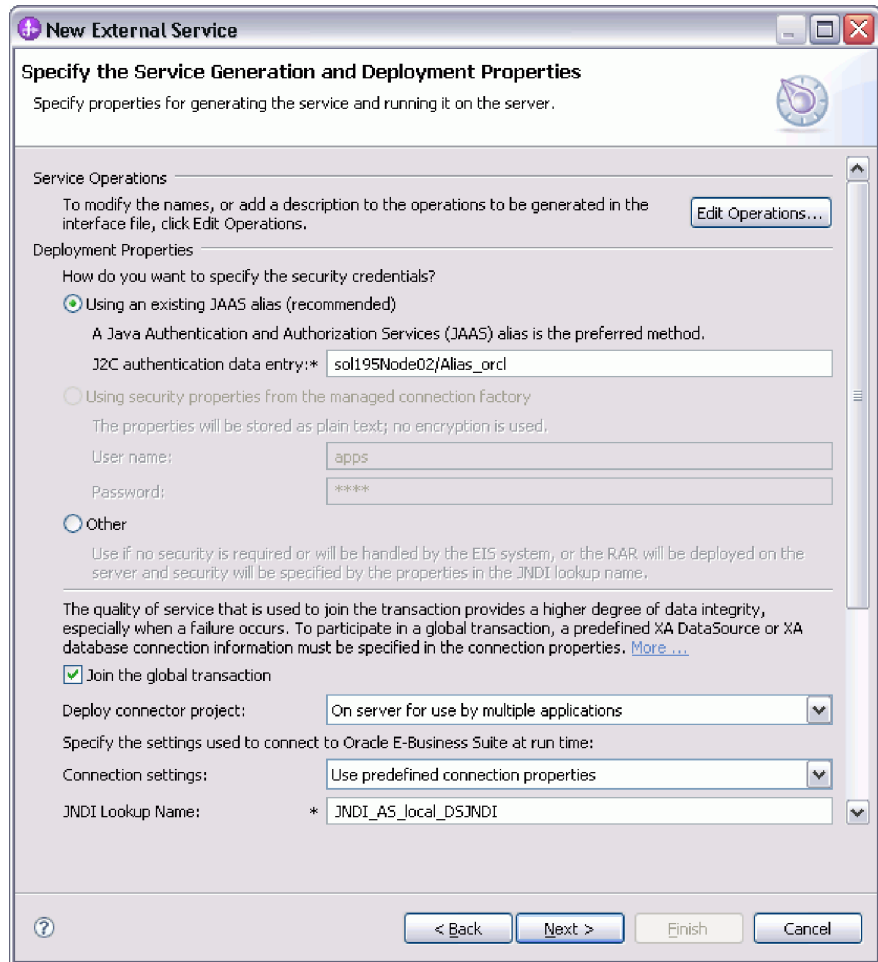
2. In the **Deployment Properties** area, specify how you want the adapter to get the user name and password at run time.
 - To use a J2C authentication alias, click **Using an existing JAAS alias (recommended)** and enter the name of the alias in the **J2C authentication data entry** field. You can specify an existing authentication alias or create one at any time before deploying the module. The name is case-sensitive and includes the node name.
 - To use the managed connection properties, click **Using security properties from the managed connection factory** and enter values in the **User name** and **Password** fields.
 - To administer the user name and password from another mechanism, click **Other**.

Note: If you use local connection information to establish the database connection, a security credential is required. You can select either **Using an**

existing JAAS alias (recommended) or **Using security properties from the managed connection factory** as the security mechanism. If you use an existing data source on the server, the security credential is not required. In this case, you can select **Other** as the security mechanism. Also, if you set the **J2C authentication data entry** field or set the **User name** and **Password** fields, they override the user name and password in the data source.

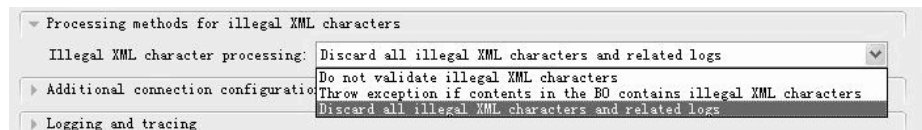
3. In the **Deploy connector project** field, specify whether to include the adapter files in the module. Select one of the following values:
 - **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded 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.
4. If you selected **On server for use by multiple adapters** in the previous step, specify the connection properties used at run time.
 - If you manually created and configured a managed connection factory or activation specification on the server or if you have already deployed an application that connects to the same database using the same managed connection factory or activation specification properties, you can reuse the managed connection factory or activation specification by specifying the name of its Java Naming and Directory Interface (JNDI) data source:
 - a. From the **Connection settings** list, select **Use predefined connection properties**.
 - b. In the **JNDI Lookup Name** field, type the name of the JNDI data source for an existing managed connection factory or activation specification.

The following figure displays the typical settings for reusing a managed connection factory or activation specification for a stand-alone deployment of the adapter.



- c. Click **Next** to complete this task.
 - If this is the first application that connects to the database with a specific user name and password, or if you want to administer the user name and password separately from other applications, select **Specify connection properties**.
5. In the **Connection Properties** area, specify how you want the adapter to establish database connection at run time.
 - To use an existing data source on the server:
 - a. From the **Database connection information** list, select **Specify predefined DataSource**.
 - b. In the **Database system connection information** area, enter the name of an existing JNDI data source in the **DataSource JNDI name** field. For more information about this property, see “Data source JNDI name (DataSourceJNDIName)” on page 262.
 - To specify the connection information to be saved in the adapter properties:
 - a. From the **Database connection information** list, select **Specify database connection information**.
 - b. In the **Database system connection information** area, enter values in the **Database URL** and **JDBC driver class name** fields. For more information about these properties, see “Database URL (DatabaseURL)” on page 263 and “JDBC driver class (JDBCdriverClass)” on page 268.

6. Review and, if necessary, change the values of the required connection properties. The fields are initialized with the connection information you specified when you started the wizard. You can change the values to specify a different user name and password at run time. You can also connect to an alternate Oracle database, although the schema names must be the same in both databases. The format of the connection properties is database-specific. For more information about the properties, see “Activation specification properties” on page 256.
7. Optionally, specify advanced properties by clicking **Advanced**. Expand each of the advanced sections to review the properties.
 - **Processing methods for illegal XML characters**
 - a. From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:
 - **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
 - **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
 - **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.



- **Event polling configuration**
 - a. In **Interval between polling periods**, type the number of milliseconds that the adapter waits between polling periods. For more information, see “Interval between polling periods (PollPeriod)” on page 271.
 - b. In **Maximum events in polling period**, type the number of events to deliver in each polling period. For more information, see “Maximum events in polling period (PollQuantity)” on page 271.
 - c. In **Time between retries in case of case of system connection failure (in milliseconds)**, type the number of milliseconds to wait before trying to connect after a connection failure during polling. For more information, see “Retry interval if connection fails (RetryInterval)” on page 272.
 - d. In **Maximum number of retries in case of system connection failure**, type the number of times to retry the connection before reporting a polling error. For more information, see “Number of times to retry the system connection (RetryLimit)” on page 272.
 - e. If you want the adapter to stop if polling errors occur, select **Stop the adapter when an error is encountered while polling**. If you do not select this option, the adapter logs an exception but continues to run. For more information, see “Stop the adapter when an error is encountered while polling (StopPollingOnError)” on page 274.
 - f. Select the calendar based scheduling option to create calendar based polling for inbound activities. You can schedule your business activities, when you create a new calendar in IBM Integration Designer. The option of working with the calendar based scheduling feature is only possible

with IBM Integration Designer as the tooling environment. The following figure helps you to schedule a calendar polling option.

Event polling configuration

Interval between polling periods (milliseconds): 2000

Maximum events in polling period: 10

Time between retries in case of system connection failure (in milliseconds): 60000

Maximum number of retries in case of system connection failure: 0

Stop the adapter when an error is encountered while polling

Retry EIS connection on startup

Polling based on calendar: [] Browse... New...

Time interval for polling unchanged files (in milliseconds): 0

Figure 21. Polling based on calendar

You can either select a blank calendar or create a new calendar for a module or library. When you select a blank calendar, you will not be able to set pre-defined time intervals. You have to define your time intervals. When you create a calendar using a pre-defined template, you can define time intervals for each template.

- 1) Click **New** to create a new calendar entry for a module or library.
You can choose an existing calendar, or create a new calendar instance.
 - Click **Browse** to select an existing calendar module. Or click **New** to create a module for the new calendar.
 - Click **Browse** to choose a folder for the calendar. (Optional).
 - Enter a name for the new calendar.
 - Click **Next** if you want to generate the calendar, through a predefined template. Or, click **Finish**, to create a non template calendar.
- 2) Click **Browse** to select an existing calendar for a module or library. In the **Select a Business Calendar** screen you can search for all currently existing calendar files (*.cal) in the IBM Integration Designer workspace.
 - In the **Name** field, type the calendar name or click the calendar in the **Matching business calendars** screen. Click **OK** to open the external service wizard.
 - In the **WebSphere Integration workspace**, select the **Calendar** module, and browse **Integration logic->Calendars**, to view or modify the calendar schedules. You can modify the intervals and exceptions, or add new entries for these elements. For more information, refer to the information at <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/index.jsp?topic=/com.ibm.wbpm.wid.bpel.doc/topics/cbuscal.html>.

Note: You must deploy the Business Calendar module to the same IBM Business Process Manager or WebSphere Enterprise Service Bus instance, along with the inbound application. If you do not map these two connections to the same server instance, the

inbound application using the business calendar will by default, poll as there is no calendar configured.

- g. Optionally, you can select **Retry EIS connection on startup** if you want the adapter to retry a failed connection when starting. For more information, see “Retry EIS connection on startup (RetryConnectionOnStartup)” on page 272
- **Event delivery configuration**

Event delivery configuration

Type of delivery: ORDERED

Ensure assured-once event delivery (may reduce performance)

Do not process events that have a time stamp in the future

Event types to process:

Adapter Instance for event filtering:

Retry limit for failed events: 5

Number of connections for event delivery

Minimum: 1

Maximum: 1

- a. In **Type of delivery**, select the delivery method. The methods are described in “Delivery type (DeliveryType)” on page 264.
 - b. If you want to ensure that events are delivered only once and to only one export, select **Ensure once-only delivery**. This option might reduce performance but does not result in duplicate or missing event delivery. For more information, see “Ensure once-only event delivery (AssuredOnceDelivery)” on page 265.
 - c. By default, the adapter processes all events that it finds when it polls. If you do not want it to process events that have timestamps later than the current time, select **Do not process events that have a timestamp in the future**. For more information, see “Do not process events that have a timestamp in the future (FilterFutureEvents)” on page 265.
 - d. In **Event types to process**, type a comma-separated list of the business objects for which you want events delivered. Leave this field blank to receive events for all business object types.

For example, if you want to receive events only when the Customer and Order tables, but not other tables, are changed in the database, set this field to Customer,Order.

For more information, see “Event types to process (EventTypeFilter)” on page 267.
 - e. In **Adapter Instance for event filtering**, type the connector ID for which you want the events delivered. For more information, see “Adapter Instance for event filtering (AdapterInstanceEventFilter)” on page 259.
 - f. In **Retry limit for failed events**, specify the number of times to try to deliver an event after a delivery failure. For more information, see “Retry limit for failed events (FailedEventRetryLimit)” on page 268.
 - g. Under **Number of connections for event delivery**, specify the minimum and maximum number of connections to use to deliver events. For more information, see “Minimum connections (MinimumConnections)” on page 269 and “Maximum connections (MaximumConnections)” on page 269.
- **Additional connection configuration**

- a. Set **Additional JDBC driver connection properties**. See more information about the property in “Additional JDBC driver connection properties [name:value;name:value] (JDBCDriverConnectionProperties)” on page 260.
 - b. Set **SQL query to verify the connection**. See more information about the property in “SQL query to verify the connection (PingQuery)” on page 270.
 - c. In **Query timeout**, type the length of time, in seconds, that the adapter must wait for a response to a database query. See more information about the property in “Query timeout (seconds) (QueryTimeOut)” on page 271.
 - d. Set **Return business object even when the stored procedure result set is empty**. See more information about the property in “Return business object even when the stored procedure result set is empty (ReturnDummyBOForSP)” on page 273.
- **Event configuration**

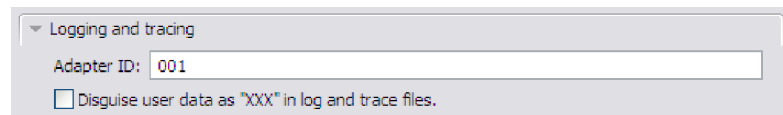
- a. In **Event order by**, indicate the order in which events are retrieved and processed. This is a comma-separated list of column names from the event table, plus the keywords that control the sort order for each column. Use asc for ascending order and desc for descending order. For more information, see “Event order by (EventOrderBy)” on page 265.
- b. In **Event table name**, accept the default name of the table that contains the event store, or type a different table name. For more information, see “Event table name (EventTableName)” on page 266.
- c. In **Stored procedure to run before poll**, the name of a stored procedure or stored function to run before the actual poll query is called. For more information, see “Stored procedure to run before polling (SPBeforePoll)” on page 274.

- d. In **Stored procedure to run after poll**, specify the name of a stored procedure or stored function to run after each polling cycle. For more information, see “Stored procedure to run after polling (SPAAfterPoll)” on page 274.
- e. In **Event query type for processing events**, select the type of event processing you want to use:
 - To use the standard event processing provided by the adapter, select **Standard**.
 - To provide your own queries to customize event processing, select **User-Defined (Dynamic)**. If you select this option, complete the additional fields described in the following table.

Field	What to specify	For more information
User-defined delete query	The name of the query, stored procedure, or stored function that is run after each event is processed to delete records that can be deleted after the event is delivered.	“User-defined delete query (CustomDeleteQuery)” on page 261
User-defined event query	The name of the query, stored procedure, or stored function that performs the polling for events.	“User-defined event query (CustomEventQuery)” on page 261
User-defined update query for failed event delivery	The name of the query, stored procedure, or stored function that is run when an event is not delivered successfully.	“User-defined update query for failed event delivery (CustomUpdateQueryForFailedEvent)” on page 261

Field	What to specify	For more information
User-defined update query	The name of the query, stored procedure, or stored function that is run after each event is processed to prevent the event from being picked up for processing in a subsequent event cycle.	"User-defined update query (CustomUpdateQuery)" on page 262

- **Logging and tracing**



- a. If you have multiple instances of the adapter, set Adapter ID to a value that is unique for this instance.
 - b. If you want to disguise potentially sensitive user information in log and trace files, select the **Disguise user data as "XXX" in log and trace files** field.
8. Click **Next**.
9. In the Specify the Location Properties window, specify the name of the module you want to create. This can be a new or existing module.
- If the module name that you want appears in the **Module** list, select the name.

Important: If the module contains an interface or business object with the same name as any you are now configuring, the original interface or business object in the module is replaced by the new version.

- Otherwise, create a module:
 - a. Click **New**.
 - b. In the Select a Business Integration Project Type window, select **Create a module project** and click **Next**.
 - c. In the Create a Module window, type a name for the module. For example, OracleInboundModule.
 - d. If you want the service description files (the .export and .wsdl files) to be located in the default folder in the module, leave **Use default location** selected. If you want to specify a different folder in the module, clear the option and then click **Browse** to specify a different folder in **Location**.
 - e. If you want the module to be automatically opened in the assembly diagram in IBM Integration Designer when the wizard closes, select **Open module assembly diagram**. Otherwise, clear this option.
 - f. Click **Finish** to create the module.

10. Specify the namespace you want to use for your business objects.
 - If you want the business objects in the module to use the default namespace, leave **Use default namespace** selected.
 - To specify a different namespace, clear the option and type a different value in **Namespace**.
11. Optionally, specify the folder within the new module where the service description must be saved. In **Folder**, type the folder name or browse to an existing folder. If you do not specify a folder name, the artifacts (the export, XSD, and WSDL files) are stored in the root folder of the module, that is, the folder with the module name.
12. In **Name**, accept the default name of the import or type a different name.
13. Optionally, if you want to save the business objects in a library where they can be used by other modules, select **Save business objects to a library** and specify the location of the library in **Library**.
14. Optionally, in **Description**, type a descriptive comment about the module.
15. When you are finished setting properties, click **Finish**.
16. If the Model Changed window is displayed, click **Yes**.

Results

The wizard exits. The module is created in the project and artifacts are generated.

What to do next

In some instances, you might need to use the assembly editor to complete the configuration. Then you can test or deploy your module.

Related concepts

“User authentication” on page 54

The adapter supports several methods for supplying the user name and password that are needed to connect to the Oracle database. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

Completing the configuration

In some situations, manual configuration steps are needed to complete the configuration of your business objects.

About this task

Perform this task when you need to customize the artifacts generated by the wizard. You might do this in the following situations:

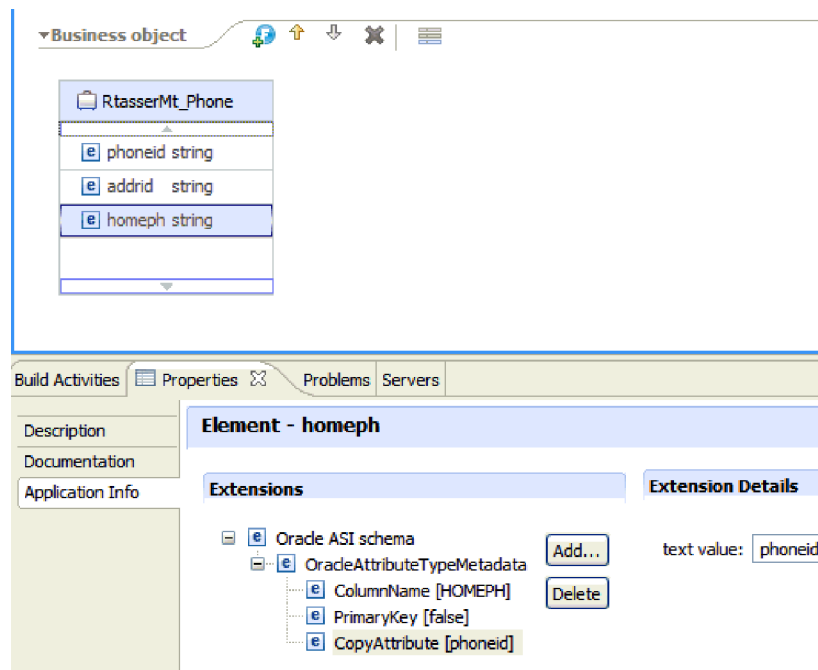
- To set the CopyAttribute parameter for a column so that its value is set to the same value as another column.
- To add or remove attributes from a business object. For example, you can simplify your business object design by removing the simple attribute corresponding to any database column that you do not need to reference.
- To configure additional parents for a table business object that has multiple parents. The wizard configures only one parent for a table business object.

This topic provides detailed instructions for setting the CopyAttribute parameter to a table business object. Other changes to business object structures can be accomplished using similar techniques.

The CopyAttribute parameter is contained in the properties of the attribute for the column that you want to populate with values and application-specific information from another column. For example, if you want the contact column of a new row in the table to contain the same value as the e-mail column, set the CopyAttribute parameter of the contact attribute to e-mail. You can use the assembly editor in IBM Integration Designer to set the value.

Procedure

1. In the Business Integration perspective of IBM Integration Designer, expand the module name, expand **Data Type**, and then locate the table business object. The business object name is the name of the database schema plus the name of the database table. An optional namespace might be included at the beginning of the name.
2. Right-click the business object name and select **Open**. The assembly editor displays the business object, which has a field for each column.
3. In the assembly editor, select the column you want to set to match another column.
4. In the Properties view, select **Application Info**. If the Properties view is not visible, right-click the column name and click **Show in Properties**.
5. Expand **Oracle ASI schema**, and then expand **OracleAttributeTypeMetadata**.
6. Right-click **OracleAttributeTypeMetadata** and then select **New > oracleasi:CopyAttribute**.
7. Select the **CopyAttribute** property.
8. In the Extension Details area, set the text value to the name of the column that contains the information to copy. The column can be in the current business object or its parent business object. To copy from a column in the current business object, set the value to the column name, for example, phoneid. To copy from a column in the parent business object, prefix the column name with two periods (..), for example, ..phone. The following figure shows the assembly editor with the CopyAttribute property set to a column in the current table.



Results

The business object is configured to use the CopyAttribute property to set the business object attribute and properties for a database column based on information in another column.

What to do next

You can now test and deploy the module.

Chapter 5. Modifying artifacts

Often business requirements mandate changes to the backend enterprise information system (EIS) data structures. These changes call for regeneration and reconfiguration of artifacts (import and export) that were previously generated using the external service wizard.

A few business scenarios where the output from one enterprise service discovery flow can be reused in successive flows, are outlined as follows:

- When you want to add a new object to the object set.
- When you want to modify configurations on selected objects like changes to operations, operation names, and service level settings like security, transactions, and reliability.
- When you want to remove a discovered object from the object set.
- When you want to rediscover existing objects in the service to synchronize the service if the objects in the back-end system have been updated.

To modify existing artifacts, you can invoke the wizard in one of the following ways. The external service wizard is initialized with previously configured settings.

- In the assembly editor, select the component you want to modify, right-click, and select **Edit Binding**.
- In the Business Integration view, select the component you want to modify, right-click, and select **Edit Binding**.
- Select the component in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

Note: The Edit Binding option is available for artifacts generated using IBM Integration Designer 7.0 only. If you are importing a project interchange from an earlier version of IBM Integration Designer, the Edit Binding option is not available. If you have made any manual changes to the configuration, running the wizard again will overwrite these changes.

Modifying service import

Modify an import component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service import interface. The wizard automatically populates the existing information for the selected import interface. You can modify the objects and services, and then regenerate the import component with the modified data.

Procedure

1. Invoke the external service wizard for the selected service interface import component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.

- Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected import interface.

2. In the Find Objects in the Enterprise System window, make the required changes. For more information about discovering objects, see “Discovering database objects” on page 101.

Note: To change the connection properties for the external service wizard, click **Back** and change the properties in the Specify the Discovery Properties window. For more information see, “Setting connection properties for the external service wizard” on page 83.

a. You can select and configure the following objects:



- To select and configure business objects for tables, views, and synonyms or nicknames for use in your module, see “Selecting and configuring tables, views, and synonyms or nicknames for outbound processing” on page 106.

Note: If a table selected during the previous enterprise service discovery is deleted in the database, the adapter generates the Object not found exception.

- To select and configure business objects corresponding to stored procedures and stored functions in the database, see “Selecting and configuring stored procedures and stored functions” on page 114.

Note: If the stored procedure definition in the database is changed, you must reconfigure the stored procedure and ensure that validation is successful.

Note: Ensure that the sequence number of specific overloaded SP/SF is not changed when you make changes in the database. If you remove or insert an overloaded SP/SF before the selected SP/SF, the artifacts are either overwritten or made redundant during the edit binding operation.

- To select and configure query business objects, see “Selecting and configuring query business objects” on page 119.
- b. To modify the configuration of an object from the **Selected objects** list, select the object name and then click the  (Edit) icon.
 - c. To remove an object from the **Selected objects** list, select the object name and then click the  (Remove) button.

3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
4. In the Specify Composite Properties window, specify properties that apply to all business objects. For more information, see “Setting global properties for operations” on page 122.
5. Click **Next**.
6. In the Service Generation window, modify the service operations if required.
7. Click **Finish**. The artifacts are regenerated.

Note: If the service export interface that you want to modify contains a Java component added to it, ensure that the Java component is regenerated

manually in Integration Designer after completing the regeneration of the artifacts to resolve the Java component exception.

8. Complete any other required manual configuration. For more information, see “Completing the configuration” on page 131.

Results

The artifacts are regenerated.

What to do next

You can test and deploy your module.

Modifying service export

Modify an export component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service export interface. The wizard automatically populates the existing information for the selected export interface. You can modify the objects and services, and then regenerate the export component with the modified data.

Procedure

1. Invoke the external service wizard for the selected service interface export component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.
 - Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected export interface.

2. In the Find Objects in the Enterprise System window, make the required changes. For more information about discovering objects, see “Discovering database objects” on page 133.



Note: To change the connection properties for the external service wizard, click **Back** and change the properties in the Specify the Discovery Properties window. For more information, see “Setting connection properties for the external service wizard” on page 83.

a. You can select and configure the following objects:

- To select and configure business objects for tables, views, and synonyms or nicknames for use in your module, see “Selecting and configuring tables, views, and synonyms or nicknames for inbound processing” on page 137.

Note: If a table selected during the previous enterprise service discovery is deleted in the database, the adapter generates the Object not found exception.

Note: Ensure that the sequence number of specific overloaded SP/SF is not changed when you make changes in the database. If you remove or insert an overloaded SP/SF before the selected SP/SF, the artifacts are either overwritten or made redundant during the edit binding operation.

- b. To modify the configuration of an object from the **Selected objects** list, select the object name and then click the  (Edit) icon.
 - c. To remove an object from the **Selected objects** list, select the object name and then click the  (Remove) button.
3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
 4. In the Specify Composite Properties window, specify properties that apply to all business objects. For more information, see “Setting global properties for operations” on page 145.
 5. Click **Next**.
 6. In the Service Generation window, modify the service operations if required.
 7. Click **Finish**. The artifacts are regenerated.

Note: If the service export interface that you want to modify contains a Java component added to it, ensure that the Java component is regenerated manually in Integration Designer after completing the regeneration of the artifacts to resolve the Java component exception.

8. Complete any other required manual configuration. For more information, see “Completing the configuration” on page 131.

Results

The artifacts are regenerated.

What to do next

You can test and deploy your module.

Chapter 6. Changing interaction specification properties

To change interaction specification properties for your adapter module after generating the service, use the assembly editor in IBM Integration Designer.

Before you begin

You must have used the external service wizard to generate a service for the adapter.

About this task

You might want to change interaction specification properties after you have generated a service for the adapter. Interaction specification properties, which are optional, are set at the method level, for a specific operation on a specific business object. The values you specify appear as defaults in all parent business objects generated by the external service wizard. You can change these properties before you export the EAR file. You cannot change these properties after you deploy the application.

To change the interaction specification properties, use the following procedure:

Procedure

1. From the Business Integration perspective of IBM Integration Designer, expand the module name.
2. Expand **Assembly Diagram** and double-click the interface.
3. Click the interface in the assembly editor. The module properties are displayed.
4. Click the **Properties** tab. You can also right-click the interface in the assembly diagram and click **Show in Properties**.
5. Under **Binding**, click **Method bindings**. The methods for the interface are displayed, one for each combination of business object and operation.
6. Select the method whose interaction specification property you want to change.
7. Change the property in the **Generic** tab. Repeat this step for each method whose interaction specification property you want to change.

Results

The interaction specification properties associated with your adapter module are changed.

What to do next

Deploy the module.

Related reference

“Interaction specification properties” on page 249

Interaction specification, or `InteractionSpec`, properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, some properties for outbound operations can be changed by the user. For example, you might increase the value of the interaction specification property that specifies the maximum number of records to be returned by a `RetrieveAll` operation, if your `RetrieveAll` operations do not return complete information. To change these properties after the application is deployed, use the assembly editor in IBM Integration Designer. The properties reside in the method binding of the import.

Chapter 7. Deploying the module

Deploy a module to place the files that make up your module and adapter into an operational environment for production or testing. In IBM Integration Designer, the integrated test environment features runtime support for IBM Business Process Manager or WebSphere Enterprise Service Bus, or both, depending on the test environment profiles that you selected during installation.

Deployment environments

There are test and production environments into which you can deploy modules and adapters.

In IBM Integration Designer, you can deploy your modules to one or more servers in the test environment. This is typically the most common practice for running and testing business integration modules. However, you can also export modules for server deployment on IBM Business Process Manager or WebSphere Enterprise Service Bus as EAR files using the administrative console or command-line tools.

Deploying the module for testing

In IBM Integration Designer, you can deploy a module that includes an embedded adapter to the test environment and work with server tools that enable you to perform such tasks as editing server configurations, starting, and stopping servers and testing the module code for errors. The testing is generally performed on the interface operations of your components, which enables you to determine whether the components are correctly implemented and the references are correctly wired.

Adding external dependencies

Dependent JARs have to be added to the libraries directory or packaged in the EAR.

About this task

The JARs are set in the class path and these dependent libraries have to be made available for run time when the module is deployed. There are two ways to make the dependent libraries available, one for either stand-alone deployment or embedded deployment and the other for embedded deployment only.

Adding external software dependencies on the server

The adapter needs certain files installed on IBM Business Process Manager or WebSphere Enterprise Service Bus server to be able to communicate with the Oracle database.

Before you begin

You need not perform this task if the Oracle database is installed on the same computer system as IBM Business Process Manager or WebSphere Enterprise Service Bus. The files are already available to the adapter.

About this task

An adapter needs the Oracle JDBC driver-specific files to be able to communicate with it.

Procedure

1. Obtain the Oracle JDBC driver-specific files or native libraries for your database software and operating system from your database administrator or from the database software Website

The following table lists the Oracle JDBC driver files needed for Oracle database software.

Table 14. Oracle JDBC driver files for Oracle database software

Database software	Driver	JDBC driver files	Native System Libraries
Oracle	Thin driver	ojdbc6.jar or ojdbc14.jar depending on the JRE version	None

Note: The Java Runtime Environment (JRE) version required by the Oracle JDBC driver must not be higher than the JRE version in the runtime environment. For example, if the JRE version is 1.5 in the runtime environment, then the Oracle JDBC Driver should be "ojdbc14.jar" which supports both Java Development Kit (JDK) 1.4 and JDK 1.5; if the JRE version is 1.6 in the runtime environment, then the Oracle JDBC Driver should be "ojdbc6.jar" which supports JDK version 1.6.

2. Copy the files to the server.
 - In a testing environment in Integration Designer, copy the files to the `${WAS_INSTALL_ROOT}/runtimes/bi_v7/lib/ext` directory.
 - In a production environment, copy the files to the `${WAS_INSTALL_ROOT}/lib/ext` directory of IBM Business Process Manager or WebSphere Enterprise Service Bus.

Adding external software dependencies when the adapter is bundled

You must copy the dependent JAR files to the EAR application before you can run your adapter applications. You must use this method only for embedded deployment.

About this task

To obtain the required files and copy them to the EAR application, use the following procedure:

Procedure

1. From the appropriate module, go to the workspace and copy the JAR files to the directory. For example if the name of the module is `ModuleName`, then go to the workspace and copy the JAR files to the `ModuleNameApp/EarContent` directory.
2. Modify the adapter RAR's manifest file, `manifest.mf`, with the list of JAR files required by the adapter. Add the JAR files in the following format: `Class-Path: dependantjar1.jar, dependantjar2.jar`.
3. Copy the native libraries to the run time bin directory and deploy the application.

Results

The vendor software libraries are now a part of your run time environment.

Generating and wiring a target component for testing inbound processing

Before deploying to the test environment a module that includes an adapter for inbound processing, you must first generate and wire a target component. This target component serves as the *destination* to which the adapter sends events.

Before you begin

You must have generated an export module, using the external service wizard.

About this task

Generating and wiring a target component for inbound processing is required in a testing environment only. It is not necessary when deploying the adapter in a production environment.

The target component receives events. You *wire* the export to the target component (connecting the two components) using the assembly editor in IBM Integration Designer. The adapter uses the wire to pass event data (from the export to the target component).

Procedure

1. Create the target component.
 - a. From the Business Integration perspective of IBM Integration Designer, expand **Assembly Diagram** and double-click the export component. If you did not change the default value, the name of the export component is the name of your adapter + **InboundInterface**.

An interface specifies the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions. The **InboundInterface** contains the operations required by the adapter to support inbound processing and is created when you run the external service wizard.
 - b. Create a new component by expanding **Components**, selecting **Untyped Component**, and dragging the component to the Assembly Diagram.

The cursor changes to the placement icon.
 - c. Click the component to have it displayed in the Assembly Diagram.
2. Wire the components.
 - a. Click and drag the export component to the new component.
 - b. Save the assembly diagram. Click **File > Save**.
3. Generate an implementation for the new component.
 - a. Right-click on the new component and select **Generate Implementation > Java**.
 - b. Select **(default package)** and click **OK**. This creates an endpoint for the inbound module.

The Java implementation is displayed in a separate tab.
 - c. **Optional:** Add print statements to print the data object received at the endpoint for each of the endpoint methods.

- d. Click **File > Save** to save the changes.

What to do next

Continue deploying the module for testing.

Preparing to test outbound operations

Before you can test your module's outbound processing with the IBM Integration Designer test client, you might need to modify some of your business objects.

About this task

This step is performed in the IBM Integration Designer test client. Open IBM Integration Designer it from the Business Integration perspective by right-clicking the name of your project and then clicking **Test > Test Module**.

Procedure

- **Query business objects**

If your query business object was created without a WHERE clause (for example, it was defined with a SELECT statement like `Select * from Customer`), unset the `jdbcwhereclause` attribute of the query business object before testing in the test client.

- **Table, view, and synonyms or nicknames business objects**

Before testing the RetrieveAll operation, you need to unset any attribute whose value you are not setting as part of your test.

- **Query business objects**

Before testing the RetrieveAll operation, you need to unset any attribute whose value you are not setting as part of your test.

Adding the module to the server

In IBM Integration Designer, you can add modules to one or more servers in the test environment.

Before you begin

If the module you are testing uses an adapter to perform inbound processing, generate and wire a *target component* to which the adapter sends the events.

About this task

In order to test your module and its use of the adapter, you need to add the module to the server.

Procedure

1. *Conditional:* If there are no servers in the **Servers** view, add and define a new server by performing the following steps:
 - a. Place your cursor in the **Servers** view, right-click, and select **New > Server**.
 - b. From the Define a New Server window, select the server type.
 - c. Configure servers settings.
 - d. Click **Finish** to publish the server.
2. Add the module to the server.

- a. Switch to the servers view. In IBM Integration Designer, select **Windows > Show View > Servers**.
- a. Start the server. In the **Servers** tab in the lower-right pane of the IBM Integration Designer screen, right-click the server, and then select **Start**.
3. When the server status is *Started*, right-click the server, and select **Add and Remove Projects**.
4. In the Add and Remove Projects screen, select your project and click **Add**. The project moves from the **Available projects** list to the **Configured projects** list.
5. Click **Finish**. This deploys the module on the server.
The Console tab in the lower-right pane displays a log while the module is being added to the server.

What to do next

Test the functionality of your module and the adapter.

Testing the module for outbound processing using the test client

Test the assembled module and adapter for outbound processing using the IBM Integration Designer integration test client.

Before you begin

You need to add the module to the server first.

About this task

Testing a module is performed on the interface operations of your components, which enables you to determine whether the components are correctly implemented and the references are correctly wired.

Procedure

1. Select the module you want to test, right-click on it, and select **Test > Test Module**.
2. For information about testing a module using the test client, see the *Testing modules and components* topic in the IBM Integration Designer information center.

What to do next

If you are satisfied with the results of testing your module and adapter, you can deploy the module and adapter to the production environment.

Deploying the module for production

Deploying a module created with the external service wizard to IBM Business Process Manager or WebSphere Enterprise Service Bus in a production environment is a two-step process. First, you export the module in IBM Integration Designer as an enterprise archive (EAR) file. Second, you deploy the EAR file using the IBM Business Process Manager or WebSphere Enterprise Service Bus administrative console.

Adding external software dependencies on the server

The adapter needs certain files installed on IBM Business Process Manager or WebSphere Enterprise Service Bus server to be able to communicate with the Oracle database.

Before you begin

You need not perform this task if the Oracle database is installed on the same computer system as IBM Business Process Manager or WebSphere Enterprise Service Bus. The files are already available to the adapter.

About this task

An adapter needs the Oracle JDBC driver-specific files to be able to communicate with it.

Procedure

1. Obtain the Oracle JDBC driver-specific files or native libraries for your database software and operating system from your database administrator or from the database software Website

The following table lists the Oracle JDBC driver files needed for Oracle database software.

Table 15. Oracle JDBC driver files for Oracle database software

Database software	Driver	JDBC driver files	Native System Libraries
Oracle	Thin driver	ojdbc6.jar or ojdbc14.jar depending on the JRE version	None

Note: The Java Runtime Environment (JRE) version required by the Oracle JDBC driver must not be higher than the JRE version in the runtime environment. For example, if the JRE version is 1.5 in the runtime environment, then the Oracle JDBC Driver should be "ojdbc14.jar" which supports both Java Development Kit (JDK) 1.4 and JDK 1.5; if the JRE version is 1.6 in the runtime environment, then the Oracle JDBC Driver should be "ojdbc6.jar" which supports JDK version 1.6.

2. Copy the files to the server.
 - In a testing environment in Integration Designer, copy the files to the `${WAS_INSTALL_ROOT}/runtimes/bi_v7/lib/ext` directory.
 - In a production environment, copy the files to the `${WAS_INSTALL_ROOT}/lib/ext` directory of IBM Business Process Manager or WebSphere Enterprise Service Bus.

Installing the RAR file (for modules using stand-alone adapters only)

If you chose not to embed the adapter with your module, but instead choose to make the adapter available to all deployed applications in the server instance, you need to install the adapter in the form of a RAR file to the application server. A RAR file is a Java archive (JAR) file that is used to package a resource adapter for the Java 2 Connector (J2C) architecture.

Before you begin

You must set **Deploy connector project** to **On server for use by multiple adapters** in the Specify the Service Generation and Deployment Properties window of the external service wizard.

About this task

Installing the adapter in the form of a RAR file results in the adapter being available to all J2EE application components running in the server run time.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **Install RAR**.

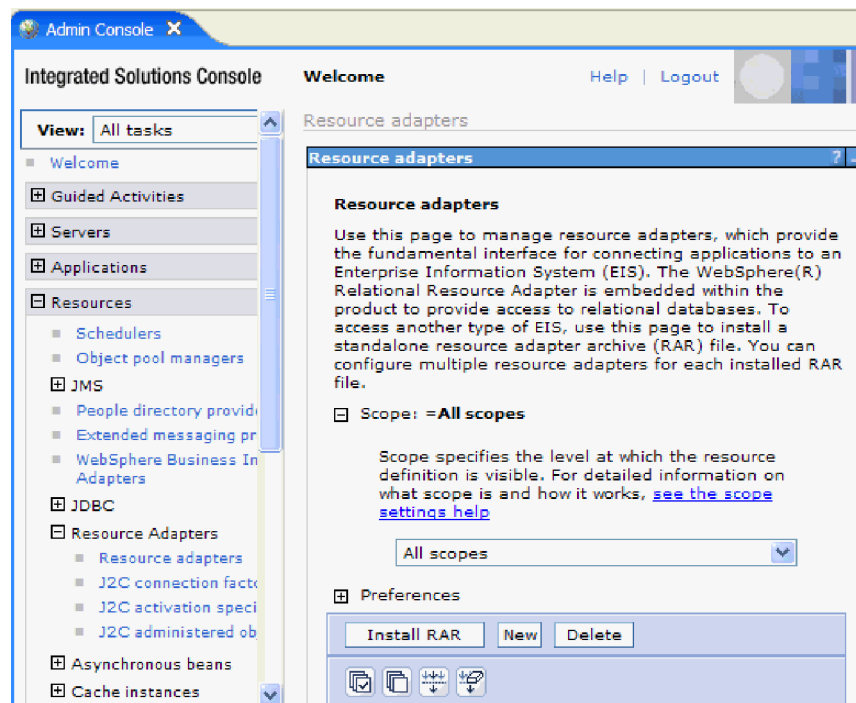


Figure 22. The Install RAR button on the Resource adapters page

6. In the Install RAR file page, click **Browse** and navigate to the RAR file for your adapter.
The RAR files are typically installed in the following path:
`IID_installation_directory/ResourceAdapters/adapter_name/adapter.rar`
7. Click **Next**.
8. Optional: In the Resource adapters page, change the name of the adapter and add a description.
9. Click **OK**.

10. Click **Save** in the **Messages** box at the top of the page.

What to do next

The next step is to export the module as an EAR file that you can deploy on the server.

Exporting the module as an EAR file

Using IBM Integration Designer, export your module as an EAR file. By creating an EAR file, you capture all of the contents of your module in a format that can be easily deployed to IBM Business Process Manager or WebSphere Enterprise Service Bus.

Before you begin

Before you can export a module as an EAR file, you must have created a module to communicate with your service. The module should be displayed in the IBM Integration Designer Business Integration perspective.

About this task

To export the module as an EAR file, perform the following procedure.

Procedure

1. Right-click the module and select **Export**.
2. In the Select window, expand **Java EE**.
3. Select **EAR file** and click **Next**.
4. Optional: Select the correct EAR application. The EAR application is named after your module, but with “App” added to the end of the name.
5. Browse for the folder on the local file system where the EAR file will be placed.
6. To export the source files, select the **Export source files** check box. This option is provided in case you want to export the source files in addition to the EAR file. Source files include files associated with Java components, data maps, and so on.
7. To overwrite an existing file, click **Overwrite existing file**.
8. Click **Finish**.

Results

The contents of the module are exported as an EAR file.

What to do next

Install the module in the administrative console. This deploys the module to IBM Business Process Manager or WebSphere Enterprise Service Bus.

Installing the EAR file

Installing the EAR file is the last step of the deployment process. When you install the EAR file on the server and run it, the adapter, which is embedded as part of the EAR file, runs as part of the installed application.

Before you begin

You must have exported your module as an EAR file before you can install it on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

To install the EAR file, perform the following procedure. For more information about clustering adapter module applications, see the <http://www.ibm.com/software/webservers/appserv/was/library/>.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > New Application > New Enterprise Application**.

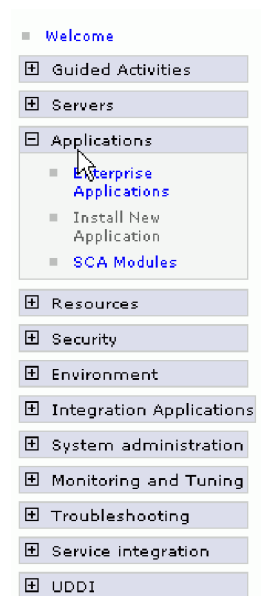


Figure 23. Preparing for the application installation window

5. Click **Browse** to locate your EAR file and click **Next**. The EAR file name is the name of the module followed by "App."
6. Optional: If you are deploying to a clustered environment, complete the following steps.
 - a. On the **Step 2: Map modules to servers** window, select the module and click **Next**.
 - b. Select the name of the server cluster.
 - c. Click **Apply**.
7. Click **Next**. In the Summary page, verify the settings and click **Finish**.
8. Optional: If you are using an authentication alias, complete the following steps:
 - a. Expand **Security** and select **Business Integration Security**.

- b. Select the authentication alias that you want to configure. You must have administrator or operator rights to change the authentication alias configurations.
- c. Optional: If it is not already specified, type the **User name**.
- d. If it is not already specified, type the **Password**.
- e. If it is not already specified, type the password again in the **Confirm Password** field.
- f. Click **OK**.

Results

The project is now deployed and the Enterprise Applications window is displayed.

What to do next

If you want to set or reset any properties or you would like to cluster adapter project applications, make those changes using the administrative console before configuring troubleshooting tools.

Chapter 8. Administering the adapter module

When you are running the adapter in a stand-alone deployment, use the administrative console of the server to start, stop, monitor, and troubleshoot the adapter module. In an application that uses an embedded adapter, the adapter module starts or stops when the application is started or stopped.

Changing configuration properties for embedded adapters

To change the configuration properties after you deploy the adapter as part of a module, you use the administrative console of the runtime environment. You can update resource adapter properties (used for general adapter operation), managed connection factory properties (used for outbound processing), and activation specification properties (used for inbound processing).

Setting resource adapter properties for embedded adapters

To set resource adapter properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter module must be deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

Custom properties are default configuration properties shared by all IBM WebSphere Adapters.

To configure properties using the administrative console, use the following procedure:

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. From the Enterprise Applications list, click the name of the adapter module whose properties you want to change. The **Configuration** page is displayed.

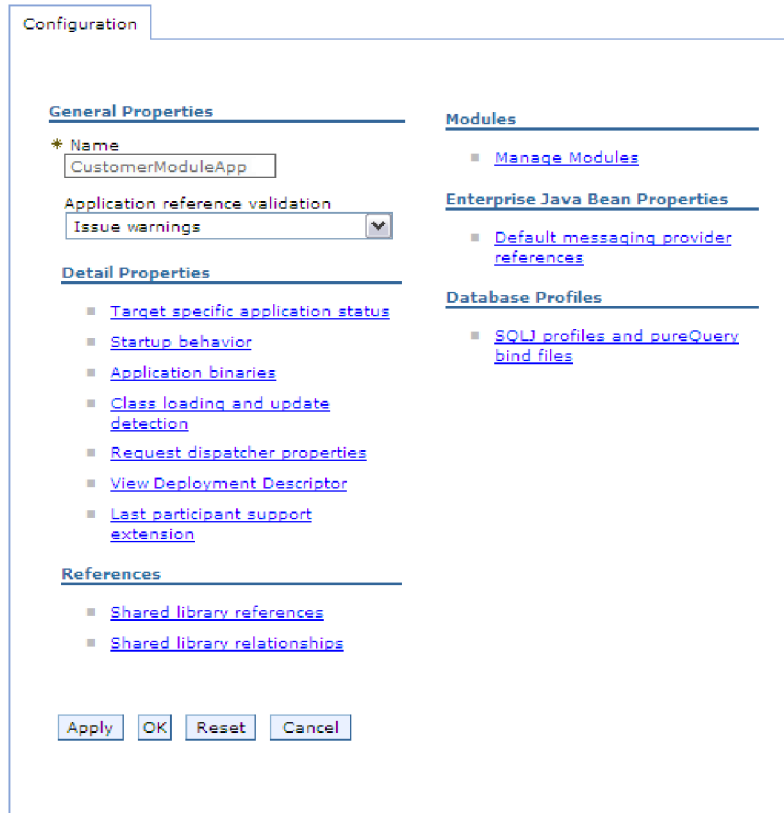


Figure 24. The Manage Modules selection in the Configuration tab

6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for Oracle**.
8. From the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **Custom properties**.
10. For each property you want to change, perform the following steps.

Note: See “Resource adapter properties” on page 225 for more information about these properties.

- a. Click the name of the property. The **Configuration** page for the selected property is displayed.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
11. In the Messages area, click **Save**.

Results

The resource adapter properties associated with your adapter module are changed.

Related reference

“Resource adapter properties” on page 225

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

Setting managed (J2C) connection factory properties for embedded adapters

To set managed connection factory properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter module must be deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

You use managed connection factory properties to configure the target Oracle database instance.

Note: In the administrative console, the properties are referred to as "J2C connection factory properties."

To configure properties using the administrative console, use the following procedure.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. In the Enterprise Applications list, click the name of the adapter module whose properties you want to change.

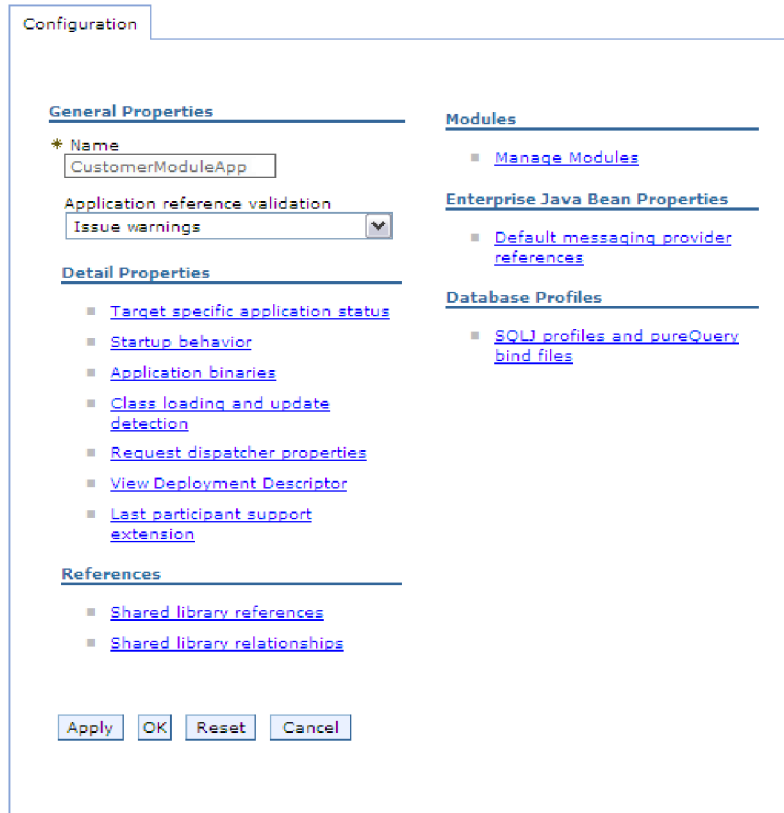


Figure 25. The Manage Modules selection in the Configuration tab

6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for Oracle**.
8. In the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **J2C connection factories**.
10. Click the name of the connection factory associated with your adapter module.
11. In the **Additional Properties** list, click **Custom properties**.
 Custom properties are those J2C connection factory properties that are unique to IBM WebSphere Adapter for Oracle E-Business Suite. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.
12. For each property you want to change, perform the following steps.

Note: See “Managed connection factory properties” on page 230 for more information about these properties.

 - a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
13. In the Messages area, click **Save**.

Results

The managed connection factory properties associated with your adapter module are changed.

Related reference

“Managed connection factory properties” on page 230

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Oracle E-Business Suite.

Setting activation specification properties for embedded adapters

To set activation specification properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the message endpoint property you want to configure, and then change or set the value.

Before you begin

Your adapter module must be deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

You use activation specification properties to configure the endpoint for inbound processing.

To configure properties using the administrative console, use the following procedure.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. From the Enterprise Applications list, click the name of the adapter module whose properties you want to change.

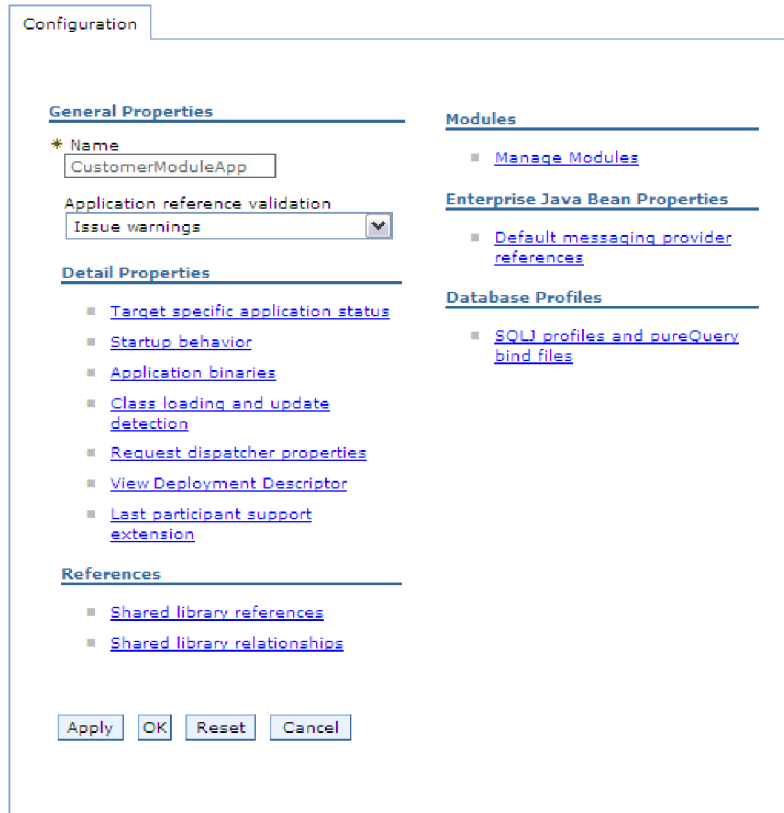


Figure 26. The Manage Modules selection in the Configuration tab

6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for Oracle**.
8. From the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **J2C activation specifications**.
10. Click the name of the activation specification associated with the adapter module.
11. From the **Additional Properties** list, click **J2C activation specification custom properties**.
12. For each property you want to change, perform the following steps.

Note: See “Activation specification properties” on page 256 for more information about these properties.

- a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
13. In the Messages area, click **Save**.

Results

The activation specification properties associated with your adapter module are changed.

Related reference

“Activation specification properties” on page 256

Activation specification properties are properties that hold the inbound event processing configuration information for an export.

Changing configuration properties for stand-alone adapters

To set configuration properties after you install a stand-alone adapter, use the administrative console of the runtime environment. Provide the general information about the adapter and then set the resource adapter properties (which are used for general adapter operation). If the adapter is used for outbound operations, create a connection factory and then set the properties for it. If the adapter is used for inbound operations, create an activation specification and then set the properties for it.

Setting resource adapter properties for stand-alone adapters

To set resource adapter properties for your stand-alone adapter after it has been installed on IBM Business Process Manager or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter must be installed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

Custom properties are default configuration properties shared by all IBM WebSphere Adapters.

To configure properties using the administrative console, use the following procedure:

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for Oracle**.
6. In the **Additional Properties** list, click **Custom properties**.
7. For each property you want to change, perform the following steps.

Note: See “Resource adapter properties” on page 225 for more information about these properties.

- a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
8. In the Messages area, click **Save**.

Results

The resource adapter properties associated with your adapter are changed.

Related reference

“Resource adapter properties” on page 225

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

Setting managed (J2C) connection factory properties for stand-alone adapters

To set managed connection factory properties for your stand-alone adapter after it has been installed on IBM Business Process Manager or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter must be installed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

You use managed connection factory properties to configure the target Oracle database instance.

Note: In the administrative console, the properties are referred to as "J2C connection factory properties."

To configure properties using the administrative console, use the following procedure:

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for Oracle**.
6. In the **Additional Properties** list, click **J2C connection factories**.
7. If you are going to use an existing connection factory, skip ahead to select from the list of existing connection factories.

Note: If you have selected **Specify connection properties** when you use the external service wizard to configure the adapter module, you do not need to create a connection factory.

If you are creating a connection factory, perform the following steps:

- a. Click **New**.
- b. In the **General Properties** section of the **Configuration** tab, type a name for the connection factory. For example, you can type AdapterCF.

- c. Type a value for **JNDI name**. For example, you can type `com/eis/AdapterCF`.
- d. Optional: Select an authentication alias from the **Component-managed authentication alias** list.
- e. Click **OK**.
- f. In the Messages area, click **Save**.

The newly created connection factory is displayed.

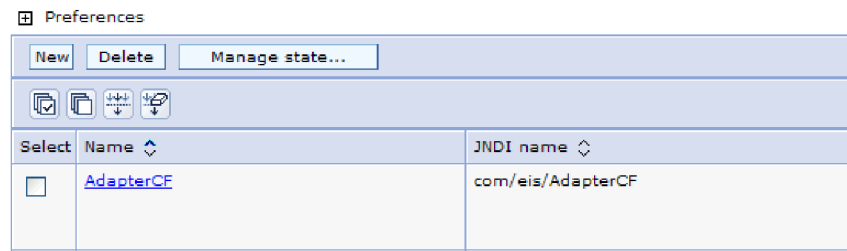


Figure 27. User-defined connection factories for use with the resource adapter

8. In the list of connection factories, click the one you want to use.
9. In the **Additional Properties** list, click **Custom properties**.
Custom properties are those J2C connection factory properties that are unique to WebSphere Adapter for Oracle E-Business Suite. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.
10. For each property you want to change, perform the following steps.

Note: See “Managed connection factory properties” on page 230 for more information about these properties.

 - a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
11. After you have finished setting properties, click **Apply**.
12. In the Messages area, click **Save**.

Results

The managed connection factory properties associated with your adapter are set.

Related reference

“Managed connection factory properties” on page 230

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Oracle E-Business Suite.

Setting activation specification properties for stand-alone adapters

To set activation specification properties for your stand-alone adapter after it has been installed on IBM Business Process Manager or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the message endpoint property you want to configure, and then change or set the value.

Before you begin

Your adapter must be installed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

You use activation specification properties to configure the endpoint for inbound processing.

To configure properties using the administrative console, use the following procedure.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for Oracle**.
6. In the **Additional Properties** list, click **J2C activation specifications**.
7. If you are going to use an existing activation specification, skip ahead to select from an existing list of activation specifications.

Note: If you have selected **Use predefined connection properties** when you use the external service wizard to configure the adapter module, you must create an activation specification.

If you are creating an activation specification, perform the following steps:

- a. Click **New**.
 - b. In the **General Properties** section of the **Configuration** tab, type a name for the activation specification. For example, you can type AdapterAS.
 - c. Type a value for **JNDI name**. For example, you can type com/eis/AdapterAS.
 - d. Optional: Select an authentication alias from the **Authentication alias** list.
 - e. Select a message listener type.
 - f. Click **OK**.
 - g. Click **Save** in the **Messages** box at the top of the page.
The newly created activation specification is displayed.
8. In the list of activation specifications, click the one you want to use.
 9. In the Additional Properties list, click **J2C activation specification custom properties**.
 10. For each property you want to set, perform the following steps.

Note: See “Activation specification properties” on page 256 for more information about these properties.

- a. Click the name of the property.
- b. Change the contents of the **Value** field or type a value, if the field is empty.
- c. Click **OK**.

11. After you have finished setting properties, click **Apply**.
12. In the Messages area, click **Save**.

Results

The activation specification properties associated with your adapter are set.

Related reference

“Activation specification properties” on page 256

Activation specification properties are properties that hold the inbound event processing configuration information for an export.

Starting the application that uses the adapter

Use the administrative console of the server to start an application that uses the adapter. By default, the application starts automatically when the server starts.

About this task

Use this procedure to start the application, whether it is using an embedded or a stand-alone adapter. For an application that uses an embedded adapter, the adapter starts when the application starts. For an application that uses a stand-alone adapter, the adapter starts when the application server starts.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > Application Types > WebSphere enterprise applications**.

Note: The administrative console is labeled “Integrated Solutions Console”.

5. Select the application that you want to start. The application name is the name of the EAR file you installed, without the .EAR file extension.
6. Click **Start**.

Results

The status of the application changes to Started, and a message stating that the application has started displays at the top of the administrative console.

Stopping the application that uses the adapter

Use the administrative console of the server to stop an application that uses the adapter. By default, the application stops automatically when the server stops.

About this task

Use this procedure to stop the application, whether it is using an embedded or a stand-alone adapter. For an application with an embedded adapter, the adapter stops when the application stops. For an application that uses a stand-alone adapter, the adapter stops when the application server stops.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > Application Types > WebSphere enterprise applications**.

Note: The administrative console is labeled “Integrated Solutions Console”.

5. Select the application that you want to stop. The application name is the name of the EAR file you installed, without the .EAR file extension.
6. Click **Stop**.

Results

The status of the application changes to Stopped, and a message stating that the application has stopped is displayed at the top of the administrative console.

Monitoring performance using Performance Monitoring Infrastructure

Performance Monitoring Infrastructure (PMI) is a feature of the administrative console that allows you to dynamically monitor the performance of components in the production environment, including IBM WebSphere Adapter for Oracle E-Business Suite. PMI collects adapter performance data, such as average response time and total number of requests, from various components in the server and organizes the data into a tree structure. You can view the data through the Tivoli® Performance Viewer, a graphical monitoring tool that is integrated with the administrative console in IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

You can monitor the performance of your adapter by having PMI collect data at the following points:

- At outbound processing to monitor outbound requests.
- At inbound event retrieval to monitor the retrieval of an event from the event table.
- At inbound event delivery to monitor the delivery of an event to the endpoint or endpoints.

Before you enable and configure PMI for your adapter, you must first set the level of tracing detail and run some events from which to gather performance data.

To learn more about how PMI can help you monitor and improve the overall performance of your adapter environment, search for PMI on the IBM Business Process Manager or WebSphere Enterprise Service Bus website:
<http://www.ibm.com/software/webservers/appserv/was/library/>.

Configuring Performance Monitoring Infrastructure

You can configure Performance Monitoring Infrastructure (PMI) to gather adapter performance data, such as average response time and total number of requests. After you configure PMI for your adapter, you can monitor the adapter performance using Tivoli Performance viewer.

Before you begin

Before you can configure PMI for your adapter, you must first set the level of tracing detail and run some events to gather the performance data.

1. To enable tracing and to receive event data, the trace level must be set to either fine, finer, finest, or all. After *=info, add a colon and a string, for example:

```
*=info: WBILocationMonitor.CEI.ResourceAdapter.  
*=finest: WBILocationMonitor.LOG.ResourceAdapter.*=finest:
```

For instructions on setting the trace level, see “Enabling tracing with the Common Event Infrastructure” on page 189.

2. Generate at least one outbound request or inbound event to produce performance data that you can configure.

Procedure

1. Enable PMI for your adapter.
 - a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
 - b. From the list of servers, click the name of your server.
 - c. Select the Configuration tab, and then select the **Enable Performance Monitoring (PMI)** check box.
 - d. Select **Custom** to selectively enable or disable statistics.

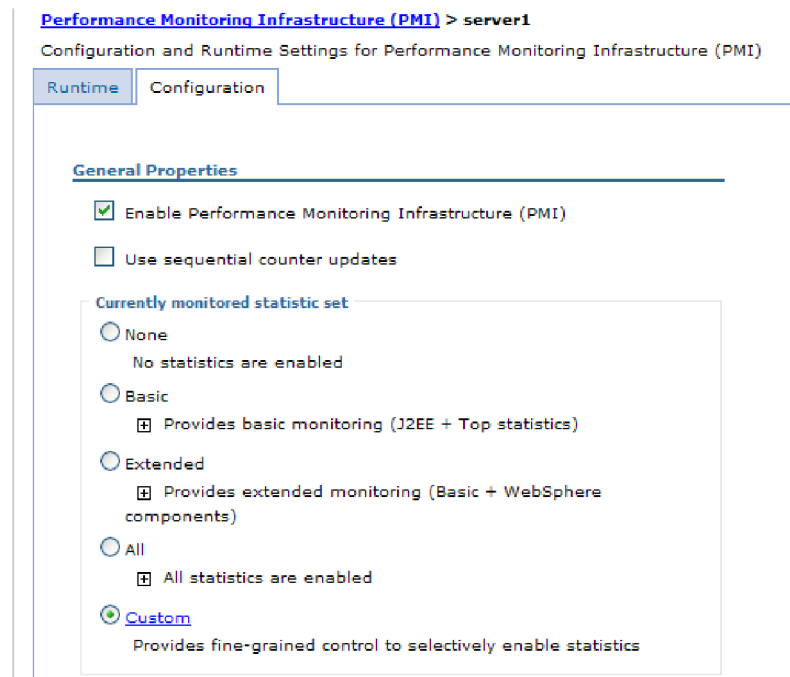


Figure 28. Enabling Performance Monitoring Infrastructure

- e. Click **Apply** or **OK**.
 - f. Click **Save**. PMI is now enabled.
2. Configure PMI for your adapter.
 - a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
 - b. From the list of servers, click the name of your server.

- c. Select **Custom**.
- d. Select the **Runtime** tab. The following figure shows the Runtime tab.

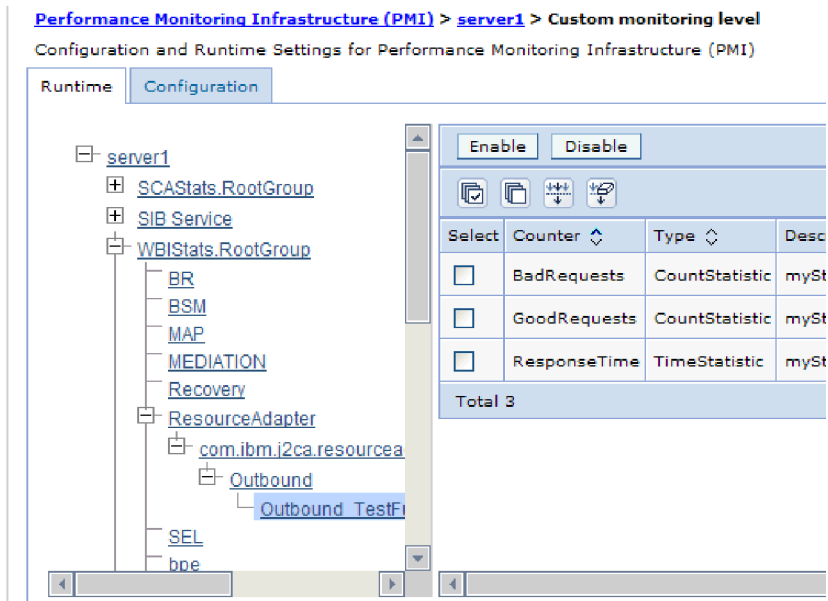


Figure 29. Runtime tab used for configuring PMI

- e. Click **WBISStats.RootGroup**. This is a PMI sub module for data collected in the root group. This example uses the name WBISStats for the root group.
- f. Click **ResourceAdapter**. This is a sub module for the data collected for the JCA adapters.
- g. Click the name of your adapter, and select the processes you want to monitor.
- h. In the right pane, select the check boxes for the statistics you want to gather, and then click **Enable**.

Results

PMI is configured for your adapter.

What to do next

Now you can view the performance statistics for your adapter.

Viewing performance statistics

You can view adapter performance data through the graphical monitoring tool, Tivoli Performance Viewer. Tivoli Performance Viewer is integrated with the administrative console in IBM Business Process Manager or WebSphere Enterprise Service Bus.

Before you begin

Configure Performance Monitoring Infrastructure for your adapter.

Procedure

1. In the administrative console, expand **Monitoring and Tuning**, expand **Performance Viewer**, then select **Current Activity**.
2. In the list of servers, click the name of your server.
3. Under your server name, expand **Performance Modules**.
4. Click **WBIStatsRootGroup**.
5. Click **ResourceAdapter** and the name of your adapter module.
6. If there is more than one process, select the check boxes for the processes whose statistics you want to view.

Results

The statistics are displayed in the right panel. You can click **View Graph** to view a graph of the data, or **View Table** to see the statistics in a table format.

The following figure shows adapter performance statistics.

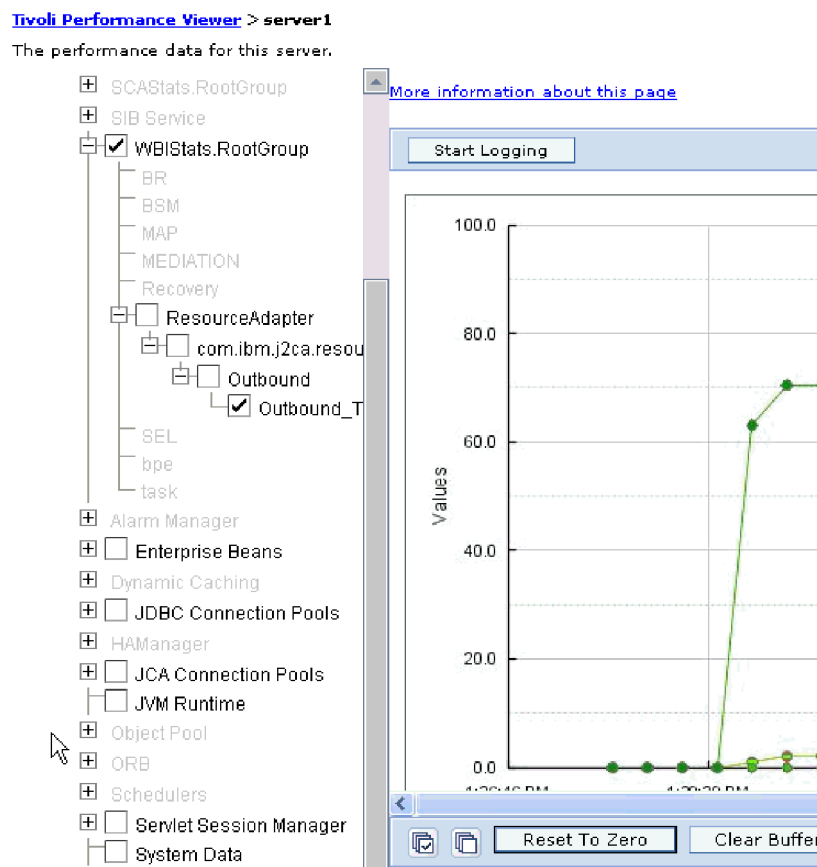


Figure 30. Adapter performance statistics, using graph view

Enabling tracing with the Common Event Infrastructure

The adapter can use the Common Event Infrastructure (CEI), a component embedded in the server, to report data about critical business events such as the starting or stopping of a poll cycle. Event data can be written to a database or a trace log file depending on configuration settings.

About this task

Use this procedure to report CEI entries in the trace log file by using the Common Base Event Browser within the administrative console.

Procedure

1. In the administrative console, click **Troubleshooting**.
2. Click **Logs and Trace**.
3. From the list of servers, click the name of your server.
4. In the **Change Log Detail Levels** box, click the name of the CEI database (for example, `WBIEventMonitor.CEI.ResourceAdapter.*`) or the trace log file (for example, `WBIEventMonitor.LOG.ResourceAdapter.*`) to which you want the adapter to write event data.
5. Select the level of detail about business events that you want the adapter to write to the database or trace log file, and (optionally) adjust the granularity of detail associated with messages and traces.
 - **No Logging**. Turns off event logging.
 - **Messages Only**. The adapter reports an event.
 - **All Messages and Traces**. The adapter reports details about an event.
 - **Message and Trace Levels**. Settings for controlling the degree of detail the adapter reports about the business object payload associated with an event. If you want to adjust the detail level, select one of the following options:
 - Fine**. The adapter reports the event but none of the business object payload.
 - Finer**. The adapter reports the event and the business object payload description.
 - Finest**. The adapter reports the event and the entire business object payload.
6. Click **OK**.

Results

Event logging is enabled. You can view CEI entries in the trace log file or by using the Common Base Event Browser within the administrative console.

Adding dependency libraries to the deployed resource adapter

The deployed resource adapter running in the WebSphere Application Server requires the same dependency libraries as it does in IBM Integration Designer to process requests. The method for adding these library files depends on the mode of the resource adapter deployment: standalone or embedded in the EAR file.

Stand-alone deployment

The dependency libraries may be added to the resource adapter deployed stand-alone either during initial deployment of the RAR file or by configuring the Resource Adapter properties after deployment. To set the values during initial deployment of the RAR file, specify Class path and Native path locations. Class path is used to point to JAR files, and Native path is used to point to native libraries, such as *.dll, *.so. To set the dependency library path files after the adapter has been installed on WebSphere Application Server, use the administrative console to modify the values for the Resource Adapter.

EAR deployment

For the rare case when the connector needs to be embedded in the EAR file, the dependant libraries are added as shared libraries. Define the appropriate shared library containing external dependencies and associate them with the EAR file.

About this task

There are two methods to do this task:

- Using enhanced EAR editor in IBM Integration Designer
- Using administrative console of the IBM Business Process Manager or WebSphere Enterprise Service Bus

Using enhanced EAR editor

You can use the EAR editor in IBM Integration Designer to add the dependency libraries.

About this task

To create shared libraries using the EAR editor, use the following procedure.

Procedure

1. Open Enhanced EAR editor.
2. Click the **Deployment** tab.
3. Expand **Shared Library** section.
4. Click **Add** to add new shared library.
5. Specify the shared library parameters and click **OK**.
6. Deploy the EAR to the server.

Results

The dependent libraries are added as shared libraries.

Using administrative console of the WebSphere Application Server

You can use the administrative console of the IBM Business Process Manager or WebSphere Enterprise Service Bus to add the dependency libraries.

Before you begin

Ensure that the dependent files are available on the server machine in the separate folder. If needed, copy the dependent files on the server machine.

Procedure

1. Define WebSphere variables to point to appropriate folders.
2. Define the shared library through the server administrative console; you can specify it using WebSphere variables defined in step 1.
3. Deploy the EAR to the server.
4. Configure the EAR to reference defined shared library.

Results

The dependent libraries are added as shared libraries.

Chapter 9. Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly.

Log and Trace Analyzer

The adapter creates log and trace files that can be viewed with the Log and Trace Analyzer.

The Log and Trace Analyzer can filter log and trace files to isolate the messages and trace information for the adapter. It can also highlight the adapter's messages and trace information in the log viewer.

The adapter's component ID for filtering and highlighting is a string composed of the characters 0EBSRA plus the value of the adapter ID property. For example, if the adapter ID property is set to 001, the component ID is 0EBSRA001.

If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter. For example, when you set the adapter ID property of two instances of WebSphere Adapter for Oracle E-Business Suite to 001 and 002. The component IDs for those instances, 0EBSRA001 and 0EBSRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to 0EBSRAInstanc.

For outbound processing, the adapter ID property is located in both the resource adapter and managed connection factory property groups. If you update the adapter ID property after using the external service wizard to configure the adapter for outbound processing, be sure to set the resource adapter and managed connection factory properties consistently. It prevents inconsistent marking of the log and trace entries. For inbound processing, the adapter ID property is located only in the resource adapter properties, so this consideration does not apply.

For more information, see the “Adapter ID (AdapterID)” on page 226 property.

Configuring logging and tracing

Configure logging and tracing to suit your requirements. Enable logging for the adapter to control the status of event processing. Change the adapter log and trace file names to separate them from other log and trace files.

Configuring logging properties

Use the administrative console to enable logging and to set the output properties for a log, including the location, level of detail, and output format of the log.

About this task

Before the adapters can log monitored events, you must specify the service component event points that you want to monitor, what level of detail you require for each event, and format of the output used to publish the events to the logs.

Use the administrative console to perform the following tasks:

- Enable or disable a particular event log
- Specify the level of detail in a log
- Specify where log files are stored and how many log files are kept
- Specify the format for log output

If you set the output for log analyzer format, you can open trace output using the Log Analyzer tool, which is an application included with your IBM Process Server. This is useful if you are trying to correlate traces from two different server processes, because it allows you to use the merge capability of the Log Analyzer.

Note: For more information about monitoring on a IBM Process Server, including service components and event points, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.admin.doc/topics/welcome_wps_mon.html.

You can change the log configuration statically or dynamically. Static configuration takes effect when you start or restart the application server. Dynamic or run time configuration changes apply immediately.

When a log is created, the detail level for that log is set from the configuration data. If no configuration data is available for a particular log name, the level for that log is obtained from the parent of the log. If no configuration data exists for the parent log, the parent of that log is checked, and so on, up the tree, until a log with a non-null level value is found. When you change the level of a log, the change is propagated to the child logs, which recursively propagate the change to their child log, as necessary.

To enable logging and set the output properties for a log, use the following procedure.

Procedure

1. In the navigation pane of the administrative console, select **Servers > Application Servers**.
2. Click the name of the server that you want to work with.
3. Under **Troubleshooting**, click **Logs and trace**.
4. Click **Change Log Detail Levels**.
5. Specify when you want the change to take effect:
 - For a static change to the configuration, click the **Configuration** tab.
 - For a dynamic change to the configuration, click the **Runtime** tab.
6. Click the names of the packages whose logging level you want to modify. The package names for WebSphere Adapters start with **com.ibm.j2ca.***:
 - For the adapter base component, select **com.ibm.j2ca.base.***.
 - For the adapter base component and all deployed adapters, select **com.ibm.j2ca.***.

- For the core component that is common between WebSphere Adapter for JDBC and WebSphere Adapter for Oracle E-Business Suite, select the **com.ibm.j2ca.dbadapter.core.***
- For the WebSphere Adapter for Oracle E-Business Suite only, select the **com.ibm.j2ca.oracleebs.*** package.

7. Select the logging level.

Logging Level	Description
Fatal	The task cannot continue or the component cannot function.
Severe	The task cannot continue, but the component can still function. This logging level also includes conditions that indicate an impending fatal error, that is, situations that strongly suggest that resources are on the verge of being depleted.
Warning	A potential error has occurred or a severe error is impending. This logging level also includes conditions that indicate a progressive failure, for example, the potential leaking of resources.
Audit	A significant event has occurred that affects the server state or resources.
Info	The task is running. This logging level includes general information outlining the overall progress of a task.
Config	The status of a configuration is reported or a configuration change has occurred.
Detail	The subtask is running. This logging level includes general information detailing the progress of a subtask.

8. Click **Apply**.
9. Click **OK**.
10. To have static configuration changes take effect, stop and then restart the IBM Process Server.

Results

Log entries from this point forward contain the specified level of information for the selected adapter components.

Changing the log and trace file names

To keep the adapter log and trace information separate from other processes, use the administrative console to change the file names. By default, log and trace information for all processes and applications on a IBM Process Server is written to the `SystemOut.log` and `trace.log` files.

Before you begin

You can change the log and trace file names at any time after the adapter module has been deployed to an application server.

About this task

You can change the log and trace file names statically or dynamically. Static changes take effect when you start or restart the application server. Dynamic or run time changes apply immediately.

Log and trace files are in the *install_root/profiles/profile_name/logs/server_name* folder.

To set or change the log and trace file names, use the following procedure.

Procedure

1. In the navigation pane of the administrative console, select **Applications > Enterprise Applications**.
2. In the Enterprise Applications list, click the name of the adapter application. This is the name of the EAR file for the adapter, but without the ear file extension. For example, if the EAR file is named `Accounting_OutboundApp.ear`, then click **Accounting_OutboundApp**.
3. In the Configuration tab, in the Modules list, click **Manage Modules**.
4. In the list of modules, click IBM WebSphere Adapter for Oracle.
5. In the Configuration tab, under Additional Properties, click **Resource Adapter**.
6. In the Configuration tab, under Additional Properties, click **Custom properties**.
7. In the Custom Properties table, change the file names.
 - a. Click either **logFilename** to change the name of the log file or **traceFilename** to change the name of the trace file.
 - b. In the Configuration tab, type the new name in the **Value** field. By default, the log file is called `SystemOut.log` and the trace file is called `trace.log`.
 - c. Click **Apply** or **OK**. Your changes are saved on your local machine.
 - d. To save your changes to the master configuration on the server, use one of the following procedures:
 - **Static change:** Stop and restart the server. This method allows you to make changes, but those changes do not take effect until you stop and start the server.
 - **Dynamic change:** Click the **Save** link in the Messages box above the Custom properties table. Click **Save** again when prompted.

First-failure data capture (FFDC) support

The adapter supports first-failure data capture (FFDC), which provides persistent records of failures and significant software incidents that occur during run time in IBM Business Process Manager or WebSphere Enterprise Service Bus.

The FFDC feature runs in the background and collects events and errors that occur at run time. The feature provides a means for associating failures to one another, allowing software to link the effects of a failure to their causes, and thereby facilitate the quick location of the root cause of a failure. The data that is captured can be used to identify exception processing that occurred during the adapter run time.

When a problem occurs, the adapter writes exception messages and context data to a log file, which is located in the *install_root/profiles/profile/logs/ffdc* directory.

For more information about first-failure data capture (FFDC), see the IBM Business Process Manager or WebSphere Enterprise Service Bus documentation.

Tracing the XML Gateway Web Service status

Use Oracle E-Business Suite administration console to find the XML Gateway Web Service status.

The messages in return Business Object indicate whether the XML Gateway Web Services are invoked successfully or not. If succeeded, it means the submitted document is received by XML Gateway Web Services server side and put into the queue for consequent asynchronous process. In order to check whether the requests are successfully processed by XML Gateway internal implementation, log on to Oracle E-Business Suite administration console and find out the details in the Transaction Monitor.

Resolving namespace conflict in Web Service/JMS interface

When using the Web Service/JMS interface to connect to two different Oracle E-Business Suite servers, if the selected root DTD and the root Element are similar, the generated artifacts will have the same namespace in the top BO, which results in a naming conflict.

To avoid this problem, set a different BO prefix.

Resolving connection information lost during EMD

When using different interfaces of WebSphere Adapter for Oracle E-Business Suite and accessing the EMD several times, the connection information may be lost at the "Specify the Service Generation and Deployment Properties" page. For example, when you are configuring the module for the first time, and choose the XML Gateway JMS processing option, to reach the "Specify the Service Generation and Deployment Properties" page, and then go back to the "Specify the Discovery Properties" page, to choose Advanced Queue for processing, and reach the "Specify the Service Generation and Deployment Properties" page again, you may find that the "Database URL" is empty, when you choose "Specify XA database connection information" as "Database connection information".

To avoid this, choose only one type of interface during the configuration, and not switch between interface options.

Disabling end point applications of the passive adapter

Problem

In the active-passive configuration mode of the adapters, the endpoint application of the passive adapter instance also listens to the events or messages even if the `enableHASupport` property is set to True.

Cause

By default, in WebSphere Application Server, version 7.0, the `alwaysactivateAllMDBs` property in the JMS activation specification is set to True. This enables the endpoint application of all the adapter (active or passive) instances to listen to the events.

Solution

To stop the endpoint application of the passive adapter instance from listening to the events, you must set the `alwaysActivateAllMDBs` property value to `False`. The JMS activation specification is associated with one or more MDBs and provides the necessary configuration to receive events. If the `alwaysActivateAllMDBs` property is set to `False`, then the endpoint application of only the active adapter instance receives the events.

Perform the following procedure, to set the `alwaysActivateAllMDBs` property to `False`.

1. Log on to the administrative console.
2. Go to **Resources > JMS > Activation specifications**.
3. Click the activation specification corresponding to the application from the list.
4. Click **Custom properties** under **Additional properties**.
5. Click `alwaysActivateAllMDBs`.
6. Change the value to `False`.
7. Click **Apply** and **OK**.

Result

The endpoint application of only the active adapter instance listens to the events.

Self-help resources

Use the resources of IBM software support to get the most current support information, obtain technical documentation, download support tools and fixes, and avoid problems with WebSphere Adapters. The self-help resources also help you diagnose problems with the adapter and provide information about how to contact IBM software support.

Support website

The WebSphere Adapters software support website at http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family provides links to many resources to help you learn about, use, and troubleshoot WebSphere Adapters, including:

- Flashes (alerts about the product)
- Technical information including the product information center, manuals, IBM Redbooks[®], and whitepapers
- Educational offerings
- Technotes

Recommended fixes

A list of recommended fixes you must apply is available at the following location: <http://www.ibm.com/support/docview.wss?fdoc=aimadp&rs=695&uid=swg27010397>.

Technotes

Technotes provide the most current documentation about WebSphere Adapter for Oracle E-Business Suite, including the following topics:

- Problems and their currently available solutions
- Answers to frequently asked questions

- How to information about installing, configuring, using, and troubleshooting the adapter
- *IBM Software Support Handbook*

For a list of technotes for WebSphere Adapters, visit this address:

<http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>.

Plug-in for IBM Support Assistant

WebSphere Adapter for Oracle E-Business Suite provides a plug-in for IBM Support Assistant, which is a free, local software serviceability workbench. The plug-in supports the dynamic trace feature. For information about installing or using IBM Support Assistant, visit this address:

<http://www.ibm.com/software/support/isa/>.

Solutions to common problems

Some of the problems you encounter while running WebSphere Adapter for Oracle E-Business Suite with your database are described, along with solutions and workarounds. These problems and solutions are in addition to those documented as technotes on the software support Web site.

For a complete list of technotes about WebSphere Adapters, see <http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>.

RecordNotFoundException for RetrieveAll operation in test client

Problem

When performing a RetrieveAll operation in the IBM Integration Designer test client, the RecordNotFoundException exception is generated when data is expected from the query. The following message is generated: RecordNotFoundException: Record not found in EIS.

Cause

This exception can occur if the WHERE clause for the SELECT statement does not set all the attributes of the business object. Leaving the attribute blank, which is the default value, is not the same as explicitly unsetting the value.

Solution

In the test client, set the values of the attributes that are required to <unset>. Repeat the RetrieveAll operation. If the exception is generated again, it is likely that no matching records exist in the database table.

CLOB datatypes of 4K or larger cannot be inserted into Oracle 9i or 10g databases

Problem

You get the following exception when inserting CLOB (character large object) values of 4 K and larger into Oracle 9i or 10g databases:

- Oracle 9i: ResourceAdapt E
com.ibm.j2ca.dbadapter.core.runtime.DBOperationHandler.
executePreparedCUDStatement CWYDB0301E: An operation on the database failed with a SQL exception for the following reason: No more data to read from socket.
- Oracle 10g: ResourceAdapt E
com.ibm.j2ca.dbadapter.core.runtime.DBOperationHandler.
executePreparedCUDStatement CWYDB0301E: An operation on the database failed with a SQL exception for the following reason: ORA-01460: unimplemented or unreasonable conversion requested

Cause

You are using an older driver that does not correctly support CLOBs larger than 4K.

Solution

Use the Oracle Thin driver from Oracle 10.1.0.2 or a later release.

Some generated business objects have no attributes for Oracle database objects

Problem

For some business objects that are generated from an Oracle database object, the generated business object has no attributes for the table columns.

Cause

Under certain conditions, the Oracle JDBC driver does not return the column information for a database object. The following bugs are currently filed with Oracle for these issues:

- 2281705. DATABASEMETADA.GETCOLUMNS does not return underlying table if there is a synonym
- 2696213. JDBC GETPROCEDURECOLUMNS does not return columns for the synonym of a procedure

Additionally, column information is not returned if a private synonym that references an object in another schema is used.

Solution

For tables that have a synonym, generate the business object using the synonym for the table.

For synonyms of a procedure, generate the business object using the original procedure that the synonym is based on.

For private synonyms that reference an object in another schema, either use the original table or create a synonym in the current schema.

ResourceException exceptions during outbound processing

If you get a ResourceException exception, examine the root cause field to determine the cause. Common problems have the following root causes:

- SQLException exception

If the SQLException exception includes the text UserID or password is invalid, then the user ID or password specified for the outbound connection is not correct.

For example:

```
javax.resource.ResourceException: ORA-01017; invalid username/password;  
logon denied.
```

- ConnectException exception

If the text included with the ConnectException exception includes text like is not reachable or could not establish the connection, then the database server might not be operational or there might be a network problem that prevents a connection.

For example:

```
java.sql.SQLException: Io exception: The Network Adapter could not  
establish the connection
```

ResourceException exception during inbound processing

This exception indicates that there is a repeated problem connecting to the database. To polls for events, the adapter must connect top the database. If the connection fails, the adapter waits a configurable amount of time before trying to connect again. The adapter tries a configurable number of times before it stops polling. When the adapter stops polling, it generates the ResourceException exception.

Class loader violation occurs when starting the external service wizard

Problem

It is not possible to use the external service wizard after using a connection to the database in the Data perspective. At the end of the second panel of the wizard, the following exception is generated:

```
com.ibm.adapter.framework.api.ImportException  
Reason: class loading constraint violated (class:  
oracle/jdbc/driver/OracleConnection  
method: getWrapper()Loracle/jdbc/OracleConnection;) at pc:0
```

The error occurs in both of the following situations:

- When you establish a connection to the database through the external service wizard, an error occurs when you attempt to connect to the database from the Data perspective.
- When you establish a connection to the database through the Data perspective, the error occurs when you attempt to connect to database through the external service wizard.

Cause

The error occurs because the Data perspective and the wizard use their own class loaders. Once the DLL, which is the native library used by the JDBC driver, is loaded in the Data perspective, it cannot be loaded again in the wizard. JVMs have an inherent restriction that only allows one class loader to load native libraries at any given time. So if class loader A loads DLL B, then no other class loader can load DLL B until class loader A is released and garbage is collected. Because you cannot really control garbage collection, it typically means that if you want to load DLL B with another class loader, you need to restart the JVM. This limitation is a known one and is documented for IBM Business Process Manager or WebSphere Enterprise Service Bus.

Solution

The only solution is to restart IBM Integration Designer when this error occurs.

Closed connection error occurs when using XA with Oracle 10g

Problem

When the Adapter for Oracle E-Business Suite is used to perform an XA transaction using Oracle 10g, the adapter generates a closed connection exception: `javax.resource.ResourceException: Closed Connection`.

Cause

This is a known issue with the Oracle 10g database driver. The following bug has been filed with Oracle for this issue: 3488761 Connection closed error from `OracleConnection.getConnection()` - 10G drivers.

Solution

The bug has been fixed in the Oracle 10g Release 2 driver. As a workaround, you can use the Oracle 9i JDBC Thin drivers to connect to the database for XA transactions.

Error occurs while starting a transaction on Oracle

Problem

When the Adapter for Oracle E-Business Suite is used to perform an XA transaction using Oracle database, the following error is generated: `WTRN0078E: An attempt by the transaction manager to call start on a transactional resource has resulted in an error. The error code was XAER_RMERR`.

Cause

For the Oracle database server to support XA transactions, some commands need to be run.

Solution

Two scripts that are included in the Oracle directory should be run. This activity mostly needs to be performed by your Oracle database administrator, because you must be logged on to Oracle as `SYSOPER` or `SYSDBA` in order to have the necessary permissions for these scripts to work. The scripts are:


```
<ORACLE_HOME>javavm\install
file: initxa.sql
file: initjvm.sql
```

The `initxa.sql` script configures the database for XA. Once it runs successfully, your database is configured for XA. The script might run successfully the first time you try. Unfortunately, it probably will not run successfully because some of the database memory spaces are too small.

To fix this, run the `initjvm.sql` script. It probably fails too, but in doing so, it indicates which parameters need to be adjusted. The parameters are stored in this file:

```
<ORACLE_HOME>\database
file: init<DATABASE_SID>.ora
```

Table 16 shows two parameters that typically need to be increased. Your particular database configuration might require adjustment of different parameters.

Table 16. Typical parameter sizes

Parameter Name	Minimum Value
<code>java_pool_size</code>	12000000
<code>shared_pool_size</code>	24000000

Outbound fails due to an exception in global transactions

Problem

Outbound fails due to an exception in the global transaction and the global transaction cannot be rolled back successfully.

Cause

JCA WebSphere Adapter for Oracle E-Business Suite version 6.2 and version 7.5 supports business objects for exceptions `IntegrityConstraintViolationException`, `MatchesExceededLimitException`, `MissingDataException`, `MultipleMatchingRecordsException`, `ObjectNotFoundException`, `RecordNotFoundException`, and `UniqueConstraintViolatedException`. When any of these exceptions are thrown, the EIS binding encapsulates the exceptions as corresponding fault business objects like `IntegrityConstraintFault`, `MatchesExceededLimitFault`, `MissingDataFault`, `MultipleMatchingRecordsFault`, `ObjectNotFoundFault`, `RecordNotFoundFault`, and `UniqueConstraintFault`.

As soon as the EIS binding receives the fault business object, it throws a `ServiceBusinessException` instead of a `ServiceRuntimeException`. In global transactions, if a component throws a `ServiceRuntimeException`, the transaction manager will roll back the global transaction. However, if a component throws a `ServiceBusinessException`, the transaction manager commits global transaction.

Solution

1. Delete all fault business object binding in the `.import` file.

```

<methodBinding inDataBindingType=
"com.ibm.xmlns.prod.websphere.j2ca.jdbc.db2admincustomerbg.Db2adminCustomerBGDataBin
ding"
    method="createDb2adminCustomerBG"
outDataBindingType="com.ibm.xmlns.prod.websphere.j2ca.jdbc.db2admincustomerbg.Db2admin
CustomerBGDataBinding">
    <faultBinding fault="INTEGRITY_CONSTRAINT_VIOLATION"
faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
    <faultBinding fault="MISSING_DATA"
faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
    <faultBinding fault="OBJECT_NOTFOUND_EXCEPTION"
faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
    <faultBinding fault="UNIQUECONSTRAINT_VIOLATION"
faultBindingType="com.ibm.j2ca.extension.emd.runtime.WBIFaultDataBindingImpl"/>
    <interaction>
        <properties>
            <functionName>Create</functionName>
        </properties>
    </interaction>
</methodBinding>

```

2. Define fault handlers to deal with business faults as shown below.

```

DataObject bo = null;
try {
    bo = (DataObject) locateService_JDBCOutboundInterfacePartner().invoke("createDb2adminAddressBG
} catch (ServiceBusinessException e) {
    // TODO Auto-generated catch block
    throw new ServiceRuntimeException(e);
}

```

Whenever WebSphere Adapter for Oracle E-Business Suite outbound process has any exception, a ServiceRuntimeException is generated and the global transaction is rolled back. Otherwise, the global transaction is committed.

A closer look at the transaction (XID) column in the event table

If the adapter is configured for assured once delivery, use the status column with the XID column to determine whether the event has been processed:

- If the XID column contains 0, the event has not yet been picked up for processing.
- If the XID column contains a transaction ID (that is, it does not contain 0), then the adapter has started to process the event but has not completed processing. You might see this combination when the adapter or application server stops while the event is being processed. The transaction manager will either COMMIT or ROLLBACK these transactions during recovery.

Handling unexpected results from a query SQL statement

Should you receive unexpected results from a query, turn on tracing and look at the query SQL in the log. Turning on tracing is especially helpful when you are in the test client to see if you forgot to *unset* all the unnecessary attributes. It is also practical to turn on tracing to determine if you did not specify your business object correctly.

Adapter returns version conflict exception message

Problem

When you install multiple adapters with different versions of CWYBS_AdapterFoundation.jar, and if a lower version of the CWYBS_AdapterFoundation.jar is loaded during runtime, the adapter will return

the ResourceAdapterInternalException error message, due to a version conflict. For example, when you install Oracle E-Business Suite adapter version 7.0.0.3 and WebSphere Adapter for Oracle E-Business Suite version 7.5, the following error message is displayed: IBM WebSphere Adapter for Oracle E-Business Suite has loaded file: /C:/IBM/WebSphere/ProcServer7/profiles/ProcSrv01/installedConnectors/CWYOE_OracleEBS.rar/CWYBS_AdapterFoundation.jar with version 7.0.0.3. However, the base level of this jar required is version 7.5. When you install multiple adapters with different CWYBS_AdapterFoundation.jar versions, the adapter returns the ResourceAdapterInternalException message, due to a version conflict. To overcome this conflict, you must migrate all adapters to the same version level. For further assistance, contact WebSphere Adapters Support for help.

Solution

Migrate all adapters to the same version level.

For further assistance, visit http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family.

The Add and Remove buttons in the EMD wizard do not function properly

Problem

When the EMD is run with the same Oracle E-Business Suite RAR file consecutively, and the imported Oracle E-Business Suite driver libraries are selected to be removed, all the libraries, even those not selected, are removed. For example, the Oracle driver libraries consists of a driver JAR file and a license JAR file. During the second run, if you try to remove one of the JAR files, both the JAR files are removed.

Cause

This is caused because of an issue with IBM Integration Designer.

Error displayed when using external service discovery wizard

Problem

Error is displayed when using the external service discovery wizard.

1. Add a Oracle E-Business Suite driver file in the Locate the Required Files and Libraries page.
2. Click **Next** and move to the Specify the Discovery Properties panel.
3. Click **Back** and go to the Locate the Required Files and Libraries page. Then add another Oracle E-Business Suite driver file.
4. Click **Next** and move to the Specify the Discovery Properties panel.

IBM WebSphere Adapter for Oracle E-Business Suite returns the following error: 'Failure in connection to EIS. Failed to load the driver: oracle.jdbc.driver.OracleDriver. Check that the Oracle E-Business Suite driver classes are in the class path'.

Cause

This is an Eclipse limitation for class loading.

Solution

Close the connector project and reopen it to add the Oracle E-Business Suite driver files again.

Chapter 10. Reference

Detailed information about business objects, adapter properties (enterprise service discovery properties, resource adapter properties, managed (J2C) connection factory properties, activation specification properties, and interaction specification properties), messages, and related product information is provided for your reference.

Business object information

A business object is a structure that contains application-specific information (metadata) about how the adapter must process the business object as well as the operation to be performed on the business object. The name of the business object is generated by the external service wizard in accordance with the naming convention for the adapter.

Business object attributes

Business object attributes define the content of a business object and are built from the list of columns in the database object.

A business object is simply a container for the data specified in the attributes. Each attribute has a name, type, cardinality, and several other properties. The external service wizard sets the attribute name to the name of the column. The adapter adds the attribute cardinality, type, and application-specific information. The structure of the data in the database is defined by the business object, but data in the database is in the business object attributes.

Table 17 lists the properties of a business object attribute and describes their interpretation and settings.

Table 17. Attribute properties

Properties	Interpretation and settings
Cardinality	<p>An integer specifying the cardinality of a business object. Each business object attribute that represents a child or an array of child business objects has the value of single or multiple (an unbounded integer) cardinality.</p> <p>In both single- and multiple-cardinality relationships, the relationship between the parent and child business objects is described by the application-specific information of the key attribute in the business object storing the relationship.</p>
Foreign Key	<p>When arrays of child business objects whose cardinality is n are retrieved, foreign keys are used in the WHERE clause of SELECT statements.</p> <p>The RetrieveAll operation overrides the use of keys and foreign keys.</p> <p>Note: The adapter does not support specifying an attribute that represents a child business object as a foreign key.</p>
Name	<p>This property represents the unique name of the attribute, if it is a simple attribute, or the name of the business object, if it is a child business object.</p>

Table 17. Attribute properties (continued)

Properties	Interpretation and settings
MinOccurs MaxOccurs	If the column is not a primary key and is not null able, the MinOccurs and MaxOccurs attributes are required, and their values are set to at least 1.
Primary Key	<p>Indicates whether this attribute is a primary key. At least one simple attribute in each business object must be specified as the primary key.</p> <p>If the primary key property is set to true for a simple attribute, the adapter adds that attribute to the WHERE clause of the SELECT statement and SQL UPDATE statements that it generates while processing the business object. The RetrieveAll operation overrides the use of primary and foreign keys.</p> <p>Note: The adapter does not support specifying an attribute that represents a child business object or an array of child business objects as a primary key attribute.</p>
Required	Specifies whether an attribute must contain a value. If this property is set to true for a container whose cardinality is single (1), then the adapter requires that the parent business object contain a child business object for this attribute. Business objects that are passed to the adapter for Create, Update, and Delete operations must also contain a child business object. Cardinality is single (1) for simple attributes and multiple (n) for container attributes. The adapter causes a Create operation to fail if a business object does not have a valid value or a default value for a required attribute. It also fails if no data is available upon retrieval from the database for this object.
Type	<p>For simple attributes, this property specifies the type of the attribute, such as Integer, String, Date, Timestamp, Boolean, Double, or Float. The supported types for simple attributes and their mapping to the Oracle type of a database object are described in Table 18 on page 209.</p> <p>For attributes that specify a child business object, this property specifies the name of the business object.</p>

The type of each database object, returned as the Oracle metadata, maps to the business object attribute types as listed in Table 18 on page 209. Only the Oracle types listed are supported by the adapter. Any columns with types that are not listed are not added to the business object. An informational message is produced explaining the problem, for example, The column named xxxx in the table named yyyy is not of a supported type and is not added to the business object.

Note: The default data type mapping varies based on the different Oracle JDBC driver versions. If the Oracle metadata does not map to the same data type during the configuration of the Oracle database objects, select the appropriate data type manually in the Specify the Configuration Properties for 'object' window. After the generation of the business object, if you find the Oracle metadata not mapped to the same business object attribute type, update the attribute data type manually in the XSD file for the business object.

Table 18. Oracle metadata column type and business object attribute types

Oracle metadata column type	Business object attribute type
CHAR LONG VARCHAR2	String
NUMBER	Decimal
TIMESTAMP	DateTime (String data type is displayed by default)
DATE	Date (String data type is displayed by default)
FLOAT	Double
BLOB	hexBinary
CLOB	String
NCHAR NVARCHAR2	String
RAW LONG RAW	hexBinary
STRUCT or ARRAY	The adapter processes these data types as child business objects of the table or query business objects. Note: The adapter supports complex types for the Oracle table and query business objects only. If the table contains any complex data type, such as an array, structure, nested structure or table, the type name and the sub attribute details are also automatically discovered and displayed. Note: The adapter treats an empty complex column as null irrespective of setting it to null or unset.
BOOLEAN	The adapter supports the boolean datatype for SP/SF with Record type parameter.

Related concepts

“Business object hierarchies” on page 33

Define the relationships between database tables using parent-child relationships and data ownership in hierarchical business objects.

Attribute application-specific information

Application-specific information (ASI) for business object attributes differs depending on whether the attribute is a simple attribute, or is an attribute that represents a child or an array of child business objects. The application-specific information for an attribute that represents a child differs depending on whether the parent-child relationship is stored in the child or in the parent.

Application-specific information for simple attributes

For simple attributes, the format for application-specific information consists of a number of parameters and their values. The only parameter that is required for a simple attribute is the column name. The application-specific information for simple attributes is described in Table 19 on page 210.

Table 19. Application-specific information for simple attributes

Parameter	Type	Description	Default value
BLOB	Boolean	Indicates whether the database column that corresponds to this attribute has the BLOB data type. While displaying BLOB data, the adapter displays the number of bytes as a hexadecimal value. The attribute type is hexBinary. If True, the column data type is BLOB.	None
ByteArray	Boolean	Specifies whether the column is a binary data type. If True, the adapter reads and writes binary data to the database and sends that data as a string to the application server. The adapter sets binary data on the business object. The attribute type is hexBinary.	False
ChildBOType	String	If the attribute is a complex data type, use this application-specific information to specify the actual type: <ul style="list-style-type: none"> • Struct • Array • ResultSet 	None
ChildBOTypeName	String	When the value of the ChildBOType application-specific information is either Struct or Array, this parameter value represents the name of the user-defined type. This value is case sensitive.	
CLOB	Boolean	Indicates whether the database column that corresponds to this attribute has the CLOB data type. This value applies only to attributes of type String. If True, the column data type is CLOB. The CLOB attribute has a String Type whose length is used to define the length of the CLOB.	None
ColumnName	String	The name of the database column corresponding to this attribute. This is the only required parameter.	None

Table 19. Application-specific information for simple attributes (continued)

Parameter	Type	Description	Default value
CopyAttribute	String	<p>A user-specified value that refers to another attribute name from within the same business object or parent business object.</p> <p>If the value set in the application-specific information refers to the name of another attribute within the same business object, then the adapter uses the value of the other attribute to set the value of this attribute (on which application-specific information is defined) before it adds the business object to the database during a Create operation.</p> <p>For example, if you want the contact column of a new row in the table to contain the same value as the email column, set the CopyAttribute parameter of the contact attribute to email.</p> <p>The value cannot reference an attribute in a child business object, but it can reference an attribute in the parent business object by preceding the name with two periods. For example, you can reference the ccode attribute in a parent business object as ..ccode.</p> <p>If you do not include this parameter in the application-specific information, the adapter uses the value of the current attribute without copying the value from another attribute.</p>	None
DateType	String	<p>Specifies that the corresponding element is a date or time stamp type. Specify one of the following values:</p> <ul style="list-style-type: none"> • Date • Timestamp <p>When setting the value of an attribute of the DateType type, use the following formats:</p> <ul style="list-style-type: none"> • For Date, use yyyy-MM-dd • For Timestamp, use yyyy-MM-dd hh:mm:ss.fffffff <p>Note: The adapter uses the Timestamp value present in the database. To learn more about the Timestamp method, go to the Sun Web site at http://java.sun.com/j2se/1.5.0/docs/api/ and search for <i>Timestamp</i>.</p>	None

Table 19. Application-specific information for simple attributes (continued)

Parameter	Type	Description	Default value
FixedChar	Boolean	<p>Specifies whether the attribute is of fixed length when the columns in the table are of type CHAR, not VARCHAR. For example, when set to true, if a particular attribute is linked to a column that is of type CHAR, the adapter pads the attribute value with blanks to the maximum length of the attribute when querying the database.</p> <p>This parameter must be updated manually in the XSD file for the business object. Open the business object by using an XML or text editor to edit the XSD file. Two changes must be made, as follows:</p> <ol style="list-style-type: none"> 1. Remove the <code>type="string"</code> added by default to the <code><element></code> tag for the object attribute. 2. Add a new <code><simpletype></code> section before the <code></element></code> tag, as in this example: <pre><xsd:simpleType> <xsd:restriction base="xsd:string"> <xsd:maxLength value="10"/> </xsd:restriction> </xsd:simpleType></pre> <p>Save the object definition, and ensure that no validation errors occur in the XSD file after it has been updated.</p> <p>Following this table, see the Example of FixedChar parameter in the business object XSD file.</p>	false
ForeignKey	String	<p>The value of this property depends on whether the parent/child relationship is stored in the parent business object or in the child.</p> <p>If the relationship is stored in the parent, the value includes both the type of the child business object and the name of the attribute in the child to be used as the foreign key (<i>Child_BO_name/Child_Property_Name</i>).</p> <p>If the relationship is stored in the child, set the value to include only the name of the attribute in the parent to be used as the foreign key.</p> <p>If an attribute is not a foreign key, do not include this parameter in the application-specific information.</p>	None
OrderBy	String	<p>If a value is specified and the attribute is in a child business object, the adapter uses the value of the attribute in the ORDER BY clause of retrieval queries.</p> <p>The adapter can retrieve child business objects in either ascending order (ASC) or descending order (DESC). If you do not include this parameter in the application-specific information, the adapter does not specify the retrieval order.</p>	None
PrimaryKey	Boolean	<p>If the column associated with this attribute is a primary key in the corresponding table in the database, PrimaryKey is to True,</p>	None

Table 19. Application-specific information for simple attributes (continued)

Parameter	Type	Description	Default value
SPParameterType	String	Specifies the type of stored procedure Possible values are: <ul style="list-style-type: none"> • IP (input only) • OP (output only) • IO (input and output) • RS (result set) 	None
UniqueIdentifier (UID)	String	The adapter uses this parameter to generate the unique ID for the business object. It supports the generation of sequences and identity columns. The format of this parameter is as follows: UID=AUTO Sequence_Name For a sequence, set the UID attribute to the name of the sequence. For an identity column, set the UID attribute to AUTO. If the attribute does not require a unique ID, do not include this parameter in the application-specific information.	None
DateFormat	String	Allows you to customize the format of the Date and Timestamp data types in the Application Info section of the Properties view in IBM Integration Designer.	None

The format of attribute application-specific information is shown in the following example section of an XSD file:

Example section of an XSD file

```
<element name="pkey" nillable="true"
  minOccurs="0" maxOccurs="1">
  <annotation xml:space="preserve">
    <appinfo
      source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/
metadata">
      <oraclesi:OracleAttributeTypeMetadata
        xmlns:oraclesi="http://www.ibm.com/xmlns/prod/websphere/j2ca/
oracle/metadata">
        <oraclesi:ColumnName>PKEY</oraclesi:ColumnName>
        <oraclesi:PrimaryKey>true</oraclesi:PrimaryKey>
        <oraclesi:FixedChar>true</oraclesi:FixedChar>
        </oraclesi:OracleAttributeTypeMetadata>
      </appinfo>
    </annotation>
    <simpleType>
      <restriction base="string">
        <maxLength value="10"/>
      </restriction>
    </simpleType>
  </element>
  <element name="ccode" type="string" nillable="true"
    minOccurs="0" maxOccurs="1">
```

```

                <annotation xml:space="preserve">
<appinfo
source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
<oracleasi:OracleAttributeTypeMetadata
xmlns:oracleasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/
metadata">
<oracleasi:ColumnName>CCODE</oracleasi:ColumnName>
<oracleasi:PrimaryKey>false</oracleasi:PrimaryKey>
<oracleasi:ForeignKey>custinfoobj/ccode</oracleasi:ForeignKey>
</oracleasi:OracleAttributeTypeMetadata>
</appinfo>
</annotation>
        </element>
        <element name="fname" type="string" nillable="true"
minOccurs="0" maxOccurs="1">
                <annotation xml:space="preserve">
<appinfo
source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
<oracleasi:OracleAttributeTypeMetadata
xmlns:oracleasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/
metadata">
<oracleasi:ColumnName>FNAME</oracleasi:ColumnName>
<oracleasi:PrimaryKey>false</oracleasi:PrimaryKey>
</oracleasi:OracleAttributeTypeMetadata>
</appinfo>
</annotation>
        </element>
        <element name="lname" type="string" nillable="true"
minOccurs="0" maxOccurs="1">
                <annotation xml:space="preserve">
<appinfo
source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
<oracleasi:OracleAttributeTypeMetadata
xmlns:oracleasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/
metadata">
<oracleasi:ColumnName>LNAME</oracleasi:ColumnName>
<oracleasi:PrimaryKey>false</oracleasi:PrimaryKey>
</oracleasi:OracleAttributeTypeMetadata>
</appinfo>
</annotation>
        </element>
        <element name="custinfoobj"
type="rtassercustinfo:RtasserCustinfo"
nillable="true" minOccurs="0"
maxOccurs="1">
                <annotation xml:space="preserve">
<appinfo
source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
<oracleasi:OracleAttributeTypeMetadata
xmlns:oracleasi="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/
metadata">
<oracleasi:Ownership>true</oracleasi:Ownership>
<oracleasi:KeepRelationship>false</oracleasi:KeepRelationship>
</oracleasi:OracleAttributeTypeMetadata>
</appinfo>
</annotation>
        </element>

```

Example of FixedChar parameter in the business object XSD file

```

<element name="pkey" nillable="true"
minOccurs="0" maxOccurs="1">
                <annotation xml:space="preserve">
<appinfo
source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/
metadata">

```

```

<oraclesi:OracleAttributeTypeMetadata
  xmlns:oraclesi="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
  <oraclesi:ColumnName>PKEY</oraclesi:ColumnName>
  <oraclesi:PrimaryKey>true</oraclesi:PrimaryKey>
  <oraclesi:FixedChar>true</oraclesi:FixedChar>
  </oraclesi:OracleAttributeTypeMetadata>
</appinfo>
</annotation>
<simpleType>
  <restriction base="string">
    <maxLength value="10"/>
  </restriction>
</simpleType>
</element>

```

Application-specific information for attributes of type child business object

Two application-specific information parameters are used for attributes that refer to child business objects (complex, as opposed to simple, attributes). When you set this application-specific information, specify the parameters listed in Table 20.

Table 20. Application-specific information for attributes of type child business object

Parameter	Type	Description	Default value
KeepRelationship	Boolean	If True, this parameter prevents the deletion of a child business object during an Update operation.	None
Ownership	Boolean	This parameter specifies that a child business object is owned by the parent. If True, then Create, Update, and Delete operations on the child business object are allowed. If False, then no updates can be applied to the child business object. When its parent is created, the existence of the child is validated to ensure that relationship integrity is maintained in the database.	None

Example of ownership in the business object XSD file

```

<element name="addressobj"
  type="rtasseraddress:RtasserAddress"
  nillable="true"
  minOccurs="0"
  maxOccurs="unbounded">
  <annotation xml:space="preserve">
    <appinfo
      source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
      <oraclesi:OracleAttributeTypeMetadata
        xmlns:oraclesi="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
        <oraclesi:Ownership>true</oraclesi:Ownership>
      </oraclesi:OracleAttributeTypeMetadata>
    </appinfo>
  </annotation>
</element>

<element name="custinfoobj"
  type="rtassercustinfo:RtasserCustinfo"
  nillable="true"
  minOccurs="0"

```

```

maxOccurs="1">
  <annotation xml:space="preserve">
<appinfo
source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
<oraclesi:OracleAttributeTypeMetadata
xmlns:oraclesi="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/
metadata">
<oraclesi:Ownership>false</oraclesi:Ownership>
</oraclesi:OracleAttributeTypeMetadata>
</appinfo>
</annotation>
</element>

```

An example of the XSD definition file for single- and multiple-cardinality child business objects is provided here. The element `custInfoObj` is a single-cardinality child business object, and `addressObj` is a multiple-cardinality child business object.

Another example XSD file for single- and multiple-cardinality child business objects

```

<element name="addressobj"
          type="rtasseraddress:RtasserAddress"
nillable="true"
          minOccurs="0"
          maxOccurs="unbounded">
  <annotation xml:space="preserve">
<appinfo
source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
<oraclesi:OracleAttributeTypeMetadata
xmlns:oraclesi="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/
metadata">
<oraclesi:Ownership>true</oraclesi:Ownership>
</oraclesi:OracleAttributeTypeMetadata>
</appinfo>
</annotation>
</element>

<element name="custinfoobj"
          type="rtassercustinfo:RtasserCustinfo"
nillable="true"
          minOccurs="0"
          maxOccurs="1">
  <annotation xml:space="preserve">
<appinfo
source="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/metadata">
<oraclesi:OracleAttributeTypeMetadata
xmlns:oraclesi="http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/
metadata">
<oraclesi:Ownership>false</oraclesi:Ownership>
</oraclesi:OracleAttributeTypeMetadata>
</appinfo>
</annotation>
</element>

```

Application-specific information for operations

The adapter uses application-specific information at the operation level to perform operations, such as to retrieve and update information in the database. The adapter retrieves and updates database tables using SQL queries, stored procedures, or stored functions, as specified in the business objects.

If you choose to add stored procedures or stored functions to the business objects, set the operation application-specific information (ASI) as specified in Table 21 on page 217.

Table 21. Operation application-specific information

Operation ASI for StoredProcedure parameters element	Set by wizard	Description
Parameters	Yes	Lists the stored procedure parameters.
PropertyName	Yes	Set to the name of the business object attribute that you select.
ResultSet	No	If the stored procedure returns a result set, set this parameter to True in the business object definition.
ReturnValue	Yes	If the stored procedure has a return value, this parameter contains one of these values: <ul style="list-style-type: none"> • The string RS. This value indicates that the procedure returns a result set, which is used to create the multiple-cardinality container corresponding to this business object. • The name of the business object attribute. This value indicates that procedure returns the value that is to be assigned to that particular attribute in the business object at run time. If the attribute is another child business object, the adapter returns an error.
StoredProcedure	Yes	Set to the stored procedure name.
StoredProcedureType	Yes	You choose from a list of types. For information about valid stored procedure types, see “Stored procedure type” on page 42
Type	Yes	Set to the type of the stored procedure parameter. Possible values are: <ul style="list-style-type: none"> • IP (input only) • OP (output only) • IO (input and output) • RS (result set)

Business object-level application-specific information

Application-specific information in business object definitions provides the adapter with application-dependent instructions for how to process business objects. The adapter parses the application-specific information from the business object or from its attributes or operations to generate queries for Create, Update, Retrieve, and Delete operations.

Application-specific information for table and view business objects

Application-specific information at the business-object level is used to specify the name of the corresponding database table and to provide information necessary to perform a physical or logical Delete operation.

The external service wizard sets the `TableName` application-specific information attribute to a value in the form of `SchemaName.TableName`. It prompts you for the information necessary to perform a physical or logical Delete operation and then sets the business object-level application-specific information shown in Table 22 on page 218.

Table 22. Business object application-specific information (ASI) for a table business object

Application-specific information	Type	Description
TableName	String	The name of the database table corresponding to this business object.
StatusColumnName	String	Indicates whether the adapter is to logically or physically delete data in the table. If the StatusColumnName parameter is not set, data is physically deleted. If the parameter is set, it specifies the name of the column that indicates a logically deleted row. You specify this parameter when you select the table object in the external service wizard. This parameter applies to both Update and Delete operations.
StatusValue	String	The value that indicates that a column is logically deleted. You specify this value when you select the table object in the external service wizard.

To illustrate how the adapter determines whether to do a logical or physical delete in response to an Update or Delete operation, assume that a Customer business object has the business object application-specific information shown in Table 23.

Table 23. Sample parameters for business object application-specific information for a table business object

Application-specific information	Value
TableName	customer
StatusColumnName	status
StatusValue	deleted

Assume that the adapter receives a request to delete a customer. Because the business object includes the StatusColumnName parameter in its application-specific information, the adapter performs a logical delete operation. It does this by placing the string “deleted”, which is specified in the StatusValue parameter, in the status column, which is the column specified in the StatusColumnName parameter.

Such a request causes the adapter to issue the following SQL statement:

```
UPDATE customer set status = 'deleted' where pkey = . . . .
```

However, if the StatusColumnName parameter is not set, the customer records are physically deleted. The adapter issues the following SQL statement:

```
DELETE from customer where pkey = . . . .
```

Application-specific information for stored procedure business objects

For business objects that are based on stored procedures, the external service wizard sets the business object-level application-specific information SPName to a value in the form of *SchemaName* + *SPName*. It sets the business object-level application-specific information, which is listed in Table 24 on page 219. The attributes of the business object are created based on the stored procedure input/output parameters. If the stored procedure has one returned value, a

corresponding business object attribute is created. If the returned value or any of the input/output parameters are complex data types, the wizard creates child business objects for them.

The discovery of database objects in the external service wizard can support nested structures and arrays. If these child business objects are generated from returned result sets, their names are in the form of *Prefix + SchemaName + SPName + RetRS + Number*. For example, if one stored procedure returns two result sets, the wizard creates two child business objects for them. Their names are *Prefix + SchemaName + SPName + RetRS1* and *Prefix + SchemaName + SPName + RetRS2*.

When the child business objects are generated from input/output parameters with a complex data type of *ResultSet*, *Struct*, or *Array*, these child business object names are in the form of *Prefix+SchemaName+SPName+ParameterName*. For those child business objects that correspond to nested structures and arrays, their business object names are in the form of *Prefix+SchemaName+SPName+ParameterName+ColumnName*.

Table 24. Business object application-specific information (ASI) for business objects based on stored procedures

Application-specific information	Type	Description
SPName	String	The name of the stored procedure or stored function
ResultSet	Boolean	Indicates whether the stored procedure or stored function returns a result set. If true, the stored procedure returns one or more result sets. If false, the stored procedure or stored function does not return a result set.
MaxNumberOfRetRS	String	The maximum number of returned result sets that are handled by the adapter run time
ReturnValue	String	Set to the name of the corresponding business object attribute if the stored procedure has a return value. If the returned value is of simple data type, the attribute is also of simple data type. If the returned value is a result set, this attribute points to a child business object.

Application-specific information for query business objects

For query business objects, there is one business object-level application-specific information, as shown in Table 25.

Table 25. Business object application-specific information (ASI) for query business objects

Application-specific information	Type	Description
SelectStatement	String	The complete SELECT statement that performs the query. You specify the statement in the external service wizard.

The external service wizard also generates business graphs for all business objects, because all are top level. The name of the business graph is the business object name followed by "BG." For example, a business object with the name *OracleSchema1Customer*, would have the business graph named *OracleSchema1CustomerBG*. The operations set in the business object are also set in the business graph.

When the wizard generates a stored procedure business object, it creates a child business object if necessary, such as for ResultSet, Struct, and Array. Creating parent-child relationships between table business objects is done manually using the Business Object Editor.

The wizard handles business objects based on synonym/nicknames like objects based on tables and views, even when a synonym is of a stored procedure.

Naming conventions

When the external service wizard generates a business object, it provides a name for the business object that reflects the naming convention for the adapter. Typically, the business object name indicates the structure of the business object.

When the external service wizard creates names for a business object, it replaces any special character except the underscore (_) in the business object name with U followed by its Unicode number. For example, the business object name for the Order_Item table in the database is Order_Item. The business object name for the Shipping-Address table is ShippingU45Address.

Business object names have no semantic value to the adapter or the database; that is, they derive no information nor meaning from the business object name. If one name is replaced by another, the adapter behavior remains the same.

Business object names can carry database-specific metadata. A name can use a string like Oracle or %AppName% as a prefix to help distinguish between two types of business objects: application-specific and generic. The remainder of the name can describe the table or stored procedure that the business object represents. For example, if the business object definition is generated for the Employee Table in a database application, such as Human Resources (HR), then the respective business object name is HREmployee.

For business objects that do not correspond to database objects, such as business objects for database queries, if you give the business object the same name as a table or stored procedure business object, a different number is appended at the end of each name to differentiate them and avoid overwriting.

Globalized characters are supported in any business object name.

You can rename business objects by using the refactoring functionality in IBM Integration Designer. For more details, refer to the IBM Integration Designer documentation.

The following table describes the naming conventions that the wizard uses for the business object.

Table 26. Business object naming conventions

Element	Naming convention
Business graph	The business graph that contains the parent business object is named for the contained business object, followed by the string "BG"; for example, the business graph that contains the SalesCustomer business object is named SalesCustomerBG.

Table 26. Business object naming conventions (continued)

Element	Naming convention
Business objects for: <ul style="list-style-type: none"> • Tables • Views • Stored procedures • Stored functions • Synonyms and nicknames 	For those business objects that are based on tables, views, stored procedures, and synonyms and nicknames, the external service wizard generates the name of the business object in the form of <i>Prefix + SchemaName + ObjectName</i> , where: <ul style="list-style-type: none"> • <i>Prefix</i> is the value as specified in the external service connection property named Prefix. A prefix is not required, and if not specified, no prefix is added to the business object name. • <i>SchemaName</i> is the name of the schema to which the object belongs. • <i>ObjectName</i> is the name of the table, view, stored procedure, stored function, or synonym/nickname. A number is appended if necessary to differentiate the business object from another business object with the same name. For example, using the prefix Campaign12 for the Customer table in the Sales schema, the business object name is Campaign12SalesCustomer.
Query business objects	For query business objects, the external service wizard generates the name of the business object in the form of <i>Prefix + QueryBOName</i> , where: <ul style="list-style-type: none"> • <i>Prefix</i> is the prefix you specify in the wizard. A prefix is not required, and if not specified, no prefix is added to the business object name. • <i>QueryBOName</i> is the value you specified when you configured the business object in the wizard. A number is appended if necessary to differentiate the business object from another business object with the same name.

Related tasks

“Setting connection properties for the external service wizard” on page 83
 For the external service wizard to connect to the database instance to discover database objects, connection properties must be specified.

Fault business objects

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

The WebSphere Adapter for Oracle E-Business Suite enables faults for you. Manual configuration of faults is not required. Some faults contain the matchCount attribute which provides additional information about the error. For others, WBIFault contains all the information needed to handle the fault.

The adapter provides the following fault business objects that the wizard creates:

- IntegrityConstraintFault

When processing a Create, Update, or Execute operation (other than on stored procedure), the adapter throws an integrity constraint fault if the Oracle database throws the SQLException exception for an integrity constraint violation. For example, if a foreign key is not found, the adapter throws this fault.
- MatchesExceededLimitFault

When processing a RetrieveAll operation, the adapter returns this fault if the number of records returned from the database query exceeds the value for the MaxRecords property in the interaction specification.

To increase the number of records that are returned, increase the value of the MaxRecords property in the interaction specification properties for the RetrieveAll operation.

The business object for this fault has only one property, `matchCount`, which is a string that contains the number of matches.

- `MissingDataFault`

If the business object that is passed to the outbound operation does not have all the required attributes, the adapter throws this fault.

- `MultipleMatchingRecordsFault`

When processing a `Retrieve` or `Update` operation, the adapter throws this fault if the query returns more than one record for the specified keys. The business object for this fault has one property, `matchCount`, a string that contains the number of matches.

- `ObjectNotFoundFault`

This fault can be thrown during a `Create` or `Update` operation. During the `Create` operation, the adapter retrieves the single cardinality child object if the ownership is false for this child object. If the retrieval of the child object does not return results, this fault is thrown.

- `RecordNotFoundFault`

When retrieving data, the adapter returns this fault if the record is not found in the database for the keys specified. This fault can occur for the `Retrieve`, `RetrieveAll`, `Delete`, and `Update` operations.

- `UniqueConstraintFault`

When processing a `Create` or `Update` operation, the adapter returns this fault if it receives a `SQLException` exception from the Oracle database because of a unique constraint violation.

Related concepts

“Create operation” on page 6

The Create operation creates rows in database tables corresponding to the business object in the request. When given a hierarchical business object, the Create operation recursively traverses the business object, creating rows corresponding to each business object in the hierarchy.

“Retrieve operation” on page 8

The Retrieve operation extracts data from a database for a hierarchy of business objects.

“RetrieveAll operation” on page 9

The adapter uses the RetrieveAll operation to retrieve an array of business objects from the database. The adapter uses different processes depending on whether the RetrieveAll operation is for database table business objects or for user-specified SQL business objects.

“Update operation” on page 12

In an Update operation, the source business object is compared to a business object that is retrieved from the database using the primary keys specified in the top-level, source business object.

“ApplyChanges operation” on page 13

The ApplyChanges operation provides both delta and after-image support for modifying or deleting a business object. The ApplyChanges operation is available only if you use business graphs.

“Delete operation” on page 15

The Delete operation is performed by pruning the incoming business object and then retrieving the complete business object from the database. The Delete operation is then applied recursively on each business object in the hierarchy.

“Execute operation” on page 16

The Execute operation is used to run stored procedures and stored functions. The external service wizard generates the required stored procedure business object that corresponds to the stored procedure or stored function definition in the database. The adapter uses the Execute operation to process the stored procedure business object.

“Exists operation” on page 17

The Exists operation determines whether the database contains records that match the attributes set in a business object.

“Business faults” on page 47

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Configuration properties

IBM WebSphere Adapter for Oracle E-Business Suite has several categories of configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter, managed connection factory, and activation specification properties after you deploy the application to IBM Business Process Manager or WebSphere Enterprise Service Bus.

Outbound configuration properties

IBM WebSphere Adapter for Oracle E-Business Suite has several categories of outbound connection configuration properties, which you set with the external

service wizard while generating or creating objects and services. You can change the resource adapter and managed connection factory properties after you deploy the module to IBM Business Process Manager or WebSphere Enterprise Service Bus using IBM Integration Designer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Guide to information about properties

The properties used to configure WebSphere Adapter for Oracle E-Business Suite are described in detail in tables included in each of the configuration properties topics, such as Resource adapter properties, Managed connection factory properties, and so on. To help you use these tables, information about each row you might see is explained here.

The following table explains the meaning of each row that might be displayed in the table for a configuration property.

Row	Explanation
Required	<p>A required field (property) must have a value in order for the adapter to work. Sometimes the external service wizard provides a default value for required properties.</p> <p>Removing a default value from a required field on the external service wizard <i>will not change that default value</i>. When a required field contains no value at all, the external service wizard processes the field using its assigned default value, and that default value is displayed on the administrative console.</p> <p>Possible values are Yes and No.</p> <p>Sometimes a property is required only when another property has a specific value. When this is the case, the table will note this dependency. For example,</p> <ul style="list-style-type: none"> • Yes, when the EventQueryType property is set to Dynamic • Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.
Default	<p>The predefined value that is set by the external service wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p>
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	<p>Describes the property type. Valid property types include:</p> <ul style="list-style-type: none"> • Boolean • String • Integer

Row	Explanation
Usage	<p>Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented:</p> <p>For Rational® Application Developer for WebSphere Software version 6.40 or earlier, the password:</p> <ul style="list-style-type: none"> • Must be uppercase • Must be 8 characters in length <p>For versions of Rational Application Developer for WebSphere Software later than 6.40, the password:</p> <ul style="list-style-type: none"> • Is not case sensitive • Can be up to 40 characters in length. <p>This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.</p>
Example	<p>Provides sample property values, for example:</p> <p>"If Language is set to JA (Japanese), code page number is set to 8000".</p>
Globalized	<p>If a property is globalized, it has national language support, meaning that you can set the value in your national language.</p> <p>Valid values are Yes and No.</p>
Bidi supported	<p>Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.</p> <p>Valid values are Yes and No.</p>

Connection properties for the wizard

External service wizard connection properties are used to establish a connection between the external service wizard, a tool that is used to create business objects, and the database.

Resource adapter properties

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following table lists the resource adapter properties and their purpose. A complete description of each property is provided in the sections that follow the table. For information about how to read the property details table, see “Guide to information about properties” on page 224.

Table 27. Resource adapter properties for the Adapter for Oracle E-Business Suite

Name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing.
Database vendor	DatabaseVendor	The type of database that the adapter uses for special processing.

Table 27. Resource adapter properties for the Adapter for Oracle E-Business Suite (continued)

Name		Description
In the wizard	In the administrative console	
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files.
Query time out	QueryTimeOut	The maximum number of seconds a query can take for all SQL statements.
Return business object even when the stored procedure result set is empty	ReturnDummyBOForSP	Specifies whether to return output parameters when the result set is empty.
(Not available)	enableHASupport	Do not change this property.
(Not available)	LogFileSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
SQL query to verify the connection	PingQuery	The SQL query used to test the reliability of the connection to the database.
(Not available)	TraceFileSize	Deprecated
(Not available)	TraceFileName	Deprecated
(Not available)	TraceNumberOfFiles	Deprecated

Adapter ID (AdapterID)

This property identifies a specific deployment or instance of the adapter.

Table 28. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 28. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in the log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, 0EBSRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is 0EBSRA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for Oracle E-Business Suite to 001 and 002. The component IDs for those instances, 0EBSRA001 and 0EBSRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to 0EBSRAInstanc.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

Database vendor (DatabaseVendor)

This property specifies the type of database that is used. The type is determined by the database vendor name.

Table 29. Database vendor details

Required	Yes
Possible values	Oracle
Default	ORACLE
Property type	String
Usage	<p>Some SQL statements require special processing, which varies by database type. For example, the Struct and Array data types in Oracle require special processing. This property specifies the RDBMS that is used, which determines the database type.</p> <p>For other databases, the adapter does not perform any special processing. Make sure that the correct driver is specified in the JDBCDriverClass property.</p> <p>Specify Oracle as the value that corresponds to your database vendor.</p>
Globalized	No

Table 29. Database vendor details (continued)

Bidi supported	No
----------------	----

Disguise user data as "XXX" in log and trace files (HideConfidentialTrace)

This property specifies whether to replace user data in log and trace files with a string of X's to prevent unauthorized disclosure of potentially sensitive data.

Table 30. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

Do not change this property. It must be set to true.

Query timeout (seconds) (QueryTimeOut)

This property specifies the maximum number of seconds a query can take to run all SQL statements.

Table 31. Query timeout details

Required	No
Default	No default value
Unit of measure	Seconds
Property type	Integer
Usage	<p>If the query takes longer than the number of seconds specified, the database generates an SQL exception that is captured. The associated message is logged in the log file.</p> <p>If a value is not specified, no timeout is set on the query.</p>
Globalized	No
Bidi supported	No

Return business object even when the stored procedure result set is empty (ReturnDummyBOForSP)

This property specifies whether to return output parameters when the result set is empty.

Table 32. Return business object even when the stored procedure result set is empty details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>The Retrieve Stored Procedure (RetrieveSP) operation returns a result set. If the result set is empty and the ReturnDummyBOForSP property is set to False, no business objects are created, and the output parameters returned by the procedure call cannot be retrieved.</p> <p>However, if the ReturnDummyBOForSP property is set to True, a dummy business object is created and the returned parameters (including the out and inout parameters) from the stored procedure are populated into the corresponding attributes</p>
Globalized	No
Bidi supported	No

SQL query to verify the connection (PingQuery)

This property specifies the SQL query that is used to test the reliability of the connection to the database.

Table 33. Ping query details

Required	No
Property type	String
Default	No default value
Usage	<p>This property contains the SQL query statement that you want to run to determine whether the adapter can connect to the database.</p> <p>The adapter runs the ping query every time it receives a SQLException exception while performing the outbound operation.</p> <p>The adapter does not try to recover the connection. If the ping query indicates that the connection to the database is no longer valid, the adapter notifies the container. It is the responsibility of the connection pool manager to remove the stale connection from the pool, which allows subsequent outbound requests to be processed.</p>
Globalized	No
Bidi supported	No

Managed connection factory properties

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Oracle E-Business Suite.

You set managed connection factory properties using the external service wizard during adapter configuration. You can change them using the IBM Integration Designer assembly editor or, after deployment, with the IBM Business Process Manager or WebSphere Enterprise Service Bus administrative console.

The following table lists and describes the managed connection factory properties. A complete description of each property is provided in the sections that follow the table. For information about how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 224.

Note: The external service wizard refers to these properties as managed connection factory properties, while the administrative console refers to them as J2C connection factory properties.

Table 34. Managed connection factory properties for Adapter for Oracle E-Business Suite

Property name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing.
Additional JDBC driver connection properties [name:value;name:value]	JDBCDriverConnectionProperties	Additional properties for connecting to the database using the JDBC driver, which are used in addition to the UserName and Password properties.
Auto commit	AutoCommit	The AutoCommit value to use on the connection
DataSource JNDI name	DataSourceJNDIName	This property is deprecated. Use the XADataSourceJNDIName or PoolDataSourceJNDIName properties in place of this
Database URL	DatabaseURL	The database URL that is used to connect to the database.
Database vendor	DatabaseVendor	The type of database that the adapter uses for special processing.
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files.
JDBC driver class	JDBCDriverClass	The class name of the JDBC driver that is used to connect to the database.
“Maximum retries on connection failure (connectionRetryLimit)” on page 239	connectionRetryLimit	Specifies the maximum number of times the adapter tries to reestablish an outbound connection to the EIS.
Password	Password	Password for the corresponding user name
Query timeout	QueryTimeOut	The maximum number of seconds a query can take for all SQL statements.

Table 34. Managed connection factory properties for Adapter for Oracle E-Business Suite (continued)

Property name		Description
In the wizard	In the administrative console	
"Connection Retry interval (in milliseconds) (connectionRetryInterval)" on page 240	connectionRetryInterval	Specifies the time interval between attempts to reconnect to the EIS if the connection fails.
Return business object even when the stored procedure result set is empty	ReturnDummyBOForS	Specifies whether to return output parameters when the result set is empty.
SQL query to connection	PingQuery	The SQL query used to test the reliability of the connection to the database.
User name	UserName	The database user name.
"XA DataSource name (XADataSourceName)" on page 239	XADataSourceName	The name of the XA data source to use to establish a connection to the database for XA (distributed) transactions
"Database connection information (ConnectionType)" on page 241	ConnectionType	Specifies how the adapter establishes connection to the database.
"Connection pool DataSource JNDI name (PoolDataSourceJNDIName)" on page 243	PoolDataSourceJNDIName	The JNDI name of the connection pool data source to be used to establish a connection to the database.
"XA DataSource JNDI name (XADataSourceJNDIName)" on page 244	XADataSourceJNDIName	The JNDI name of the XA data source used to establish a connection to the database.

Table 35. Managed connection factory properties for XML gateway interface

Property name		Description
In the wizard	In the administrative console	
"Oracle E-Business Suite Connection Type (EBSConnectionType)" on page 244	EBSConnectionType	Specifies the Oracle E-Business Suite connection type.
"Oracle E-Business Suite Transport (EBSTransport)" on page 245	EBSTransport	Specifies the Oracle E-Business Suite transport.
"Oracle E-Business Suite Version (EBSVersion)" on page 245	EBSVersion	Specifies the Oracle E-Business Suite version.
"Password (Password)" on page 246	Password	Specifies the password required when invoking the web or JMS service.
"Host Name (EBSHostName)" on page 246	EBSHostName	Specifies the Oracle E-Business Suite Host name or IP Address when invoking the JMS service.
"System ID (EBSSystemID)" on page 246	EBSSystemID	Specifies the Oracle E-Business Suite system ID when invoking the JMS service.
"User Name (UserName)" on page 248	UserName	Specifies the username for invoking the web or JMS service.
"WSDL URL (WSDLURL)" on page 248	WSDLURL	Specifies the WSDL URL of the XML gateway interface.

Table 35. Managed connection factory properties for XML gateway interface (continued)

Property name		Description
In the wizard	In the administrative console	
"Port Number (EBSPortNumber)" on page 246	EBSPortNumber	Specifies the Oracle E-Business Suite port number when invoking the JMS.
"Schema Name (EBSSchemaName)" on page 245	EBSSchemaName	Specifies the schema name of XML gateway JMS advanced queue.
"Queue Name (EBSQueueName)" on page 245	EBSQueueName	Specifies the XML gateway JMS advanced queue name.

Adapter ID (AdapterID)

This property identifies a specific deployment or instance of the adapter.

Table 36. Adapter ID details

Required	Yes
Default	001
Property type	String
Usage	<p>This property identifies the adapter instance in the log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, 0EBSRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is 0EBSRA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for Oracle E-Business Suite to 001 and 002. The component IDs for those instances, 0EBSRA001 and 0EBSRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to 0EBSRAInstanc.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

Additional JDBC driver connection properties [name:value;name:value] (JDBCDriverConnectionProperties)

This property contains additional information for connecting to the database using the JDBC driver.

Table 37. Additional JDBC driver connection properties details

Row	Explanation
Required	No
Possible values	Database connection properties are database-specific.
Default	No default value
Property type	String
Usage	These connection properties are used in addition to the UserName and Password properties to customize the database connection used by the adapter. Specify the connection properties as one or more <i>name:value</i> pairs separated by the semicolon character (;).
Example	The following value of this property specifies a login timeout interval: loginTimeout:20; ConnectionRetryCount:5; ConnectionRetryDelay:5
Globalized	Yes
Bidi supported	No

Auto commit (AutoCommit)

This property specifies whether AutoCommit is set for the connection.

Table 38. Auto commit details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	This property is ignored for XA (distributed) transactions.
Globalized	No
Bidi supported	No

Database URL (DatabaseURL)

This property specifies the JDBC driver-specific URL for creating a connection to the database.

Table 39. Database URL details

Required	Yes, when the connection is of the type LocalConnectionProps or ConnectionProps.
Default	No default value
Property type	String

Table 39. Database URL details (continued)

Usage	<p>In the external service wizard, compose the database URL by entering in database-specific fields. For example, the database URL for an Oracle database is composed of the system ID (SID), server host name, and database port number. In the administrative console, type the complete database URL value.</p> <p>If your database server supports IPv6, you can specify the host name portion of the database URL in IPv6 format.</p> <p>If you specify the host name as an IP address in IPv6 format, enclose the IP address in square brackets ([]).</p>
Examples	<p>The following example illustrates typical database URL values for a common</p> <p>database:jdbc:oracle:thin:@9.26.248.148:1521:dev</p>
Globalized	Yes
Bidi supported	Yes

Database vendor (DatabaseVendor)

This property specifies the type of database that is used. The type is determined by the database vendor name.

Table 40. Database vendor details

Required	Yes
Possible values	Oracle
Default	ORACLE
Property type	String
Usage	<p>Some SQL statements require special processing, which varies by database type. For example, the Struct and Array data types in Oracle require special processing. This property specifies the RDBMS that is used, which determines the database type.</p> <p>For other databases, the adapter does not perform any special processing. Make sure that the correct driver is specified in the JDBCDriverClass property.</p> <p>Specify Oracle as the value that corresponds to your database vendor.</p>
Globalized	No
Bidi supported	No

Disguise user data as "XXX" in log and trace files (HideConfidentialTrace)

This property specifies whether to replace user data in log and trace files with a string of X's to prevent unauthorized disclosure of potentially sensitive data.

Table 41. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False

Table 41. Disguise user data as "XXX" in log and trace files details (continued)

Default	False
Property type	Boolean
Usage	<p>If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No
Bidi supported	No

Throw exception when no records are found (ErrorOnEmptyResultSet)

This property specifies whether to generate a fault or return an empty business object when no records are found.

Table 42. Throw exception when no records are found details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	If you set this property to False, the RetrieveAll operation does not generate a RecordNotFoundException exception when no records are found.
Globalized	No
Bidi supported	No

JDBC driver class (JDBCDriverClass)

This property specifies the class name of the JDBC driver that is used to connect to the database.

Table 43. JDBC driver class details

Row	Explanation
Required	Yes, when the connection is of the type LocalConnectionProps or ConnectionProps.
Possible values	The value specified when you set the connection properties to connect to the database instance.
Default	No default value
Property type	String

Table 43. JDBC driver class details (continued)

Usage	<p>In the external service wizard, the JDBC driver class is displayed if you select a common database software and driver combination, such as type 4 drivers for recent versions of Oracle. For type 2 drivers, you must type the database class name.</p> <p>If you select a type 2 driver or a generic driver, you must type the JDBC driver class name.</p> <p>In the administrative console, type the database-specific name of the driver.</p>
Examples	<p>Values for the JDBC driver class is displayed in both the external service wizard and the administrative console. The following examples depict the JDBC driver class properties in both the external service wizard and the administrative console. In the external service wizard:</p> <ul style="list-style-type: none"> To connect to an Oracle 10 database using the type 4 driver, select Oracle Thin Driver. <p>In the administrative console:</p> <p>Oracle Thin JDBC driver <code>oracle.jdbc.driver.OracleDriver</code></p>
Globalized	No
Bidi supported	No

Password (Password)

This property specifies the password for the database user name.

Table 44. Password details

Required	<p>No. For inbound processing, if you set the Authentication alias or the DataSourceJNDIName, the password is not mandatory. However, if you set the DataSourceJNDIName, and the Password field, the value specified for the Password takes precedence.</p> <p>For outbound processing, if you set the Authentication alias, XADataSourceJNDIName or PoolDataSourceJNDIName property, the password is not mandatory. However, if you set the XADataSourceJNDIName or PoolDataSourceJNDIName property, and the Password field, the value specified for the Password takes precedence.</p>
Default	No default value
Property type	String
Usage	<p>For inbound processing, setting this property overrides the password specified for the data source on the server using the Authentication alias or DataSourceJNDIName property.</p> <p>For outbound processing, setting this property overrides the password specified for the data source on the server using the Authentication alias, or XADataSourceJNDIName or PoolDataSourceJNDIName property.</p> <p>If you specify JAAS as the security credential, the authentication alias will override this property.</p>

Table 44. Password details (continued)

Globalized	Yes
Bidi supported	Yes

Query timeout (seconds) (QueryTimeout)

This property specifies the maximum number of seconds a query can take to run all SQL statements.

Table 45. Query timeout details

Required	No
Default	No default value
Unit of measure	Seconds
Property type	Integer
Usage	If the query takes longer than the number of seconds specified, the database generates an SQL exception that is captured. The associated message is logged in the log file. If a value is not specified, no timeout is set on the query.
Globalized	No
Bidi supported	No

Return business object even when the stored procedure result set is empty (ReturnDummyBOForSP)

This property specifies whether to return output parameters when the result set is empty.

Table 46. Return business object even when the stored procedure result set is empty details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	The Retrieve Stored Procedure (RetrieveSP) operation returns a result set. If the result set is empty and the ReturnDummyBOForSP property is set to False, no business objects are created, and the output parameters returned by the procedure call cannot be retrieved. However, if the ReturnDummyBOForSP property is set to True, a dummy business object is created and the returned parameters (including the out and inout parameters) from the stored procedure are populated into the corresponding attributes
Globalized	No
Bidi supported	No

SQL query to verify the connection (PingQuery)

This property specifies the SQL query that is used to test the reliability of the connection to the database.

Table 47. Ping query details

Required	No
Property type	String
Default	No default value
Usage	<p>This property contains the SQL query statement that you want to run to determine whether the adapter can connect to the database.</p> <p>The adapter runs the ping query every time it receives a SQLException exception while performing the outbound operation.</p> <p>The adapter does not try to recover the connection. If the ping query indicates that the connection to the database is no longer valid, the adapter notifies the container. It is the responsibility of the connection pool manager to remove the stale connection from the pool, which allows subsequent outbound requests to be processed.</p>
Globalized	No
Bidi supported	No

User name (UserName)

This property specifies the database user name that is used to access the database.

Table 48. User name details

Required	<p>No. For inbound processing, if you set the Authentication alias or the DataSourceJNDIName, the User name property is not mandatory. However, if you set the DataSourceJNDIName, and User name field, the value specified for the User name takes precedence.</p> <p>For outbound processing, if you set the Authentication alias, XADataSourceJNDIName or PoolDataSourceJNDIName property, the User name is not mandatory. However, if you set the XADataSourceJNDIName or PoolDataSourceJNDIName property, and the User name field, the value specified for the User name takes precedence.</p>
Default	No default value
Property type	String
Usage	<p>For inbound processing setting this property overrides the user name specified for the data source on the server using the DataSourceJNDIName property or the Authentication alias.</p> <p>For outbound processing, setting this property overrides the user name specified for the data source on the server using the XADataSourceJNDIName property, PoolDataSourceJNDIName property or the Authentication alias.</p> <p>If you specify JAAS as the security credential, the authentication alias will override this property.</p>

Table 48. User name details (continued)

Globalized	Yes
Bidi supported	Yes

XA DataSource name (XADataSourceName)

This property specifies the name of the XA data source used to establish a connection to the database for XA (distributed) transactions.

Table 49. XA data source name details

Required	No
Default	No default value
Property type	String
Usage	This property is used to make an XA connection to an Oracle database.
Examples	A typical value for an Oracle database: <code>oracle.jdbc.xa.client.OracleXADataSource</code>
Globalized	No
Bidi supported	No

Maximum retries on connection failure (connectionRetryLimit)

This property specifies the number of times the adapter attempts to reestablish a connection to the EIS, when the adapter encounters an error related to the outbound connection.

Table 50. Maximum retries in case of system connection failure details

Required	No
Possible values	Integer equal to and greater than zero
Default	0
Property type	Integer

Table 50. Maximum retries in case of system connection failure details (continued)

Usage	<p>When the adapter encounters an error related to the outbound connection, it retries to establish a physical connection (when physical connection is not established) for the number of times specified for this property with a time delay specified in the property "Connection Retry interval (in milliseconds) (connectionRetryInterval)."</p> <p>If the value is set to 0, the adapter does not validate the connection to the EIS and it executes the outbound operation. If the EIS connection is invalid, the outbound operation fails. The adapter does not attempt to reconnect to the EIS.</p> <p>If the value is set to greater than 0, then during each request the adapter validates if the connection to the EIS is active.</p> <ul style="list-style-type: none"> • If the connection is valid, the operation is completed. • If connection is invalid, the adapter terminates the current managed connection and a new managed connection is created (new physical connection). If the adapter is successful in reestablishing the connection to the EIS, the outbound operation is completed otherwise the adapter generates the ResourceException after trying to reconnect for the specified number of times.
Globalized	No
Bidi supported	No

Connection Retry interval (in milliseconds) (connectionRetryInterval)

This property specifies the time interval between attempts to reconnect to the EIS if the connection fails.

Table 51. Retry interval if connection failure details

Required	No
Possible values	Integer equal to and greater than zero
Default	60000
Unit of measure	Milliseconds
Property type	Integer
Usage	<p>When the adapter encounters an error while establishing a connection to the EIS, this property specifies the time interval the adapter waits between attempts to reestablish a connection.</p> <p>By default, this property disabled and it is enabled only when the value of the "Maximum retries on connection failure (connectionRetryLimit)" on page 239property is set to greater than zero.</p>
Globalized	No
Bidi supported	No

Database connection information (ConnectionType)

This property specifies how the adapter establishes connection to the database.

Table 52. Database connection information

Required	Yes
Possible values	XADataSourceJNDI, XAConnectionProps, PoolDataSourceJNDI, or LocalConnectionProps
Default	No default value
Property type	String

Table 52. Database connection information (continued)

Usage	<p>This property specifies how the adapter establishes database connection at run time. This property can have the following values:</p> <ul style="list-style-type: none"> • XADataSourceJNDI- Indicates that the database connection is established using the XADataSourceJNDIName property that corresponds to the predefined XA data source. • XAConnectionProps - Indicates that the database connection is established using the XADataSourceName and DatabaseURL or XADatabaseName properties. • PoolDataSourceJNDI - Indicates that the database connection is established using the poolDataSourceJNDIName property that corresponds to the predefined data source. • LocalConnectionProps - Indicates that the database connection is established using the DatabaseURL and JDBCClass properties. <p>For a new application, this property is automatically set by external service wizard. For a migrated application, this property is set during the migration process according to the ManagedConnectionFactory properties.</p> <ul style="list-style-type: none"> • If the DataSourceJNDIName property is set, the value of this property is set to XADataSourceJNDI. • If the XADataSourceName is set, the value of this property is set to XAConnectionProps. • Otherwise, the value of this property is set to LocalConnectionProps. <p>If this property is not set, adapter uses backward compatibility mode to establish database connection. In the backward compatibility mode, the properties for connecting to the database are used in the following order:</p> <ol style="list-style-type: none"> 1. If the DataSourceJNDIName property is set, the adapter uses it to establish the connection to the database. 2. If the DataSourceJNDIName property is not set, and the XADataSourceName and XADatabaseName properties are set, the adapter uses them to establish the connection. The DataSourceJNDIName property represents an XA or connection pool data source. If you define a JNDI data source on the server that supports XA transactions and then specify that data source when you configure the adapter, you can connect to any type of database that supports XA transactions. If you use an XA data source and database, the adapter supports XA transaction only for DB2[®] and Oracle databases. 3. If the DataSourceJNDIName, XADataSourceName, and XADatabaseName properties are not set, then the adapter uses the DatabaseURL, JDBCClass, UserName, and Password properties to establish the connection.
Globalized	No
Bidi supported	No

Connection pool DataSource JNDI name (PoolDataSourceJNDIName)

The JNDI name of the connection pool data source to be used to establish a connection to the database.

Table 53. Connection pool data source

Required	No
Default	No default value
Property type	String
Usage	Use this property to specify the JNDI name of a connection pool data source in IBM Business Process Manager or WebSphere Enterprise Service Bus that specifies connection information for the target database. If the "Database connection information (ConnectionType)" on page 241 property is set to "PoolDataSourceJNDI" the adapter use this property to establish the connection to the database. To improve the performance of outbound operations, specify the name of a data source that is enabled for prepared statement caching. If the other valid authentication properties are also set, they override authentication properties in the data source.
Globalized	No
Bidi supported	No

JDBC driver class (JDBCDriverClass)

This property specifies the class name of the JDBC driver that is used to connect to the database.

Table 54. JDBC driver class details

Row	Explanation
Required	Yes, when the connection is of the type LocalConnectionProps or ConnectionProps.
Possible values	The value specified when you set the connection properties to connect to the database instance.
Default	No default value
Property type	String
Usage	<p>In the external service wizard, the JDBC driver class is displayed if you select a common database software and driver combination, such as type 4 drivers for recent versions of Oracle. For type 2 drivers , you must type the database class name.</p> <p>If you select a type 2 driver or a generic driver, you must type the JDBC driver class name.</p> <p>In the administrative console, type the database-specific name of the driver.</p>

Table 54. JDBC driver class details (continued)

Examples	<p>Values for the JDBC driver class is displayed in both the external service wizard and the administrative console. The following examples depict the JDBC driver class properties in both the external service wizard and the administrative console. In the external service wizard:</p> <ul style="list-style-type: none"> To connect to an Oracle 10 database using the type 4 driver, select Oracle Thin Driver. <p>In the administrative console:</p> <p>Oracle Thin JDBC driver oracle.jdbc.driver.OracleDriver</p>
Globalized	No
Bidi supported	No

XA DataSource JNDI name (XADataSourceJNDIName)

This property specifies the JNDI name of the XA data source used to establish a connection to the database.

Table 55. XA Data source JNDI name details

Required	No
Default	No default value
Property type	String
Usage	<p>Use this property to specify the JNDI name of a XA data source in IBM Business Process Manager or WebSphere Enterprise Service Bus that specifies connection information for the target database. If the "Database connection information (ConnectionType)" on page 241 is set to "XADataSourceJNDI" the adapter uses this property to establish the connection to the database. To improve the performance of outbound operations, specify the name of a data source that is enabled for prepared statement caching. If the other valid authentication properties are also set, they override authentication properties in the data source.</p>
Globalized	Yes
Bidi supported	No

Oracle E-Business Suite Connection Type (EBSConnectionType)

This property specifies the Oracle E-Business Suite connection type.

Table 56. Oracle E-Business Suite Connection Type details

Required	Yes
Possible values	XML Gateway
Property type	String
Usage	Integrates Oracle E-Business Suite using XML Gateway connection type. If blank, then the runtime is for the database operation.
Globalized	Yes
Bidi supported	No

Oracle E-Business Suite Transport (EBSTransport)

This property specifies the Oracle E-Business Suite transport.

Table 57. Oracle E-Business Suite Transport details

Required	Yes
Possible values	Web Service
Property type	String
Usage	Integrates Oracle E-Business Suite using Web Service transport. For XML Gateway process, this property should be set as 'XML Gateway', if no value is set, the JDBC connection type will be used as default.
Globalized	Yes
Bidi supported	No

Oracle E-Business Suite Version (EBSVersion)

This property specifies the Oracle E-Business Suite version.

Table 58. Oracle E-Business Suite Version details

Required	Yes
Possible values	11.5, 12.0, 12.1
Property type	String
Usage	Connects to the selected version of Oracle E-Business Suite.
Globalized	Yes
Bidi supported	No

Schema Name (EBSSchemaName)

This property specifies the schema name of XML gateway JMS advanced queue.

Table 59. Schema Name details

Required	Yes
Default value	applsys
Property type	String
Usage	Specifies the schema name of XML Gateway JMS Advanced Queue
Globalized	No
Bidi supported	No

Queue Name (EBSQueueName)

This property specifies the schema name of XML gateway JMS advanced queue.

Table 60. Queue Name details

Required	Yes
Default value	wf_jms_in
Property type	String
Usage	Specifies the XML gateway JMS advanced queue name

Table 60. Queue Name details (continued)

Globalized	No
Bidi supported	No

Port Number (EBSPortNumber)

This property specifies the Oracle E-Business Suite port number when invoking the JMS.

Table 61. Port Number details

Required	Yes
Property type	Integer
Usage	Specifies the Oracle E-Business Suite port number when invoking the JMS.
Globalized	No
Bidi supported	No

Host Name (EBSHostName)

This property specifies the Oracle E-Business Suite host name or IP Address when invoking the JMS.

Table 62. Host Name details

Required	Yes.
Property type	String
Globalized	Yes
Bidi supported	No

System ID (EBSSystemID)

This property specifies the Oracle E-Business Suite system ID when invoking the JMS.

Table 63. System ID details

Required	Yes.
Property type	String
Globalized	Yes
Bidi supported	No

Password (Password)

This property specifies the password required when invoking the web service.

Table 64. Password details

Required	If you set the Authentication alias, the password is not mandatory.
Property type	String

Table 64. Password details (continued)

Usage	If you specify JAAS as the security credential, the authentication alias will override this property.
Globalized	Yes
Bidi supported	No

Password (Password)

This property specifies the password for the database user name.

Table 65. Password details

Required	<p>No. For inbound processing, if you set the Authentication alias or the DataSourceJNDIName, the password is not mandatory. However, if you set the DataSourceJNDIName, and the Password field, the value specified for the Password takes precedence.</p> <p>For outbound processing, if you set the Authentication alias, XADataSourceJNDIName or PoolDataSourceJNDIName property, the password is not mandatory. However, if you set the XADataSourceJNDIName or PoolDataSourceJNDIName property, and the Password field, the value specified for the Password takes precedence.</p>
Default	No default value
Property type	String
Usage	<p>For inbound processing, setting this property overrides the password specified for the data source on the server using the Authentication alias or DataSourceJNDIName property.</p> <p>For outbound processing, setting this property overrides the password specified for the data source on the server using the Authentication alias, or XADataSourceJNDIName or PoolDataSourceJNDIName property.</p> <p>If you specify JAAS as the security credential, the authentication alias will override this property.</p>
Globalized	Yes
Bidi supported	Yes

User name (UserName)

This property specifies the database user name that is used to access the database.

Table 66. User name details

Required	No. For inbound processing, if you set the Authentication alias or the DataSourceJNDIName, the User name property is not mandatory. However, if you set the DataSourceJNDIName, and User name field, the value specified for the User name takes precedence. For outbound processing, if you set the Authentication alias, XADataSourceJNDIName or PoolDataSourceJNDIName property, the User name is not mandatory. However, if you set the XADataSourceJNDIName or PoolDataSourceJNDIName property, and the User name field, the value specified for the User name takes precedence.
Default	No default value
Property type	String
Usage	For inbound processing setting this property overrides the user name specified for the data source on the server using the DataSourceJNDIName property or the Authentication alias. For outbound processing, setting this property overrides the user name specified for the data source on the server using the XADataSourceJNDIName property, PoolDataSourceJNDIName property or the Authentication alias. If you specify JAAS as the security credential, the authentication alias will override this property.
Globalized	Yes
Bidi supported	Yes

User Name (UserName)

This property specifies the username required when invoking the web service.

Table 67. User Name details

Required	If you set the Authentication alias, the User name is not mandatory
Property type	String
Usage	If you specify JAAS as the security credential, the authentication alias will override this property.
Globalized	Yes
Bidi supported	No

WSDL URL (WSDLURL)

This property specifies the WSDL URL of the XML Gateway interface.

Table 68. WSDL URL details

Required	Yes
Property type	String
Globalized	Yes

Table 68. WSDL URL details (continued)

Bidi supported	No
----------------	----

Related tasks

“Setting managed (J2C) connection factory properties for embedded adapters” on page 177

To set managed connection factory properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Interaction specification properties

Interaction specification, or InteractionSpec, properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, some properties for outbound operations can be changed by the user. For example, you might increase the value of the interaction specification property that specifies the maximum number of records to be returned by a RetrieveAll operation, if your RetrieveAll operations do not return complete information. To change these properties after the application is deployed, use the assembly editor in IBM Integration Designer. The properties reside in the method binding of the import.

Table 69 lists and describes the interaction specification property that you set. For information about how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 224.

Table 69. Interaction specification property for the Adapter for Oracle E-Business Suite

Property name	Description
“Maximum records for RetrieveAll operation”	Maximum number of result sets to return during a RetrieveAll operation

Maximum records for RetrieveAll operation

This property specifies the maximum number of records to return for a RetrieveAll operation.

Table 70. Maximum records for RetrieveAll operation details

Required	Yes
Default	100

Table 70. Maximum records for RetrieveAll operation details (continued)

Usage	<p>Use this property to control the number of records returned by the RetrieveAll operation. If the number of matches in the database exceeds the value of this property, the adapter throws the exception MatchesExceededLimitException which is wrapped as a fault, MatchesExceededLimitFault and is returned to the client.</p> <p>Note: The MatchesExceededLimitFault is stored in the trace file and the MatchesExceededLimitException is not stored in log or trace files.</p> <ul style="list-style-type: none"> • If the value is -1, the RetrieveAll operation returns all records matching the query. The value for this property is set to -1 internally when you select the Return all records for RetrieveAll operation check box. • If the value is zero or less than zero except -1, the adapter generates the fault MatchesExceededLimitFault. • If the value is greater than zero and the number of matches in the database exceeds the value of this property, the adapter generates the fault MatchesExceededLimitFault. If the RetrieveAll operation does not return all the records, increase this value. For example, if you set the value to 50 and the table contains 100 records, the adapter generates the fault MatchesExceededLimitFault. • If the value is greater than zero and the number of matches in the database is lesser than the value of this property, the RetrieveAll operation returns all records. For example, if you set the value to 50 and the table contains 25 records, the RetrieveAll operation returns all the 25 records.
Property type	Integer
Globalized	No
Bidi supported	No

Related tasks

Chapter 6, “Changing interaction specification properties,” on page 163

To change interaction specification properties for your adapter module after generating the service, use the assembly editor in IBM Integration Designer.

Inbound configuration properties

WebSphere Adapter for Oracle E-Business Suite has several categories of inbound connection configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter and activation specification properties after you deploy the module using IBM Integration Designer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Guide to information about properties

The properties used to configure WebSphere Adapter for Oracle E-Business Suite are described in detail in tables included in each of the configuration properties topics, such as Resource adapter properties, Managed connection factory properties, and so on. To help you use these tables, information about each row you might see is explained here.

The following table explains the meaning of each row that might be displayed in the table for a configuration property.

Row	Explanation
Required	<p>A required field (property) must have a value in order for the adapter to work. Sometimes the external service wizard provides a default value for required properties.</p> <p>Removing a default value from a required field on the external service wizard <i>will not change that default value</i>. When a required field contains no value at all, the external service wizard processes the field using its assigned default value, and that default value is displayed on the administrative console.</p> <p>Possible values are Yes and No.</p> <p>Sometimes a property is required only when another property has a specific value. When this is the case, the table will note this dependency. For example,</p> <ul style="list-style-type: none"> • Yes, when the EventQueryType property is set to Dynamic • Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.
Default	<p>The predefined value that is set by the external service wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p>
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	<p>Describes the property type. Valid property types include:</p> <ul style="list-style-type: none"> • Boolean • String • Integer
Usage	<p>Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented:</p> <p>For Rational Application Developer for WebSphere Software version 6.40 or earlier, the password:</p> <ul style="list-style-type: none"> • Must be uppercase • Must be 8 characters in length <p>For versions of Rational Application Developer for WebSphere Software later than 6.40, the password:</p> <ul style="list-style-type: none"> • Is not case sensitive • Can be up to 40 characters in length. <p>This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.</p>
Example	<p>Provides sample property values, for example:</p> <p>"If Language is set to JA (Japanese), code page number is set to 8000".</p>
Globalized	<p>If a property is globalized, it has national language support, meaning that you can set the value in your national language.</p> <p>Valid values are Yes and No.</p>

Row	Explanation
Bidi supported	Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file. Valid values are Yes and No .

Connection properties for the wizard

External service wizard connection properties are used to establish a connection between the external service wizard, a tool that is used to create business objects, and the database.

Resource adapter properties

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following table lists the resource adapter properties and their purpose. A complete description of each property is provided in the sections that follow the table. For information about how to read the property details table, see “Guide to information about properties” on page 224.

Table 71. Resource adapter properties for the Adapter for Oracle E-Business Suite

Name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing.
Database vendor	DatabaseVendor	The type of database that the adapter uses for special processing.
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files.
Query time out	QueryTimeout	The maximum number of seconds a query can take for all SQL statements.
Return business object even when the stored procedure result set is empty	ReturnDummyBOForSP	Specifies whether to return output parameters when the result set is empty.
(Not available)	enableHASupport	Do not change this property.
(Not available)	LogFileSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
SQL query to verify the connection	PingQuery	The SQL query used to test the reliability of the connection to the database.
(Not available)	TraceFileSize	Deprecated
(Not available)	TraceFileName	Deprecated
(Not available)	TraceNumberOfFiles	Deprecated

Adapter ID (AdapterID)

This property identifies a specific deployment or instance of the adapter.

Table 72. Adapter ID details

Required	Yes
Default	001
Property type	String
Usage	<p>This property identifies the adapter instance in the log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, 0EBSRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is 0EBSRA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for Oracle E-Business Suite to 001 and 002. The component IDs for those instances, 0EBSRA001 and 0EBSRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to 0EBSRAInstanc.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

Database vendor (DatabaseVendor)

This property specifies the type of database that is used. The type is determined by the database vendor name.

Table 73. Database vendor details

Required	Yes
Possible values	Oracle
Default	ORACLE
Property type	String

Table 73. Database vendor details (continued)

Usage	<p>Some SQL statements require special processing, which varies by database type. For example, the Struct and Array data types in Oracle require special processing. This property specifies the RDBMS that is used, which determines the database type.</p> <p>For other databases, the adapter does not perform any special processing. Make sure that the correct driver is specified in the JDBCDriverClass property.</p> <p>Specify Oracle as the value that corresponds to your database vendor.</p>
Globalized	No
Bidi supported	No

Disguise user data as "XXX" in log and trace files (HideConfidentialTrace)

This property specifies whether to replace user data in log and trace files with a string of X's to prevent unauthorized disclosure of potentially sensitive data.

Table 74. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

Do not change this property. It must be set to true.

Query timeout (seconds) (QueryTimeOut)

This property specifies the maximum number of seconds a query can take to run all SQL statements.

Table 75. Query timeout details

Required	No
Default	No default value

Table 75. Query timeout details (continued)

Unit of measure	Seconds
Property type	Integer
Usage	If the query takes longer than the number of seconds specified, the database generates an SQL exception that is captured. The associated message is logged in the log file. If a value is not specified, no timeout is set on the query.
Globalized	No
Bidi supported	No

Return business object even when the stored procedure result set is empty (ReturnDummyBOForSP)

This property specifies whether to return output parameters when the result set is empty.

Table 76. Return business object even when the stored procedure result set is empty details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	The Retrieve Stored Procedure (RetrieveSP) operation returns a result set. If the result set is empty and the ReturnDummyBOForSP property is set to False, no business objects are created, and the output parameters returned by the procedure call cannot be retrieved. However, if the ReturnDummyBOForSP property is set to True, a dummy business object is created and the returned parameters (including the out and inout parameters) from the stored procedure are populated into the corresponding attributes
Globalized	No
Bidi supported	No

SQL query to verify the connection (PingQuery)

This property specifies the SQL query that is used to test the reliability of the connection to the database.

Table 77. Ping query details

Required	No
Property type	String
Default	No default value

Table 77. Ping query details (continued)

Usage	<p>This property contains the SQL query statement that you want to run to determine whether the adapter can connect to the database.</p> <p>The adapter runs the ping query every time it receives a SQLException exception while performing the outbound operation.</p> <p>The adapter does not try to recover the connection. If the ping query indicates that the connection to the database is no longer valid, the adapter notifies the container. It is the responsibility of the connection pool manager to remove the stale connection from the pool, which allows subsequent outbound requests to be processed.</p>
Globalized	No
Bidi supported	No

Activation specification properties

Activation specification properties are properties that hold the inbound event processing configuration information for an export.

You set activation specification properties using the external service wizard during adapter configuration and can change them using the IBM Integration Designer assembly editor or, after deployment, the IBM Business Process Manager or WebSphere Enterprise Service Bus administrative console.

The following table lists and describes the activation specification properties. A complete description of each property is provided in the sections that follow the table. For information about how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 224.

Table 78. Activation specification properties for Adapter for Oracle E-Business Suite

Property name		Description
In the wizard	In the administrative console	
Adapter Instance for event filtering	AdapterInstanceEventFilter	Identifier that determines whether this adapter instance processes specific events in the event store.
Additional JDBC driver connection properties [name:value;name:value]	JDBCDriverConnectionProperties	Additional properties for connecting to the database using the JDBC driver, which are used in addition to the UserName and Password properties.
Business object namespace	BusinessObjectNameSpace	The namespace for the business object definitions
User-defined delete query	CustomDeleteQuery	The name of the query, stored procedure, or stored function that is run after each event is processed to delete records that can be deleted after the event is delivered.

Table 78. Activation specification properties for Adapter for Oracle E-Business Suite (continued)

Property name		Description
In the wizard	In the administrative console	
User-defined event query	CustomEventQuery	The name of the query, stored procedure, or stored function that performs the polling for events.
User-defined update query	CustomUpdateQuery	The name of the query, stored procedure, or stored function that is run after each event is processed to prevent the event from being picked up for processing in a subsequent event cycle.
User-defined update query for failed event delivery	CustomUpdateQueryForFailedEvent	The name of the query, stored procedure, or stored function that is run when an event is not delivered successfully.
Data source JNDI name	DataSourceJNDIName	The name of the JNDI data source used to establish a connection to the database.
Database URL	DatabaseURL	The database URL that is used to connect to the database.
Database vendor	DatabaseVendor	The type of database that the adapter uses for special processing.
Do not process events that have a timestamp in the future	FilterFutureEvents	Specifies whether the adapter filters out future events by comparing the timestamp on each event with the system time.
Ensure once-only event delivery	AssuredOnceDelivery	Specifies whether the adapter provides assured once delivery of events.
Event order by	EventOrderBy	The order in which events are retrieved and processed
Event query type	EventQueryType	Determines whether to use the standard event store or custom query
Event table name	EventTableName	Name of the database table that contains events generated by the database for inbound processing
Event types to process	EventTypeFilter	A delimited list of event types that indicates to the adapter which events it should deliver. The adapter supports event filtering based on operations, in addition to the BO Name.
Retry limit for failed events	FailedEventRetryLimit	The number of times the adapter attempts to redeliver an event before marking the event as failed.

Table 78. Activation specification properties for Adapter for Oracle E-Business Suite (continued)

Property name		Description
In the wizard	In the administrative console	
Interval between polling periods	PollPeriod	The length of time that the adapter waits between polling periods.
JDBC driver class	JDBCDriverClass	The class name of the JDBC driver that is used to connect to the database.
Maximum connections	MaximumConnections	The maximum number of connections that the adapter can use for inbound event delivery.
Minimum connections	MinimumConnections	The minimum number of connections that the adapter can use for inbound event delivery.
Maximum number of retries in case of system connection failure	RetryLimit	The number of times the adapter tries to reestablish an inbound connection after an error.
Password	Password	Password for authorizing the user to retrieve events from the database.
Poll quantity	PollQuantity	The number of events the adapter delivers to the export during each poll period.
Query time out	QueryTimeOut	The maximum number of seconds a query can take for all SQL statements.
Retry EIS connection on startup	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the Oracle database if it cannot connect at startup.
Time between retries in case of system connection failure (milliseconds)	RetryInterval	The length of time that the adapter waits between attempts to reestablish connection after an error during inbound operations.
Return dummy business object for RetrieveSP	ReturnDummyBOForSP	Specifies whether to return output parameters when the result set is empty.
SQL query to verify the connection	PingQuery	The SQL query used to test the reliability of the connection to the database.
Stop the adapter when an error is encountered while polling	StopPollingOnError	Specifies whether the adapter stops polling for events when it encounters an error during polling.
Stored procedure to run after polling	SPAAfterPoll	The name of the stored procedure that you want to be run after each poll cycle
Stored procedure to run before polling	SPBeforePoll	The name of the stored procedure that you want to be run before the actual poll query is called

Table 78. Activation specification properties for Adapter for Oracle E-Business Suite (continued)

Property name		Description
In the wizard	In the administrative console	
Type of delivery	DeliveryType	Determines the order in which events are delivered by the adapter to the export.
User name	UserName	The database user name to use for inbound events
“Database connection information (ConnectionType)” on page 275	ConnectionType	Specifies how the adapter establishes connection to the database.

Adapter Instance for event filtering (AdapterInstanceEventFilter)

This property controls whether the adapter instance processes specific events in the event store.

Table 79. Adapter Instance for event filtering details

Required	No
Default	null
Property type	String
Usage	<p>This property helps you migrate from WebSphere Business Integration Adapter for Oracle Applications to WebSphere Adapter for Oracle E-Business Suite. WebSphere Business Integration Adapter for Oracle Applications allows you to perform load balancing on high-volume event types by allowing multiple adapter instances to process events of the same type. When load balancing is not required, a single adapter instance processes all events of a given type. This property is to enable seamless migration for WBIA customers to JCA for customers who are currently taking advantage of the connectorID filtering.</p> <p>WebSphere Adapter for Oracle E-Business Suite typically does not require load balancing in this way, but supports it so that you can migrate without modifying the database triggers or other mechanisms that write events to the event store.</p> <p>The AdapterInstanceEventFilter property corresponds to the ConnectorID property of the WebSphere Business Integration Adapter for Oracle Applications.</p> <p>To use this feature, the database triggers or other mechanisms that create events in the event store must assign the appropriate value to the connectorId column.</p> <p>Table 80 on page 260 shows the interaction between the AdapterInstanceEventFilter property and the value in the connectorId column in the event store.</p> <p>If the EventTypeFilter and AdapterInstanceEventFilter properties are both set, the adapter processes only events that meet both criteria. That is, it processes only those events whose type is specified in the EventTypeFilter property and whose connectorId column matches the AdapterInstanceEventFilter property.</p>
Example	See Table 80 on page 260.
Globalized	Yes
Bidi supported	Yes

Table 80. Interaction of the AdapterInstanceEventFilter property with the connectorId column in the event store

AdapterInstanceEventFilter property	connectorId column of an event	Result
null	null	The adapter processes the event.
null	Instance1	The adapter processes the event, because the connectorId column is not checked.
Instance1	Instance1	The adapter processes the event.
Instance1	Instance2	The adapter does not process the event, because the instance IDs do not match.
Instance1	null	The adapter does not process the event, because the instance IDs do not match.

Additional JDBC driver connection properties [name:value;name:value] (JDBCDriverConnectionProperties)

This property contains additional information for connecting to the database using the JDBC driver.

Table 81. Additional JDBC driver connection properties details

Row	Explanation
Required	No
Possible values	Database connection properties are database-specific.
Default	No default value
Property type	String
Usage	These connection properties are used in addition to the UserName and Password properties to customize the database connection used by the adapter. Specify the connection properties as one or more <i>name:value</i> pairs separated by the semicolon character (;).
Example	The following value of this property specifies a login timeout interval: loginTimeout:20; ConnectionRetryCount:5; ConnectionRetryDelay:5
Globalized	Yes
Bidi supported	No

Business object namespace (BusinessObjectNameSpace)

This property specifies the namespace for the business object definitions.

Table 82. Business object namespace property characteristics

Required	No
Default	http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle
Property type	String
Usage	This value is added as a prefix to the business object name to keep business object names logically separated.
Example	The following example shows the Schema1Customer business object with the default namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/oracle/Schema1Customer
Bidi supported	No

User-defined delete query (CustomDeleteQuery)

Use this property to specify the SQL statement, stored procedure, or stored function to run after each event is processed to delete records that can be deleted after the event is delivered.

Table 83. Custom delete query details

Required	No
Default	No default value
Property type	String
Usage	Use this property to specify the SQL statement, stored procedure, or stored function to run when the EventQueryType property is set to Dynamic.
Globalized	Yes
Bidi supported	Yes

User-defined update query for failed event delivery (CustomUpdateQueryForFailedEvent)

Use this property to specify the SQL statement, stored procedure, or stored function to run to update event status when an event is not delivered successfully.

Table 84. Custom update query for failed event details

Required	No
Default	No default value
Property type	String
Usage	Use this property to specify the SQL statement, stored procedure, or stored function to run when the EventQueryType property is set to Dynamic. If an event is delivered successfully, the property "User-defined update query (CustomUpdateQuery)" on page 262 is used to update event status. If an event is not delivered successfully, the property "User-defined update query for failed event delivery (CustomUpdateQueryForFailedEvent)" is used to update event status.
Globalized	Yes
Bidi supported	Yes

User-defined event query (CustomEventQuery)

Use this property to specify the SQL statement, stored procedure, or stored function to run to poll for events in custom event processing.

Table 85. Custom event query details

Required	No
Default	No default value
Property type	String
Usage	Use this property to specify the SQL statement, stored procedure, or stored function to run during each poll cycle when the EventQueryType property is set to Dynamic.

Table 85. Custom event query details (continued)

Examples	<p>In the following example, the custom event query runs an SQL statement that returns the event ID, object key, and object name of every record in the MY_EVENT_TABLE event store whose status column has the value 0:</p> <pre>select event_id, object_key, object_name from MY_EVENT_TABLE where status = 0</pre> <p>The following example limits the returned event records to the value of the PollQuantity property:</p> <pre>select event_id, object_key, object_name from MY_EVENT_TABLE where status = 0 and rownum < POLL QUANTITY</pre> <p>The following example runs a stored procedure with two parameters:</p> <pre>CALL MY_EVENT_STORED_PROC (?,?)</pre> <p>The following example runs a stored function with one parameter and one return value:</p> <pre>? = CALL MY_EVENT_FUNCTION(?)</pre>
Globalized	Yes
Bidi supported	Yes

User-defined update query (CustomUpdateQuery)

Use this property to specify the SQL statement, stored procedure, or stored function to run after each event is processed so that the same event does not get picked up for processing in the subsequent event cycle.

Table 86. Custom update query details

Required	No
Default	No default value
Property type	String
Usage	Use this property to specify the SQL statement, stored procedure, or stored function to run when the EventQueryType property is set to Dynamic.
Globalized	Yes
Bidi supported	Yes

Data source JNDI name (DataSourceJNDIName)

This property specifies the name of the JNDI data source to use to establish a connection to the database.

Table 87. Data source JNDI name details

Required	No
Default	No default value
Property type	String

Table 87. Data source JNDI name details (continued)

Usage	<p>Use this property to specify the JNDI name of a data source in IBM Business Process Manager or WebSphere Enterprise Service Bus that specifies connection information for the target database.</p> <p>To improve the performance of inbound or outbound operations, specify the name of a data source that is enabled for prepared statement caching.</p> <p>If the UserName and Password properties are also set, they override the user name and password in the data source.</p> <p>Do not confuse the data source JNDI name property with the JNDI name of a managed connection factory or activation specification on the server. The following list highlights important differences between the types of JNDI names:</p> <ul style="list-style-type: none"> • Data source JNDI name <ul style="list-style-type: none"> – Specifies a connection to a database – Is used instead of saving user name and password in adapter properties – Is saved as an adapter property • JNDI name of the managed connection factory or activation specification <ul style="list-style-type: none"> – Specifies a connection to a managed connection factory or activation specification on the server – Is used instead of specifying the value of each managed connection factory or activation specification property in the wizard – Is saved as the connection target in the import file
Globalized	Yes
Bidi supported	No

Database URL (DatabaseURL)

This property specifies the JDBC driver-specific URL for creating a connection to the database.

Table 88. Database URL details

Required	Yes, when the connection is of the type LocalConnectionProps or ConnectionProps.
Default	No default value
Property type	String
Usage	<p>In the external service wizard, compose the database URL by entering in database-specific fields. For example, the database URL for an Oracle database is composed of the system ID (SID), server host name, and database port number. In the administrative console, type the complete database URL value.</p> <p>If your database server supports IPv6, you can specify the host name portion of the database URL in IPv6 format.</p> <p>If you specify the host name as an IP address in IPv6 format, enclose the IP address in square brackets ([]).</p>

Table 88. Database URL details (continued)

Examples	The following example illustrates typical database URL values for a common database:jdbc:oracle:thin:@9.26.248.148:1521:dev
Globalized	Yes
Bidi supported	Yes

Database vendor (DatabaseVendor)

This property specifies the type of database that is used. The type is determined by the database vendor name.

Table 89. Database vendor details

Required	Yes
Possible values	Oracle
Default	ORACLE
Property type	String
Usage	Some SQL statements require special processing, which varies by database type. For example, the Struct and Array data types in Oracle require special processing. This property specifies the RDBMS that is used, which determines the database type. For other databases, the adapter does not perform any special processing. Make sure that the correct driver is specified in the JDBCDriverClass property. Specify Oracle as the value that corresponds to your database vendor.
Globalized	No
Bidi supported	No

Delivery type (DeliveryType)

This property specifies the order in which events are delivered by the adapter to the export.

Table 90. Delivery type details

Required	No
Possible values	ORDERED UNORDERED
Default	ORDERED
Property type	String
Usage	The following values are supported: <ul style="list-style-type: none"> • ORDERED: The adapter delivers events to the export one at a time. • UNORDERED: The adapter delivers all events to the export at once.
Globalized	No
Bidi supported	No

Do not process events that have a timestamp in the future (FilterFutureEvents)

This property specifies whether the adapter filters out future events by comparing the timestamp on each event with the system time.

Table 91. Do not process events that have a timestamp in the future details

Required	Yes
Possible values	True False
Default	False
Property type	Boolean
Usage	If set to True, the adapter compares the time of each event to the system time. If the event time is later than the system time, the event is not delivered. If set to False, the adapter delivers all events.
Globalized	No
Bidi supported	No

Ensure once-only event delivery (AssuredOnceDelivery)

This property specifies whether to provide ensure once-only event delivery for inbound events.

Table 92. Ensure once-only event delivery details

Required	Yes
Possible values	True False
Default	True
Property type	Boolean
Usage	When this property is set to True, the adapter provides assured once event delivery. This means that each event will be delivered once and only once. A value of False does not provide assured once event delivery, but provides better performance. When this property is set to True, the adapter attempts to store transaction (XID) information in the event store. If it is set to False, the adapter does not attempt to store the information. This property is used only if the export component is transactional. If it is not, no transaction can be used, regardless of the value of this property.
Globalized	No
Bidi supported	No

Event order by (EventOrderBy)

The order in which events are retrieved and processed.

Table 93. Event order by details

Required	No
Possible values	A comma-separated (,) list of column names in the event store, and the order attributes asc and desc

Table 93. Event order by details (continued)

Default	event_time, event_priority
Property type	String
Usage	Specify a comma-separated list of column names from the event store, with the optional attributes for ascending or descending order.
Examples	To present events ordered first by time and then by priority, specify: event_time, event_priority To present events ordered first by object name in ascending order and then by event time in descending order, specify: object_name asc, event_time desc
Globalized	Yes
Bidi supported	Yes

Event query type (EventQueryType)

This property specifies whether to use standard or custom query processing.

Table 94. Event query type details

Required	Yes
Possible values	Standard Dynamic
Default	Standard
Property type	String
Usage	The valid values are Standard, for standard event processing, and Dynamic, for custom event processing. If this property is set to Dynamic, the CustomEventQuery, CustomUpdateQuery, and CustomDeleteQuery properties are used. If this property is set to Standard, those properties are ignored.
Globalized	No
Bidi supported	No

Event table name (EventTableName)

This property specifies the name of the table in the target database that contains the event store, which is used for inbound processing.

Table 95. Event table name details

Required	Yes
Default	WBIA_Oracle_EventStore
Property type	String
Usage	Create the event store before starting to configure the adapter. For standard event processing, the event is generated by the database through a trigger or other mechanism. For custom query processing, the adapter saves events in the event store as it receives the result of the custom queries.
Globalized	Yes
Bidi supported	Yes

Event types to process (EventTypeFilter)

This property contains a delimited list of event types that indicates to the adapter which events it should deliver.

Table 96. Event types to process details

Required	No
Possible values	A comma-delimited (,) list of business object types
Default	null
Property type	String
Usage	<p>Events are filtered by business object type and operations. If the property is set, the adapter delivers only those events that are in the list. A value of null or * indicates that no filter will be applied and that all events will be delivered to the export.</p> <p>When the default operation set (Create/Update/Delete) is not preferred, the adapter will provide an operation-based event polling capability.</p> <p>For example:</p> <p>If you select Create Update operation, the adapter will poll only those events with Create or Update operation for all the BOs. The default string generated for eventTypeFilter is *: Create Update.</p> <p>Syntax: BOName:Operation1 Operation2, BOName:Operation1 Operation2 Operation3</p> <ul style="list-style-type: none"> • "," is used for separating the business objects. The objects that are not in this list are ignored. • ":" is used for separating the business object name and the operation name. • " " is used for separating the supported operations, such as Create, Delete, and Update.
Example	<p>To receive events related to the Customer and Order business objects only, specify this value: Customer,Order</p> <ul style="list-style-type: none"> • To receive events related to the Customer and Order business objects regardless of operations, specify this value: Customer,Order or specify this value for all supported operations: Customer:Create Update Delete,Order:Create Update Delete. • To receive all events of the Customer business object and the Create and Delete events of the Order business object, specify this value: Customer,Order:Create Delete • To receive the Create and Delete events of all the business objects, specify this value: *:Create Delete. <p>Note: In this property, you can use only those operations that are chosen while selecting business objects and services.</p> <p>If the EventTypeFilter and AdapterInstanceEventFilter properties are both set, the adapter processes only events that meet both criteria. That is, it processes only those events whose type is specified in the EventTypeFilter property and whose connectorId column matches the AdapterInstanceEventFilter property.</p>
Globalized	No
Bidi supported	No

Retry limit for failed events (FailedEventRetryLimit)

This property specifies the number of times that the adapter attempts to redeliver an event before marking the event as failed.

Table 97. Retry limit for failed events details

Required	No
Possible values	Integers
Default	5
Property type	Integer
Usage	<p>Use this property to control how many times the adapter tries to send an event before marking it as failed. It accepts the following values:</p> <p>Default</p> <p>If this property is not set, the adapter tries five additional times before marking the event as failed.</p> <p>0</p> <p>The adapter tries to deliver the event an infinite number of times. When the property is set to 0, the event remains in the event store and the event is never marked as failed.</p> <p>> 0</p> <p>For integers greater than zero, the adapter retries the specified number of times before marking the event as failed.</p> <p>< 0</p> <p>For negative integers, the adapter does not retry failed events.</p>
Globalized	No
Bidi supported	No

JDBC driver class (JDBCDriverClass)

This property specifies the class name of the JDBC driver that is used to connect to the database.

Table 98. JDBC driver class details

Row	Explanation
Required	Yes, when the connection is of the type LocalConnectionProps or ConnectionProps.
Possible values	The value specified when you set the connection properties to connect to the database instance.
Default	No default value
Property type	String
Usage	<p>In the external service wizard, the JDBC driver class is displayed if you select a common database software and driver combination, such as type 4 drivers for recent versions of Oracle. For type 2 drivers, you must type the database class name.</p> <p>If you select a type 2 driver or a generic driver, you must type the JDBC driver class name.</p> <p>In the administrative console, type the database-specific name of the driver.</p>

Table 98. JDBC driver class details (continued)

Examples	<p>Values for the JDBC driver class is displayed in both the external service wizard and the administrative console. The following examples depict the JDBC driver class properties in both the external service wizard and the administrative console. In the external service wizard:</p> <ul style="list-style-type: none"> To connect to an Oracle 10 database using the type 4 driver, select Oracle Thin Driver. <p>In the administrative console:</p> <p>Oracle Thin JDBC driver oracle.jdbc.driver.OracleDriver</p>
Globalized	No
Bidi supported	No

Maximum connections (MaximumConnections)

This property specifies the maximum number of connections that the adapter can use for inbound event delivery.

Table 99. Maximum connections details

Required	No
Default	1
Property type	Integer
Usage	Only positive values are valid. Any value less than 1 is treated as 1 by the adapter.
Globalized	No
Bidi supported	No

Minimum connections (MinimumConnections)

This property specifies the minimum number of connections that the adapter can use for inbound event delivery.

Table 100. Minimum connections details

Required	No
Default	1
Property type	Integer
Usage	Only positive values are valid. Any value less than 1 is treated as 1 by the adapter.
Globalized	No
Bidi supported	No

Password (Password)

This property specifies the password for the database user name.

Table 101. Password details

Required	<p>No. For inbound processing, if you set the Authentication alias or the DataSourceJNDIName, the password is not mandatory. However, if you set the DataSourceJNDIName, and the Password field, the value specified for the Password takes precedence.</p> <p>For outbound processing, if you set the Authentication alias, XADataSourceJNDIName or PoolDataSourceJNDIName property, the password is not mandatory. However, if you set the XADataSourceJNDIName or PoolDataSourceJNDIName property, and the Password field, the value specified for the Password takes precedence.</p>
Default	No default value
Property type	String
Usage	<p>For inbound processing, setting this property overrides the password specified for the data source on the server using the Authentication alias or DataSourceJNDIName property.</p> <p>For outbound processing, setting this property overrides the password specified for the data source on the server using the Authentication alias, or XADataSourceJNDIName or PoolDataSourceJNDIName property.</p> <p>If you specify JAAS as the security credential, the authentication alias will override this property.</p>
Globalized	Yes
Bidi supported	Yes

SQL query to verify the connection (PingQuery)

This property specifies the SQL query that is used to test the reliability of the connection to the database.

Table 102. Ping query details

Required	No
Property type	String
Default	No default value
Usage	<p>This property contains the SQL query statement that you want to run to determine whether the adapter can connect to the database.</p> <p>The adapter runs the ping query every time it receives a SQLException exception while performing the outbound operation.</p> <p>The adapter does not try to recover the connection. If the ping query indicates that the connection to the database is no longer valid, the adapter notifies the container. It is the responsibility of the connection pool manager to remove the stale connection from the pool, which allows subsequent outbound requests to be processed.</p>

Table 102. Ping query details (continued)

Globalized	No
Bidi supported	No

Interval between polling periods (PollPeriod)

This property specifies the length of time that the adapter waits between polling periods.

Table 103. Interval between polling periods details

Required	Yes
Possible values	Integers greater than or equal to 0.
Default	2000
Unit of measure	Milliseconds
Property type	Integer
Usage	The poll period is established at a fixed rate, which means that if running the poll cycle is delayed for any reason (for example, if a prior poll cycle takes longer than expected to complete) the next poll cycle will occur immediately to make up for the lost time caused by the delay.
Globalized	No
Bidi supported	No

Maximum events in polling period (PollQuantity)

This property specifies the number of events that the adapter delivers to the export during each poll period.

Table 104. Maximum events in polling period details

Required	Yes
Default	10
Property type	Integer
Usage	The value must be greater than 0. If this value is increased, more events are processed per polling period and the adapter may perform less efficiently. If this value is decreased, fewer events are processed per polling period and the adapter's performance might improve slightly.
Globalized	No
Bidi supported	No

Query timeout (seconds) (QueryTimeOut)

This property specifies the maximum number of seconds a query can take to run all SQL statements.

Table 105. Query timeout details

Required	No
Default	No default value
Unit of measure	Seconds
Property type	Integer

Table 105. Query timeout details (continued)

Usage	If the query takes longer than the number of seconds specified, the database generates an SQL exception that is captured. The associated message is logged in the log file. If a value is not specified, no timeout is set on the query.
Globalized	No
Bidi supported	No

Retry interval if connection fails (RetryInterval)

When the adapter encounters an error related to the inbound connection, this property specifies the length of time the adapter waits before trying to reestablish a connection.

Table 106. Retry interval details

Required	Yes
Default	2000
Unit of measure	Milliseconds
Property type	Integer
Usage	Only positive values are valid. When the adapter encounters an error related to the inbound connection, this property specifies the length of time the adapter waits before trying to establish a new connection.
Globalized	No
Bidi supported	No

Number of times to retry the system connection (RetryLimit)

This property specifies the number of times the adapter tries to reestablish an inbound connection.

Table 107. Number of times to retry the system connection details

Required	No
Possible values	0 and positive integers
Default	0
Property type	Integer
Usage	This property controls how many times the adapter retries the connection if the adapter cannot connect to the Oracle database to perform inbound processing. A value of 0 indicates an infinite number of retries. To control whether the adapter retries if it cannot connect to the Oracle database when it is first started, use the RetryConnectionOnStartup property.
Globalized	No
Bidi supported	No

Retry EIS connection on startup (RetryConnectionOnStartup)

This property controls whether the adapter attempts to connect again to the Oracle database if it cannot connect at startup.

Table 108. Retry EIS connection on startup details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>This property indicates whether the adapter should retry the connection to the Oracle database if the connection cannot be made when the adapter is started:</p> <ul style="list-style-type: none"> • Set the property to False when you want immediate feedback about whether the adapter can establish a connection to the Oracle database, for example, when you are building and testing the application that receives events from the adapter. If the adapter cannot connect, the adapter writes log and trace information and stops. The administrative console shows the application status as Stopped. After you resolve the connection problem, start the adapter manually. • Set the property to True if you do not need immediate feedback about the connection. If the adapter cannot connect during startup, it writes log and trace information, and then attempts to reconnect, using the RetryInterval property to determine how frequently to retry and the value of the RetryLimit property to retry multiple times until that value is reached. The administrative console shows the application status as Started.
Globalized	No
Bidi supported	No

Return business object even when the stored procedure result set is empty (ReturnDummyBOForSP)

This property specifies whether to return output parameters when the result set is empty.

Table 109. Return business object even when the stored procedure result set is empty details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>The Retrieve Stored Procedure (RetrieveSP) operation returns a result set. If the result set is empty and the ReturnDummyBOForSP property is set to False, no business objects are created, and the output parameters returned by the procedure call cannot be retrieved.</p> <p>However, if the ReturnDummyBOForSP property is set to True, a dummy business object is created and the returned parameters (including the out and inout parameters) from the stored procedure are populated into the corresponding attributes</p>
Globalized	No
Bidi supported	No

Stop the adapter when an error is encountered while polling (StopPollingOnError)

This property specifies whether the adapter will stop polling for events when it encounters an error during polling.

Table 110. Stop the adapter when an error is encountered while polling details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If this property is set to True, the adapter stops polling when it encounters an error. If this property is set to False, the adapter logs an exception when it encounters an error during polling and continues polling.
Globalized	No
Bidi supported	No

Stored procedure to run after polling (SPAfterPoll)

This property specifies the name of the stored procedure or stored function to run after each polling cycle.

Table 111. Stored procedure to run after poll details

Required	No
Default	No default value
Property type	String
Usage	The stored procedure takes one parameter for poll quantity.
Globalized	Yes
Bidi supported	Yes

Stored procedure to run before polling (SPBeforePoll)

This property specifies the name of any stored procedure or stored function to run before the actual poll query is called.

Table 112. Stored procedure to run before poll details

Required	No
Default	No default value
Property type	String
Usage	The stored procedure takes one parameter for poll quantity.
Globalized	Yes
Bidi supported	Yes

User name (UserName)

This property specifies the database user name that is used to access the database.

Table 113. User name details

Required	No. For inbound processing, if you set the Authentication alias or the DataSourceJNDIName, the User name property is not mandatory. However, if you set the DataSourceJNDIName, and User name field, the value specified for the User name takes precedence. For outbound processing, if you set the Authentication alias, XADataSourceJNDIName or PoolDataSourceJNDIName property, the User name is not mandatory. However, if you set the XADataSourceJNDIName or PoolDataSourceJNDIName property, and the User name field, the value specified for the User name takes precedence.
Default	No default value
Property type	String
Usage	For inbound processing setting this property overrides the user name specified for the data source on the server using the DataSourceJNDIName property or the Authentication alias. For outbound processing, setting this property overrides the user name specified for the data source on the server using the XADataSourceJNDIName property, PoolDataSourceJNDIName property or the Authentication alias. If you specify JAAS as the security credential, the authentication alias will override this property.
Globalized	Yes
Bidi supported	Yes

Database connection information (ConnectionType)

This property specifies how the adapter establishes connection to the database.

Table 114. Database connection information

Required	Yes
Possible values	ConnectionProps or DataSourceJNDI
Default	ConnectionProps
Property type	String

Table 114. Database connection information (continued)

Usage	<p>This property specifies how adapter establish database connection at run time. This property can have the following values:</p> <ul style="list-style-type: none"> • ConnectionProps - Indicates that the database connection is established using the DatabaseURL and JDBCClass properties. • DataSourceJNDI - Indicates that the database connection is established using the DataSourceJNDIName property corresponding to predefined datasource. <p>For a new application, this property is automatically set by external service wizard. For a migrated application, this property is set during the migration process according to the ActivationSpecification properties.</p> <ul style="list-style-type: none"> • If the DataSourceJNDIName property is set, the value of this property is set to DataSourceJNDI. • If the DataSourceName is set, the value of this property is set to ConnectionProps. <p>If this property is not set, adapter uses backward compatibility mode to establish database connection. In backward compatibility mode, the properties for connecting to the database are used in the following order:</p> <ol style="list-style-type: none"> 1. If the DataSourceJNDIName property is set, the adapter uses it to establish the connection to the database. 2. If the DataSourceJNDIName is not set, then the adapter uses the DatabaseURL, JDBCClass, UserName and Password properties to establish the connection.
Globalized	No
Bidi supported	No

Globalization

WebSphere Adapter for Oracle E-Business Suite is a globalized application that can be used in multiple linguistic and cultural environments. Based on character set support and the locale of the host server, the adapter delivers message text in the appropriate language. The adapter supports bidirectional script data transformation between integration components.

Globalization and bidirectional transformation

The adapter is globalized to support single and multibyte character sets and deliver message text in the specified language. The adapter also performs bidirectional transformation, which refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.

Globalization

Globalized software applications are designed and developed for use within multiple linguistic and cultural environments rather than a single environment. WebSphere Adapters, IBM Integration Designer, IBM Business Process Manager or WebSphere Enterprise Service Bus are written in Java. The Java run time environment within the Java virtual machine (JVM) represents data in the Unicode character code set. Unicode contains encoding for characters in most known

character code sets (both single- and multi-byte). Therefore, when data is transferred between these integration system components, there is no need for character conversion.

To log error and informational messages in the appropriate language and for the appropriate country or region, the adapter uses the locale of the system on which it is running.

Bidirectional transformation

Languages such as Arabic and Hebrew are written from right to left, yet they contain embedded segments of text that are written left to right, resulting in bidirectional script. There are multiple ways that a software application might display and process bidirectional script. IBM Business Process Manager or WebSphere Enterprise Service Bus uses the Windows standard format, but an enterprise information system exchanging data with IBM Business Process Manager or WebSphere Enterprise Service Bus can use a different format. WebSphere Adapters transform bidirectional script data is passed between the two systems so that it is accurately processed and displayed on both sides of a transaction.

Bidirectional format

IBM Business Process Manager or WebSphere Enterprise Service Bus uses the bidirectional format of ILYNN (implicit, left-to-right, on, off, nominal). This is the format used by Windows. If an enterprise information system uses a different format, the adapter converts the format prior to introducing the data to IBM Business Process Manager or WebSphere Enterprise Service Bus.

The bidirectional format consists of five attributes. When you set bidirectional properties, you assign values for each of these attributes. The attributes and settings are listed in the following table.

Table 115. Bidirectional format attributes

Letter position	Purpose	Values	Description	Default setting
1	Order schema	I	Implicit (Logical)	I
		V	Visual	
2	Direction	L	Left-to-Right	L
		R	Right-to-Left	
		C	Contextual Left-to-Right	
		D	Contextual Right-to-Left	
3	Symmetric Swapping	Y	Symmetric swapping is on	Y
		N	Symmetric swapping is off	
4	Text Shaping	S	Text is shaped	N
		N	Text is not shaped (Nominal)	
		I	Initial shaping	
		M	Middle shaping	
		F	Final shaping	
		B	Isolated shaping	

Table 115. Bidirectional format attributes (continued)

Letter position	Purpose	Values	Description	Default setting
5	Numeric Shaping	H	National (Hindi)	N
		C	Contextual shaping	
		N	Numbers are not shaped (Nominal)	

The adapter transforms data into a logical, left-to-right format before sending the data to IBM Business Process Manager or WebSphere Enterprise Service Bus.

Using bidirectional properties

You can use multiple bidirectional properties to control the transformation of both content data and metadata. You can set special bidirectional properties to exclude either content data or metadata from bidirectional transformation, or to identify data that requires special treatment during a transformation.

The following table describes the types of bidirectional properties.

Table 116. Bidirectional property types

Property type	Data transformations
EIS	Controls the format for content data, or data that is sent by the enterprise information system, which is, the database.
Metadata	Controls the format for metadata, or data that provides information about the content data.
Skip	Identifies content or metadata to exclude from transformation.
Special Format	Identifies certain text, such as file paths or URLs, which require different treatment during the transformation process. Can be set for either content data or metadata.

You can set properties that control bidirectional transformation in the following areas:

- **Resource adapter properties:** These properties store default configuration settings, including the TurnBiDiOff property, which controls whether the adapter instance performs bidirectional transformation or not. Use the administrative console of the server to configure these properties.
- **Managed connection factory properties:** These properties are used at run time to create an outbound connection instance with an enterprise information system. After the managed connection factory properties are created, they are stored in the deployment descriptor.
- **Activation specification properties:** These properties hold the inbound event processing configuration information for a message endpoint. Set them when you use the external service wizard, or use the administrative console of the server.

Property scope and lookup mechanism

After you set values for bidirectional properties for an adapter, the adapter performs bidirectional transformations. It does so by using logic that relies on a hierarchical inheritance of property settings and a lookup mechanism.

Properties defined within the resource adapter are at the top of the hierarchy, while those defined within other areas or annotated within a business object are at lower levels of the hierarchy. So for example, if you set values for EIS-type bidirectional properties only for the resource adapter, those values are inherited and used by transformations that require a defined EIS-type bidirectional property, whether they arise from an inbound (activation specification) transaction or an outbound (managed connection factory) transaction.

However, if you set values for EIS-type bidirectional properties for both the resource adapter and the activation specification, a transformation arising from an inbound transaction uses the values set for the activation specification.

The processing logic uses a lookup mechanism to search for bidirectional property values to use during a transformation. The lookup mechanism begins its search at the level where the transformation arises and searches upward through the hierarchy for defined values of the appropriate property type. It uses the first valid value it finds. It searches the hierarchy from child to parent only; siblings are not considered in the search.

Properties enabled for bidirectional data transformation

WebSphere Adapter for Oracle E-Business Suite has several configuration properties that are enabled for bidirectional data transformation.

The adapter enables the exchange of bidirectional data between a client application and the database, even if the data in the database is in a different bidirectional format than is used by the runtime environment. You can use bidirectional characters when configuring the adapter and in the application-specific information of your business objects. The following sets of properties and application-specific information are enabled for bidirectional support:

- Configuration properties
 - Activation specification properties
 - Connection properties for the external service wizard
 - Managed connection factory properties
- Application-specific information
 - Business object level ASI
 - Operation level ASI
 - Attribute level ASI

The sections which follow list the specific configuration properties and application-specific information that are enabled for bidirectional transformation.

Activation specification properties

The following activation specification properties are enabled for bidirectional script data transformation:

- Custom delete query
- Custom event query
- Custom update query

- Additional JDBC driver connection properties
- Database URL
- Event order by
- Event table name
- Password
- Stored procedure to run before polling
- Stored procedure to run after polling
- User name

Connection properties used in the wizard

The following connection properties for the external service wizard are enabled for bidirectional script data transformation:

- User name
- Password

Managed connection factory properties

The following managed connection properties are enabled for bidirectional script data transformation:

- Additional JDBC driver connection properties
- Database URL
- Password
- User name

Business object application-specific information

The following business object application-specific information parameters are enabled for bidirectional script data transformation:

- TableName
- StatusColumnName
- SPName
- SelectStatement

Operation application-specific information

The following operation application-specific information parameters are enabled for bidirectional script data transformation:

- StoredProcedureName
- PropertyName in Parameters

Attribute application-specific information

The following attribute application-specific information parameters are enabled for bidirectional script data transformation:

- ColumnName

Adapter messages

View the messages issued by WebSphere Adapter for Oracle E-Business Suite at the following location.

Link to messages: http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.ref.doc/topics/welc_ref_msg_wbpm.html

The displayed Web page shows a list of message prefixes. Click a message prefix to see all the messages with that prefix:

- Messages with the prefix CWYOE are issued by WebSphere Adapter for Oracle E-Business Suite
- Messages with the prefix CWYDB are issued by the common component shared by WebSphere Adapter for JDBC and WebSphere Adapter for Oracle E-Business Suite
- Messages with the prefix CWYBS are issued by the adapter foundation classes, which are used by all the adapters

Related information

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

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

- IBM Business Process Manager, version 7.5, information center, which includes IBM Business Process Manager, IBM WebSphere Enterprise Service Bus, and IBM Integration Designer information: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/index.jsp>.
- IBM Business Process Manager, version 7.0, information center, which includes IBM Business Process Manager, IBM WebSphere Enterprise Service Bus, and IBM Integration Designer information: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp>.
- WebSphere Adapters, version 6.2.x, information center: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp>.
- IBM WebSphere Adapters, version 7.5 installation on WebSphere Application Server, version 8.0 information: <http://www-01.ibm.com/support/docview.wss?rs=695&uid=swg27011040>.

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Video samples

To help you use WebSphere Adapters, sample videos are available to detail some of the scenarios.

You can find several Adapter video demos at this FTP location
<ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/>.

To view these video samples:

1. Download the zip file of the required sample from the FTP location.
2. Extract the downloaded zip file to your system.
3. From the extracted folder, double-click the HTML file.
4. The video sample will open in your default Web browser.

Note: The video has a built in audio narration. If you do not want to listen to the audio, you can lower its tab to the minimum, and slide the tab to an audio level, whenever you want the audio back.

The following demos are currently available at this location: <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/>

- AFC: Scheduling Calendar based polling
- AFC: Overcoming an AFC version conflict situation
- Lotus® Domino®: Configuring notification rules for forms in database paths, and generating their events
- IBM i: Supporting CCSID in the RPG, COBOL, and service programs
- IBM i: Discovering the PCMLS in a local file system
- JDE: Retrieving data using multiple business functions in outbound processing
- Oracle: Integrating support for an Oracle AQ activity
- Oracle: Integrating an XML gateway through JMS in Oracle E-Business Suite
- PeopleSoft: Handling a RecordNotFound fault behavior
- PeopleSoft: Handling an event filtering
- SAP: Monitoring Tivoli ITCAM for a BPM Portfolio requirement
- SAP: Adding random numbers to the end of duplicate field names
- Siebel: Adding codepage properties to both EMD and runtime for an inbound Business Object scenario
- Siebel: Supporting more than one output argument for a Siebel Business Services Interface in an outbound scenario
- WAT: Creating a custom adapter using WebSphere Adapter Toolkit
- WAT: Creating a custom adapter outbound interface in WebSphere Adapter Toolkit
- WAT: Testing a module in WebSphere Adapter Toolkit

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, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

- activation specification properties
 - list of 256
 - setting in administrative console 179, 184
- Active-Passive 60
- adapter 77
 - configure 77
 - information resources 281
 - messages 281
 - package files 194
 - performance 186
 - project, create 81
 - related information 281
 - samples and tutorials 281
 - support and assistance 281
 - technotes 281
- adapter application
 - starting 185
 - stopping 185
- Adapter for Oracle E-Business Suite module
 - exporting as EAR file 172
 - installing EAR file on server 173
 - starting 185
 - stopping 185
- adapter implementation
 - security 53
- after-image 5
- application-specific information 217
 - adding to object 103, 134
 - for attributes of type child business object 215
 - for simple attributes 209
- ApplyChanges operation 14
- artifacts 70
- assured once delivery 21
- attribute properties 207
- attribute type, business object 208
- authentication 77
 - description 54
 - in the wizard 54
 - runtime environment 54
- authentication alias 77, 80
 - J2C 54

B

- Batch Processing 60
- business faults 47, 221
- business graph 4
- business integration adapters to JCA-compliant adapters 67
- business object 77
- business object information 207
- business object structure
 - for query business objects 30
 - for stored procedure business objects 29
 - for table or view business objects 26

- business objects 26, 217
 - assembly editor 131, 156
 - attribute types 208
 - attributes 207
 - cardinality 33
 - composite keys 131, 156
 - how to view 104, 135
 - multiple parents 131, 156
 - naming conventions 220
 - query 46
 - stored procedure 39

C

- cardinality 33, 207
- CEI (Common Event Infrastructure) 190
- changes after migration
 - to the export file 72
 - to the import file 72
 - to the wsdl file 72
- CL (control language) 77
- clustered environment
 - adapters version conflict 60
 - deployment 60
 - inbound process 60
 - inbound processes 60
 - load balancing 60
 - outbound process 60
 - outbound processes 61
- Common Event Infrastructure (CEI) 190
- compatibility matrix 1
- complex data types 39
- confidential data, disguising 53
- confidential tracing 53
- configuration overview 78
- configuration properties
 - inbound 250
- configuring
 - logging properties 194
 - Performance Monitoring Infrastructure (PMI) 187
- configuring JDBC adapter for XA 126, 127
- connector project 81
- Create operation 7
- custom properties
 - activation specification 179, 184
 - managed connection factory 177, 182
 - resource adapter 175, 181
- custom queries
 - standard SQL 23
 - stored function 23
 - stored procedure 23

D

- data types
 - complex 39
- debugging
 - self-help resources 198

- Delete operation 15
- delta 5
- deployment 173
 - production environment 170
 - test environment 165
- deployment environment 77, 165
- deployment options 56
- developerWorks 281
- disguising confidential data 53
- distributed transactions, see XA transactions 239

E

- EAR file
 - exporting 172
 - installing on server 173
- edit binding
 - import component 159
- embedded adapter
 - activation specification properties, setting 179
 - considerations for using 57
 - managed connection factory properties, setting 177
 - resource adapter properties, setting 175
 - usage considerations 56
- embedded adapters
 - changing configuration properties 175
 - setting activation specification properties 179
 - setting managed (J2C) connection factory properties 177
 - setting resource adapter properties 175
- embedded deployment 166
- enableHASupport property 60
- endpoint applicaiton
 - troubleshoot 197
- Enqueue operation 20
- event delivery 264
- event processing
 - custom 4, 22
 - standard 4, 21
- event store 24, 25
- event store set up 79
- Execute operation 16
- Exist operation
 - limitations 17
- Exists operation 17
 - for database table business objects 17
 - for database views business objects 17
 - for nicknames business objects 17
 - for synonyms business objects 17
- export file 77
- exporting module as EAR file 172
- external dependencies, adding 165, 166

- external service connection
 - properties 225, 252
- external service discovery
 - attribute information 207
- external service wizard
 - connection properties 83
 - start up 83

F

- faults
 - description 47, 221
- FFDC (first-failure data capture) 196
- files
 - SystemOut.log log file 195
 - trace.log trace file 195
- first-failure data capture (FFDC) 196
- flat business objects 33
- foreign key 207

G

- global transaction 126, 127
 - with XA DataSource JNDI name 127
 - with XADataSourceName and XADatabaseName 127

H

- HA Active-Active 60
- hardware and software requirements 1
- hardware requirements 1
- hierarchical business objects 33
- High Availability (HA)
 - clustered environments 281
- high-availability environment 60
 - Active-Active 60
 - Active-Passive 60
 - deployment 60
 - inbound processes 60
 - outbound processes 61

I

- IBM Business Process Manager
 - information 281
- IBM Business Process Manager or WebSphere Enterprise Service Bus
 - deploying to 170
- IBM Business Process Manager, version 7.0, information 281
- IBM Integration Designer
 - information 281
 - test environment 165
- IBM WebSphere Adapter for Oracle E-Business Suite
 - administering 175
- IBM WebSphere Adapter Toolkit 281
 - developerWorks resources 281
- IBM WebSphere Enterprise Service Bus
 - information 281
- implementation, Java 167
- import component 159
- import file 77

- inbound
 - configuration properties 250
- inbound processing 3, 77
- installing EAR file 173
- interaction specification properties
 - changing 163
- interface 77
- interfaces business object 48
- Invoke operation 19
- iterative development
 - connection-based editing 159
 - Edit Binding 159
 - import component 159

J

- J2C connection factory
 - see managed connection factory 230
- JAR file, adding external 165, 166
- Java 2 security 53, 54
- Java implementation 167
- JDBC driver files 82
- JRE 1
- JRE version 1

L

- load balancing 60
- local connection
 - pool datasource 128
 - with Database URL 128
- local transactions 6
- Log Analyzer 194
- log and trace
 - configure 193
- Log and Trace Analyzer, support for 193
- log and trace files 193
- log files
 - changing file name 195
 - disabling 194
 - enabling 194
 - level of detail 194
 - location 196
 - SystemOut.log 195
- logging
 - configuring properties with administrative console 194
- logging level 194

M

- managed (J2C) connection factory
 - properties
 - setting in administrative console 177, 182
- managed connection factory properties
 - details 230
 - XA data source name 239
 - XADataSourceName 239
- matrix, compatibility 1
- messages, adapter 281
- Metadata selection properties
 - how to specify (inbound) 145
 - how to specify (outbound) 122
- migration 67
 - adapter-specific artifacts 70

- migration (*continued*)
 - artifacts
 - adapter-specific 70
 - considerations
 - compatibility with earlier versions 62
 - migrating applications from WebSphere InterChange Server
 - roadmap 67
 - overview 67
 - performing migration 64
 - upgrading a project 66
 - WebSphere InterChange Server
 - migration
 - migration roadmap 67
 - WebSphere InterChange Server migration wizard 69
- modifying adapter application-specific information 131, 156
- module 77
 - adding to the server 168
 - configuring for deployment
 - overview 77
 - configuring inbound processing 133
 - configuring outbound processing 101
 - deploy for testing 165
 - monitoring performance 186
 - multiple connection 264

N

- naming conventions for business
 - objects 220
- NULL objects
 - retrieving 12

O

- operations
 - ApplyChanges 14
 - Create 7
 - Delete 15
 - Enqueue 20
 - Execute 16
 - Exists 17
 - Invoke 19
 - Publish 20
 - Retrieve 8
 - RetrieveAll 9
 - Update 12
- outbound configuration properties 224
- outbound operations
 - list of 5
- outbound processing 2, 77
 - testing the module 169

P

- package files for adapters 194
- passive adapter 197
- performance
 - prepared statement caching 61
- performance monitoring
 - infrastructure 186
 - configuring 187
 - performance statistics 188

- Performance Monitoring Infrastructure (PMI)
 - configuring 187
 - description 186
 - viewing performance statistics 188
- performance statistics 188
- PMI (Performance Monitoring Infrastructure)
 - configuring 187
 - viewing performance statistics 188
- polling 24
- PoolDataSourceJNDIName 6
- prepared statement caching 61, 243, 244
- primary key 207
- problem determination
 - self-help resources 198
 - solutions to common problems 199
- project 77
- project interchange (PI) file
 - project interchange files 66
 - projects 66
 - updating without migrating 66
- properties
 - activation specification 179, 184
 - list of 256
 - configuration properties
 - inbound 250
 - outbound 224
 - external service connection 225, 252
 - inbound configuration 250
 - managed (J2C) connection
 - factory 177, 182
 - outbound configuration 224
 - resource adapter 175, 181
- properties information
 - guide 224, 250
- Publish operation 20

Q

- query business object
 - generate from SELECT statement 46
 - structure 30

R

- RAR (resource adapter archive) file
 - description 171
 - installing on server 171
- recommended fixes 198
- RecordNotFoundException 10
- Redbooks, IBM WebSphere
 - Adapters 281
- related products, information 281
- requirements
 - hardware 1
 - software 1
- resource adapter archive (RAR) file
 - description 171
 - installing on server 171
- resource adapter properties
 - details 225, 252
 - setting in administrative console 175, 181
- Retrieve operation 8

- RetrieveAll operation
 - for database table business objects 9
 - for user-specified query business objects 11
- Retry limit property 272
- roadmap
 - configuring the module 77
- roadmap for migrating
 - WebSphere InterChange Server applications 67
- runtime environment 1
 - deploying EAR file 170

S

- samples 75, 281
- security
 - disguising sensitive data 53
 - user authentication 54
- security features
 - adapter 53
 - Java 2 security 53
- security, Java 2 54
- self-help resources 198
- sensitive data, disguising 53
- service interface queue 79
- software dependencies 82
- software dependencies, adding
 - external 165, 166
- software requirements 1
- stand-alone adapter 184
 - considerations for using 58
 - managed connection factory
 - properties, setting 182
 - resource adapter properties, setting 181
 - usage considerations 56
- stand-alone adapters
 - changing configuration
 - properties 181
 - setting activation specification
 - properties 184
 - setting managed (J2C) connection
 - factory properties 182
 - setting resource adapter
 - properties 181
- starting adapter applications 185
- stopping adapter applications 185
- stored functions
 - overview 46
- stored procedure 16
 - business object structure 29
 - definition 38
 - screen showing definition 45
- stored procedure business object 39
- stored procedures
 - overview 38
 - SQL statements 38
- structure of business objects 26
- support
 - overview 193
 - plug-in for IBM support
 - assistant 198
 - self-help resources 198
 - web site 198
- SystemOut.log file 195

T

- table
 - business object structure 26
- target component 167
- technotes 1, 198
- test environment 165
 - adding module to 168
 - deploying to 168
 - testing modules 169
- trace files
 - changing file name 195
 - disabling 194
 - enabling 194
 - level of detail 194
 - location 196
 - trace.log 195
- tracing
 - configuring properties with
 - administrative console 194
- transactions 5
 - with XADataSourceJNDIName 6
- transactions, see also XA transactions and
 - local transactions 5
- triggers on user tables 79
- troubleshooting
 - overview 193
 - self-help resources 198
- tutorials 75, 281

U

- UDF, see user-defined function 38, 46
- UNORDERED 264
- Update operation 12
- user-defined function 38, 46

V

- view
 - business object structure 26

W

- WebSphere Adapters, version 6.0,
 - information 281
- WebSphere Adapters, version 6.2.x,
 - information 281
- WebSphere Application Server
 - information 281
- WebSphere business integration
 - adapters 67
- WebSphere Business Integration Adapters
 - information 281
- WebSphere Extended Deployment 60
- wiring components 167

X

- XA transaction 126
 - with XA DataSource JNDI name 127
 - with XADataSourceName and
 - XADatabaseName 127
- XA transactions 6
 - Oracle databases 6
 - XA data source name 239

XML Gateway
interfaces 48



Printed in USA