

Developing OTDs for Oracle® Java CAPS Database Adapters

Copyright © 2008, 2011, Oracle and/or its affiliates. All rights reserved.

License Restrictions Warranty/Consequential Damages Disclaimer

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

Warranty Disclaimer

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

Restricted Rights Notice

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

U.S. GOVERNMENT RIGHTS

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

Hazardous Applications Notice

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

Trademark Notice

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

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

Third Party Content, Products, and Services Disclaimer

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

Contents

Developing OTDs for Database Adapters	5
About Database OTDs	6
Prepared Statements	6
Generating ResultSet Nodes of a Stored Procedure	6
DB2 Database Configuration Information	7
DB2 Connect Database Connection Information	9
JDBC Database Connection Information	10
Informix Database Connection Information	11
Oracle Database Connection Information	11
SQL Server Database Connectivity Information	11
Sybase Database Connectivity Information	11
VSAM Database Connectivity Information	12
Creating a Database OTD	12
Selecting the Wizard Type	12
Connecting to a Database	13
Selecting Database Objects	15
Selecting Table/Views/Aliases	16
Selecting Procedures	21
Adding Prepared Statements	26
Specifying the OTD Name	31
Review Selections	32
Editing Existing OTDs	33
▼ To Edit an Existing OTD	33

Developing OTDs for Database Adapters

The Database OTD Wizard generates OTDs by connecting to external data sources and creating corresponding Object Type Definitions. The OTD Wizard can create OTDs based on any combination of Tables, Stored Procedures, or Prepared Statements.

Field nodes are added to the OTD based on the Tables in the external data source. Java method and parameter nodes are added to provide the appropriate JDBC functionality. For more information about the Java methods, refer to your JDBC developer's reference. Each of the database adapters below provides additional support, such as the Sybase adapter which supports double-byte character set (DBCS). The DBCS is a set of characters in which each character is represented by 2 bytes. The Korean language requires double-byte character sets.

The following sections provide information and instructions on how to create a database OTD.

What You Need to Know

The following topics provide information you should know before creating an OTD for your database type.

- [“About Database OTDs” on page 6](#)
- [“Prepared Statements” on page 6](#)
 - [“Prepared Statements” on page 6](#)
- [“Generating ResultSet Nodes of a Stored Procedure” on page 6](#)
- [“DB2 Database Configuration Information” on page 7](#)
- [“DB2 Connect Database Connection Information” on page 9](#)
- [“JDBC Database Connection Information” on page 10](#)
- [“Informix Database Connection Information” on page 11](#)
- [“Oracle Database Connection Information” on page 11](#)
- [“SQL Server Database Connectivity Information” on page 11](#)
- [“Sybase Database Connectivity Information” on page 11](#)

- [“VSAM Database Connectivity Information” on page 12](#)

What You Need to Do

The following topics provide instructions for creating and editing Database OTDs.

- [“Creating a Database OTD” on page 12](#)
- [“Editing Existing OTDs” on page 33](#)

About Database OTDs

The Database OTD Wizard generates OTDs by connecting to external data sources and creating corresponding Object Type Definitions. The OTD Wizard can create OTDs based on any combination of Tables and Stored Procedures or Prepared SQL Statements. Field nodes are added to the OTD based on the Tables in the external data source. Java method and parameter nodes are added to provide the appropriate JDBC functionality. For more information about Java methods, refer to your JDBC developer’s reference guide.

Prepared Statements

A Prepared Statement OTD represents a SQL statement that has been compiled. Fields in the OTD correspond to the input values that users need to provide. Prepared statements can be used to perform insert, update, delete and query operations. A prepared statement uses a question mark (?) as a place holder for input. For example: insert into EMP_TAB (Age, Name, Dept No) values (?, ?, ?).

To execute a prepared statement, set the input parameters and call `executeUpdate()` and specify the input values if any.

When using a Prepared Statement, the `ResultsAvailable()` method will always return true. Although this method is available, you should not use it with a while loop. Doing so would result in an infinite loop at runtime and will stop all of the system’s CPU. If it is used, it should only be used with an if statement. You can process a `ResultSet` by looping through the `next()` method.

Generating ResultSet Nodes of a Stored Procedure

The OTD Wizard provides three different ways to generate the `ResultSet` nodes of a Stored Procedure. They are the By Executing, Manually, and With Assistance modes.

By Executing Mode	<p>The By Executing mode executes the specified Stored Procedure with default values to generate the ResultSets. Depending on the business logic of the Stored Procedure, zero or more ResultSets can be returned from the execution. In the case that there are multiple ResultSets and the By Executing mode does not return all ResultSets, one should use the other modes to generate the ResultSet nodes.</p> <p>If you modify the ResultSet generated by the By Executing mode, you need to make sure the indexes match the Stored Procedure. This assures your ResultSet indexes are preserved.</p>
With Assistance Mode	<p>The With Assistance mode allows users to specify a query and execute it to generate the ResultSet node. To facilitate this operation, the wizard tries to retrieve the content of the specified Stored Procedure and display it. However, content retrieval is not supported by all types of Stored Procedures. We can roughly classify Stored Procedures into two types: SQL and external. SQL Stored Procedures are created using CREATE PROCEDURE SQL statements while external Stored Procedures are created using host languages such as Java. Since external Stored Procedures do not store their execution plans in the database, content retrieval is impossible. When using this mode, highlight the execute statement up to and including the table name(s) before executing the query.</p>
Manually Mode	<p>The Manually mode is the most flexible way to generate the result set nodes. It allows users to specify the node name, original column name and data type manually. One drawback of this method is that users need to know the original column names and data types. This is not always possible. For example, the column name of 3*C in this query.</p> <pre>SELECT A, B, 3*C FROM table T</pre> <p>is generated by the database. In this case, "With Assistance" mode is a better choice.</p>

DB2 Database Configuration Information

The following topics contain information you should know when creating a DB2 OTD.

DB2 Database Connectivity Information

For DB2 OTDs, fill in the following fields for the database connection information:

- **Host Name:** The server where DB2 resides.
- **Port:** The port number of DB2.
- **Location** (AS/400 and z/OS specific): The name of the DB2 subsystem. To find the location of the DB2 subsystem, use the Database Query Tool to issue the following query: `select current server from sysibm.sysdummy1.`
- **Database** (Windows/UNIX specific): The name of the database instance.
- **Collection** (AS/400 and z/OS specific): The name that identifies a group of packages. For more information on Packages, refer to [“DB2 and Binding Packages” on page 8.](#)

- **User Name:** The user name that the adapter uses to connect to the database.
- **Password:** The password used to access the database.
- An **Optional Parameters** field (only for AS/400 and z/OS) will display in the OTD Wizard if the `DB2_ConnectionInfo.txt` file is present in the `JavaCAPS_Home\.netbeans\usrdir\modules\ext\db2adapter` directory. This field allows additional connection parameters for the DB2 OTD wizard. Please contact Oracle Corporation. for more information on Optional Parameters.

Note – This parameter is not specific to an OTD. It's only specific to the session where Java CAPS IDE runs. When you edit the OTD, the current value that shows up is the value that was entered previously in the Wizard. It may or may not be the same value you used for creating the original OTD.

Parameter examples include:

- The **showSelectableTables=false** parameter can be set to false (default is true) to see additional tables listed under a user, such as the behavior in a previous Adapter which uses an older version of the driver.
- The **AlternateID=user1** parameter allows you to set the object owner to be user1 otherwise the user in the User Name field will be used.
- Multiple parameters can be used and must be separated by a semi-colon (;).

DB2 and Binding Packages

This Adapter uses a DataDirect driver (previously known as Merant) to execute SQL calls in DB2. The DataDirect driver requires packages to be created in the DB2 System. Packages do not contain specific SQL statements like static SQL packages but rather dynamic sections, used like cursors to help facilitate the driver's executing of dynamic SQL queries and returning results.

Creating packages on the server, also known as binding packages, needs only be done once. The first user of the OTD Wizard must have bind permission to create the packages. Without bind authority the user receives an error message when the driver attempts to bind the packages and they will be unable to issue any SQL call. Packages are created automatically, under the Collection ID, when the user fills in the Wizard entries. If the Collection ID is left as blank, it will generate the packages under NULLID.

The driver creates SQL packages on the database including: DDJC330A, DDJC330B, DDJC330C, and so on. When connecting, the driver queries a system table to determine whether the default packages exist on the system. If none exist, the driver creates them.

Note – SQL applications that execute dynamic SQL against DB2 need to have packages bound on the server. In the case of some IBM native tools this may not be obvious because the packages are already installed on the database by default.

DB2 Connect Database Connection Information

The connection information to specify for a DB2 Connect database varies depending on the type of connection you use, Type 2 or Type 4.

For the Type 2 Connection (shown above):

- **Database:** The name of the database instance.
- **User Name:** The user name that the Adapter uses to connect to the database.
- **Password:** The password used to access the database.

For the Type 4 Connection (shown above):

- **Host Name:** The server where DB2 Connect resides.
- **Port:** The port number of DB2 Connect.
- **Database:** The name of the database instance that the DB2 is running on Window/UNIX. The name of the locator for the DB2 running in z/OS or AS/400.
- **User Name:** The user name that the adapter uses to connect to the database.
- **Password:** The password used to access the database.

JDBC Database Connection Information

The required database connection fields for a JDBC OTD include the following:

- **Driver Jar Files:** The location of the driver JAR file.
- **Driver Java Class Name:** The name of the Driver Manager Class.
- **URL Connection String:** The URL connection string for the driver.
- **User Name:** A valid JDBC database username.
- **Password:** The password for the user name noted above.

Informix Database Connection Information

The required database connection fields for an Informix OTD include the following:

- **Host name:** The name of the host to which you are connecting.
- **Port ID:** The host port number (1526 is the default).
- **Informix Server:** The name of the Informix server.
- **Database name:** The name of the database to which you are connecting.
- **User name:** A valid Informix database username.
- **Password:** A password for the user name noted above.

Oracle Database Connection Information

The required database connection fields for an Oracle OTD include the following:

- **Host Name:** The server where Oracle resides.
- **Port ID:** The port number of Oracle.
- **SID:** The name of the Oracle instance (equivalent to the database name).
- **User Name:** The user name that the Adapter uses to connect to the database.
- **Password:** The password used to access the database.

SQL Server Database Connectivity Information

The required database connection fields for a SQL Server OTD include the following:

- **Host Name:** The database service host name.
- **Port ID:** The database service connection port ID/number.
- **Database name:** the name of the SQL Server database.
- **User Name:** A valid SQL Server database username.
- **Password:** The password for the user name noted above.

Sybase Database Connectivity Information

The required database connection fields for a Sybase OTD include the following:

- **Host Name:** The database service host name.
- **Port ID:** The database service connection port ID/number.
- **Database Name:** The name of the Sybase database.
- **User Name:** A valid Sybase database username.
- **Password:** The password for the user name noted above.

VSAM Database Connectivity Information

The required database connection fields for a VSAM OTD include the following:

- **Host Name:** The database service host name.
- **Port ID:** The database service connection port ID/number.
- **Database Name:** The name of the VSAM database.
- **User Name:** A valid VSAM database username.
- **Password:** The password for the user name noted above.

Creating a Database OTD

The Database OTD Wizard generates OTDs by connecting to external data sources and creating corresponding Object Type Definitions. The OTD Wizard can create OTDs based on any combination of Tables and Stored Procedures or Prepared SQL Statements. Field nodes are added to the OTD based on the Tables in the external data source. Java method and parameter nodes are added to provide the appropriate JDBC functionality. For more information about the Java methods, refer to your JDBC developer's reference.

Note – Database OTDs are not messageable. The images in the following procedure reflect an Oracle OTD, and might appear different from the images you see depending on the type of database OTD you are creating.

Perform the following steps to create a new database OTD:

- [“Selecting the Wizard Type” on page 12](#)
- [“Connecting to a Database” on page 13](#)
- [“Selecting Database Objects” on page 15](#)
- [“Selecting Table/Views/Aliases” on page 16](#)
- [“Selecting Procedures” on page 21](#)
- [“Adding Prepared Statements” on page 26](#)
- [“Specifying the OTD Name” on page 31](#)
- [“Review Selections” on page 32](#)

Selecting the Wizard Type

Select the type of wizard required to build an OTD in the New Object Type Definition Wizard.

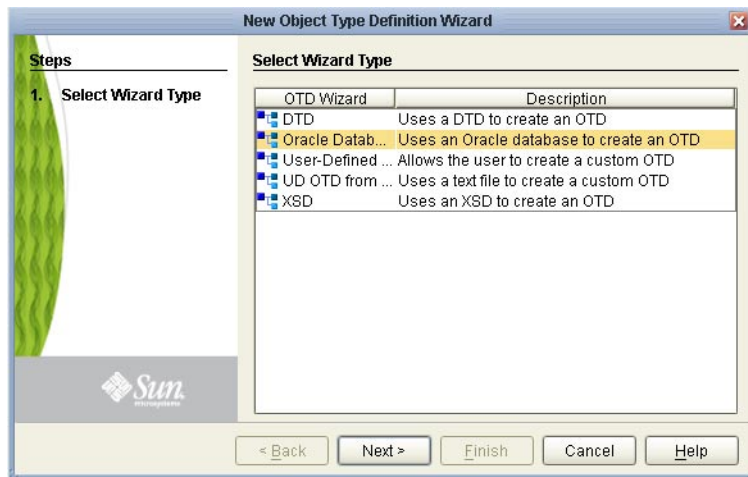
▼ To Select the Database OTD Wizard

- 1 On the Project tree, right-click the Project, point to New, and then select Object Type Definition from the shortcut menu.

The New Object Type Definition Wizard appears with the Select Wizard Type page displayed.

- 2 Select the type of database to use from the list of available types.

FIGURE 1 OTD Wizard Selection



- 3 Click Next.

The Specify Database Connection Information window appears.

Next Steps Proceed to the following step, [“Connecting to a Database” on page 13](#).

Connecting to a Database

In order to create a database OTD, you must first connect to the database for which you are creating it by specifying the connection information.

▼ To Connect to a Database

Before You Begin Complete the previous step, “[Selecting the Wizard Type](#)” on page 12.

1 In the Connection drop-down list, select the type of connection to use.

The resulting connection information fields that appear depend on your selection.

2 Specify the applicable connection information (depending on the Connection type) for your database.

The connection information for each database type is different. See the following topics for more information.

- “[DB2 Database Connectivity Information](#)” on page 7
- “[DB2 Connect Database Connection Information](#)” on page 9
- “[JDBC Database Connection Information](#)” on page 10
- “[Informix Database Connection Information](#)” on page 11
- “[Oracle Database Connection Information](#)” on page 11
- “[SQL Server Database Connectivity Information](#)” on page 11
- “[Sybase Database Connectivity Information](#)” on page 11
- “[VSAM Database Connectivity Information](#)” on page 12

3 Click Next.

The Select Database Objects window appears.

Next Steps Proceed to the following step, [“Selecting Database Objects” on page 15.](#)

Selecting Database Objects

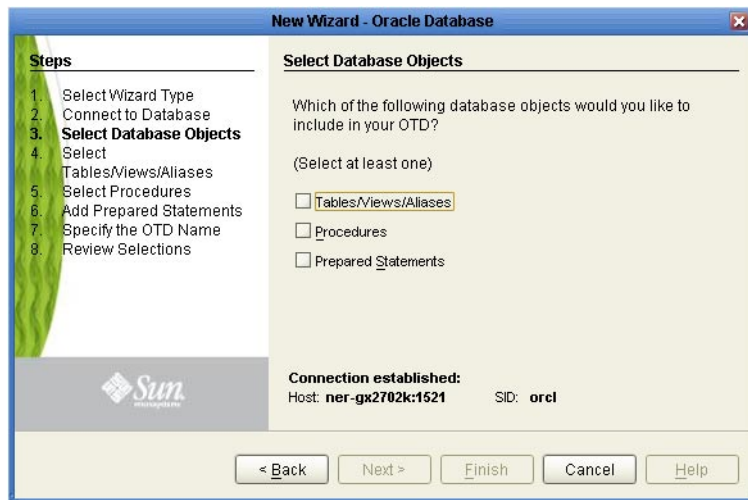
You can select any combination of Tables, Views, Procedures, or Prepared Statements that you want to include in the Database OTD.

▼ To Select Database Objects

Before You Begin Complete the previous step, [“Connecting to a Database” on page 13.](#)

- 1 **Select any combination of Tables/Views/Aliases, Procedures, and Prepared Statements.**

Note – Views are read-only and are for informational purposes only.



- 2 **Click Next to continue.**

The next window to appear depends on the combination of databases objects you selected.

Next Steps Do one of the following:

- If you selected Tables/Views/Aliases, continue to [“Selecting Table/Views/Aliases” on page 16.](#)
- If you did not select Tables/Views/Aliases but selected Procedures, skip to [“Selecting Procedures” on page 21.](#)

- If you selected only Prepared Statements, skip to [“Adding Prepared Statements”](#) on page 26.

Selecting Table/Views/Aliases

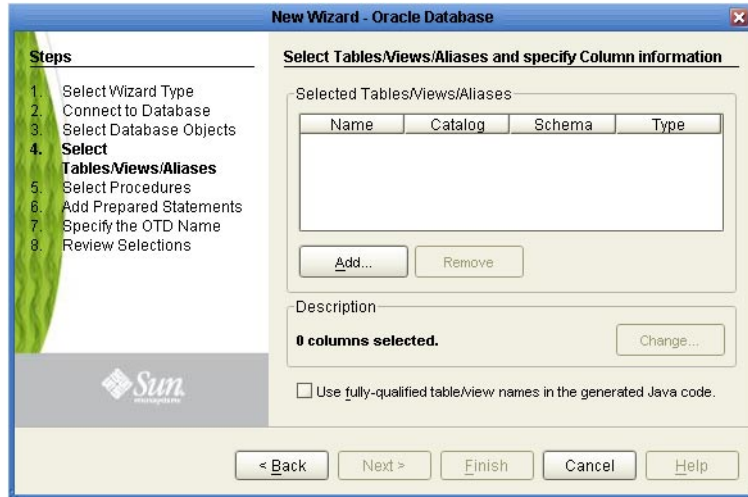
Select the types of tables or views required in the OTD.

Note – Aliases are not supported in the current release of the Oracle Adapter.

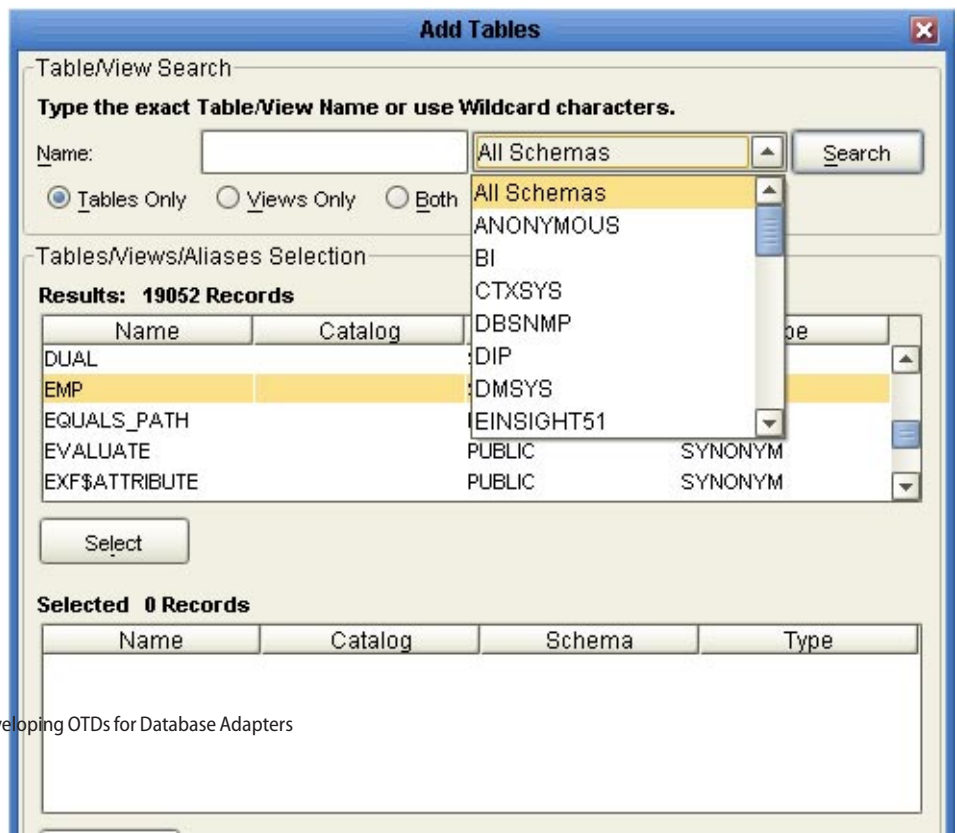
▼ To Select Table/Views/Aliases

Before You Begin Complete the previous step, “Selecting Database Objects” on page 15.

- 1 In the Select Tables/Views/Aliases window, click Add.



The Add Tables window appears.



2 In the Add Tables window, select the type of criteria to use for your search.

Search criteria can consist of table data, view only data, or both. You can include system tables in your search by selecting the check box.

3 From the Table/View Name drop down list, select the location of your database table and click Search.

Search for Table/View Names by entering a table name. The use of wildcard characters of "?", and "*" as part of your Table/View name search allow for greater search capabilities. For example, "AB?CD" or "AB*CD".

4 Select a table and then click OK.

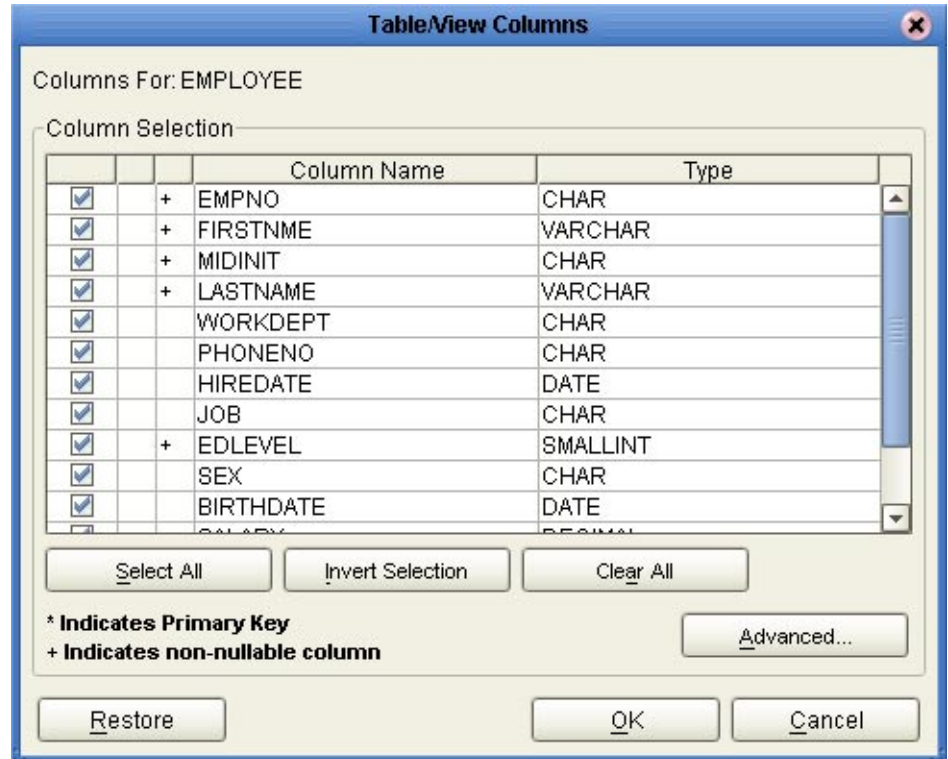
The table selected is added to the Selected window.

5 Repeat the previous two steps until you have selected all needed tables.

6 Review the tables you have selected. To make changes to the selected Table or View, click Change and do any of the following:

- a. **Select or deselect the table columns on the Table/View Columns window. Change the data type for each table by highlighting the data type and selecting a different one from the drop down.**

Note – The data type is usually listed as Other when the driver cannot detect the data type. In these situations, you should change the data type to one that is more appropriate for the type of column data.



- b. To change the data type, precision/length, or scale, click Advanced. In general, do not change the precision/length or the scale.

Note – The Informix database driver currently deployed with the Informix adapter displays non-nullable columns in the OTD wizard dialogue box regardless of whether the columns in the database accept null values or not. When using Prepared Statement packages, select Use Fully Qualified Table/View Names in the Generated Java Code.

Table/View Columns: Advanced

Additional column properties for EMPLOYEE

Column Selection

		Column	SQL type	Precision / len...	Scale
<input checked="" type="checkbox"/>	+	EMPNO	CHAR	6	0
<input checked="" type="checkbox"/>	+	FIRSTNAME	VARCHAR	12	0
<input checked="" type="checkbox"/>	+	MIDINIT	CHAR	1	0
<input checked="" type="checkbox"/>	+	LASTNAME	VARCHAR	15	0
<input checked="" type="checkbox"/>		WORKDEPT	CHAR	3	0
<input checked="" type="checkbox"/>		PHONENO	CHAR	4	0
<input checked="" type="checkbox"/>		HIREDATE	DATE	10	0
<input checked="" type="checkbox"/>		JOB	CHAR	8	0
<input checked="" type="checkbox"/>	+	EDLEVEL	SMALLINT	5	0
<input checked="" type="checkbox"/>		SEX	CHAR	1	0
<input checked="" type="checkbox"/>		BIRTHDATE	DATE	10	0
<input checked="" type="checkbox"/>		SALARY	DECIMAL	9	2
<input checked="" type="checkbox"/>		BONUS	DECIMAL	9	2
<input checked="" type="checkbox"/>		COMM	DECIMAL	9	2

* Indicates Primary Key
+ Indicates non-nullable column

Select All Invert Selection Clear All

Restore OK Cancel

- 7 When you are done making changes, click OK.
- 8 On the Select Tables/Views/Aliases window, select or deselect Use Fully-Qualified Table/View Names in the Generated Java Code.

Note – When you use fully-qualified names, the schema name or database name is prepended to the table name in the Java Code. For example, a table named `employee` in a schema named `APM` would appear as `APM.employee` in the generated Java code.

Whether to use fully-qualified names depends largely on your internal business practices. Using fully-qualified names may result in improved performance, and it allows users to access tables outside of their default schemas. However, if the database schema changes in the future, the Java code could be broken. There are many more considerations to take into account. See your database documentation for more information.

9 Click Next.

The next window to appear depends on the database objects you chose.

Next Steps Do one of the following:

- If you selected Procedures, skip to [“Selecting Procedures” on page 21](#).
- If you did not select Procedures but did select Prepared Statements, skip to [“Adding Prepared Statements” on page 26](#).
- If you did not select either, skip to [“Specifying the OTD Name” on page 31](#).

Selecting Procedures

Select the procedure required in the OTD.

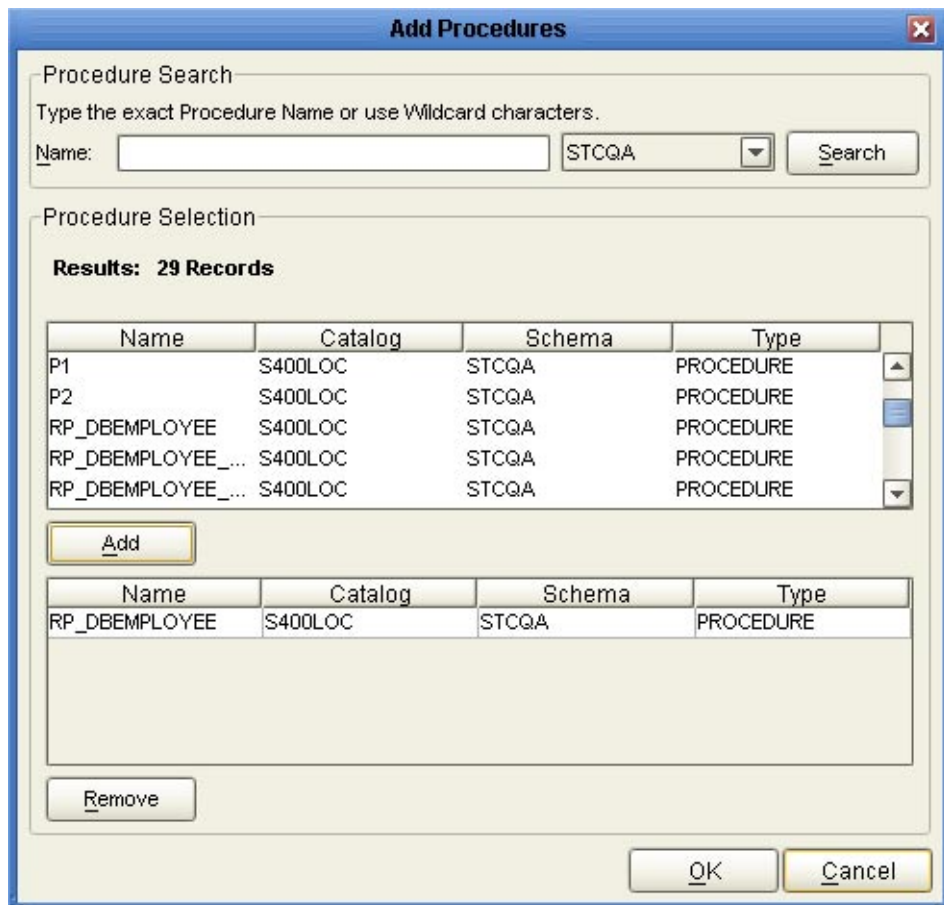
Note – A fully qualified Procedure name consists of two parts: Schema Name, and the Stored Procedure Table Name. Code that is generated in a Java Collaboration Definition appears fully qualified when the Use Fully-Qualified Procedure Names in the Generated Java Code check box is selected. When this checkbox is not selected, then only the Stored Procedure Table Name appears.

▼ To Select Procedures

Before You Begin Complete the previous step “[Selecting Database Objects](#)” on page 15, and optionally, “[Selecting Table/Views/Aliases](#)” on page 16.

- 1 On the Select Procedures window, click Add.

The Add Procedures window appears.

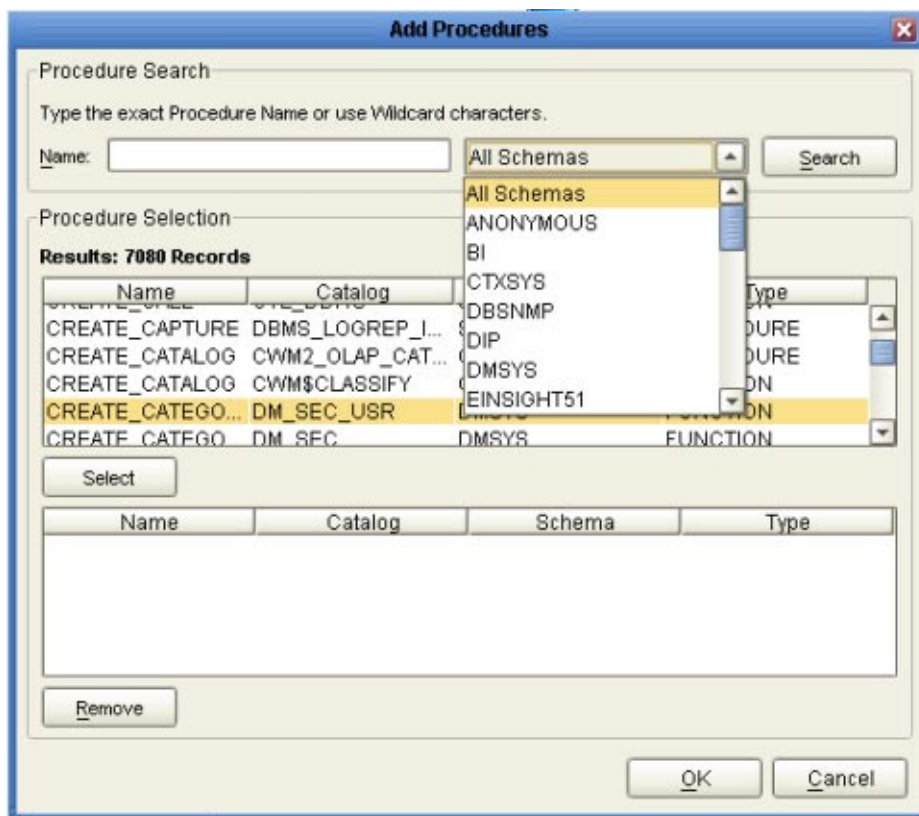


- 2 On the Add Procedures window, enter the name of a Procedure or select a table from the drop down list, and then click Search.

You can use wildcard characters in this search.

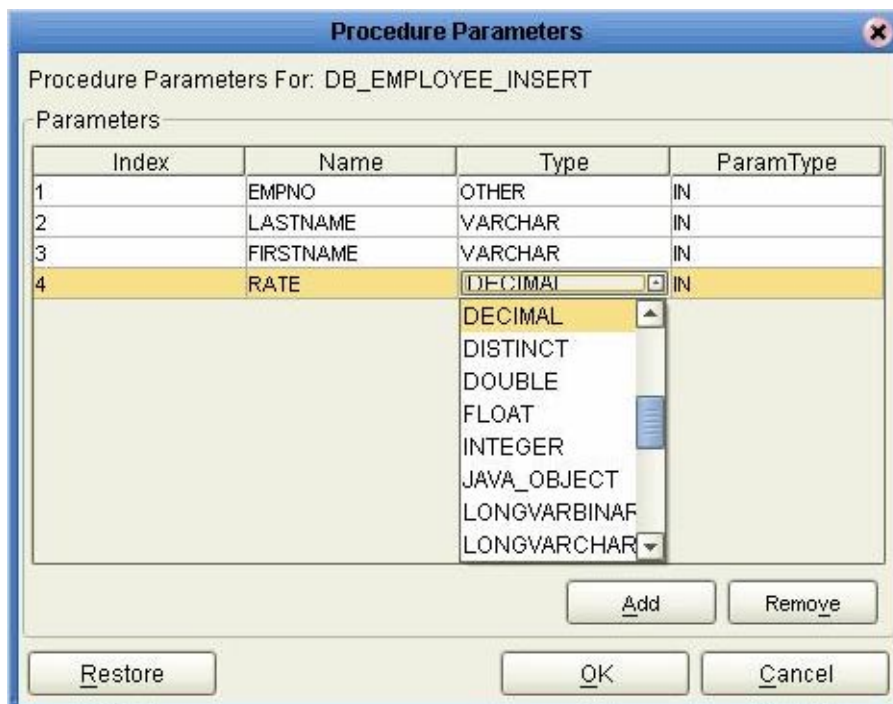
Note – For Informix, you must use lower case schema names when calling stored procedures.

- 3 In the resulting Procedure Selection list box, select a Procedure and then click OK.



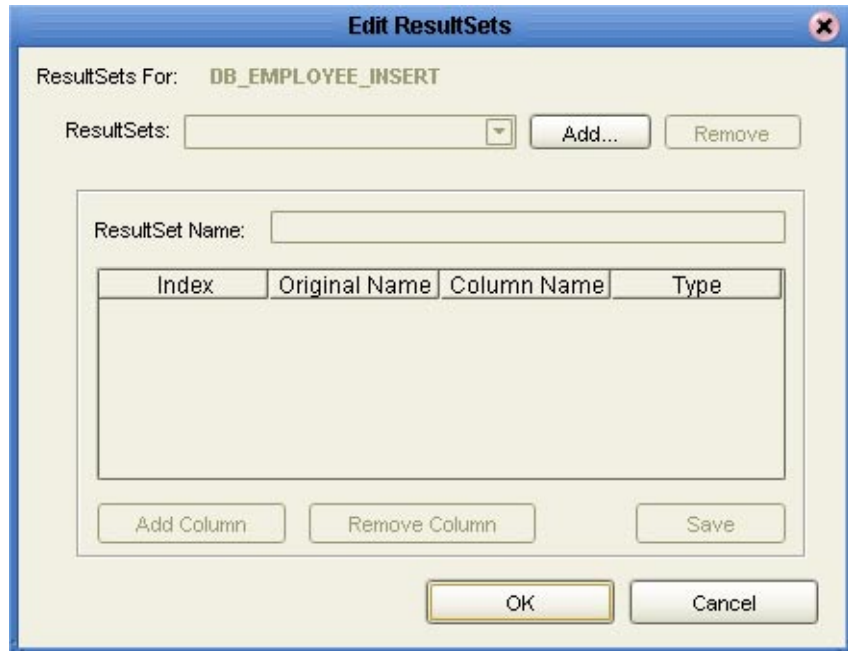
- 4 To make any changes to the selected Procedure, click Edit Parameters on the Select Procedures window.

The Procedure Parameters window appears.



- 5 To restore the data type, click Restore. When you are done, click OK.
- 6 To specify how the OTD should generate the nodes for the ResultSet, click Edit ResultSets. The Edit ResultSets window appears.

- 7 Click Add to add the type of ResultSet node you would like to generate. Click OK when you are done.



For more information about Add types, see [“Generating ResultSet Nodes of a Stored Procedure” on page 6](#).

- 8 On the Select Procedures, click Next to continue.

The Add Prepared Statements window appears.

Note – Keep the following in mind when working with ResultSets:

- Not all drivers and databases support stored procedures or stored procedures with ResultSets. Contact the driver’s vendor for more information.
- In some situations, stored procedures that uses a Select statement may not return a ResultSet when a conditional statement is used. In this case, no data is returned when the `next()` method is called, even if the `ResultSet available()` method returns “true”. This result is consistent with the type of driver we use.
-

Note – When you use Insert, Update, and Delete operations, in addition to using Select, the stored procedure will return results for each operation used. We recommend invoking the `enableResultSetsOnly()` method if you only want to return a `ResultSet` for the Select statement.

Next Steps Do one of the following:

- If you selected Prepared Statements, continue to [“Adding Prepared Statements” on page 26](#).
- If you did not select Prepared Statements, skip to [“Specifying the OTD Name” on page 31](#).

Adding Prepared Statements

A Prepared Statement OTD represents a SQL statement that has been compiled. Fields in the OTD correspond to the input values that users need to provide. Prepared statements can be used to perform insert, update, delete and query operations. A prepared statement uses a question mark (?) as a place holder for input. For example: `insert into EMP_TAB (Age, Name, Dept No) values (?, ?, ?)`.

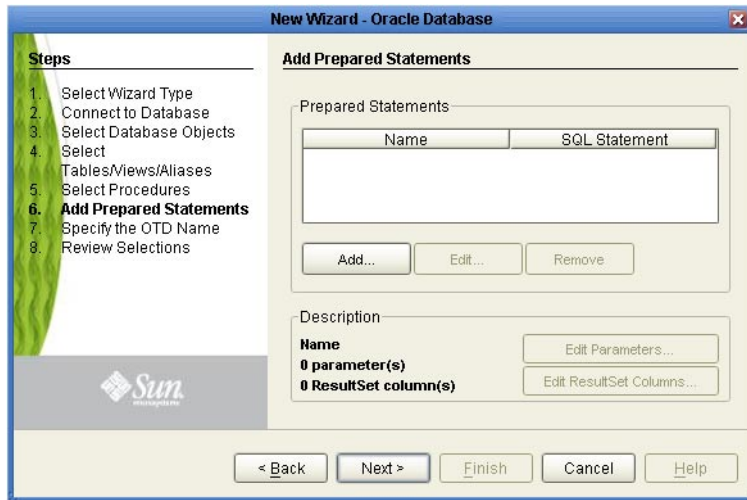
▼ To Add Prepared Statements

To execute a prepared statement, set the input parameters and call `executeUpdate()` and specify the input values if any.

Note – When using a Prepared Statement, the `ResultsAvailable()` method will always return true. Although this method is available, you should not use it with a while loop. Doing so would result in an infinite loop at runtime and will stop all of the system’s CPU. If it is used, it should only be used with an if statement. You can process a `ResultSet` by looping through the `next()` method.

Before You Begin Complete the steps under [“Selecting Database Objects” on page 15](#), and optionally under [“Selecting Table/Views/Aliases” on page 16](#) and [“Selecting Procedures” on page 21](#).

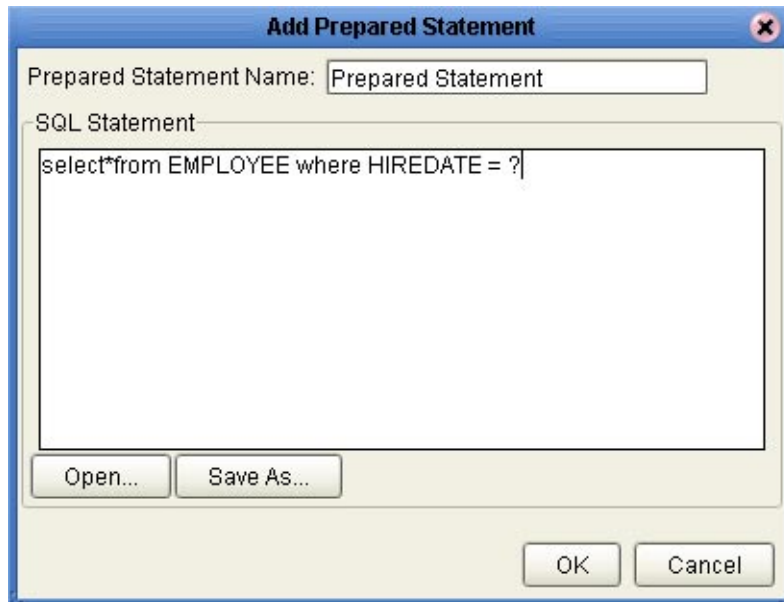
- 1 On the Add Prepared Statements window, click Add.



The Add Prepared Statement dialog appears.

- 2 Enter the name of a Prepared Statement or create a SQL statement by typing directly into the SQL Statement window.

Note – If you are creating a VSAM OTD and are not logging into the VSAM database with the default user name, you must enter the table schema name in the SQL Prepared Statement.



3 Click Save As, and provide a name for the statement.

This name will appear as a node in the OTD.

4 Click OK.

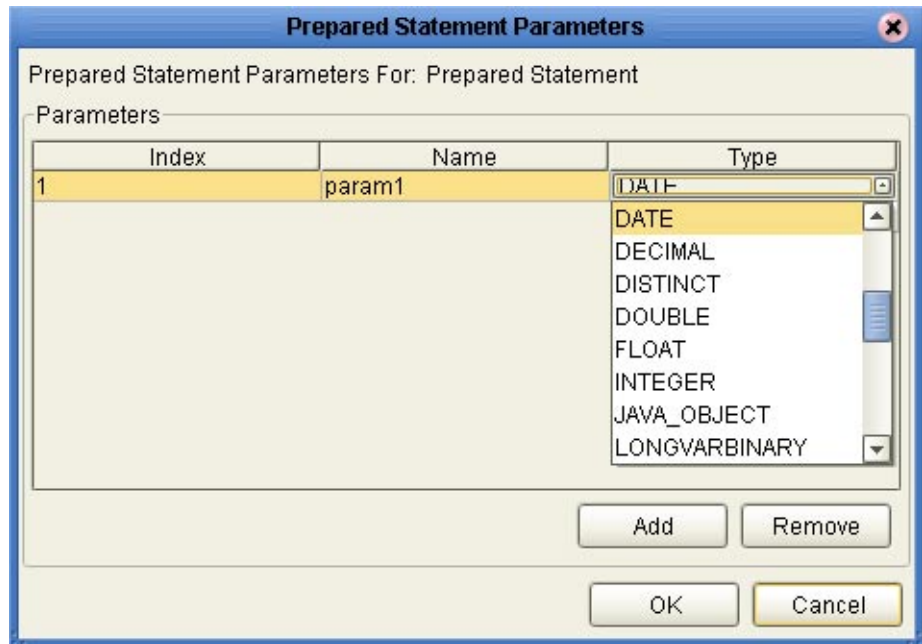
The name you assigned to the Prepared Statement appears on the Add Prepared Statements window of the wizard.

5 Repeat the previous steps as needed to add all Prepared Statements.

6 To edit statement parameters, select the statement and then click Edit Parameters.

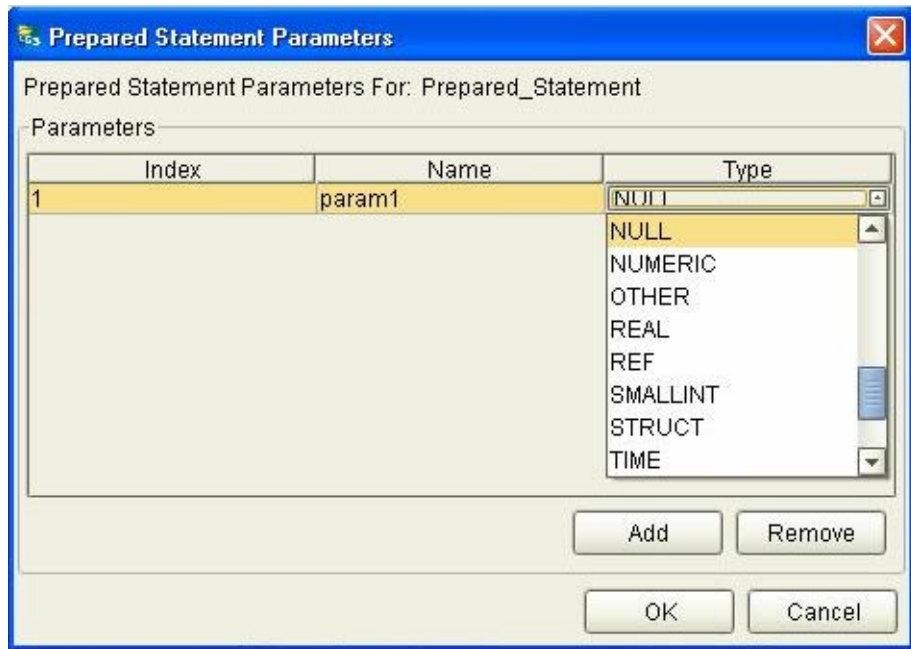
The Prepared Statement Parameters window appears.

Note – When defining a Prepared Statement with two or more tables, where multiple tables have the same column name, you must put the table name qualifier in the Prepared Statement to build the OTD.



- 7 To change the data type, click in the Type field and select a different type from the list.

- 8 To add more parameters, click Add and then enter the required information. To remove a parameter, highlight a row and click Remove.

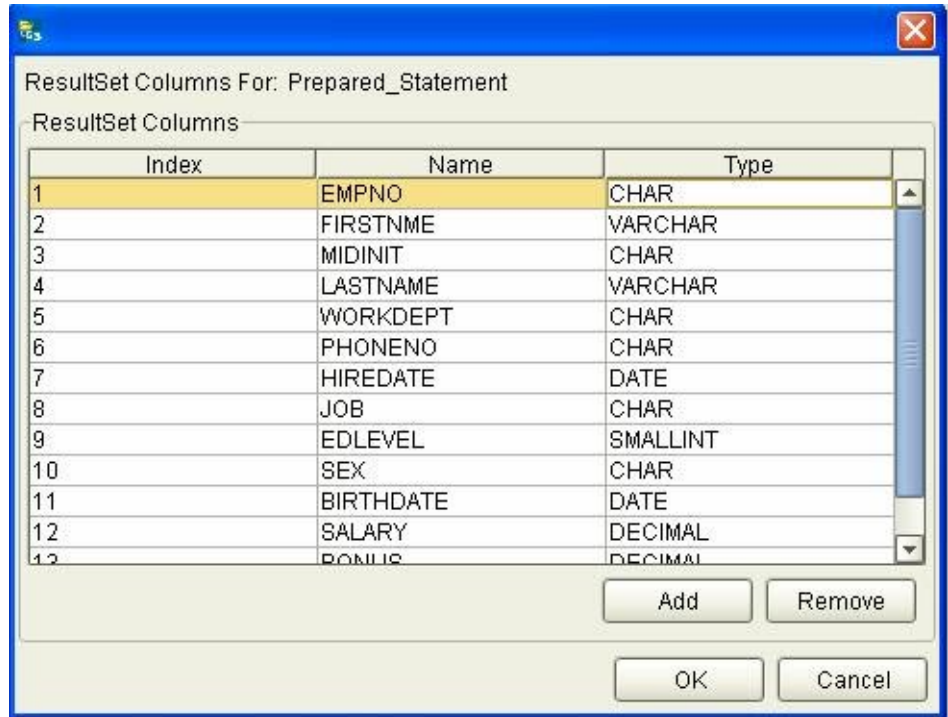


Note – Once you save the Prepared Statement, make sure that the ResultSet Column Name in the Prepared Statement parameters is a valid alphanumeric string with no special characters (for example, no brackets).

- 9 Click OK.
- 10 To edit the ResultSet Columns, click Edit ResultSet Columns.

Both the Name and Type are editable but it is recommend you do not change the Name. Doing so will cause a loss of integrity between the ResultSet and the Database.

Note – The OTD Wizard cannot create OTDs with complex prepared statements that use the same column name in different tables. This problem is resolved by modifying the SQL statement to use column name aliases.



- 11 To add a ResultSet column, click Add and enter the required information. To remove a ResultSet column, highlight the row and then click Remove.
- 12 Click OK.
- 13 On the Add Prepared Statements window, click Next.
The Specify the OTD Name window appears.

Next Steps Proceed to the following step, [“Specifying the OTD Name” on page 31](#).

Specifying the OTD Name

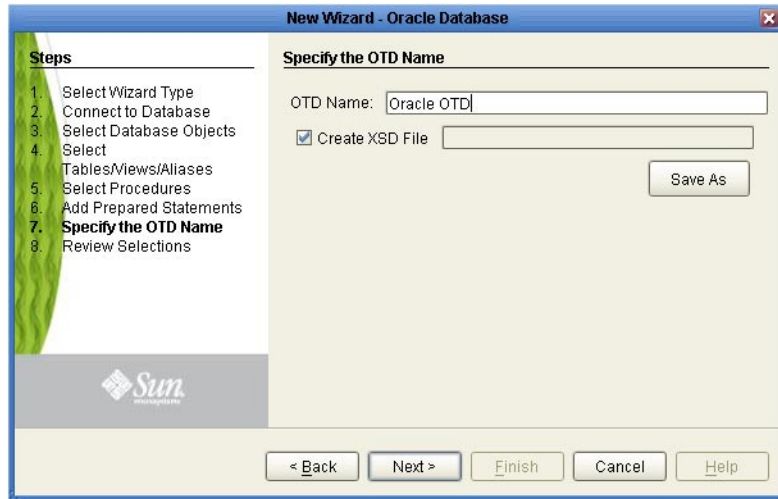
Specify the name that your OTD will display in the Java CAPS IDE.

▼ To Specify the OTD Name

Before You Begin Complete the steps one or more of these previous steps (depending on the database objects you created): [“Selecting Table/Views/Aliases” on page 16](#), [“Selecting Procedures” on page 21](#), and [“Adding Prepared Statements” on page 26](#).

1 Enter a name for the OTD.

The OTD contains the selected tables and the package name of the generated classes.



2 To create an XSD file for the OTD, do the following:

- a. Select Create XSD File.
- b. Click Save As.
- c. In the Save As dialog, navigate to the directory where you want to store the file, enter a file name, and then click Save.

3 Click Next to continue. The Review your Selections window appears.

Next Steps Proceed to the following step, [“Review Selections” on page 32](#).

Review Selections

Review the selections made for the new OTD.

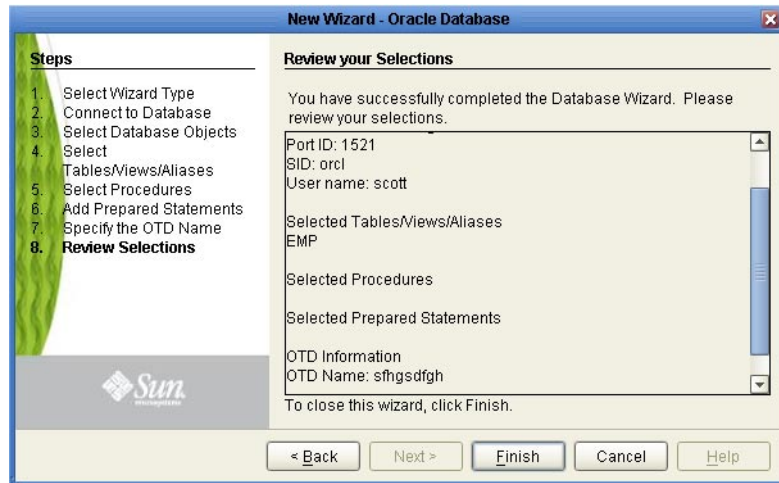
▼ To Review Your OTD Selections

Before You Begin Complete the previous step, [“Specifying the OTD Name” on page 31](#).

- 1 View the summary of the OTD. If you find you have made a mistake, click Back and correct the information.

- 2 If you are satisfied with the OTD information, click Finish to begin generating the OTD.

FIGURE 2 Database Wizard - Summary



The resulting OTD appears on the Java CAPS IDE's canvas.

Editing Existing OTDs

A single OTD can consist of many database objects. They can be a mixture of Tables, Prepared Statements, and Stored Procedures. By using the Database OTD Wizard, the OTD Edit feature allows you to:

- Add or Remove Table/Views.
- Change data types by selecting a different one from a list.
- Add or Remove columns from a Table object.
- Add or Remove Prepared Statement objects.
- Edit Prepared Statement objects.
- Add or Remove Stored Procedure objects.
- Edit Stored Procedure ResultSets.

▼ To Edit an Existing OTD

When a minor change is needed for an existing OTD, there is no need to rebuild it from scratch; instead, you can edit the OTD.

Note – The OTD must be checked out before you can edit it.

1 In the project tree, right-click the OTD and then click Edit.

The Database Connection Information Wizard appears.

2 Enter the connection information as described in “Connecting to a Database” on page 13, and click Next.

Once the connection is established, the Database Wizard opens, allowing you to make modifications to the OTD.

3 Go through each of the wizard steps to make your changes.

4 When you are done editing the OTD, click the Finish button to save the changes.



Caution – Once the OTD has been edited, you must verify that the changes are reflected in the Collaboration so that no errors occur at runtime. For example, if during the edit process, you delete a database object that is included in a Collaboration, the Collaboration could fail at activation or runtime.

When editing an OTD, you can connect to another instance of the database under the following conditions:

- The same version of the database should be used unless the newer version is compatible with the older version. For DB2, the same type of DB2 database must be used. Switching between DB2 databases on z/os, AS/400 and Windows or UNIX is not supported.
- Tables in the database must be defined with the same definition.
- The stored procedures must be identical.
- For tables/stored procedures built with qualified-name, the schema name for the tables/stored procedures must be identical in both database instances.