WebSphere® Adapters
Version 7 Release 0 Feature Pack 2

WebSphere Adapter for Oracle E-Business Suite User Guide Version 7 Release 0 Feature Pack 2

IBM

WebSphere® Adapters
Version 7 Release 0 Feature Pack 2

WebSphere Adapter for Oracle E-Business Suite User Guide Version 7 Release 0 Feature Pack 2

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 69.

October 2010

This edition applies to version 7, release 0, modification 2 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.

Contents

WebSphere Adapter for Oracle E-Business Suite documentation

With WebSphere® Adapter for Oracle E-Business Suite, you can create integrated processes that include the exchange of information with Oracle E-Business Suite, without special coding.

What is new in this release

This version includes several new features that enhance the business flexibility, user experience, and performance of the adapter.

Complete information about other supported features is available at the WebSphere Adapter for Oracle E-Business Suite information center, http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.wsadapters.jca.oracleebiz.doc/doc/stbp_ore_welcome.html which is periodically updated with the latest information.

WebSphere Adapter for Oracle E-Business Suite, supports the following new features:

Feature Pack 2

- XML Gateway interface into Oracle E-Business Suite.
- Metadata discovery for supported interfaces for Oracle E-Business Suite.
- Support for Oracle PLSQL datatype BOOLEAN used in Oracle Stored Procedure parameters.
- Support for overloaded SP/SF in Oracle database.

Note: In WebSphere Integration Developer, ensure that you have only one version of the adapter imported into your workspace. You can either have the adapter Fix Pack version 7.0.0.3 or Feature Pack version 7.0.2.0.

In the runtime environment, the application (EAR) should contain only one version of the embedded RAR file, either the adapter Fix Pack version 7.0.0.3 or Feature Pack version 7.0.2.0. The node level deployed adapter should also have only any one version of the adapter.

Feature Pack 1

- User-defined type (STRUCTS) for Oracle database for inbound and outbound processing.
- Special value to indicate return ALL records.

Note: In WebSphere Integration Developer, ensure that you have only one version of the adapter imported into your workspace. You can either have the adapter Fix Pack version 7.0.0.1 or Feature Pack version 7.0.1.0.

In the runtime environment, the application (EAR) should contain only one version of the embedded RAR file, either the adapter Fix Pack version 7.0.0.1 or Feature Pack version 7.0.1.0. The node level deployed adapter should also have only any one version of the adapter.

Support for user-defined type (STRUCTS) for Oracle database

For Oracle databases, the adapter supports complex data types such as ARRAY, TABLE, STRUCT in table and query business objects. The adapter processes these data types as child business objects of the table or query business objects.

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:

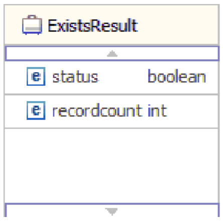
- Tables and views
- Synonyms and nicknames
- Stored procedures and stored functions

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.



ExistsResult	
status	boolean
recordcount	int

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:

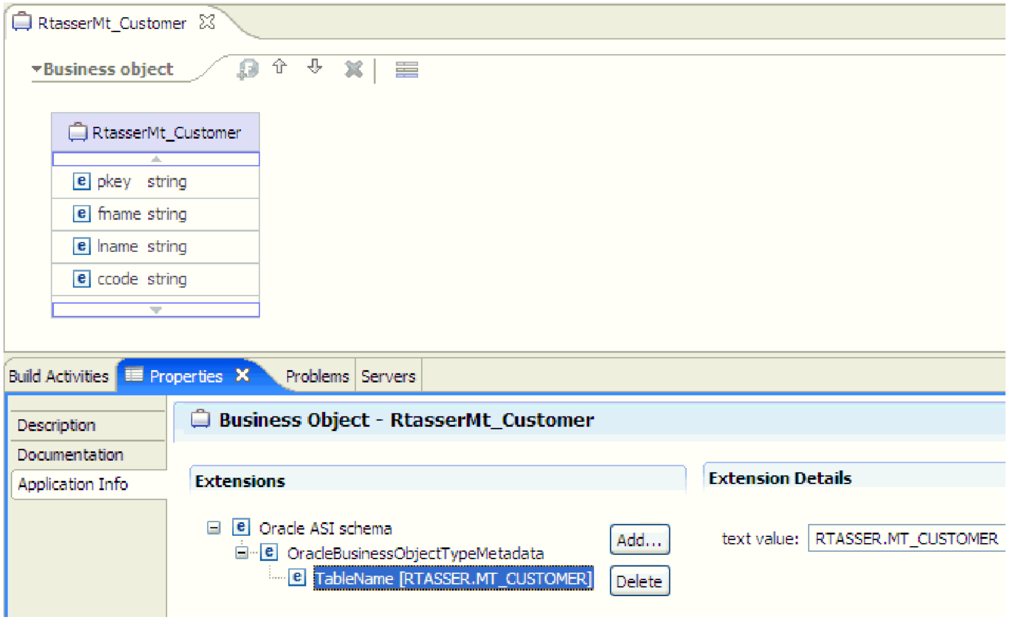
- The database table can have more columns than the corresponding business object has simple attributes; that is, some columns in the database are not represented in the business object. Only those columns needed for the processing of your business object must be included in your design.
- The business 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 1 on page 4 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.



text value: RTASSER.MT_CUSTOMER

Figure 1. A table business object with no child.

Figure 2 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.

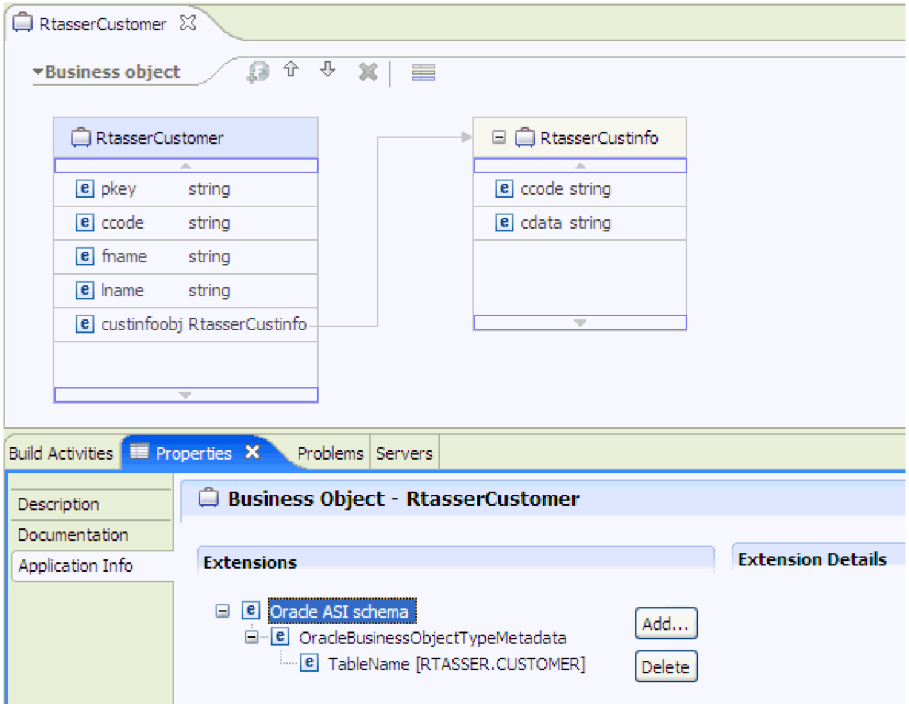
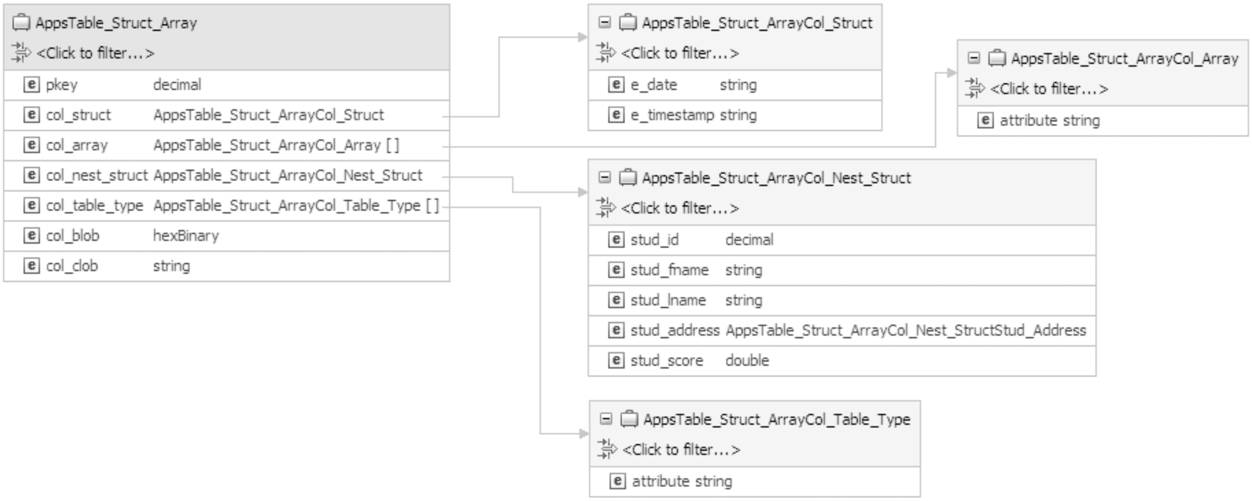


Figure 2. A table business object with one child business object.

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.



AppsTable_Struct_Array
<Click to filter…>

pkey	decimal
col_struct	AppsTable_Struct_ArrayCol_Struct
col_array	AppsTable_Struct_ArrayCol_Array [ ]
col_nest_struct	AppsTable_Struct_ArrayCol_Nest_Struct
col_table_type	AppsTable_Struct_ArrayCol_Table_Type [ ]
col_blob	hexBinary
col_clob	string

AppsTable_Struct_ArrayCol_Struct
<Click to filter…>

e_date	string
e_timestamp	string

AppsTable_Struct_ArrayCol_Array
<Click to filter…>

attribute	string

AppsTable_Struct_ArrayCol_Nest_Struct
<Click to filter…>

stud_id	decimal
stud_fname	string
stud_lname	string
stud_address	AppsTable_Struct_ArrayCol_Nest_StructStud_Address
stud_score	double

AppsTable_Struct_ArrayCol_Table_Type
<Click to filter…>

attribute	string

Figure 3. An Oracle table business object having complex data types as columns

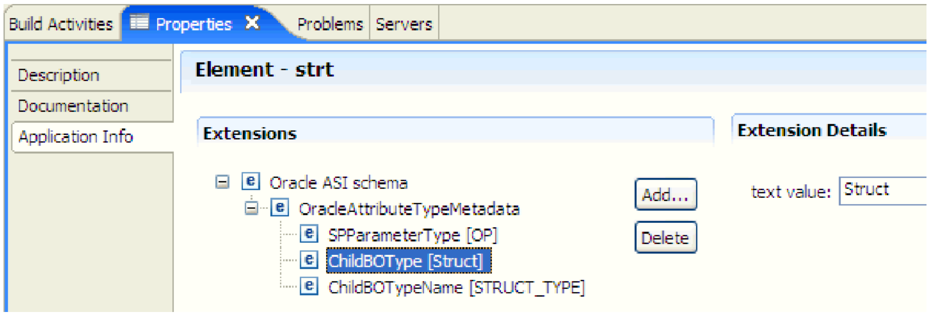
For stored procedure and stored function business objects

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.

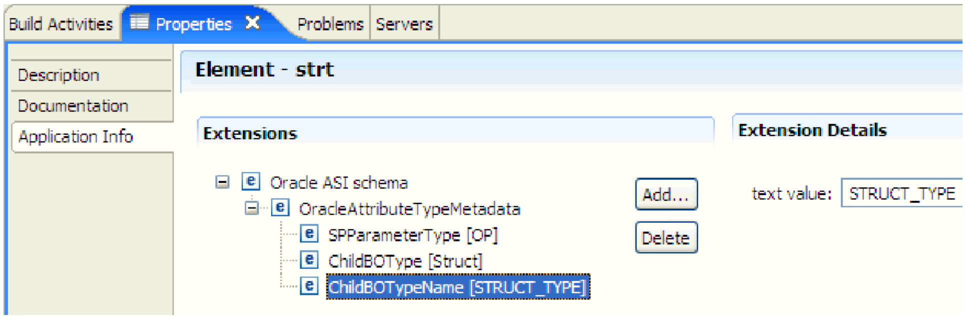
The following Properties view shows business objects generated from a stored procedure that has one input type and two output types. One of the output parameters is of the Struct data type. The external service wizard generates a business object for the Struct type and adds it as a child object to the 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.



text value: Struct

Figure 4. 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.



text value: STRUCT_TYPE

Figure 5. 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 1 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 1. Attributes of a query business object

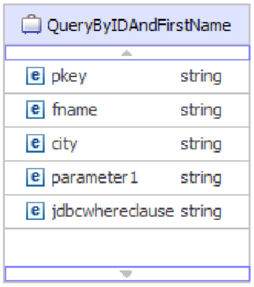
Business object attribute	Description
pkey	Corresponds to database column PKEY in the Customer table
fname	Corresponds to database column FNAME in the Customer table
city	Corresponds to database column CITY in the Address table

Table 1. 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.

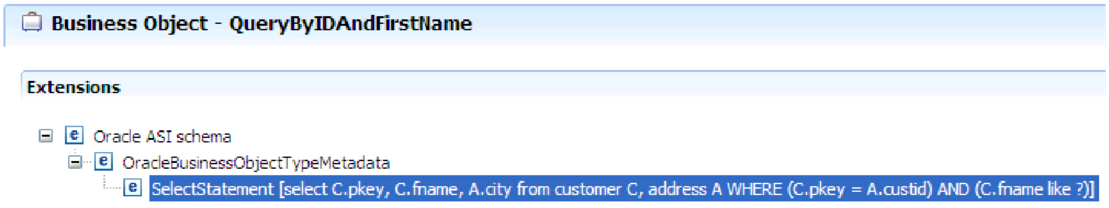


QueryByIDAndFirstName

pkey	string
fname	string
city	string
parameter1	string
jdbcwhereclause	string

Figure 6. 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.



Business Object - QueryByIDAndFirstName

Extensions

- Oracle ASI schema
  - OracleBusinessObjectTypeMetadata
    - SelectStatement [select C.pkey, C.fname, A.city from customer C, address A WHERE (C.pkey = A.custid) AND (C.fname like ?)]

Figure 7. 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 7.0, 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 WebSphere Integration Developer 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 WebSphere Integration Developer. The wizard connects to the database, discovers database objects, and displays them to you. You select the database objects for which you want to create business objects. For example, you specify which schemas you want to examine. In those schemas, you 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 which business objects you want and define the hierarchy of those objects, the wizard then generates business objects to represent the objects that you selected. It also generates other artifacts needed by the adapter.

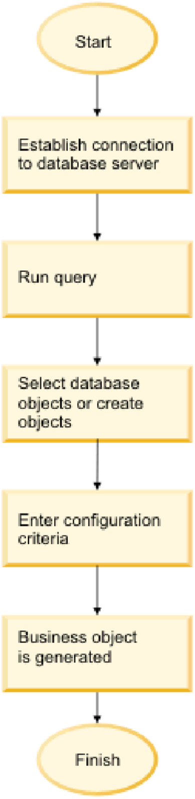

Start
Establish connection to database server
Run query
Select database objects or create objects
Enter configuration criteria
Business object is generated
Finish

Figure 8. 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 WebSphere Integration Developer, 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 WebSphere Integration Developer information center at the following link: http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp.

Related tasks

"Selecting and configuring tables, views, and synonyms or nicknames for outbound processing" on page 12
Select and configure business objects for tables, views, and synonyms or nicknames for use in your module. For inbound processing, these are the business objects that are delivered in events.

"Selecting and configuring query business objects" on page 20
Select and configure query business objects for user-defined SELECT statements for use in your module.

"Selecting and configuring tables, views, and synonyms or nicknames for inbound processing" on page 23
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 32
Business object attributes define the content of a business object and are built from the list of columns in the database object.

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 tasks

"Selecting and configuring tables, views, and synonyms or nicknames for outbound processing" on page 12
Select and configure business objects for tables, views, and synonyms or nicknames for use in your module. For inbound processing, these are the business objects that are delivered in events.

"Selecting and configuring query business objects" on page 20
Select and configure query business objects for user-defined SELECT statements for use in your module.

"Selecting and configuring tables, views, and synonyms or nicknames for inbound processing" on page 23
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 32
Business object attributes define the content of a business object and are built from the list of columns in the database object.

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 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 tasks

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

"Selecting and configuring query business objects" on page 20
Select and configure query business objects for user-defined SELECT statements for use in your module.

"Selecting and configuring tables, views, and synonyms or nicknames for inbound processing" on page 23
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 32
Business object attributes define the content of a business object and are built from the list of columns in the database object.

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

Select and configure business objects for tables, views, and synonyms or nicknames for use 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 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. You 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

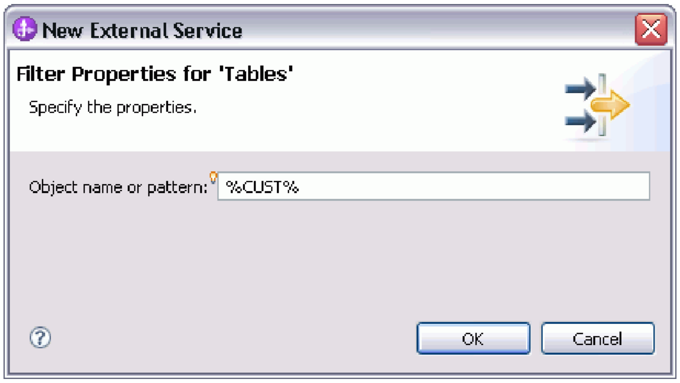
1. In the **Discovered objects** list of the Find Objects in the Enterprise System window, select one or more tables, views, or synonyms from the **Discovered objects** list, and click the **>** (Add) button to add the object or objects to the **Selected objects** list. Alternatively, you can also filter 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.



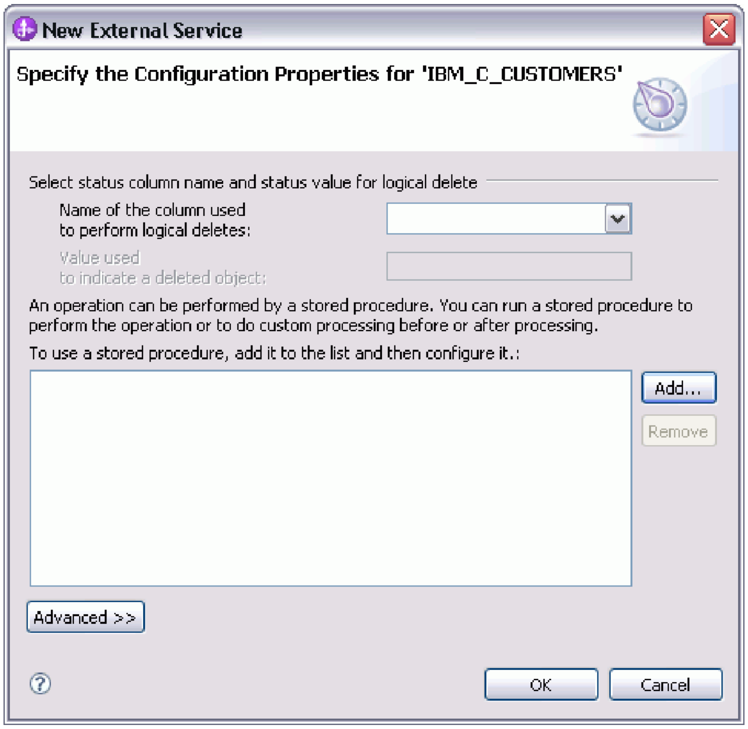
New External Service

Filter Properties for 'Tables'

Specify the properties.

Object name or pattern: %CUST%

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.


New External Service

Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

Select status column name and status value for logical delete

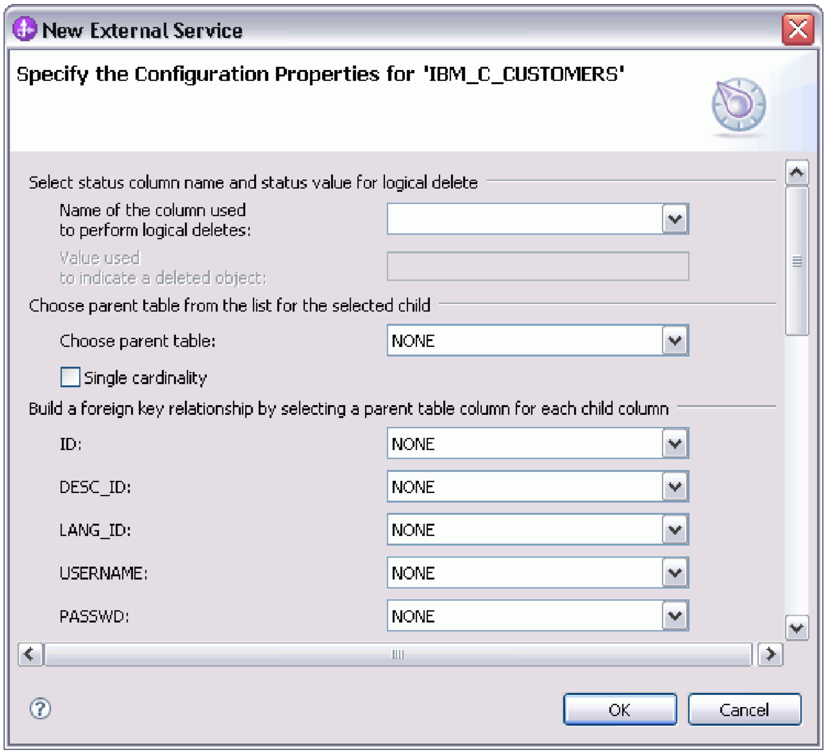
Name of the column used to perform logical deletes:

Value used to indicate a deleted object:

An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

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.


New External Service

Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

Value used to indicate a deleted object:

Choose parent table from the list for the selected child

Choose parent table: NONE

□ Single cardinality

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

ID: NONE

DESC_ID: NONE

LANG_ID: NONE

USERNAME: NONE

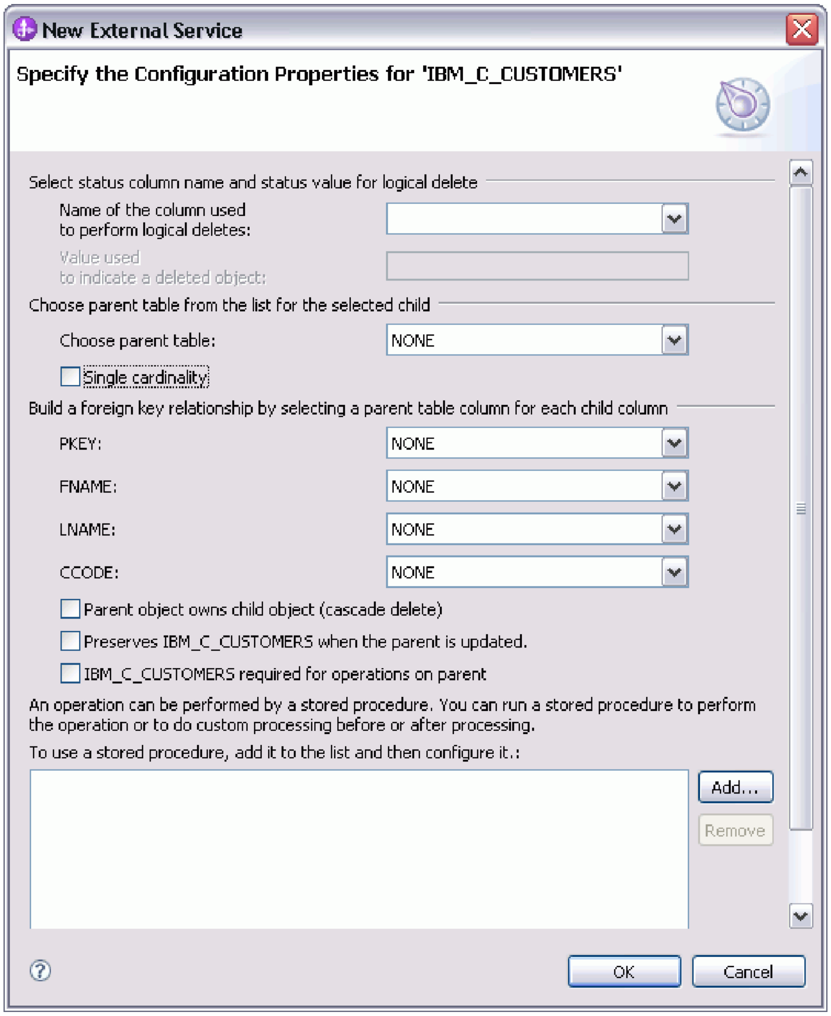
PASSWD: NONE

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.



New External Service

Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

Value used to indicate a deleted object:

Choose parent table from the list for the selected child

Choose parent table: NONE

□ Single cardinality

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

PKEY: NONE

FNAME: NONE

LNAME: NONE

CCODE: NONE

□ Parent object owns child object (cascade delete)

□ Preserves IBM_C_CUSTOMERS when the parent is updated.

□ IBM_C_CUSTOMERS required for operations on parent

An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

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 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.

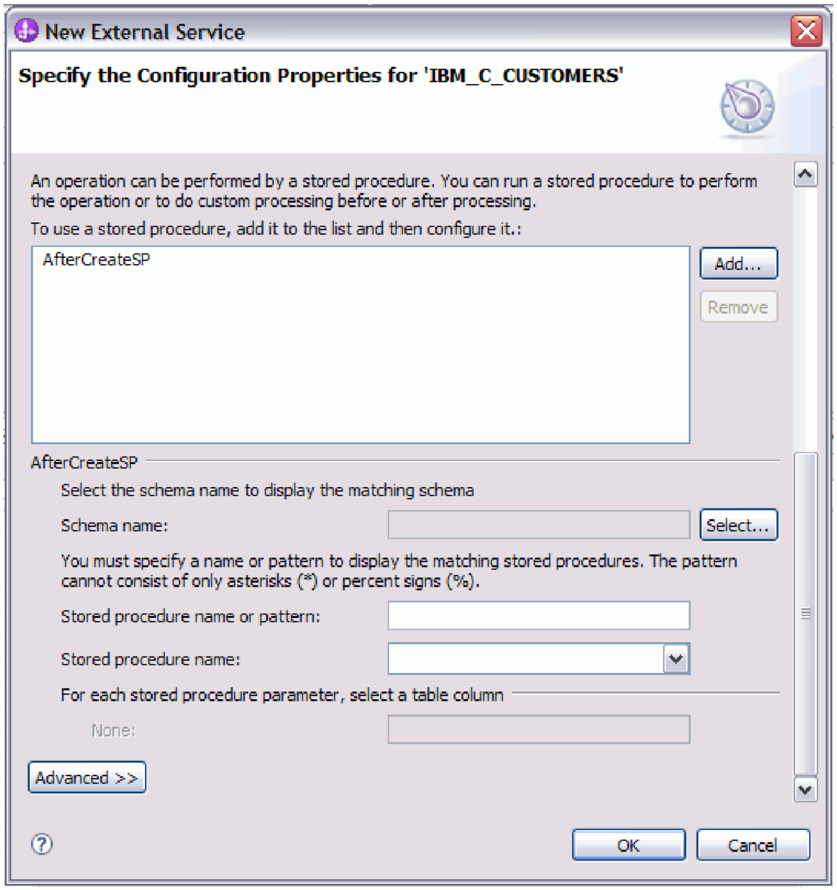
5. An operation can be performed using either a standard SQL statement generated by the adapter or using stored procedures or stored functions from the database. If you want to use stored procedures or stored functions:

   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.



New External Service

Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

AfterCreateSP

AfterCreateSP

Select the schema name to display the matching schema

Schema name:

You must specify a name or pattern to display the matching stored procedures. The pattern cannot consist of only asterisks (*) or percent signs (%).

Stored procedure name or pattern:

Stored procedure name:

For each stored procedure parameter, select a table column

None:

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 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**.

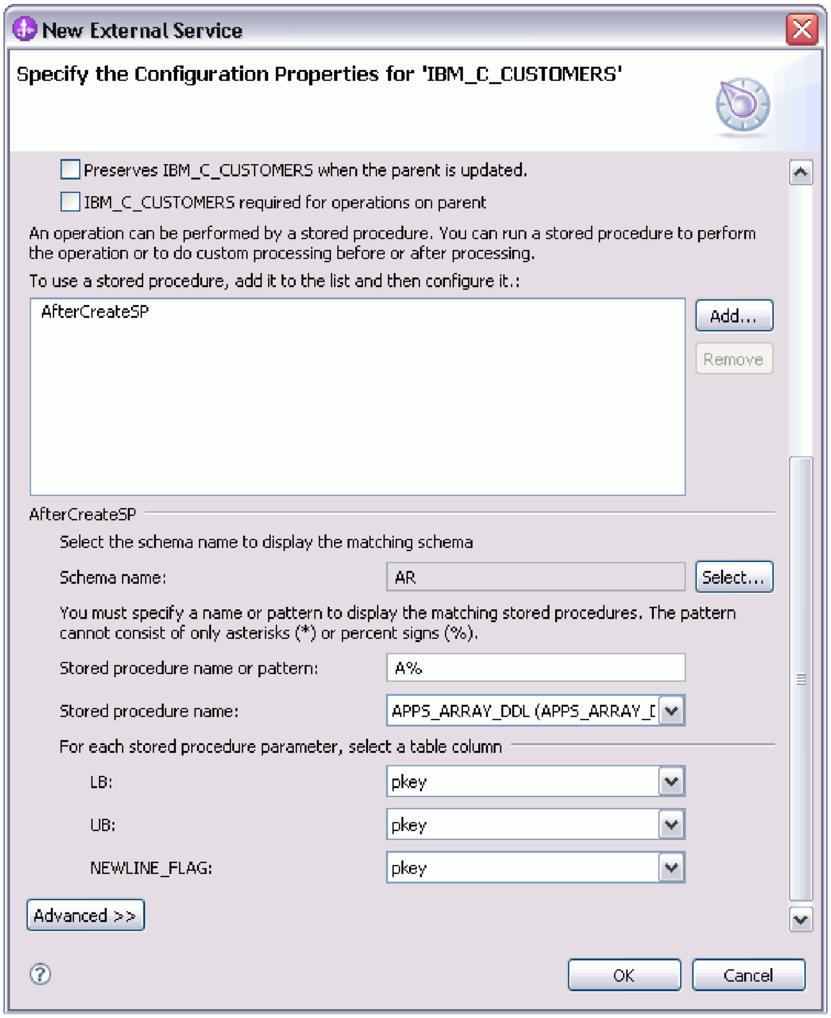
   b. Specify the name of the stored procedure or stored function.

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.



Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

□ Preserves IBM_C_CUSTOMERS when the parent is updated.

□ IBM_C_CUSTOMERS required for operations on parent

An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

AfterCreateSP

AfterCreateSP

Select the schema name to display the matching schema

Schema name: AR

You must specify a name or pattern to display the matching stored procedures. The pattern cannot consist of only asterisks (*) or percent signs (%).

Stored procedure name or pattern: A%

Stored procedure name: APPS_ARRAY_DDL (APPS_ARRAY_[

For each stored procedure parameter, select a table column

LB: pkey

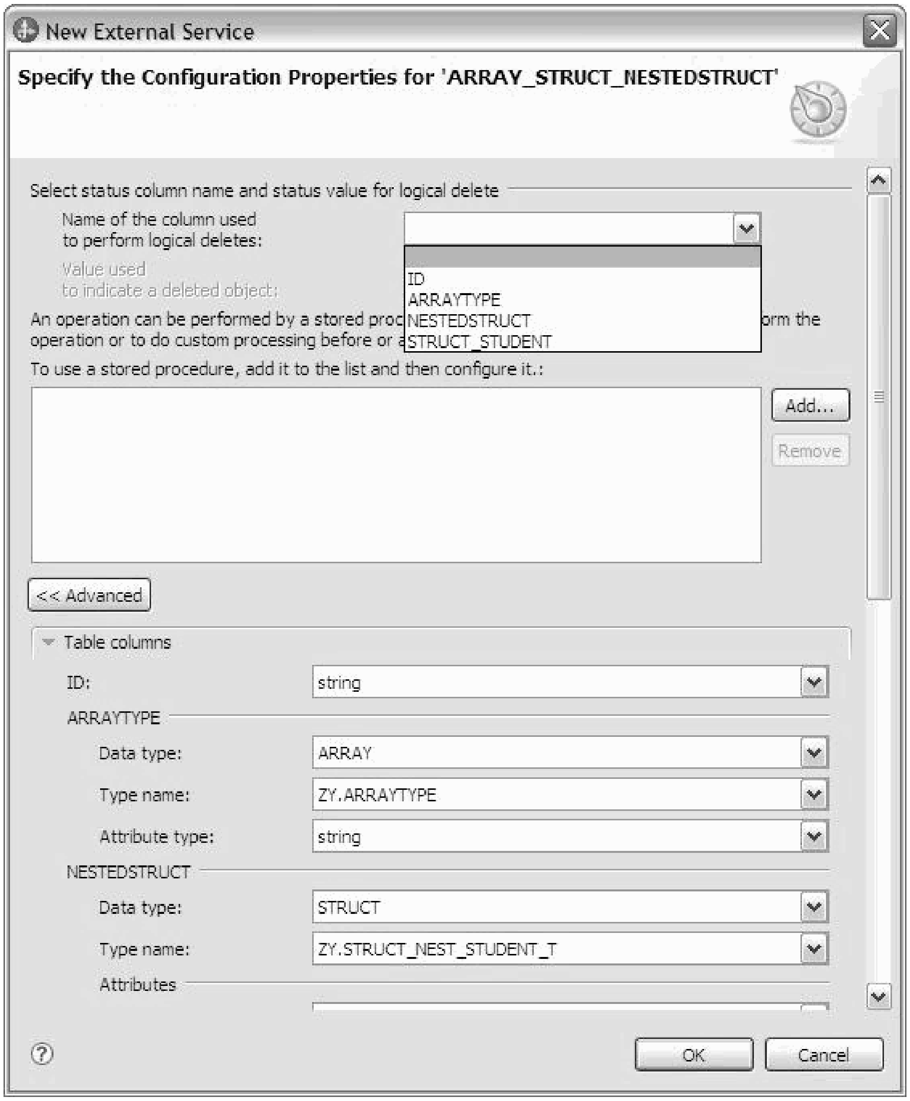
UB: pkey

NEWLINE_FLAG: pkey

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.



New External Service

Specify the Configuration Properties for 'ARRAY_STRUCT_NESTEDSTRUCT'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

Value used to indicate a deleted object:

An operation can be performed by a stored proc orm the operation or to do custom processing before or a

To use a stored procedure, add it to the list and then configure it.:

Table columns

ID: string

ARRAYTYPE

Data type: ARRAY

Type name: ZY.ARRAYTYPE

Attribute type: string

NESTEDSTRUCT

Data type: STRUCT

Type name: ZY.STRUCT_NEST_STUDENT_T

Attributes

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 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 Systemwindow to select and configure other types of business objects.

Related concepts

"Business objects" on page 2
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.

"Create operation" on page 9
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.

"Update operation" on page 10
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.

Related reference

"Business object attributes" on page 32
Business object attributes define the content of a business object and are built from the list of columns in the database object.

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

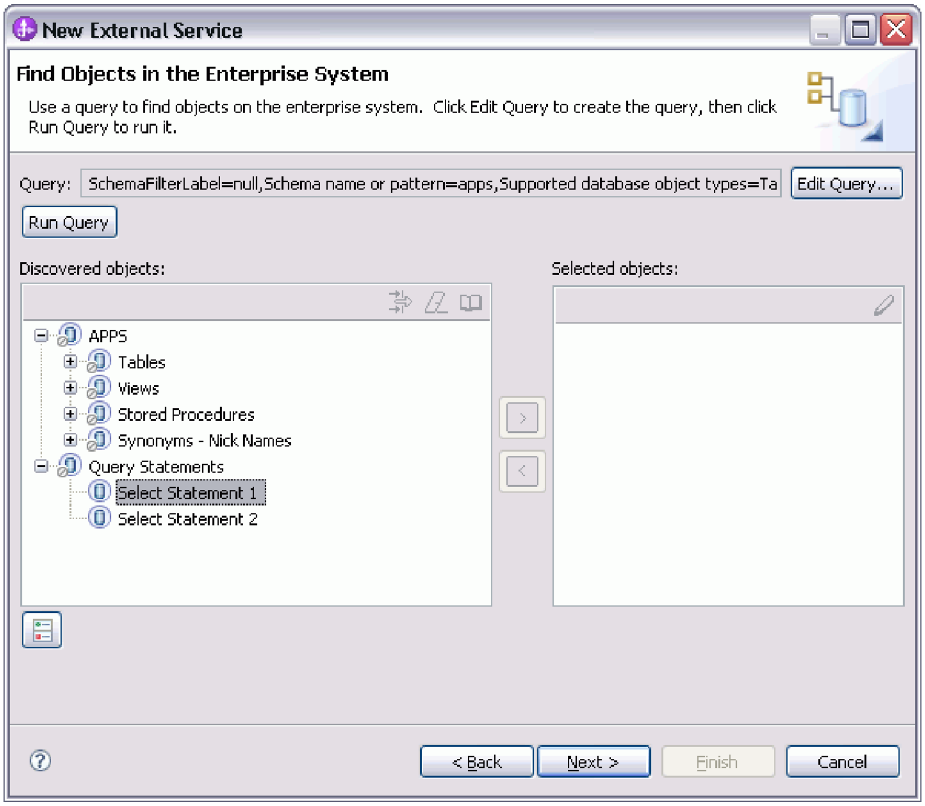
To configure query business objects, you must know the structure of the data in your database, including the tables and views. You need to know the name and data type of the columns that your module needs to access. You must also be able to write SQL SELECT statements.

About this task

This task is performed through the external service wizard. You 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.



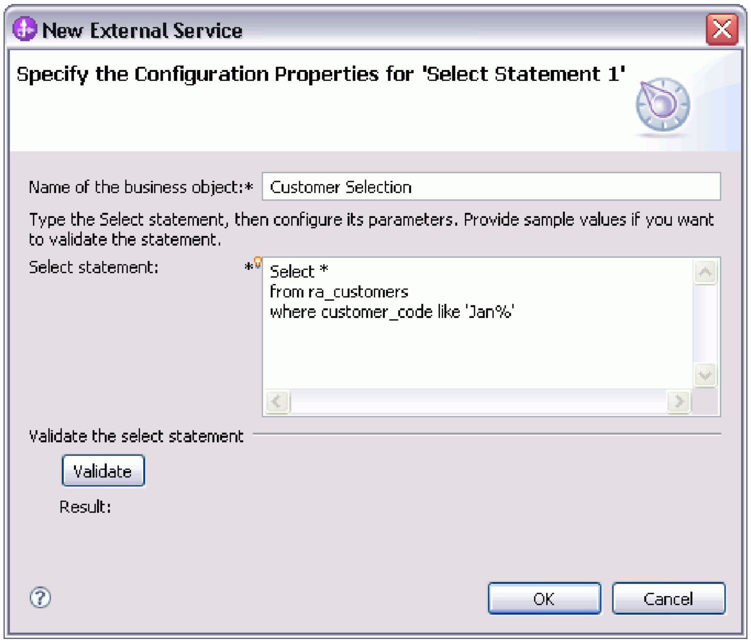
New External Service

Find Objects in the Enterprise System

Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: SchemaFilterLabel=null,Schema name or pattern=apps,Supported database object types=Ta

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 shows the Specify the Configuration Properties for 'object' window that opens when you click > (Add) for a query business object.



New External Service

Specify the Configuration Properties for 'Select Statement 1'

Name of the business object:* Customer Selection

Type the Select statement, then configure its parameters. Provide sample values if you want to validate the statement.

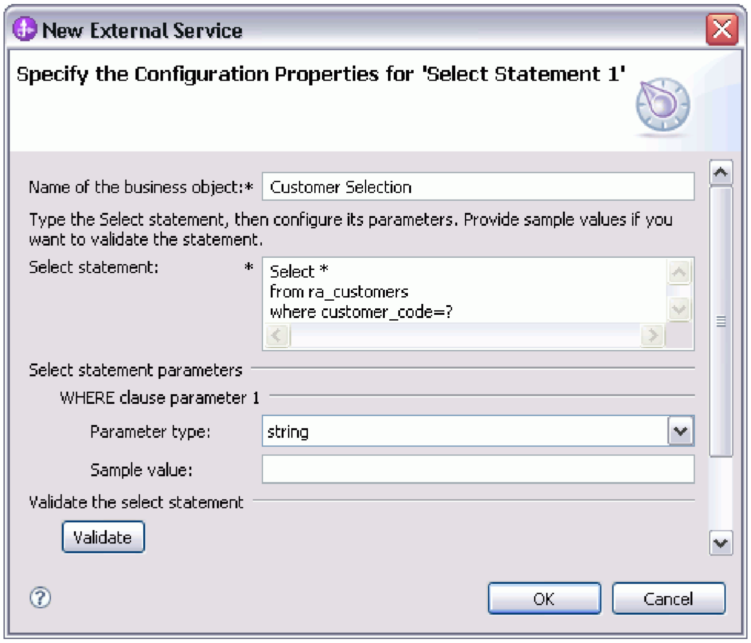
Select statement: *

```
Select *
from ra_customers
where customer_code like 'Jan%'
```

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 shows the Specify the Configuration Properties for 'object' window for a query business object that has a single parameter.



New External Service

Specify the Configuration Properties for 'Select Statement 1'

Name of the business object:* Customer Selection

Type the Select statement, then configure its parameters. Provide sample values if you want to validate the statement.

Select statement: *
```
Select *
from ra_customers
where customer_code=?
```

Select statement parameters

WHERE clause parameter 1

Parameter type: string

Sample value:

Validate the select statement

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 global properties.

Related concepts

"Business objects" on page 2
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.

"Create operation" on page 9
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.

"Update operation" on page 10
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.

Related reference

"Business object attributes" on page 32
Business object attributes define the content of a business object and are built from the list of columns in the database object.

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 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. You 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

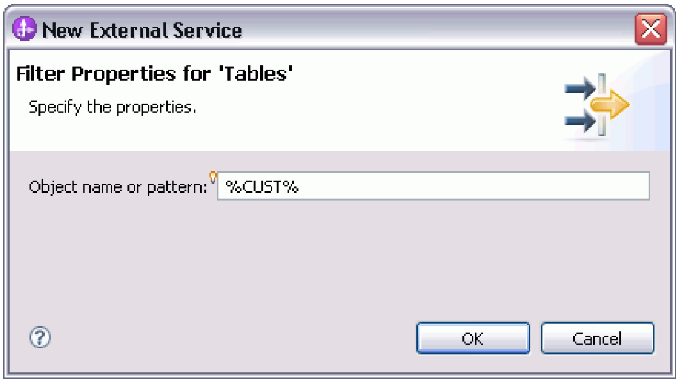
1. In the **Discovered objects** list of the Find Objects in the Enterprise System window, select one or more tables, views, or synonyms. Click the > (Add) button to add the object or objects to the **Selected objects** list. Alternatively, you can also filter the tables, views, or synonyms by specifying 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.



New External Service

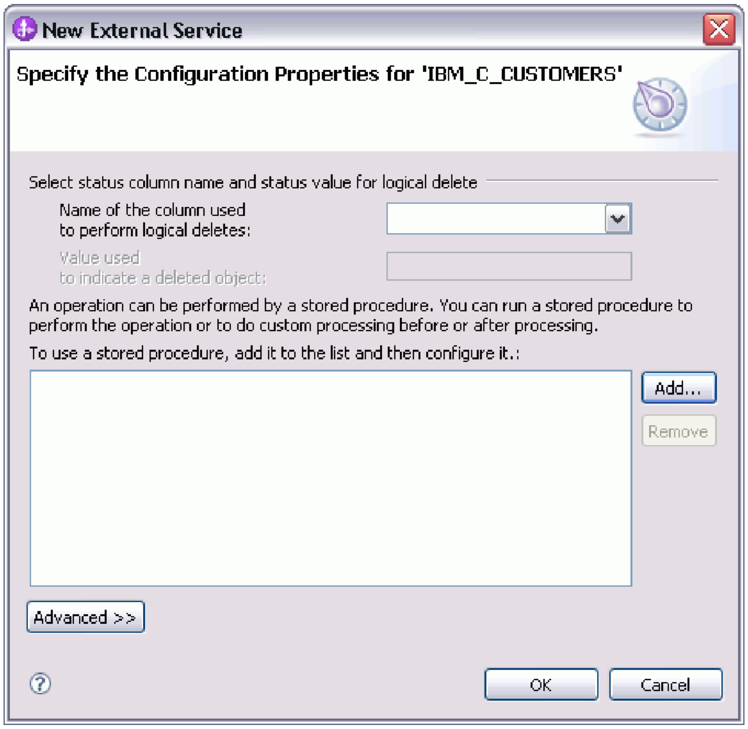
Filter Properties for 'Tables'

Specify the properties.

Object name or pattern: %CUST%

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.


New External Service

Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

Select status column name and status value for logical delete

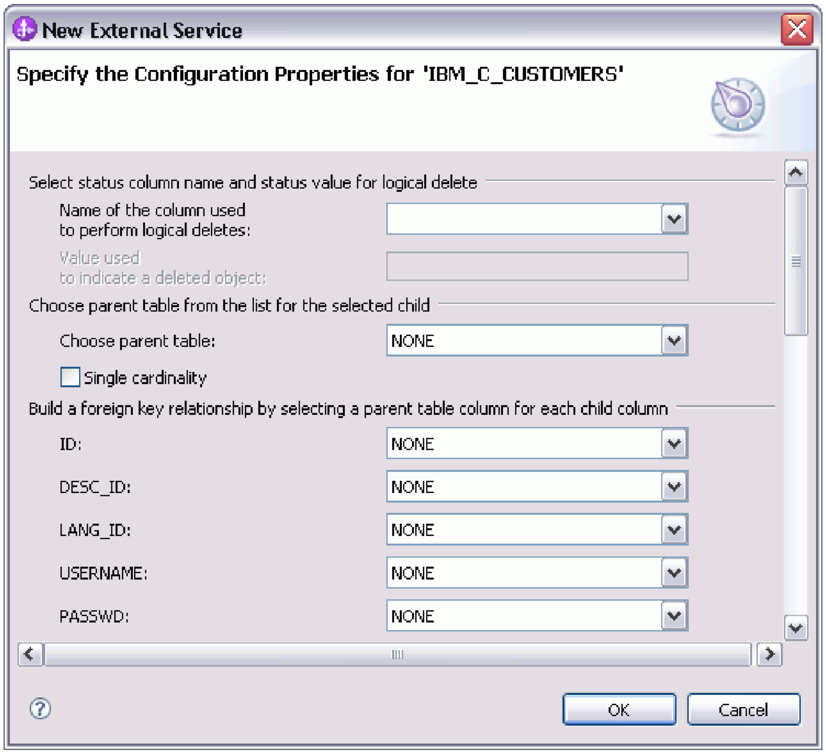
Name of the column used to perform logical deletes:

Value used to indicate a deleted object:

An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

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.


New External Service

Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

Value used to indicate a deleted object:

Choose parent table from the list for the selected child

Choose parent table: NONE

□ Single cardinality

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

ID: NONE

DESC_ID: NONE

LANG_ID: NONE

USERNAME: NONE

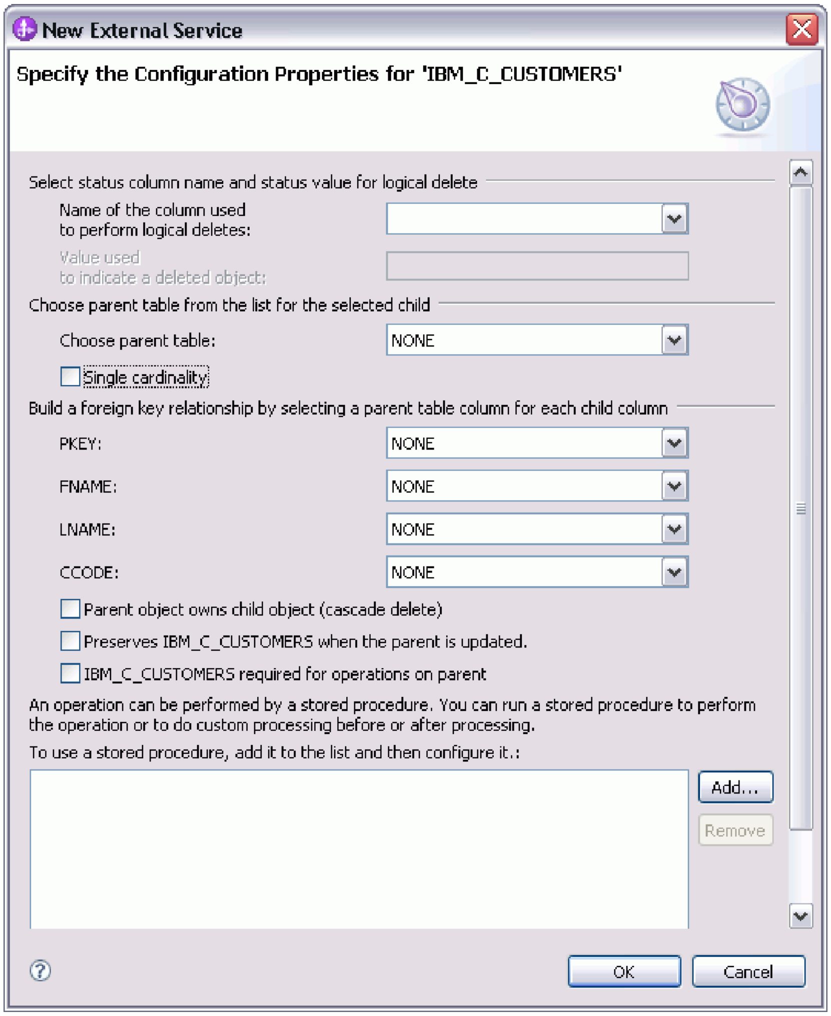
PASSWD: NONE

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.



New External Service

Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

Value used to indicate a deleted object:

Choose parent table from the list for the selected child

Choose parent table: NONE

□ Single cardinality

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

PKEY:	NONE
FNAME:	NONE
LNAME:	NONE
CCODE:	NONE

□ Parent object owns child object (cascade delete)

□ Preserves IBM_C_CUSTOMERS when the parent is updated.

□ IBM_C_CUSTOMERS required for operations on parent

An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

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 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.

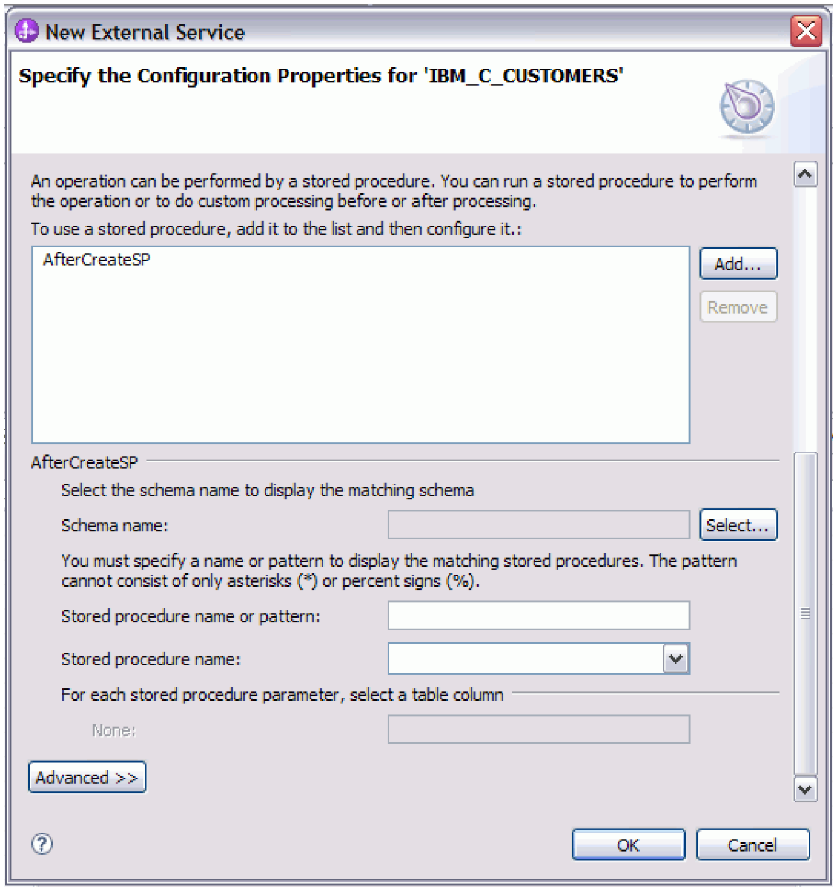
5. An operation can be performed using either a standard SQL statement generated by the adapter or using stored procedures or stored functions from the database. If you want to use stored procedures or stored functions:

   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.



Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

AfterCreateSP

AfterCreateSP

Select the schema name to display the matching schema

Schema name:

You must specify a name or pattern to display the matching stored procedures. The pattern cannot consist of only asterisks (*) or percent signs (%).

Stored procedure name or pattern:

Stored procedure name:

For each stored procedure parameter, select a table column

None:

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 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.

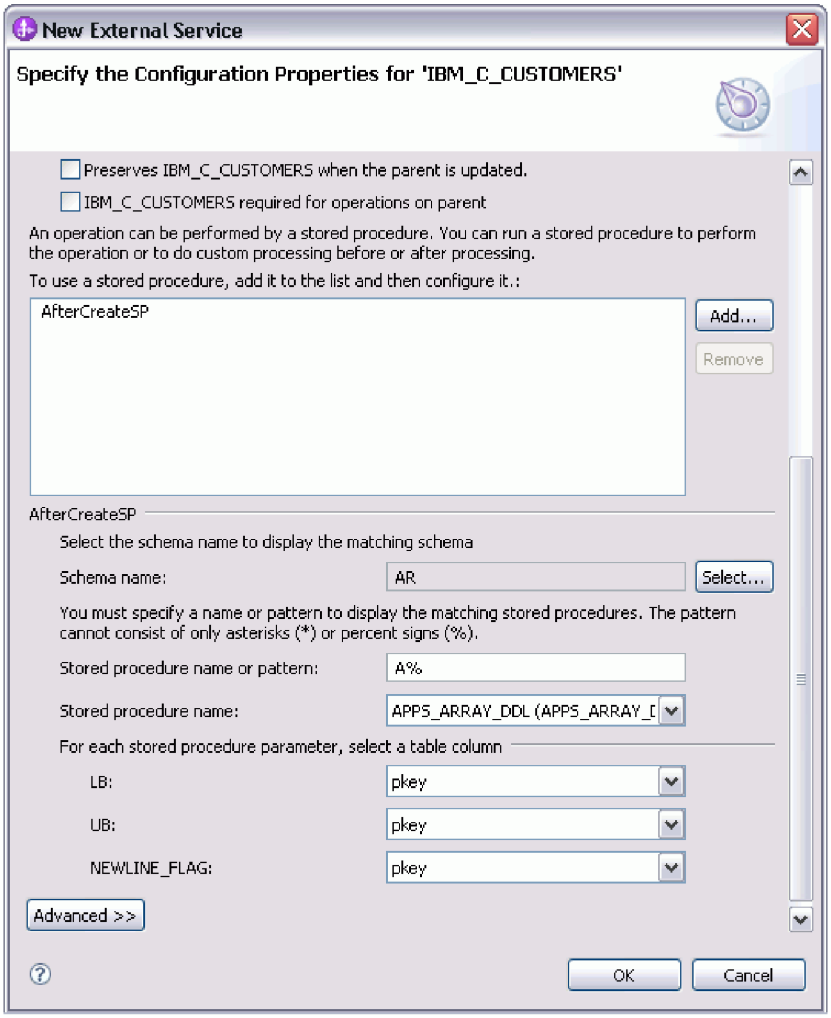
      1) In the **Stored procedure name or pattern** field, either type the name of the stored procedure or stored function, or type a name pattern. Use the question mark or underscore (? or _) to match a single character and the asterisk or percentage sign (* or %) to match multiple characters.

      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.



New External Service

Specify the Configuration Properties for 'IBM_C_CUSTOMERS'

□ Preserves IBM_C_CUSTOMERS when the parent is updated.

□ IBM_C_CUSTOMERS required for operations on parent

An operation can be performed by a stored procedure. You can run a stored procedure to perform the operation or to do custom processing before or after processing.

To use a stored procedure, add it to the list and then configure it.:

AfterCreateSP

AfterCreateSP

Select the schema name to display the matching schema

Schema name: AR

You must specify a name or pattern to display the matching stored procedures. The pattern cannot consist of only asterisks (*) or percent signs (%).

Stored procedure name or pattern: A%

Stored procedure name: APPS_ARRAY_DDL (APPS_ARRAY_[

For each stored procedure parameter, select a table column

LB: pkey

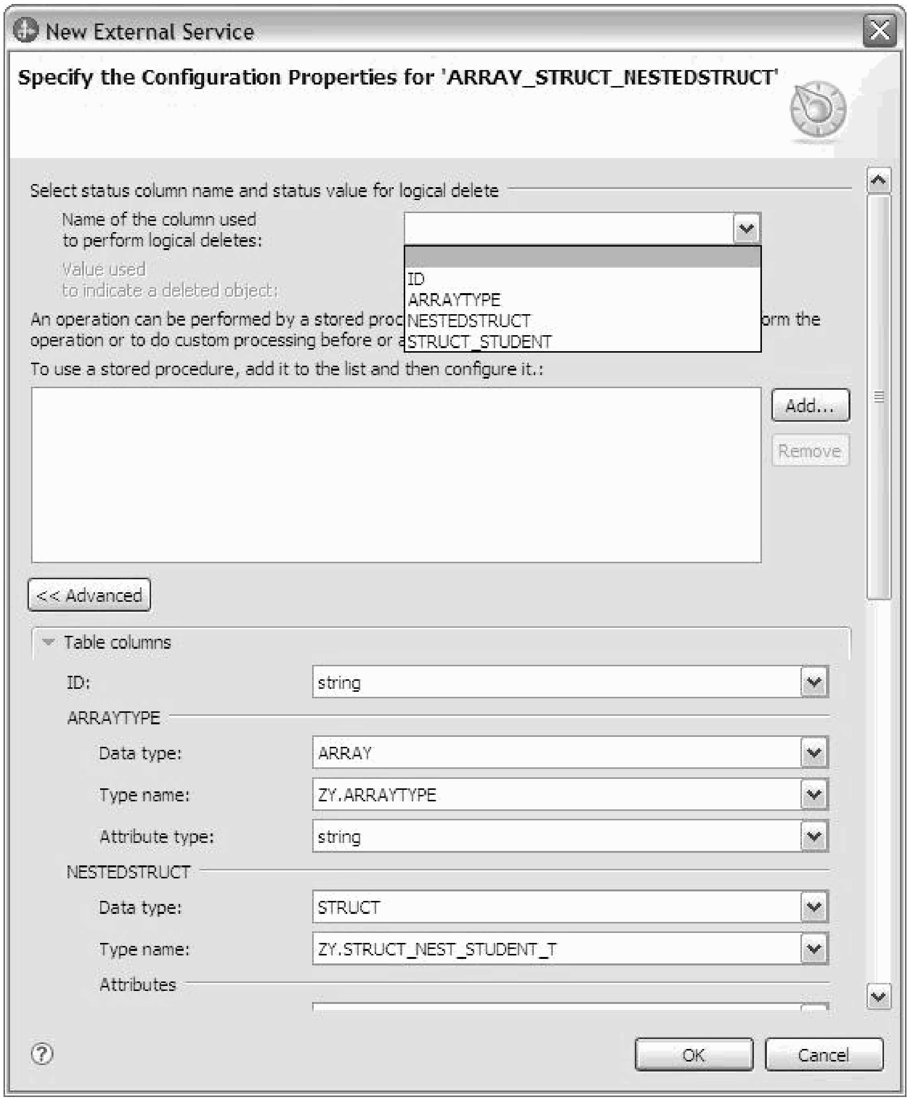
UB: pkey

NEWLINE_FLAG: pkey

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 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.



New External Service

Specify the Configuration Properties for 'ARRAY_STRUCT_NESTEDSTRUCT'

Select status column name and status value for logical delete

Name of the column used to perform logical deletes:

ID
ARRAYTYPE
NESTEDSTRUCT
STRUCT_STUDENT

Value used to indicate a deleted object:

An operation can be performed by a stored proc... orm the operation or to do custom processing before or a...

To use a stored procedure, add it to the list and then configure it.:

Table columns

ID: string

ARRAYTYPE

Data type: ARRAY

Type name: ZY.ARRAYTYPE

Attribute type: string

NESTEDSTRUCT

Data type: STRUCT

Type name: ZY.STRUCT_NEST_STUDENT_T

Attributes

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 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 objects" on page 2
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.

"Create operation" on page 9
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.

"Update operation" on page 10
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.

Related reference

"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.

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 2 lists the properties of a business object attribute and describes their interpretation and settings.

Table 2. Attribute properties

Properties	Interpretation and settings
Cardinality	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. 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.

Table 2. Attribute properties (continued)

Properties	Interpretation and settings
Foreign Key	When arrays of child business objects whose cardinality is *n* are retrieved, foreign keys are used in the WHERE clause of SELECT statements. The RetrieveAll operation overrides the use of keys and foreign keys. **Note:** The adapter does not support specifying an attribute that represents a child business object as a foreign key.
Name	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.
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	Indicates whether this attribute is a primary key. At least one simple attribute in each business object must be specified as the primary key. 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. **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.
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	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 3 on page 34. For attributes that specify a child business object, this property specifies the name of the business object.

The type of each database object, returned as the Oracle metadata, maps to the business object attribute types as listed in Table 3 on page 34. 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 3. 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 objects" on page 2
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.

"Create operation" on page 9
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.

"Update operation" on page 10
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.

Related tasks

"Selecting and configuring tables, views, and synonyms or nicknames for outbound processing" on page 12
Select and configure business objects for tables, views, and synonyms or nicknames for use in your module. For inbound processing, these are the business objects that are delivered in events.

"Selecting and configuring query business objects" on page 20
Select and configure query business objects for user-defined SELECT statements for use in your module.

"Selecting and configuring tables, views, and synonyms or nicknames for inbound processing" on page 23
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.

Support for a special value to indicate return ALL records

This feature provides a new value to the property Maximum records for RetrieveAll operation. If the value is -1, then RetrieveAll operation returns all records matching the query.

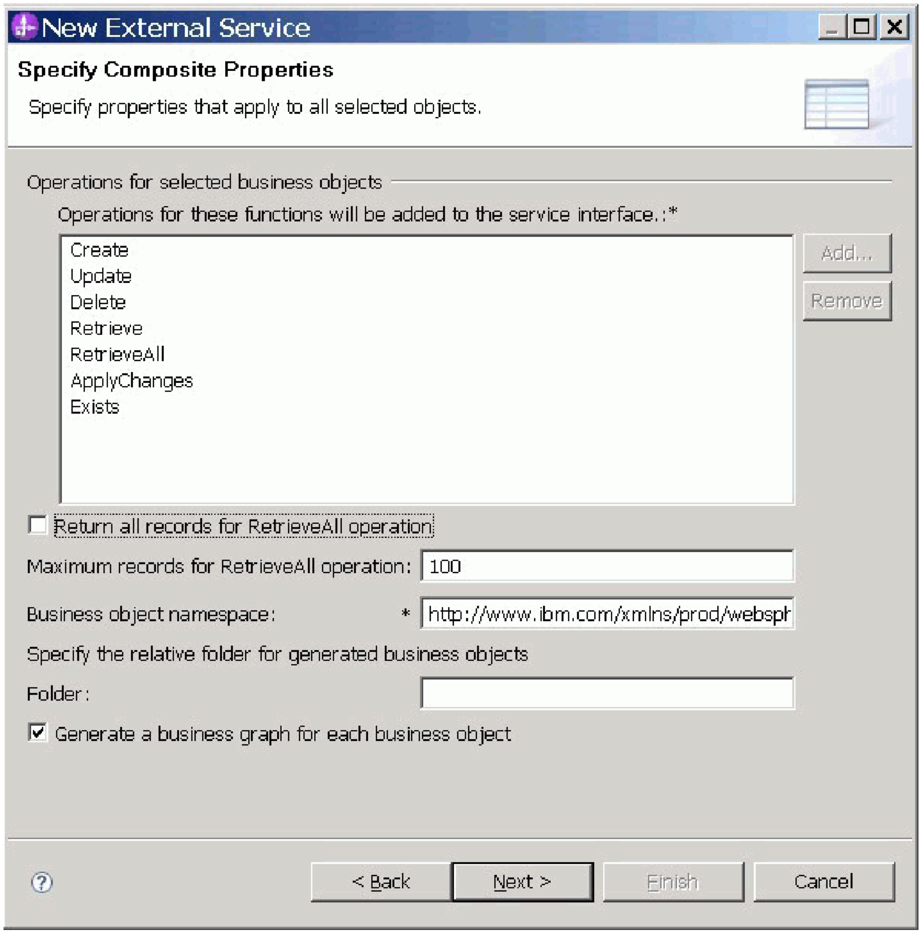
Setting global properties for operations

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.



Specify Composite Properties

Specify properties that apply to all selected objects.

Operations for selected business objects

Operations for these functions will be added to the service interface.:*

Create
Update
Delete
Retrieve
RetrieveAll
ApplyChanges
Exists

□ Return all records for RetrieveAll operation

Maximum records for RetrieveAll operation: 100

Business object namespace: * http://www.ibm.com/xmlns/prod/websph

Specify the relative folder for generated business objects

Folder:

☑ Generate a business graph for each business object

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 38.

     Note: The “Maximum records for RetrieveAll operation” on page 38 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 WebSphere Integration Developer earlier than version 6.2.x.

   Note: You must select this option if you are adding business objects to a module that was created with an earlier version of WebSphere Integration Developer. 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.

Related reference

"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 WebSphere Integration Developer. The properties reside in the method binding of the import.

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 WebSphere Integration Developer. The properties reside in the method binding of the import.

Table 4 on page 38 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.

Table 4. 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 5. Maximum records for RetrieveAll operation details

Required	Yes
Default	100
Usage	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 wrappered as a fault, MatchesExceededLimitFault and is returned to the client. **Note:** The MatchesExceededLimitFault is stored in the trace file and the MatchesExceededLimitException is not stored in log or trace files. • 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

"Setting global properties for operations" on page 35
After you select database objects in the external service wizard, you need to specify properties that apply to all business objects.

Support for XML Gateway interface

XML Gateway 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, Web Service Integration, and so on).

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. When XML Gateway interface is selected, the adapter converts the payload from the Document Type Definitions (DTD) format you specify to XSD format, and generates the request and response business objects.

The external service wizard helps you create an XML-based integrated service document that can be used as a payload content to invoke the web service during runtime. There are a few supporting terms used in this feature, which are described as follows:

- 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 some specific DTD definition and schema files (XSD). The payload XML document, which will be created based on generated Schema files (XSD), is used to take the business data for business process in the XML Gateway interface.

Configuring the module for XML Gateway interface

To configure the module for XML Gateway interface, use the external service wizard in WebSphere Integration Developer. You can configure this interface only for 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 based on your business requirement.

Before you begin

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

About this task

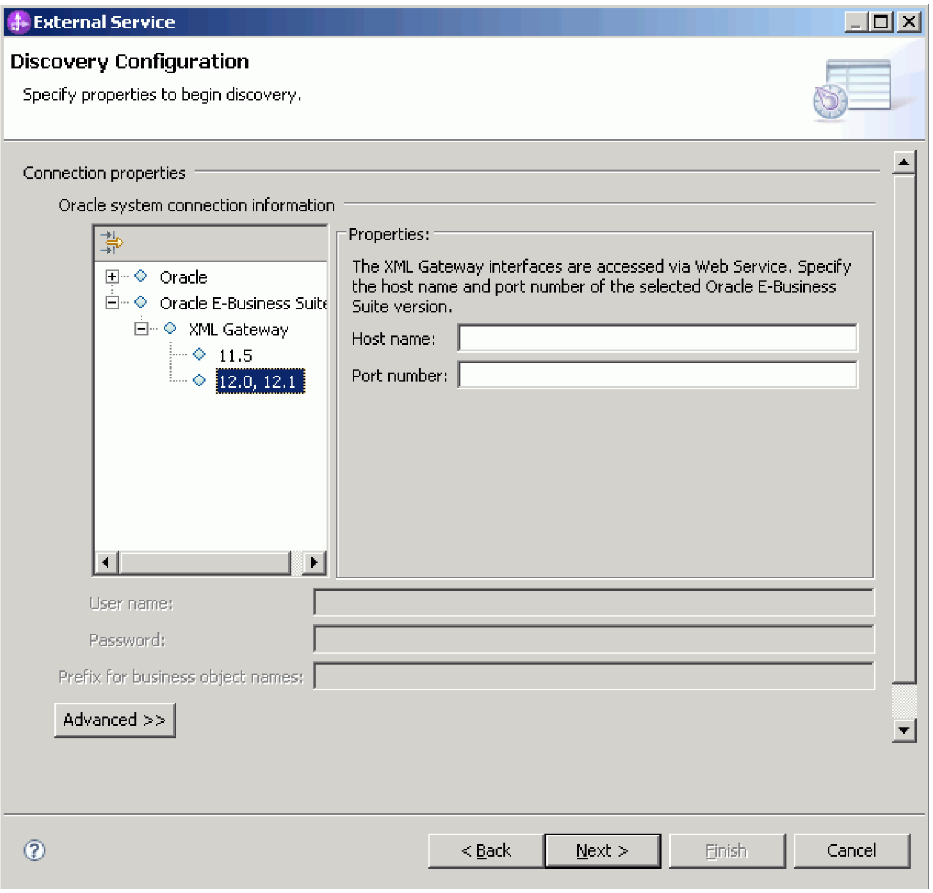
This task is to create an outbound integration application using XML Gateway interface.

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**.

   External Service

   Discovery Configuration

   Specify properties to begin discovery.

   The XML Gateway interfaces are accessed via Web Service. Specify the host name and port number of the selected Oracle E-Business Suite version.

   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 for the selected XML Gateway interface.

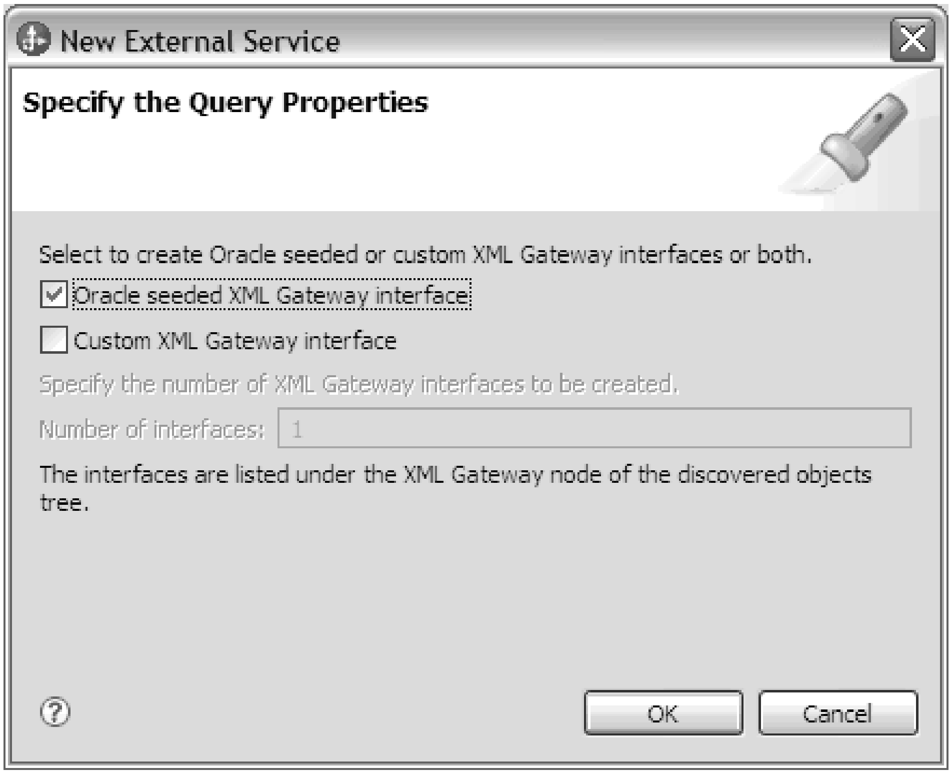
   a. In the **Host name** field, enter the host name or IP address of the system on which the Oracle E-Business Suite is installed.

b. In the **Port number** field, specify the port number to connect to the Oracle E-Business Suite.

c. 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 the 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.



New External Service

Specify the Query Properties

Select to create Oracle seeded or custom XML Gateway interfaces or both.

☑ Oracle seeded XML Gateway interface

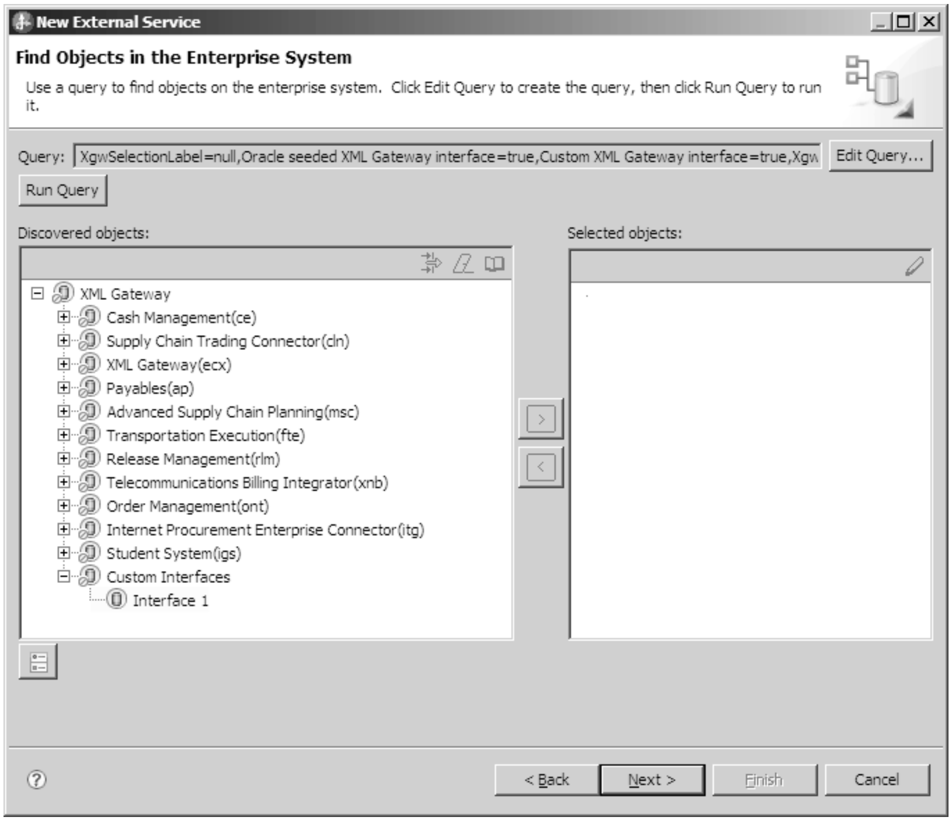
□ Custom XML Gateway interface

Specify the number of XML Gateway interfaces to be created.

Number of interfaces: 1

The interfaces are listed under the XML Gateway node of the discovered objects tree.

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. When you run a typical query, the result is displayed in the Query field, as shown in the following figure.



Find Objects in the Enterprise System

Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: XgwSelectionLabel=null,Oracle seeded XML Gateway interface=true,Custom XML Gateway interface=true,Xgw

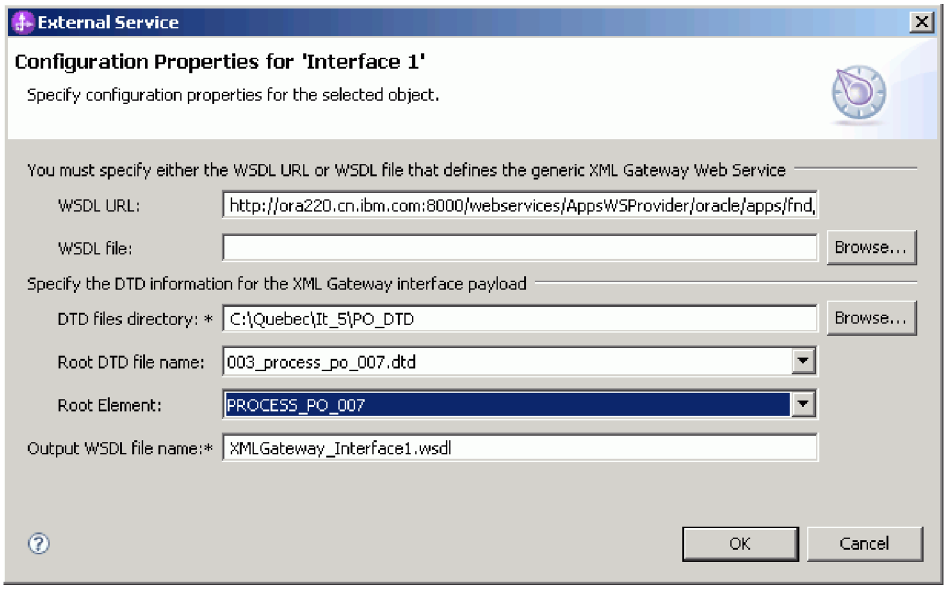
Discovered objects:

- XML Gateway
  - Cash Management(ce)
  - Supply Chain Trading Connector(cln)
  - XML Gateway(ecx)
  - Payables(ap)
  - Advanced Supply Chain Planning(msc)
  - Transportation Execution(fte)
  - Release Management(rlm)
  - Telecommunications Billing Integrator(xnb)
  - Order Management(ont)
  - Internet Procurement Enterprise Connector(itg)
  - Student System(igs)
  - Custom Interfaces
    - Interface 1

Selected objects:

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.



Configuration Properties for 'Interface 1'

Specify configuration properties for the selected object.

You must specify either the WSDL URL or WSDL file that defines the generic XML Gateway Web Service

WSDL URL: http://ora220.cn.ibm.com:8000/webservices/AppsWSProvider/oracle/apps/fnd,

WSDL file:

Specify the DTD information for the XML Gateway interface payload

DTD files directory: * C:\Quebec\It_5\PO_DTD

Root DTD file name: 003_process_po_007.dtd

Root Element: PROCESS_PO_007

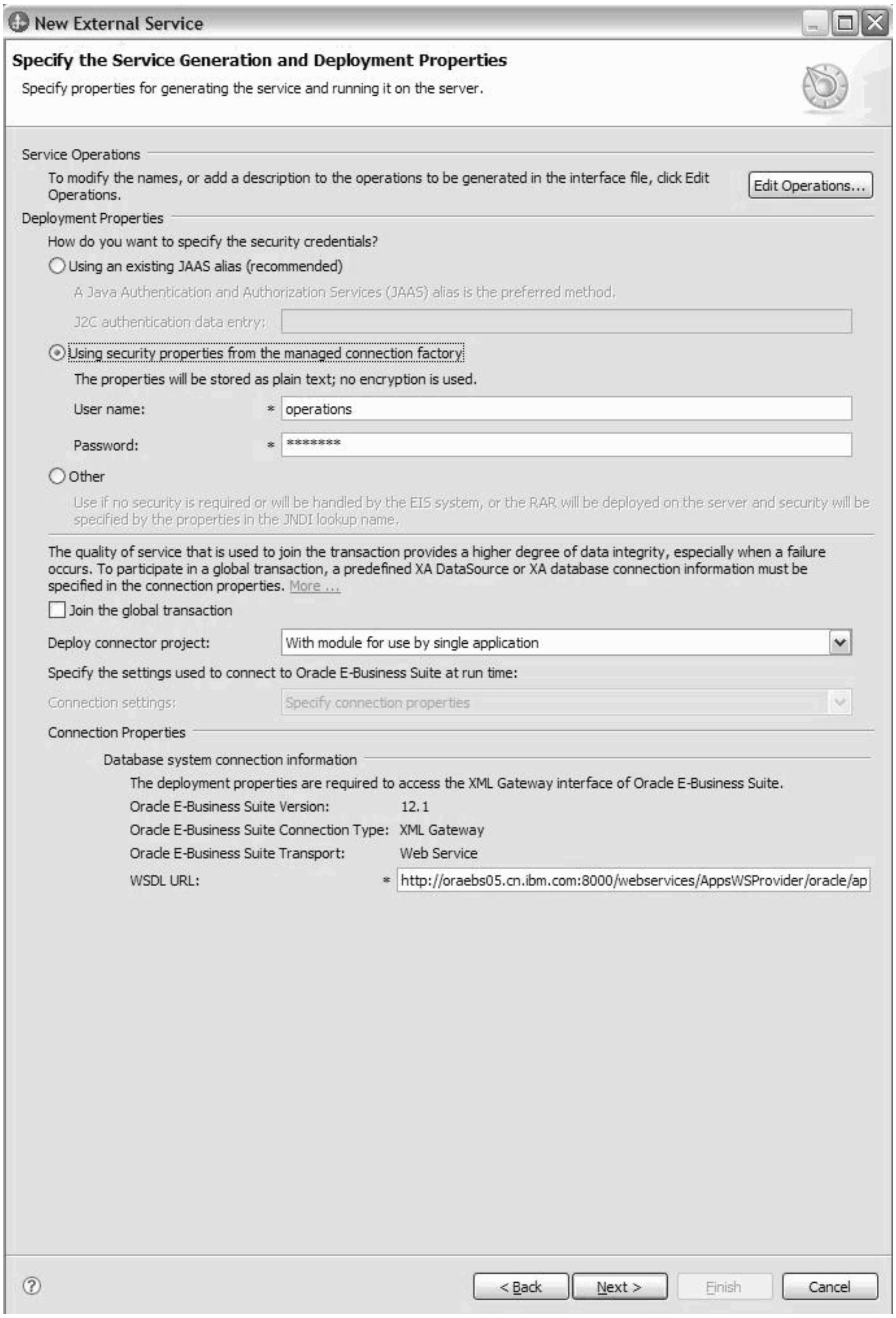
Output WSDL file name:* XMLGateway_Interface1.wsdl

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.

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: * operations

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: With module for use by single application

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

Connection settings: Specify connection properties

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: * http://oraebs05.cn.ibm.com:8000/webservices/AppsWSProvider/oracle/ap

16. 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.

Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly.

Tracing the XML Gateway Web Service status

WebSphere Adapter for Oracle E-Business Suite does not handle the runtime exception and this exception is not reflected back to adapter by the Open interface. Use Oracle E-Business Suite administration console to find the XML Gateway Web Service status.

Problem:

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.

Solution

In order to check whether the requests are successfully processed by XML Gateway internal implementation, login to Oracle E-Business Suite administration console and find out the details in the Transaction Monitor.

Managed connection factory properties for XML Gateway

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 WebSphere Integration Developer assembly editor or, after deployment, with the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

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 6. Managed connection factory properties

Property name		
In the wizard	**In the administrative console**	**Description**
Oracle E-Business Suite Connection Type	EBSConnectionType	Specifies the Oracle E-Business Suite connection type.
Oracle E-Business Suite Transport	EBSTransport	Specifies the Oracle E-Business Suite transport.
Oracle E-Business Suite Version	EBSVersion	Specifies the Oracle E-Business Suite version.

Table 6. Managed connection factory properties (continued)

Property name		
In the wizard	**In the administrative console**	**Description**
Password	Password	Specifies the password required when invoking the web service.
User Name	UserName	Specifies the username for invoking the web service.
WSDL URL	WSDLURL	Specifies the WSDL URL of the XML Gateway interface.

Oracle E-Business Suite Connection Type (EBSConnectionType)

This property specifies the Oracle E-Business Suite connection type.

Table 7. 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 8. 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 9. 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.

Table 9. Oracle E-Business Suite Version details (continued)

Globalized	Yes
Bidi supported	No

Password (Password)

This property specifies the password required when invoking the web service.

Table 10. Password details

Required	If you set the Authentication alias, the password 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

User Name (UserName)

This property specifies the username required when invoking the web service.

Table 11. 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 12. WSDL URL details

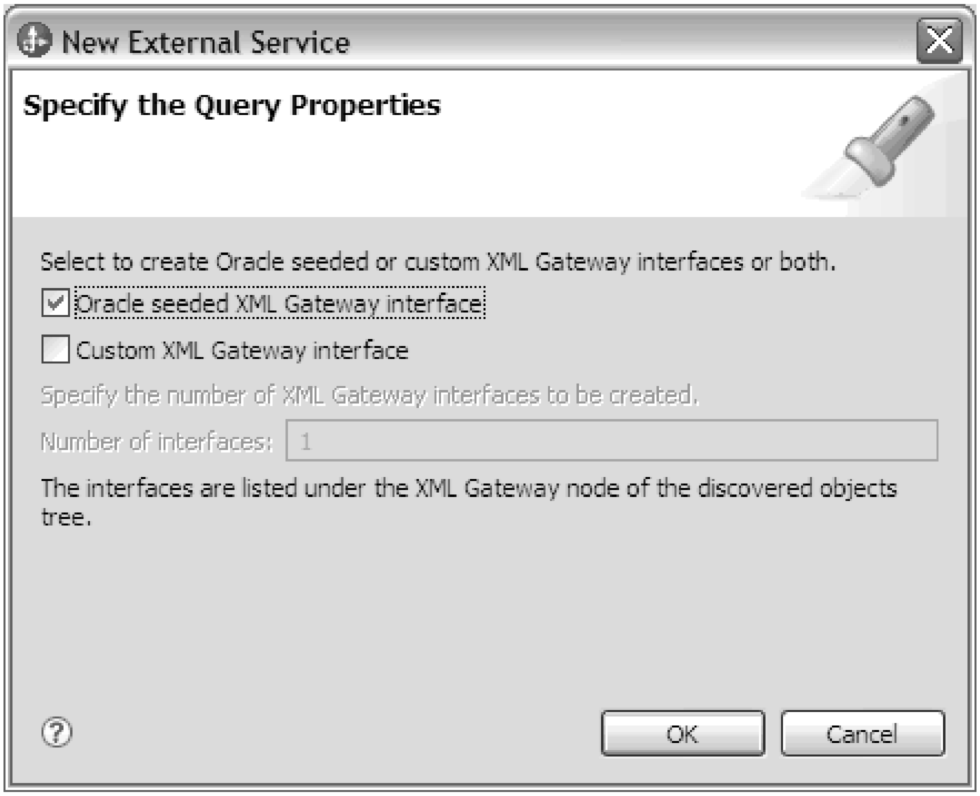
Required	Yes
Property type	String
Globalized	Yes
Bidi supported	No

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 retrieve the structure and metadata of different business service interfaces.

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

- Select the Oracle E-Business Suite XML Gateway interface type, Web Service transport and specific EBS version and enter the required security credentials to connect 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.



New External Service

Specify the Query Properties

Select to create Oracle seeded or custom XML Gateway interfaces or both.

☑ Oracle seeded XML Gateway interface

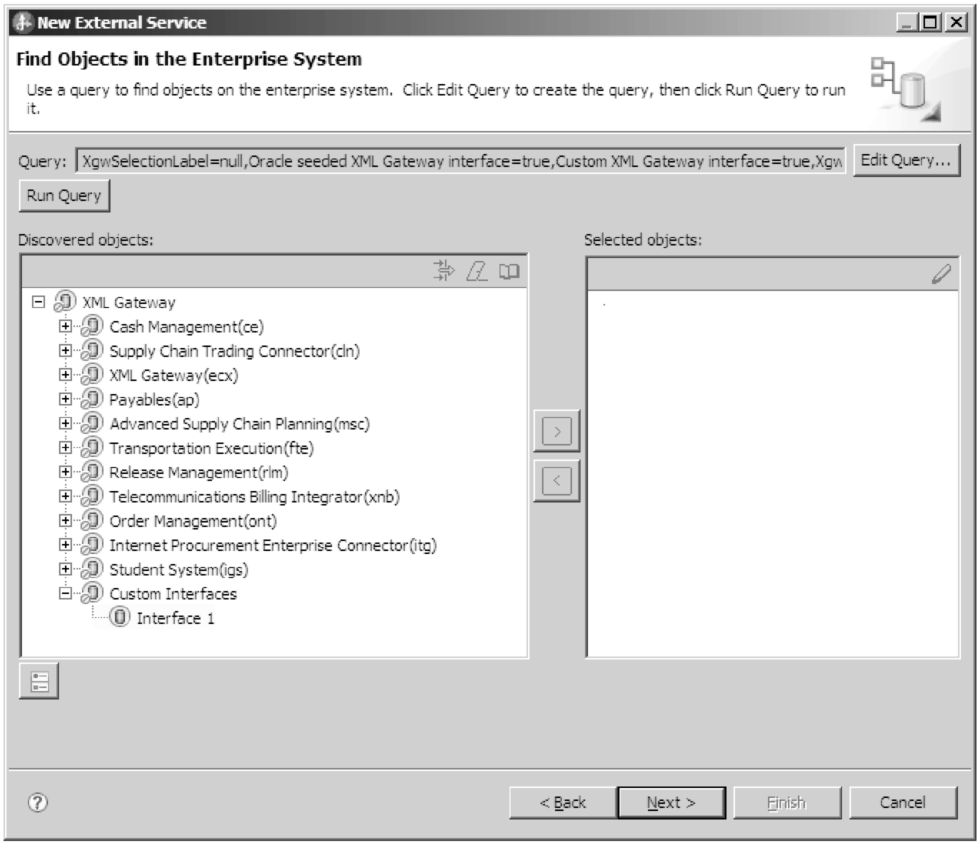
□ Custom XML Gateway interface

Specify the number of XML Gateway interfaces to be created.

Number of interfaces: 1

The interfaces are listed under the XML Gateway node of the discovered objects tree.

- 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 is displayed as shown in the following figure:



Find Objects in the Enterprise System

Use a query to find objects on the enterprise system. Click Edit Query to create the query, then click Run Query to run it.

Query: XgwSelectionLabel=null,Oracle seeded XML Gateway interface=true,Custom XML Gateway interface=true,Xgw

- XML Gateway
  - Cash Management(ce)
  - Supply Chain Trading Connector(cln)
  - XML Gateway(ecx)
  - Payables(ap)
  - Advanced Supply Chain Planning(msc)
  - Transportation Execution(fte)
  - Release Management(rlm)
  - Telecommunications Billing Integrator(xnb)
  - Order Management(ont)
  - Internet Procurement Enterprise Connector(itg)
  - Student System(igs)
  - Custom Interfaces
    - Interface 1

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

Support for Oracle PLSQL datatype BOOLEAN used in Oracle Stored Procedure parameters

Oracle PLSQL has some special data types like Record, BOOLEAN that are commonly used in Oracle E-Business Suite prebuilt stored procedures. The adapter processes these data types automatically.

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 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.

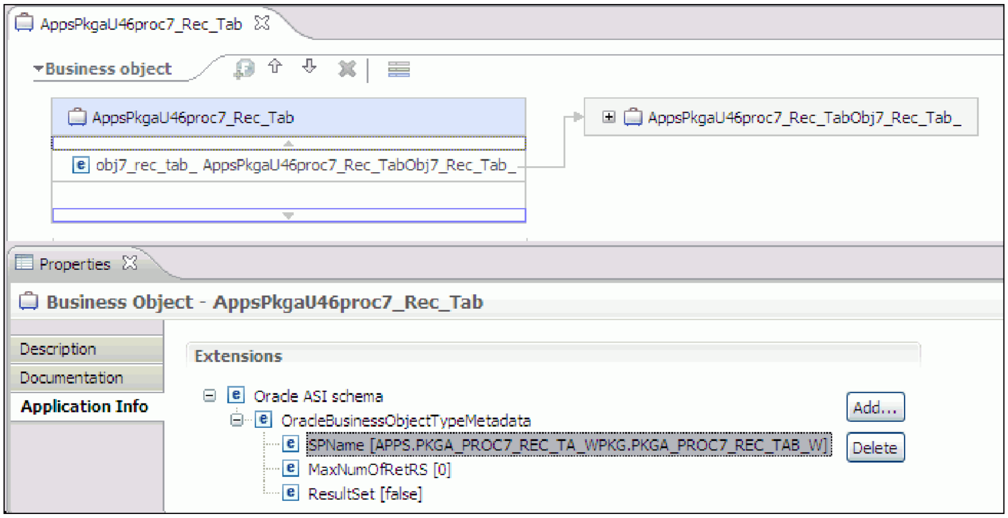
For validating the stored procedure attributes, a sample value parameter is provided with each attribute. The sample value parameter is provided for both simple and complex data type attributes. The wizard uses the sample values that you provide to validate the stored procedure before saving it. The 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 the 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 property SPComplexParameterTypeName.

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, PKGA_PROC7_REC_TAB_WPKG.PKGA_PROC7_REC_TAB_W, where "PKGA" 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.

The 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.



Business Object - AppsPkgaU46proc7_Rec_Tab

Oracle ASI schema
OracleBusinessObjectTypeMetadata
SPName [APPS.PKGA_PROC7_REC_TA_WPKG.PKGA_PROC7_REC_TAB_W]
MaxNumOfRetRS [0]
ResultSet [false]

If the stored procedure returns 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 13. 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. The wizard creates this number of business objects.

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 2 on page 32 lists the properties of a business object attribute and describes their interpretation and settings.

Table 14. Attribute properties

Properties	Interpretation and settings
Cardinality	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. 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.
Foreign Key	When arrays of child business objects whose cardinality is *n* are retrieved, foreign keys are used in the WHERE clause of SELECT statements. The RetrieveAll operation overrides the use of keys and foreign keys. **Note:** The adapter does not support specifying an attribute that represents a child business object as a foreign key.
Name	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.
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	Indicates whether this attribute is a primary key. At least one simple attribute in each business object must be specified as the primary key. 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. **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.

Table 14. Attribute properties (continued)

Properties	Interpretation and settings
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	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 3 on page 34. For attributes that specify a child business object, this property specifies the name of the business object.

The type of each database object, returned as the Oracle metadata, maps to the business object attribute types as listed in Table 3 on page 34. 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 15. 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

Table 15. Oracle metadata column type and business object attribute types (continued)

Oracle metadata column type	Business object attribute type
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 objects" on page 2
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.

"Create operation" on page 9
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.

"Update operation" on page 10
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.

Related tasks

"Selecting and configuring tables, views, and synonyms or nicknames for outbound processing" on page 12
Select and configure business objects for tables, views, and synonyms or nicknames for use in your module. For inbound processing, these are the business objects that are delivered in events.

"Selecting and configuring query business objects" on page 20
Select and configure query business objects for user-defined SELECT statements for use in your module.

"Selecting and configuring tables, views, and synonyms or nicknames for inbound processing" on page 23
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.

Support for overloaded SP/SF in Oracle database

WebSphere Adapter for Oracle E-Business Suite distinguishes the original SP/SF from the overloaded ones, and add the right parameters for the selected SP/SF.

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 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.

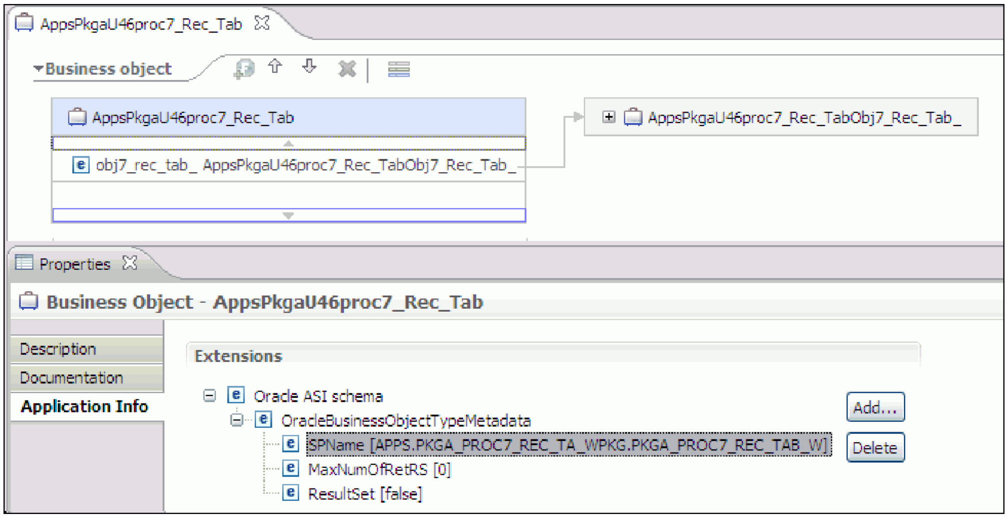
For validating the stored procedure attributes, a sample value parameter is provided with each attribute. The sample value parameter is provided for both simple and complex data type attributes. The wizard uses the sample values that you provide to validate the stored procedure before saving it. The 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 the 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 property SPComplexParameterTypeName.

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, PKGA_PROC7_REC_TAB_WPKG.PKGA_PROC7_REC_TAB_W, where "PKGA" 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.

The 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.



Oracle ASI schema
OracleBusinessObjectTypeMetadata
SPName [APPS.PKGA_PROC7_REC_TA_WPKG.PKGA_PROC7_REC_TAB_W]
MaxNumOfRetRS [0]
ResultSet [false]

If the stored procedure returns 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 16. 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. The wizard creates this number of business objects.

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.

The 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 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 17 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 61.

Table 17. 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.

Table 17. Application-specific information for stored procedures in table and view business objects (continued)

Descriptive name	Element name	Purpose
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 18. Stored procedure type element characteristics

Required	Yes
Default	None
Possible values	Can be one of: • Before*Operation*SP • After*Operation*SP • *Operation*SP *Operation* specifies one of the operation names: Create, Update, Delete, Retrieve, or RetrieveAll.
Bidirectional transformation supported	No
Property type	String
Usage notes	Stored procedure types associated with RetrieveAll apply to top-level business objects only. You can remove any selected application-specific information from the StoredProcedureType property. All the corresponding operation application-specific information property groups are also removed.
Examples	• 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 19. 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 20. Result set element characteristics

Required	Yes
Default	None
Possible values	`True` `False`
Bidirectional transformation supported	No
Property type	Boolean
Usage notes	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.

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 21. Parameters element characteristics

Required	Yes
Default	None
Contents	Each Parameters element specifies the following information: • 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: – `IP` for input only – `OP` for output only – `IO` for input and output – `RS` for result set

Table 21. Parameters element characteristics (continued)

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

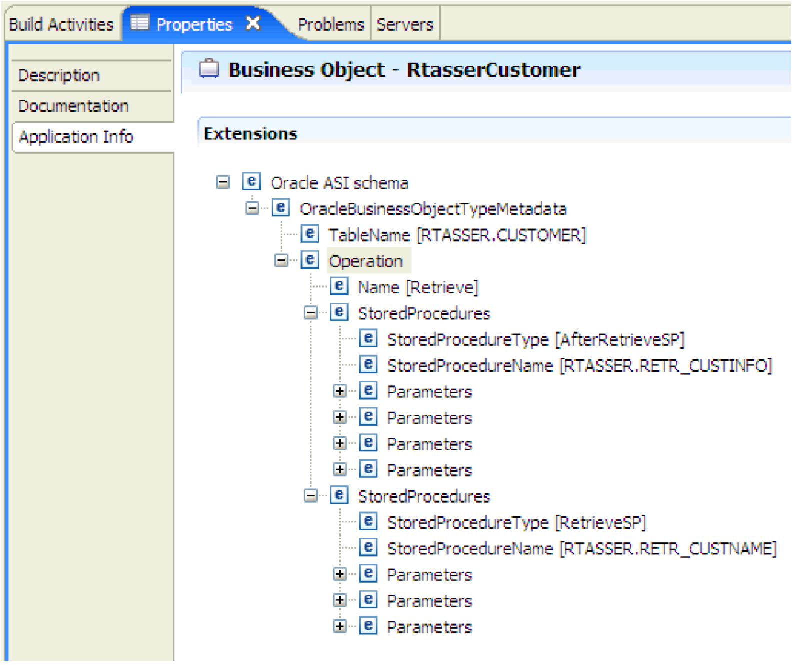
A value that indicates it is a function call, not a procedure call, because a value is returned.

Table 22. 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
Usage notes	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. 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 retuned by the function. 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. If the value of this application-specific information is another child business object, the adapter run time returns an error. 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. **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.

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.



Business Object - RtasserCustomer

Extensions

```
Oracle ASI schema
  OracleBusinessObjectTypeMetadata
    TableName [RTASSER.CUSTOMER]
    Operation
      Name [Retrieve]
      StoredProcedures
        StoredProcedureType [AfterRetrieveSP]
        StoredProcedureName [RTASSER.RETR_CUSTINFO]
        Parameters
        Parameters
        Parameters
        Parameters
      StoredProcedures
        StoredProcedureType [RetrieveSP]
        StoredProcedureName [RTASSER.RETR_CUSTNAME]
        Parameters
        Parameters
        Parameters
```

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.

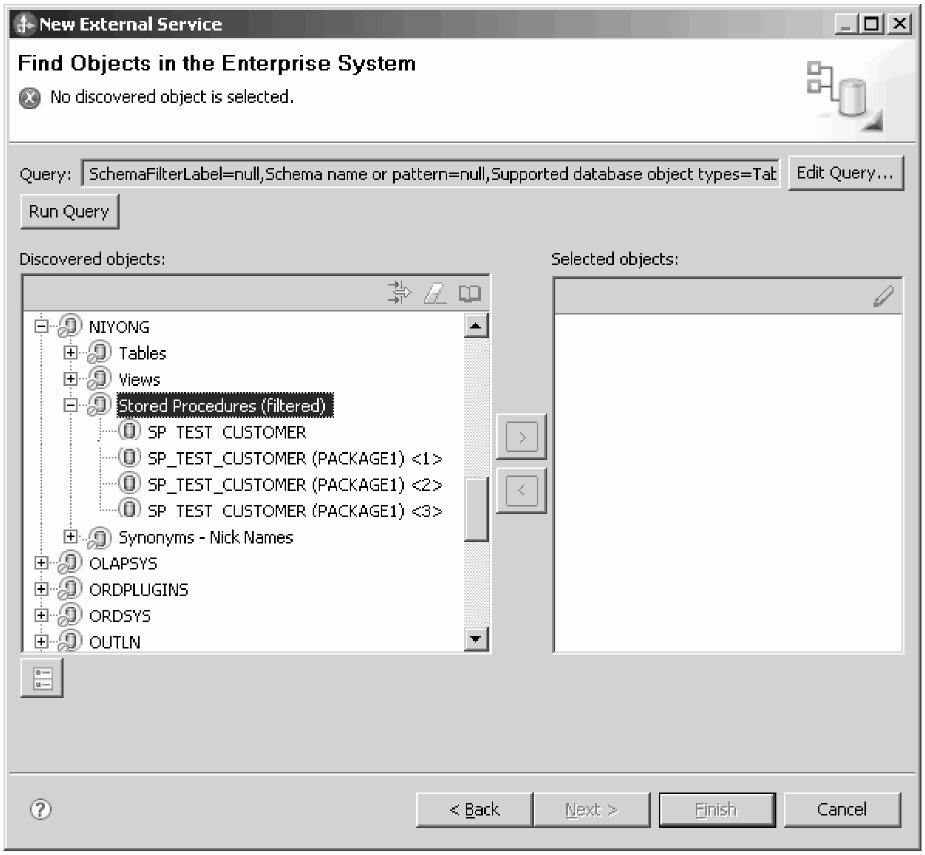
About this task

This task is performed through the external service wizard. You 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.

   The 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.



New External Service

Find Objects in the Enterprise System

No discovered object is selected.

Query: SchemaFilterLabel=null,Schema name or pattern=null,Supported database object types=Tab

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 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.

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.
6. The result of the validation is displayed in the **Result** area.

   Note: Make sure 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. Make sure 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 figure shows the window 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 global properties.

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 that can be reused in successive flows, are outlined below:

- 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 WebSphere Integration Developer 7.0 only. If you are importing a project interchange from an earlier version of WebSphere Integration Developer, 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 WebSphere Integration Developer.

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.

   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.

   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 12.

        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 61.

        Note: If the stored procedure definition in the database is changed, you must re-configure 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 20.

   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 35.

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 WebSphere Integration Developer 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.

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 WebSphere Integration Developer.

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.

   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.
   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 23.

        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.
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 WebSphere Integration Developer 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.

Results

The artifacts are regenerated.

What to do next

You can test and deploy your module.

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 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).

Printed in USA