BEA Application Explorer

Installation and Configuration Guide

Release 7.0 with Service Pack 1
Document Date: November 2002

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA Application Explorer Installation and Configuration Guide

Part Number	Date	Release
N/A	November 2002	7.0 with Service Pack 1

Table of Contents

About This Document

This document explains how to install the BEA Application Explorer, which can be used with WebLogic Integration to create the schemas required to build application view services and events.

This document is organized as follows:

- Chapter 1, "Installing the BEA Application Explorer," directs users to the information they need before installing the BEA Application Explorer and provides step-by-step installation and uninstallation instructions for UNIX and Windows operating systems.
- Chapter 2, "Configuring the BEA Application Explorer," describes how to configure the BEA Application Explorer in order to create the schemas required to build application view services and events.
- Chapter 3, "Using the BEA Application Explorer With an RDBMS," describes how to use the BEA Application Explorer. The functionality of the BEA Application Explorer is presented using the BEA WebLogic Adapter for RDBMS as an example.

What You Need to Know

This document is written for system integrators responsible for installing the BEA Application Explorer and building the metadata required by the BEA application adapters. It is assumed that readers know Web technologies and have a general understanding of Microsoft Windows and UNIX systems as well as:

- General product knowledge of the application system being explored.
- General environment knowledge, such as how to access the individual business and data objects in the application system.
- Business application knowledge in a specific application area.
- Knowledge of the business processes and data models in the application area.
- General knowledge of client/server concepts.

Extensive knowledge of the BEA application adapters is not required but may be helpful in building the appropriate metadata.

Related Information

The following documents provide additional information for the associated software components:

- *BEA WebLogic Adapter for RDBMS Installation and Configuration Guide*
- *BEA WebLogic Adapter for RDBMS User Guide*
- BEA WebLogic Server installation and user documentation, which is available at the following URL:

  `http://edocs.bea.com/more_wls.html`
- BEA WebLogic Integration installation and user documentation, which is available at the following URL:

  `http://edocs.bea.com/more_wli.html`

Contact Us!

Your feedback on the BEA Application Explorer documentation is important to us. Send us e-mail at `docsupport@bea.com` if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA Application Explorer documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA Application Explorer release.

If you have any questions about this version of BEA Application Explorer, or if you have problems installing and running BEA Application Explorer, contact BEA Customer Support through BEA WebSupport at `www.bea.com`. You can also contact Customer Support by using the contact information provided on the Customer Support Card that is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Examples*: LPT1 SIGNON OR

Convention	Item
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...`
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
. . .	Indicates one of the following in a command line: ■ That an argument can be repeated several times in a command line ■ That the statement omits additional optional arguments ■ That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: `buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...`
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER

1 Installing the BEA Application Explorer

This section directs users to the information they need before installing the BEA Application Explorer and provides step-by-step installation instructions for UNIX and Windows operating systems. It includes the following topics:

- Key Features of the BEA Application Explorer
- Before Installing the BEA Application Explorer
- Windows Installation
- UNIX Installation
- Uninstalling the BEA Application Explorer

Key Features of the BEA Application Explorer

The BEA Application Explorer incorporates in-depth knowledge of application system environments to query for metadata on specific business objects. It uses that metadata to generate the schemas required to build application view services and events.

Resource adapters—referred to in this document as adapters—connect one application to another when those applications are not originally designed to communicate with each other. For example, an order entry system built by one company may require an adapter to communicate with a customer information system built by another.

Adapters are bidirectional, that is, they can send requests to the Enterprise Information System (EIS), as well as wait, or listen, for notification of events occurring in the EIS. When the adapter makes requests of the EIS, it is referred to as a *service adapter* or *service*. When it waits for event notification, it is referred to as an e*vent adapter* or *event*.

In order for adapter services and events to properly communicate with the EIS, metadata for the interaction is required.

The BEA Application Explorer queries for metadata and creates the schemas required to build application view services and events. The schema generation encompasses the request and response schema for services and the event schema for application view events. BEA WebLogic Integration uses the generated schemas to integrate enterprise application systems.

Key features of the BEA Application Explorer include:

- Use of metadata from application systems to build application view services and events (XML schemas for database events and SQL requests and responses). The application view services and events can then be used in business process workflows created in WebLogic Integration Studio.
- Point-and-click process for generating application view schemas.
- Ability to connect to and explore the application system regardless of the availability of the BEA WebLogic Integration environment

Before Installing the BEA Application Explorer

The BEA Application Explorer requires a Java Virtual Machine (JVM) on which to run. Verify that a JVM environment is installed on the target system.

In addition to the JVM, you must install and configure the appropriate application system (for example, SAP or Siebel). The BEA Application Explorer does not need to reside on the same system as the application system being accessed, but network access is required.

Install the BEA Application Explorer using the appropriate `.exe` or `.tar` file:

- `BEABSE.EXE` file on Windows NT or Windows 2000.
- `beabse.hpux.tar`, `beabse.solaris.tar`, or `beabse.aix.tar` file on UNIX platforms.

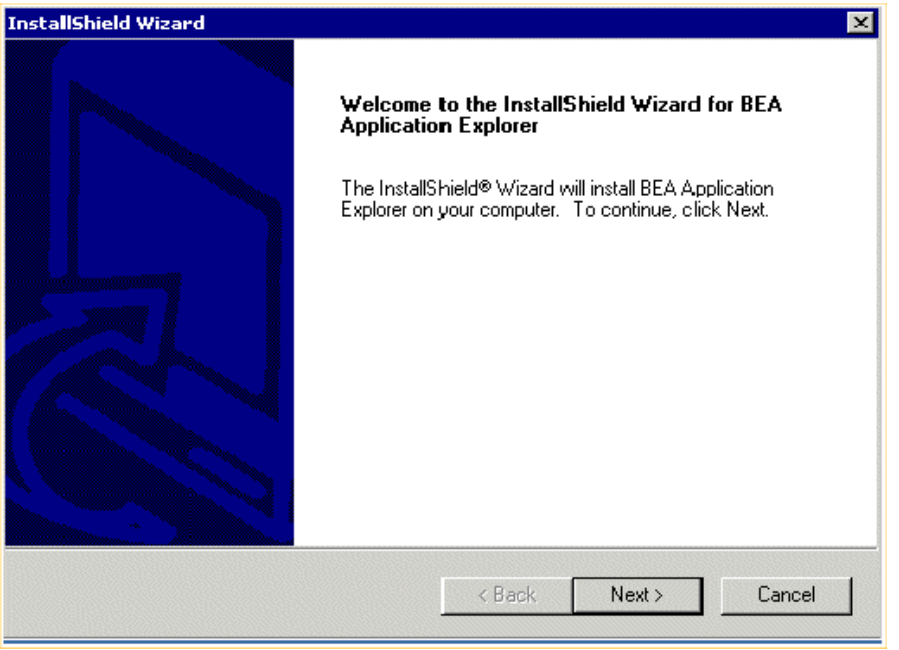
Windows Installation

To install the BEA Application Explorer:

1. Execute the file `BEABSE.EXE`.

   The InstallShield Wizard for the BEA Application Explorer opens.

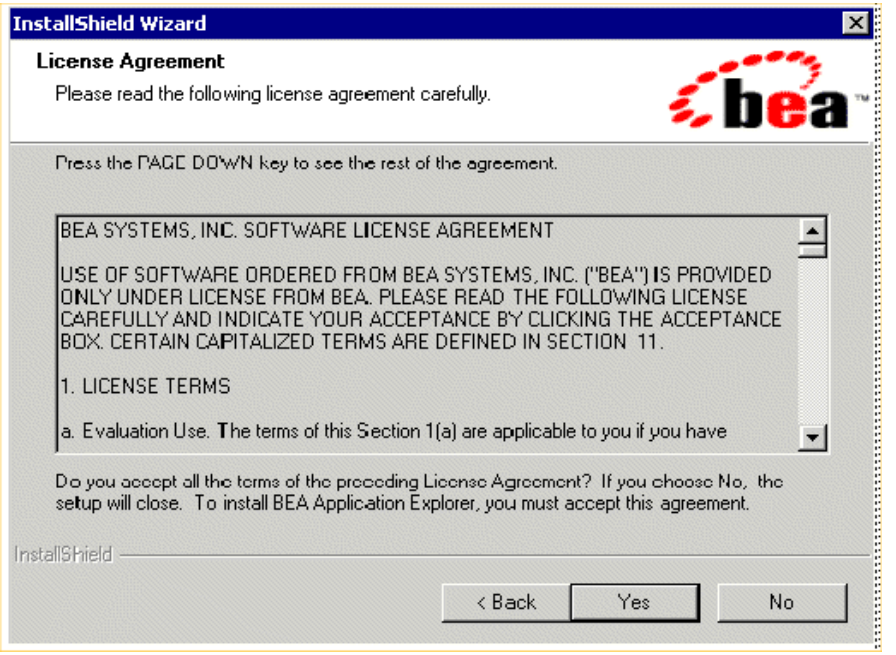
Figure 1-1 Welcome Window for Installation



2. Click Next to continue.

The License Agreement window opens.

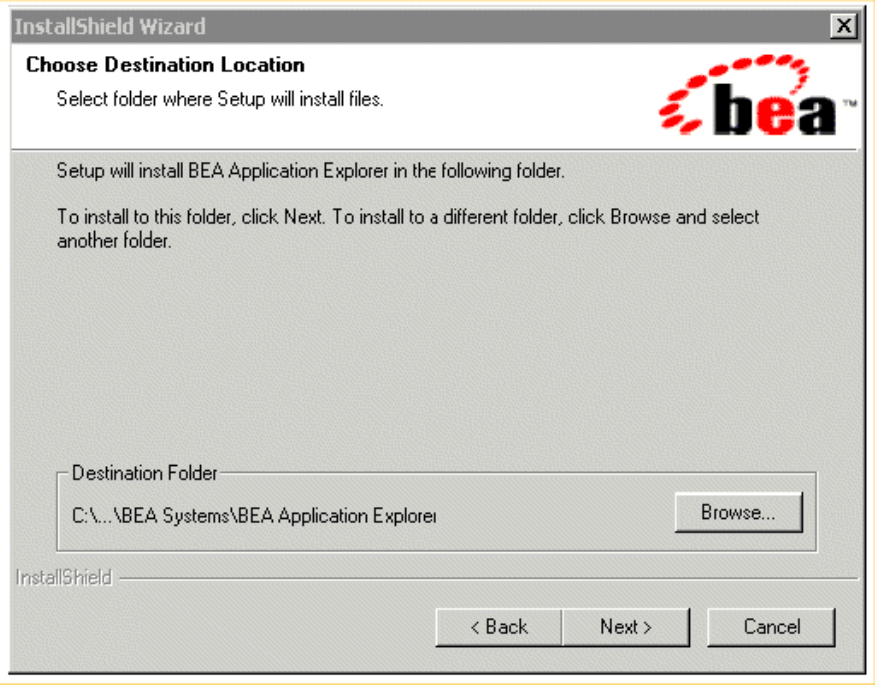
Figure 1-2 License Agreement Window



3. Read the license agreement and click Yes if you agree to the terms. If not, click No, which terminates the installation procedure.

   The next window prompts you for the location in which to install the BEA Application Explorer.

Figure 1-3 Choose Destination Location Window



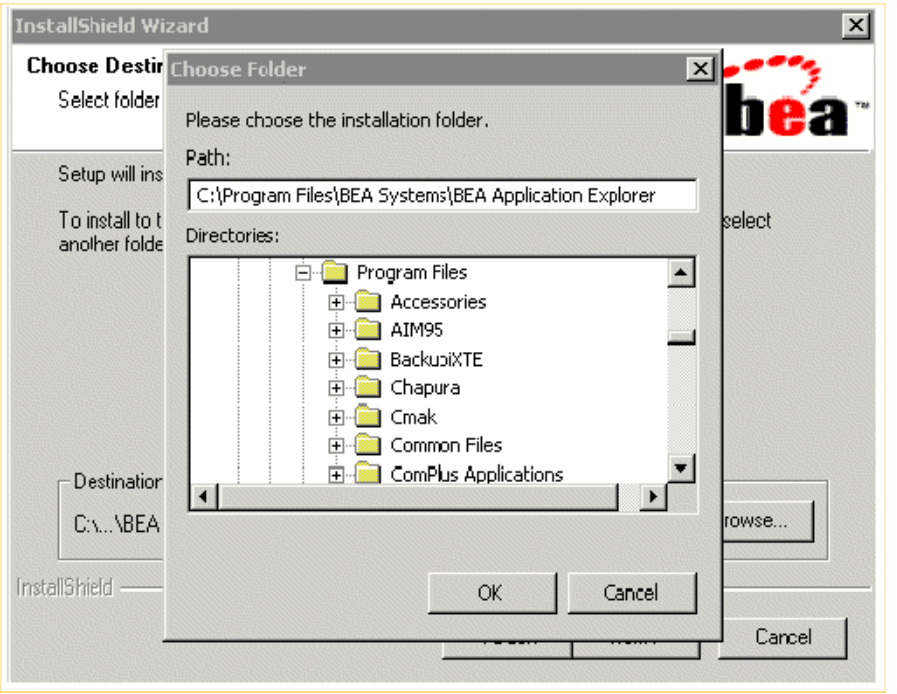
   The default location is:

   ```
   C:\Program Files\BEA Systems\BEA Application Explorer
   ```

4. To accept the default location, click Next and proceed to Step 6. If you select this location and the directory does not exist, the program creates it for you. To select a location other than the default, click Browse.

The Choose Folder window opens.

Figure 1-4 Choose Folder Window

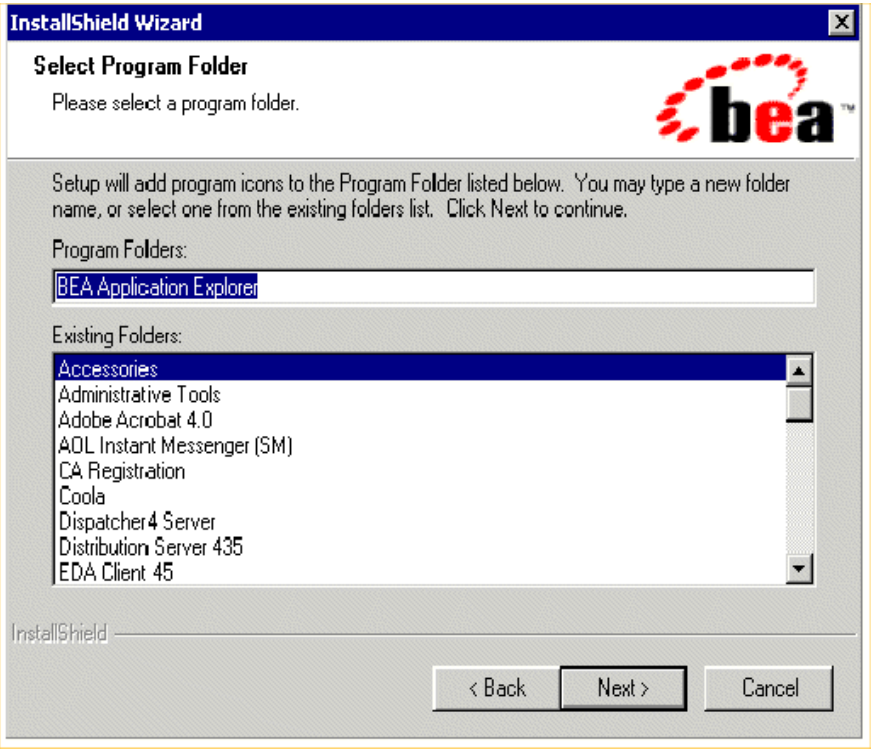


5. Select the location and click OK. Clicking Cancel leaves the installation location unchanged.

6. Click Next.

   The Select Program Folder window opens.

Figure 1-5 Select Program Folder Window

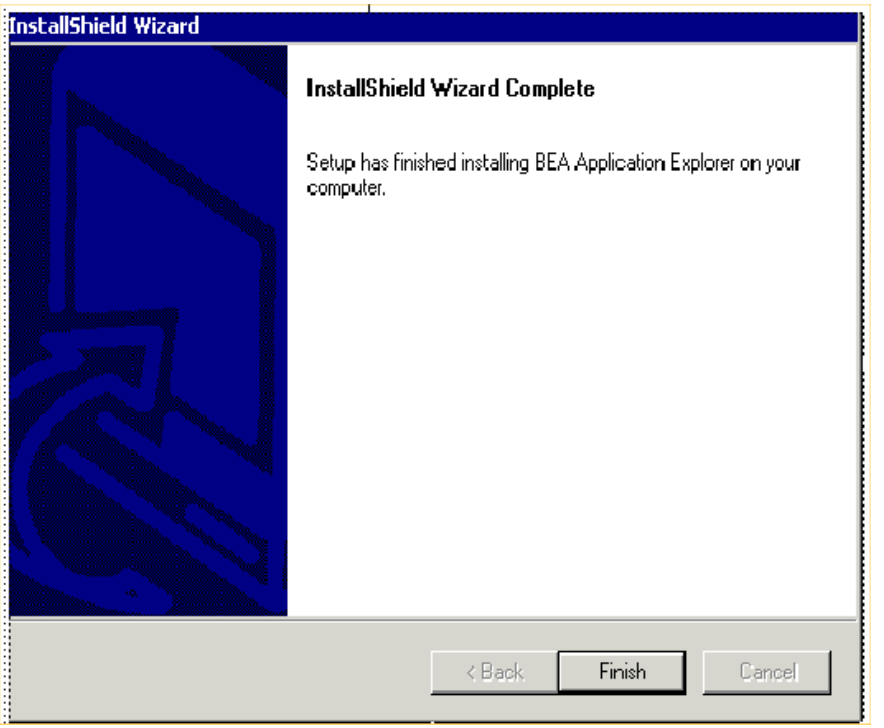


   By default, the installation procedure creates a program folder named BEA Application Explorer. You can enter a different name, but it is recommended that you use the default name.

7. Click Next to continue.

   The installation program copies the files. When the installation is complete, the InstallShield Wizard Complete window opens.

Figure 1-6 InstallShield Wizard Complete Window



8. Click Finish to exit the installation program.

UNIX Installation

To install the BEA Application Explorer:

1. Locate the appropriate `.tar` file for the UNIX operating system. Refer to the following table for the file name.

Operating System	File Name
Hewlett-Packard HP-UX	`beabse.hpux.tar`
Sun Solaris	`beabse.solaris.tar`
IBM AIX	`beabse.aix.tar`

2. Copy the appropriate `.tar` file to a local directory for installation.
3. Create a root directory for the files to be expanded to, for example, `/BEA/BAE`.
4. Issue the tar command. The following is an example of a basic tar command that expands the `.tar` file `beabse.hpux.tar` into the current directory:

   ```
   tar -xvf beabse.hpux.tar
   ```

   The following is a sample result of the tar process:

```
$ tar -xvf beabse.hpux.tar
x hpux/bin/adapters.xml, 71 bytes, 1 media blocks.
x hpux/bin/datatypes.dtd, 6861 bytes, 14 media blocks.
x hpux/bin/xdr-xsd-converter.xslt, 31730 bytes, 62 media blocks.
x hpux/bin/XMLSchema.dtd, 16207 bytes, 32 media blocks.
x hpux/lib/hpux/librfccm.o, 7281990 bytes, 14223 media blocks.
x hpux/lib/hpux/libsapjcorfc.o, 577121 bytes, 1128 media blocks.
x hpux/lib/hpux/sapjco.jar, 152597 bytes, 299 media blocks.
x hpux/lib/bea-ae.jar, 402409 bytes, 786 media blocks.
x hpux/lib/classes12.zip, 1931357 bytes, 3773 media blocks.
x hpux/lib/SiebAdapter.jar, 33626 bytes, 66 media blocks.
x hpux/lib/SiebelJI_Common.jar, 404777 bytes, 791 media blocks.
x hpux/lib/SiebelJI_enu.jar, 156003 bytes, 305 media blocks.
x hpux/lib/xalan.jar, 906248 bytes, 1771 media blocks.
x hpux/lib/xercesImpl.jar, 933730 bytes, 1824 media blocks.
x hpux/lib/xml-apis.jar, 108484 bytes, 212 media blocks.
x hpux/lib/xmlParserAPIs.jar, 78440 bytes, 154 media blocks.
```

See the appropriate operating system manual for more information on the `tar` command for that system.

Uninstalling the BEA Application Explorer

The BEA Application Explorer can be easily unistalled should you have to do so.

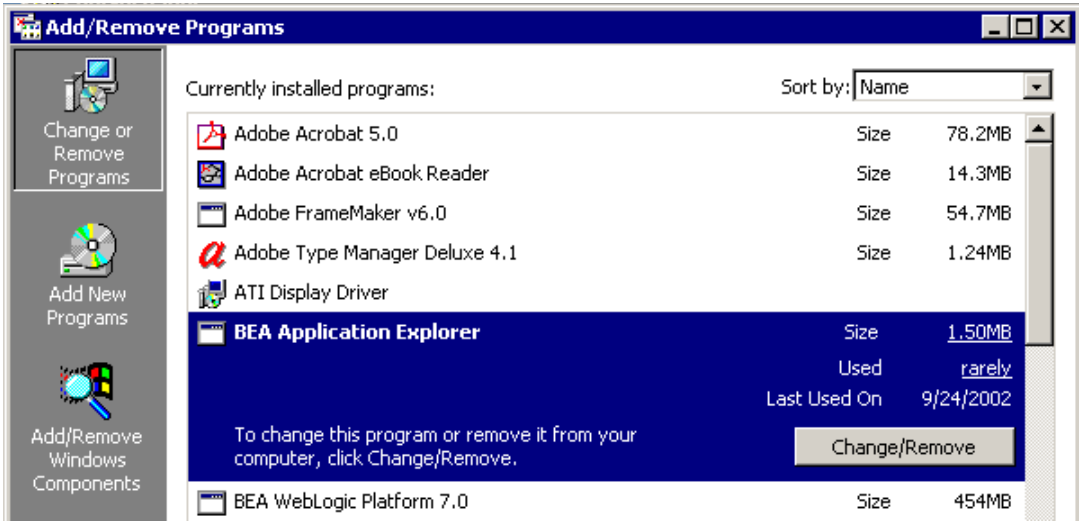
Uninstalling the BEA Application Explorer on Windows

To unistall the BEA Application Explorer:

1. Choose Start→Settings→Control Panel.
2. Double-click Add/Remove Programs.
3. Click BEA Application Explorer.

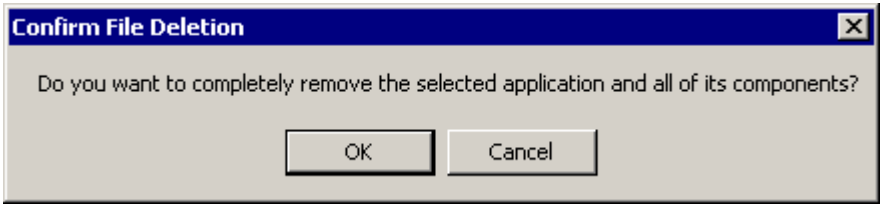
   BEA Application Explorer is highlighted, as shown in the following figure.

Figure 1-7 Add/Remove Programs Window



4. Click Change/Remove.

   The Confirm File Deletion window opens.



5. Click OK.

Uninstalling the BEA Application Explorer on UNIX

To uninstall the BEA Application Explorer on UNIX, delete the installation directory created during the installation.

2 Configuring the BEA Application Explorer

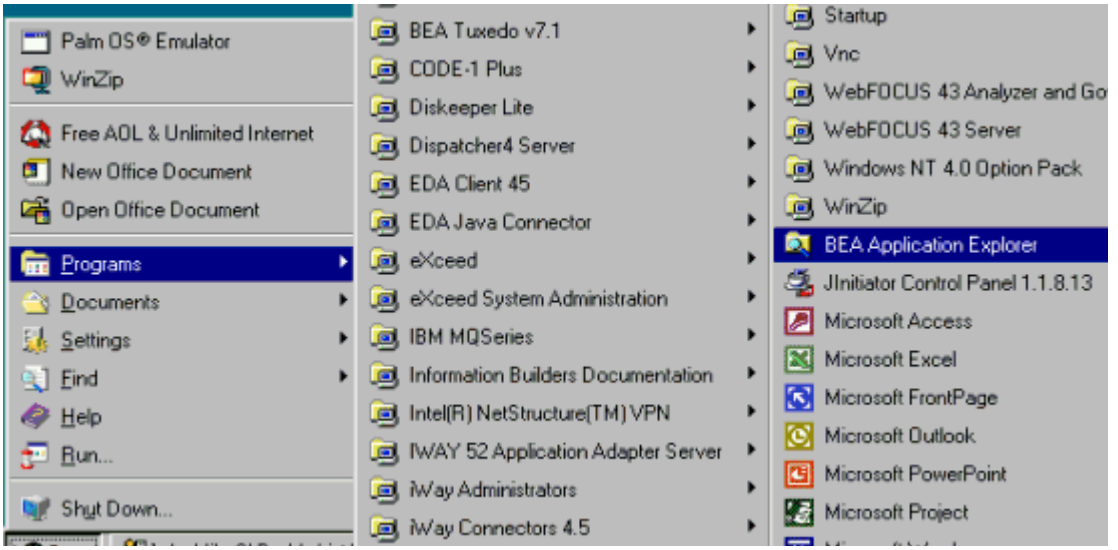
The BEA Application Explorer is an exploration tool that creates the schemas required to build application view services and events. The generated schemas aid in the integration process for BEA WebLogic Integration. In order for a BEA application adapter to properly communicate with the Enterprise Information System (EIS), metadata for the interaction is required.

This section describes how to configure the BEA Application Explorer in order to create the schemas required to build application view services and events. It includes the following topics:

Starting the BEA Application Explorer on Windows

To start the BEA Application Explorer:

1. From the Windows Start menu, choose Programs→BEA Application Explorer.

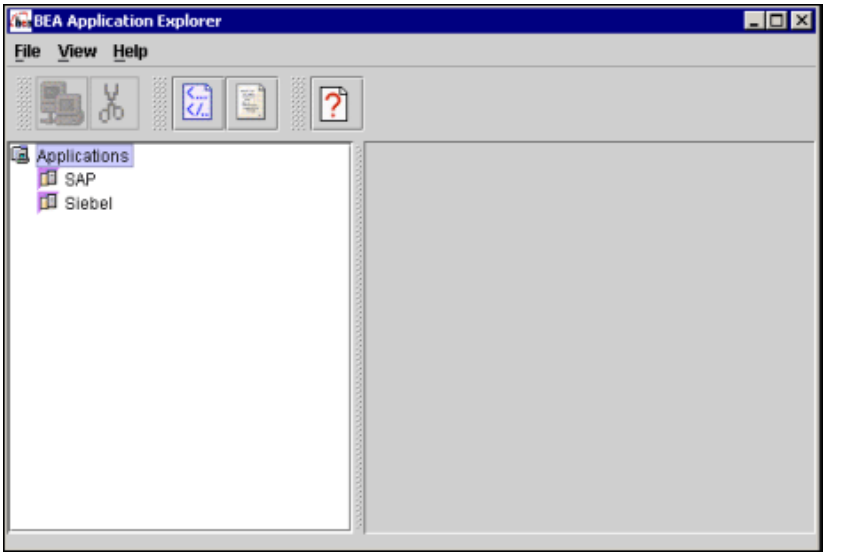


After the loading of the BEA Application Explorer starts, the initialization pane opens.

Figure 2-1 Initialization Pane

When initialization is complete, the BEA Application Explorer opens.

Figure 2-2 BEA Application Explorer Main Window



2. Proceed to "Establishing a Storage Directory for Metadata" on page 2-7.

Starting the BEA Application Explorer on UNIX

To start the BEA Application Explorer:

1. Go to the `bin` directory created during the installation process.

   `/installation_directory/operating system/``bin`

   For example:

   `/BEA/BAE/hpux/bin`

2. If it is not already included, add the JDK to your `PATH` variable by executing the command appropriate for your system. For example,

   ```
   PATH=BEA_HOME/jdk131_03/bin:$PATH
   export PATH
   ```

3. Run the `ae` shell script. For example, on HP-UX, use the following command for the `bin` directory:

   ```
   . ./ae
   ```

   Note: The ae shell script cannot be run as root. Furthermore, the `ae` shell script does not have executable permissions by default. On UNIX, you can either change permission by executing `chmod 755` or start `ae` in a new shell by executing `sh ae`.

   The following is the content of the `ae` shell script.

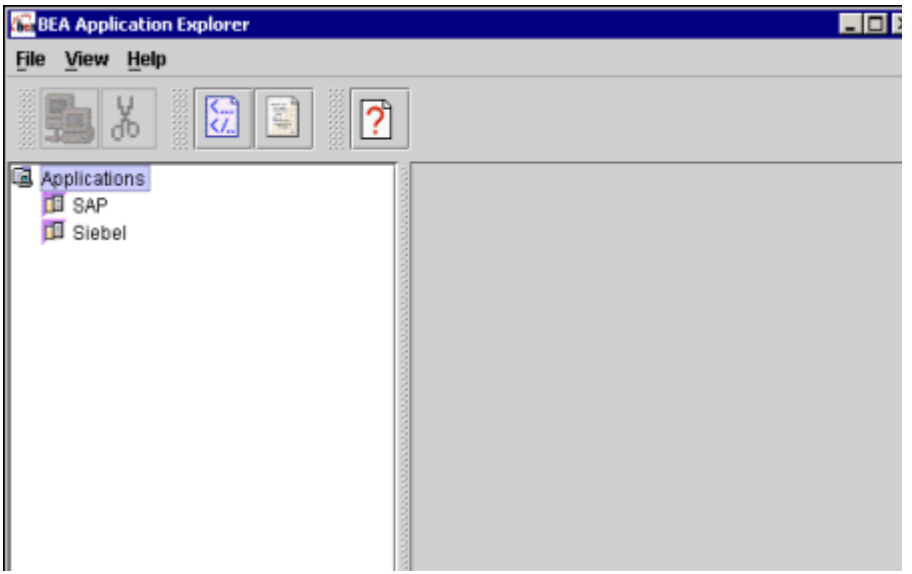
Listing 2-1 ae Shell Script

```
#! /bin/sh

if [ "$BSE_HOME" = "" ] ; then
    BSE_HOME=${PWD}/..
fi

# Allow .antrc to specify flags to java cmd
if [ "$JAVACMD" = "" ] ; then
  JAVACMD=java
fi
```

```
LOCALCLASSPATH=`echo $BSE_HOME/lib/*.zip | tr ' ' ':'`
LOCALCLASSPATH=`echo $BSE_HOME/lib/*.jar | tr ' '
':'`:$LOCALCLASSPATH

if [ "$CLASSPATH" != "" ] ; then
  LOCALCLASSPATH=$CLASSPATH:$LOCALCLASSPATH
fi
osname=`uname`
if [ "$osname" = "HP-UX" ] ; then
   chmod a+x $BSE_HOME/lib/hpux/*
   LOCALCLASSPATH=$LOCALCLASSPATH:$BSE_HOME/lib/hpux/jCO.jar
   SHLIB_PATH=$SHLIB_PATH:$BSE_HOME/lib/hpux
elif [ "$osname" = "AIX" ] ; then
   chmod a+x $BSE_HOME/lib/aix/*
   LOCALCLASSPATH=$LOCALCLASSPATH:$BSE_HOME/lib/aix/jCO.jar
   LIBPATH=$LIBPATH:$BSE_HOME/lib/aix
elif [ "$osname" = "SunOS" ] ; then
   chmod a+x $BSE_HOME/lib/solaris/*
   LOCALCLASSPATH=$LOCALCLASSPATH:$BSE_HOME/lib/solaris/jCO.jar
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$BSE_HOME/lib/solaris

elif [ "$osname" = "Linux" ] ; then
   chmod a+x $BSE_HOME/lib/linux/*
   LOCALCLASSPATH=$LOCALCLASSPATH:$BSE_HOME/lib/linux/jCO.jar
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$BSE_HOME/lib/linux
fi

if [ "$JAVA_HOME" != "" ] ; then
  if test -f $JAVA_HOME/lib/tools.jar ; then
    LOCALCLASSPATH=$LOCALCLASSPATH:$JAVA_HOME/lib/tools.jar
  fi

  if test -f $JAVA_HOME/lib/classes.zip ; then
    LOCALCLASSPATH=$LOCALCLASSPATH:$JAVA_HOME/lib/classes.zip
  fi
fi

$JAVACMD -classpath $LOCALCLASSPATH com.ibi.bse.gui.BseFlashScreen
$@
```

The BEA Application Explorer opens.

Figure 2-3 BEA Application Explorer Main Window



4. Proceed to the following section, "Establishing a Storage Directory for Metadata."

Establishing a Storage Directory for Metadata

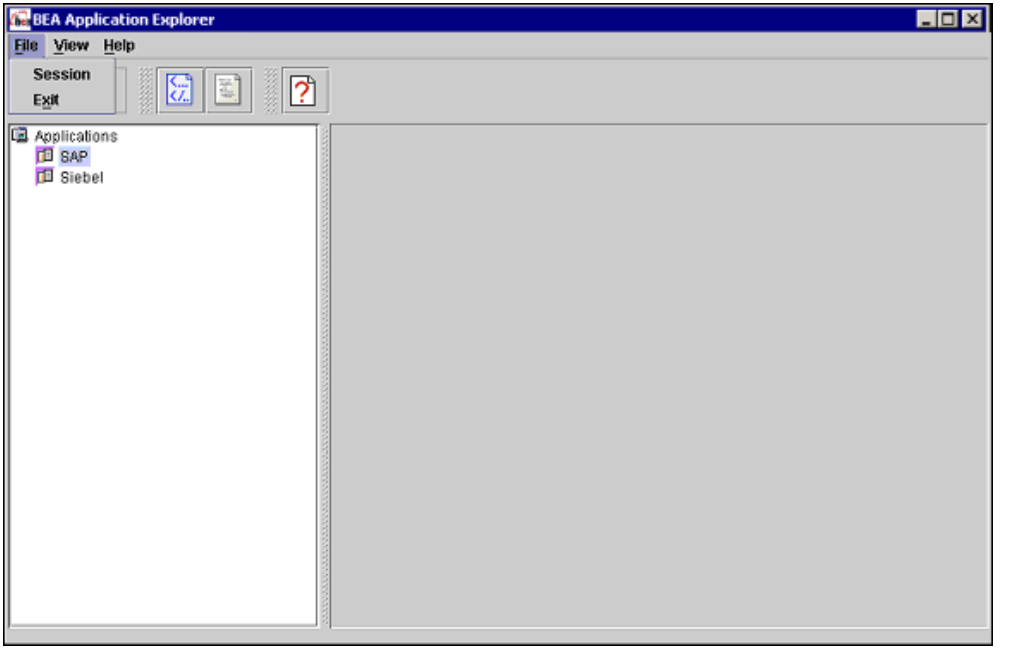
When you create metadata for use with a BEA application adapter, the metadata is stored in the file system based on the session setting.

To set up a session directory for storing metadata:

1. Choose File→Session to change the default session directory.

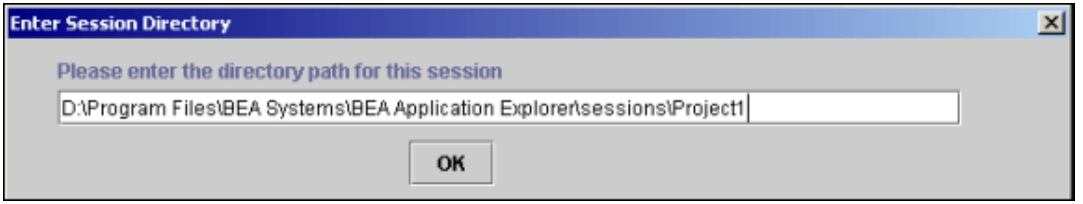
   Note: In this section, directory locations and representations are provided for a Windows environment. Please specify the appropriate information for your platform.

Figure 2-4 Changing the Default Session Directory



You are prompted for the directory path for the session.

Figure 2-5 Enter Session Directory Dialog Box

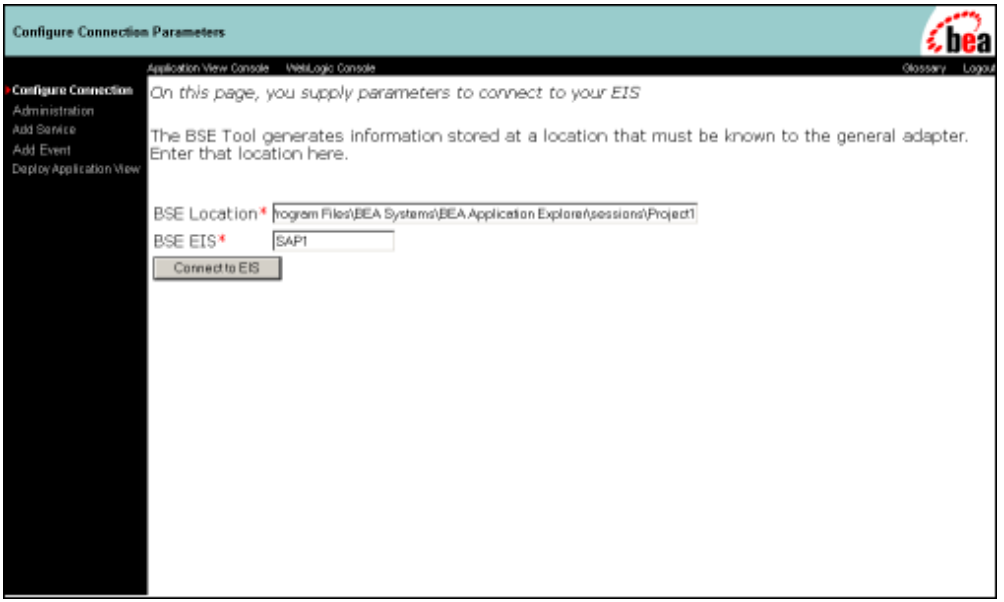


2. Click OK after changing the default session directory.

   The session directory establishes the directory structure for storing the generated metadata. Once the session directory is set, you can explore an application system for metadata. The individual adapters contain a `.JSP` page that points to the directory specified.

   The following is a sample `.JSP` page for the BEA Adapter for SAP.

Figure 2-6 Sample .JSP Page for the BEA Adapter for SAP



For more information on how the BEA Adapter for SAP uses the session directory path, see the *BEA Adapter for SAP User Guide*.

3 Using the BEA Application Explorer With an RDBMS

This section describes how to use the BEA Application Explorer. The underlying technology used to access and create the appropriate schemas differs for the individual application systems being explored, but the user interface is consistent. In this section the functionality of the BEA Application Explorer is presented using the BEA WebLogic Adapter for RDBMS as an example.

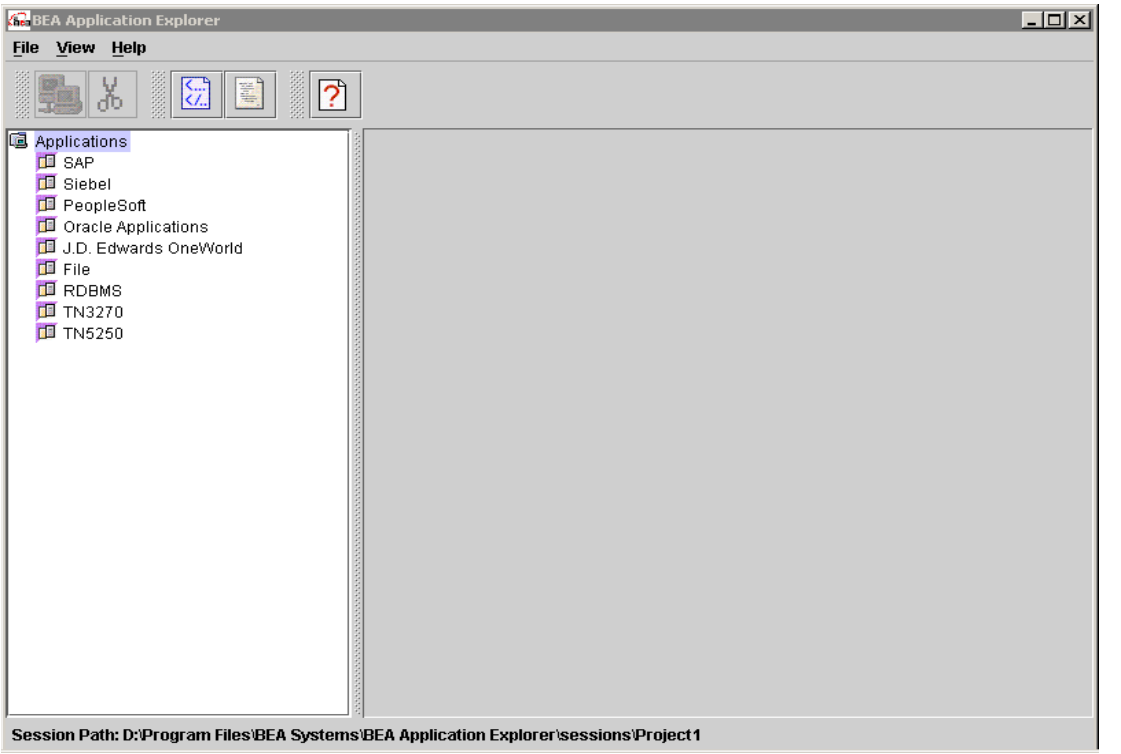
The BEA Application Explorer supports the creation of schemas based on specific tables and resulting answer sets. To obtain the metadata about the relational database management system (RDBMS) tables and answer sets, the BEA Application Explorer connects to the RDBMS using the same JDBC drivers that the BEA WebLogic Adapter for RDBMS uses.

This section includes the following topics:

Connecting to an RDBMS

Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer. When you first start the BEA Application Explorer, you can view the main panes. The left pane displays all the BEA WebLogic Adapters supported by the version of the BEA Application Explorer being used.

Figure 3-1 BEA Application Explorer Window



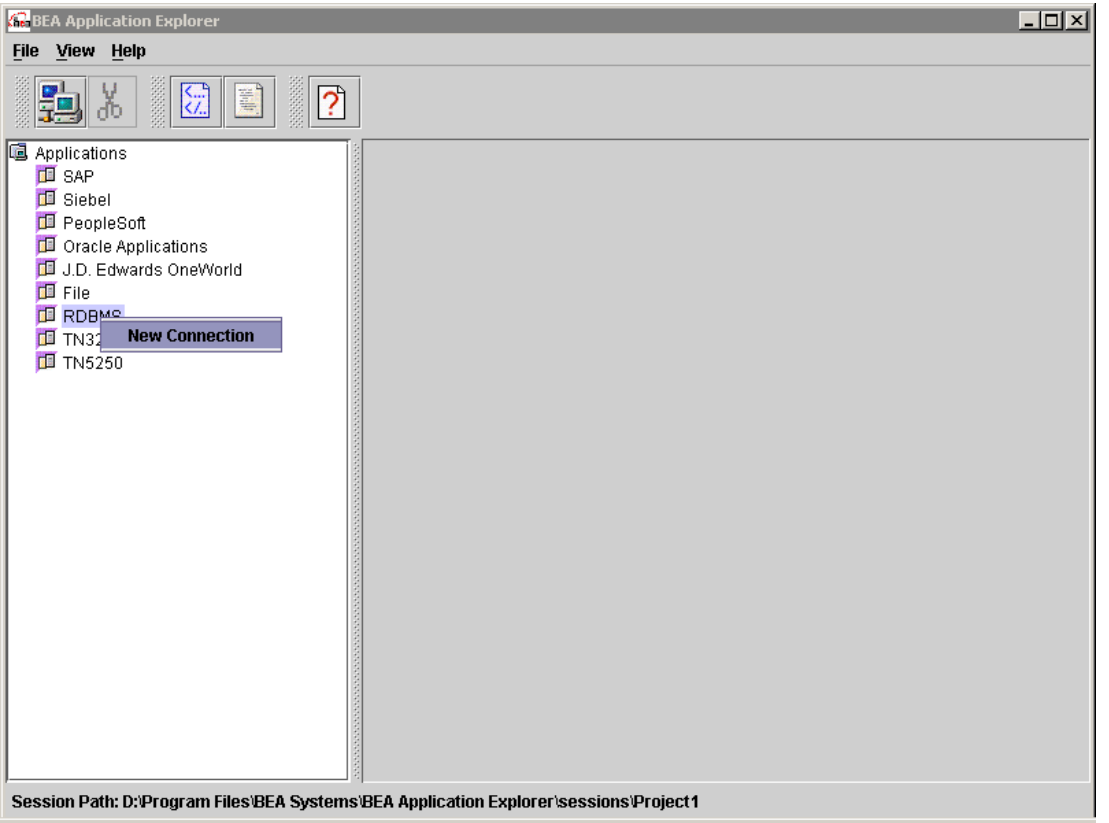
To connect to a specific RDBMS, you must first create a new connection.

1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.

   The BEA Application Explorer opens.

2. Right-click the RDBMS node.

Figure 3-2 BEA Application Explorer - New Connection



3. Select the New Connection option to create a new connection.

Note: If a connection was previously created, see "Connecting to an RDBMS Using an Existing Connection" on page 3-9.

The New Connection prompt displays.

Figure 3-3 New Connection Name Input Window

4. Enter a name for the connection. Use a descriptive name, for example, Oracle817.

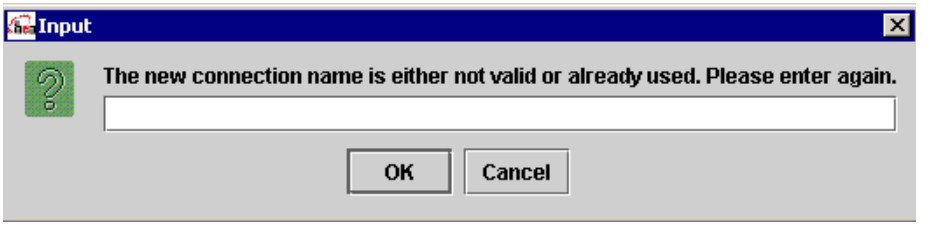
   The name entered is used to build a directory underneath the session path specified, as well as to identify the connection.

   Since the connection name is used as a directory name, it must be a valid directory name for the operating system on which the connection and schema information is stored. For example, the connection name `!@#$%^&*()` is invalid on a Windows system.

5. Click OK.

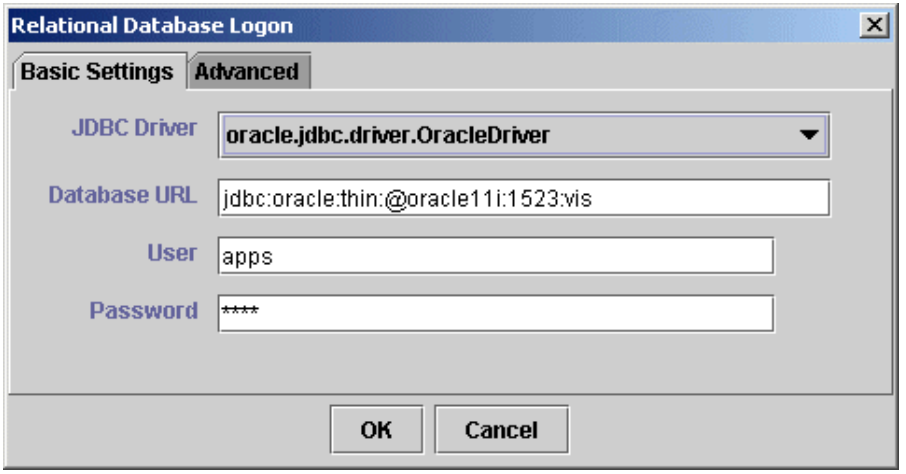
   The connection name is verified for the system. If you enter an invalid connection name, a new input box opens and asks for the connection name to be entered again.

Figure 3-4 Invalid Connection Name Message



After you enter a valid connection name, the system prompts you for connection information.

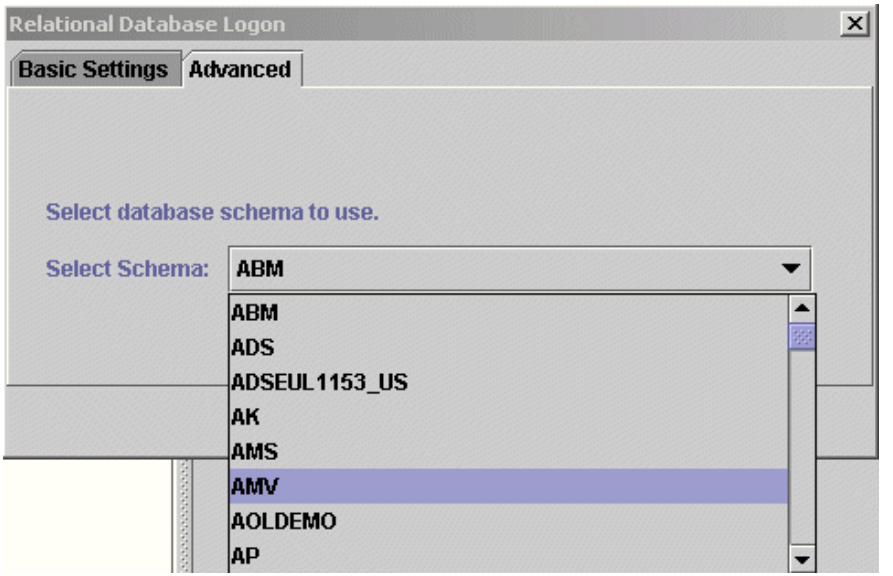
Figure 3-5 Relational Database Logon Window



6. From the drop-down list, select the appropriate JDBC driver to use and enter the appropriate information for the connection, as follows:
   - Database URL: The JDBC driver-specific URL used to connect to the RDBMS
   - User: A valid user ID for the RDBMS
   - Password: The password associated with the user ID specified.
7. Click the Advanced tab.

8. Select the schema to use, as shown in the following figure.

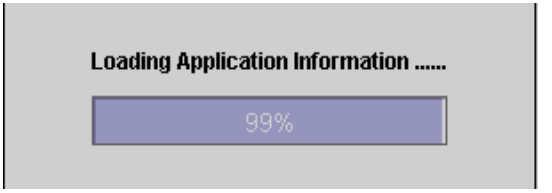
Figure 3-6 Relational Database Logon - Advanced Tab



9. Click OK.

   If the parameters are correct and the RDBMS is available, a progress bar displays, indicating that the RDBMS metadata is loading.

Figure 3-7 Loading Application Information Progress Indicator

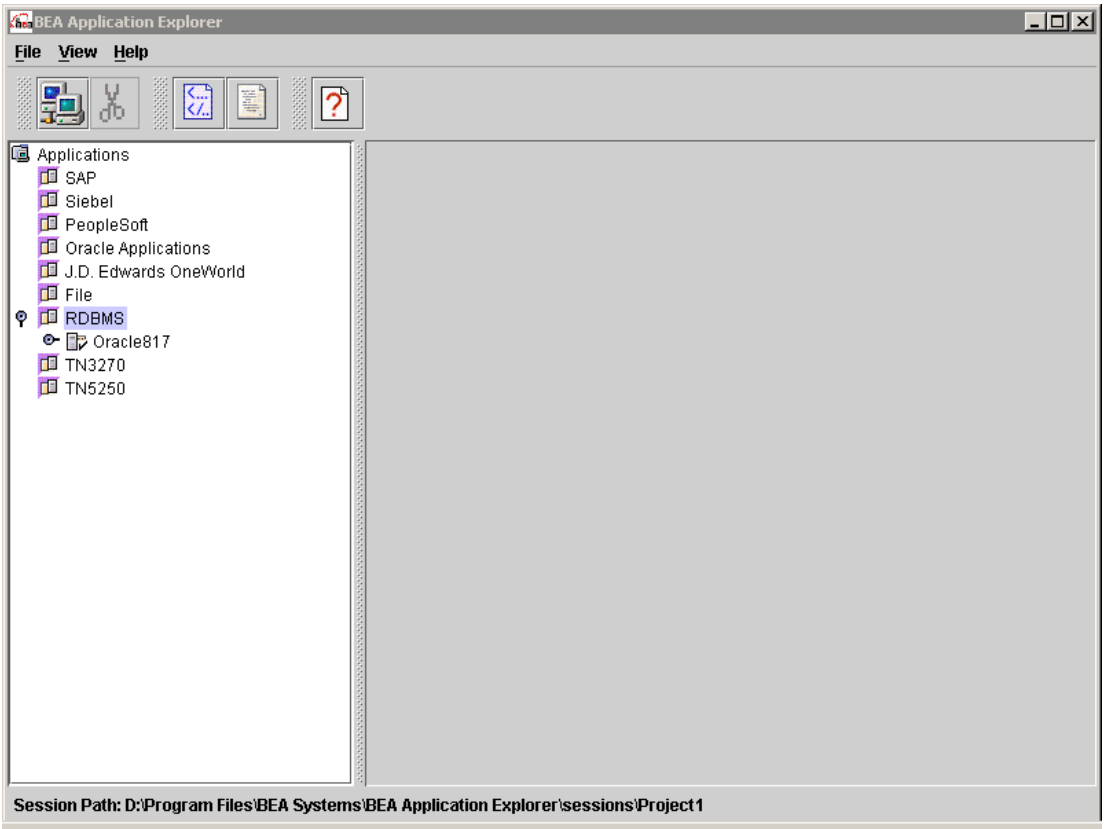


If there is a problem with the connection, a message appears.

Figure 3-8 Database Logon Error Message

After the loading of the application (RDBMS) information is complete, the connection appears as a node under the RDBMS node.

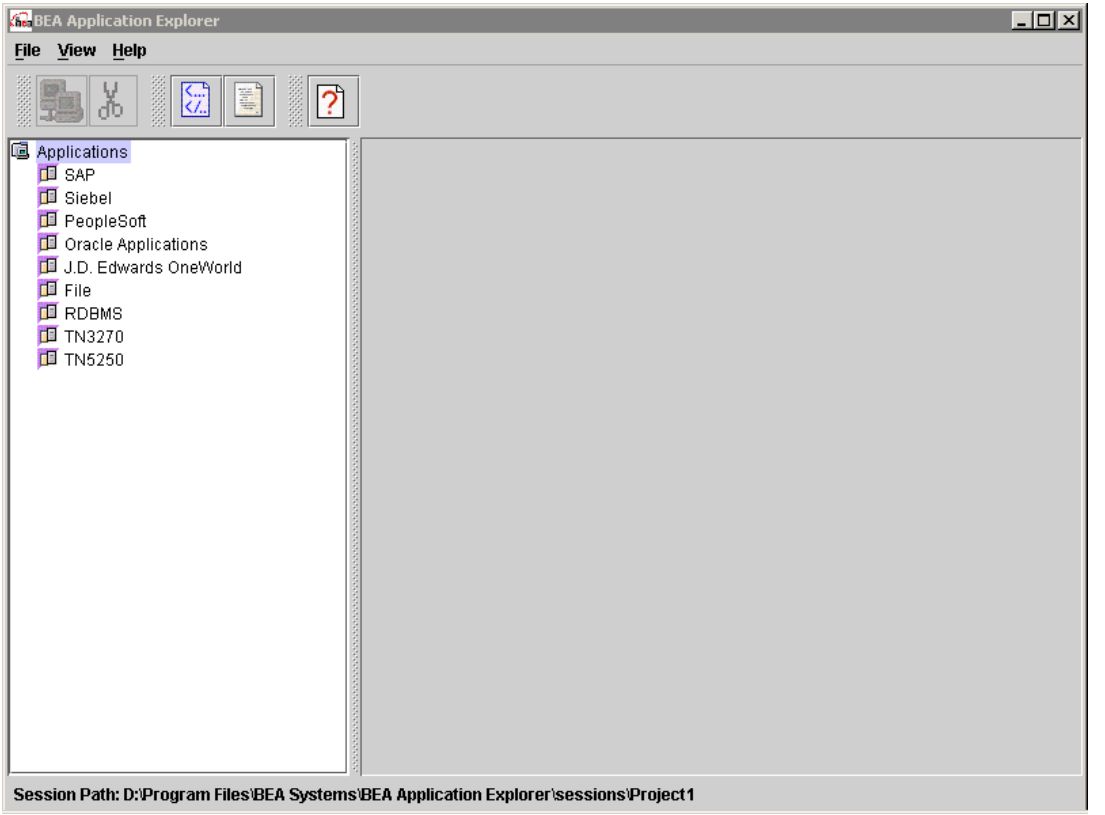
Figure 3-9 BEA Application Explorer - Connection Node



Connecting to an RDBMS Using an Existing Connection

When you first start the BEA Application Explorer, the main window opens. The left pane displays all the BEA WebLogic Adapters supported by the version of the BEA Application Explorer you are using.

Figure 3-10 BEA Application Explorer Window



To connect to a specific RDBMS with an existing connection:

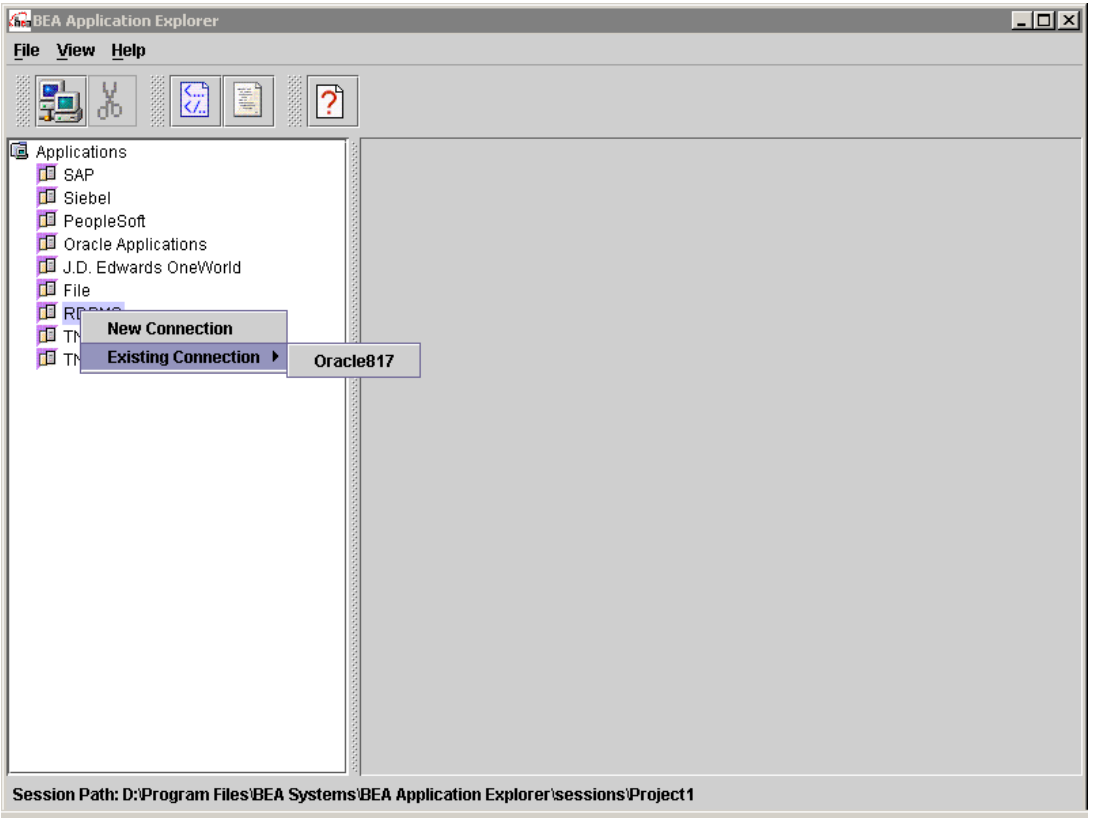
1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.

   The BEA Application Explorer opens.

2. Right-click the RDBMS node and select Existing Connection from the shortcut menu.

   A list of existing connections from which to choose opens.

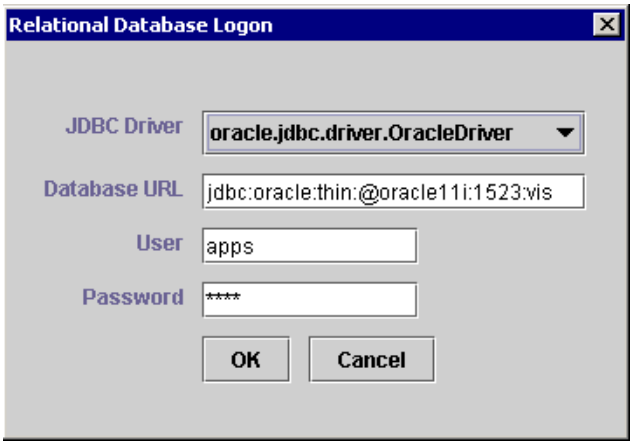
Figure 3-11 Existing Connections



3. Select the desired connection.

A confirmation window opens, showing the connection information being used.

Figure 3-12 Relational Database Logon Confirmation Window

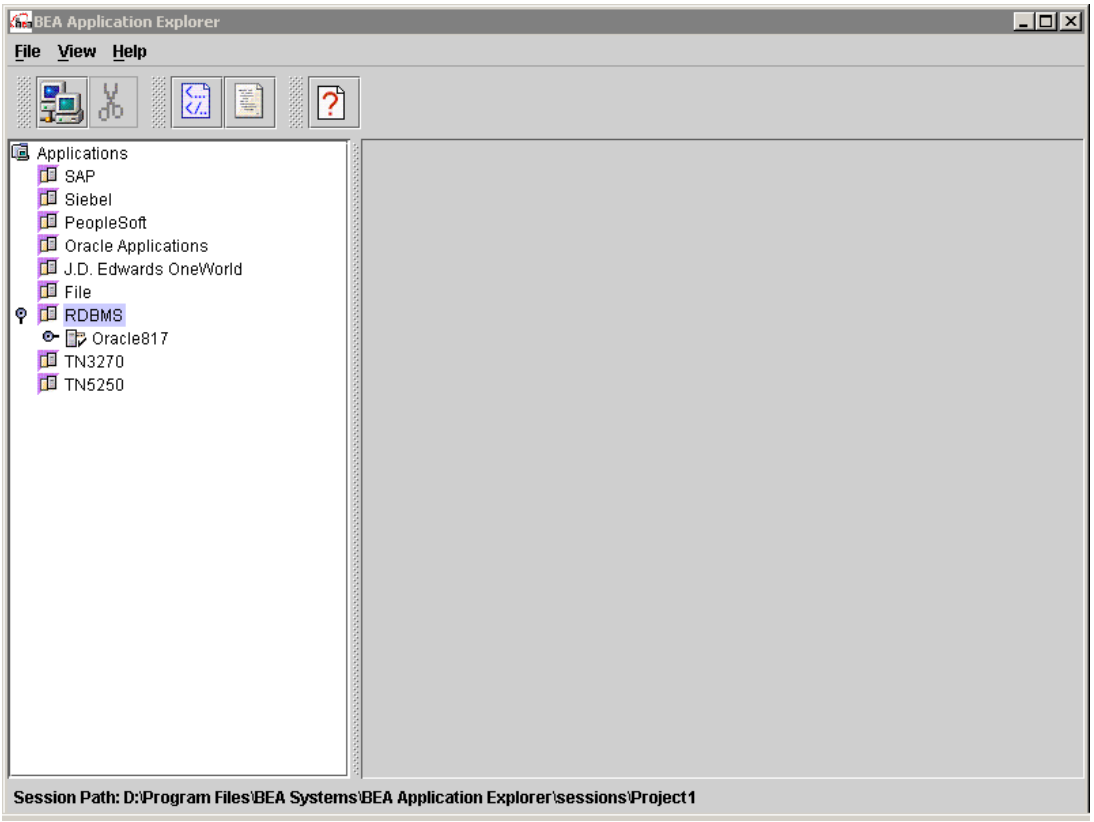


4. If this is the correct connection, click OK. If not, click Cancel.

   If the parameters are correct and the RDBMS is available, a progress bar indicates that the RDBMS metadata is loading. If there is a problem with the connection, an error message appears.

After the application (RDBMS) information loads, the connection appears as a node under the RDBMS node.

Figure 3-13 BEA Application Explorer - Connection Node



Disconnecting from an RDBMS

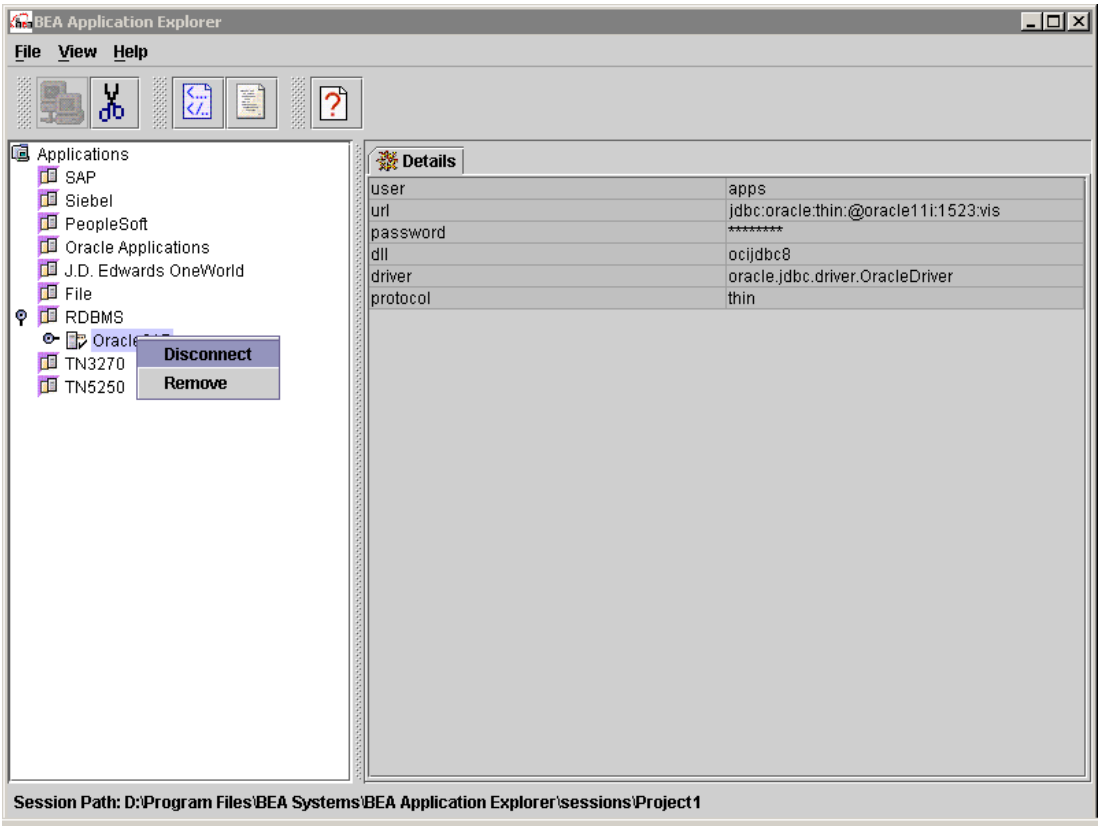
Although you can maintain multiple open connections to different RDBMSs and applications, it is prudent to close connections when they are not being used.

To close a connection to an RDBMS:

1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.

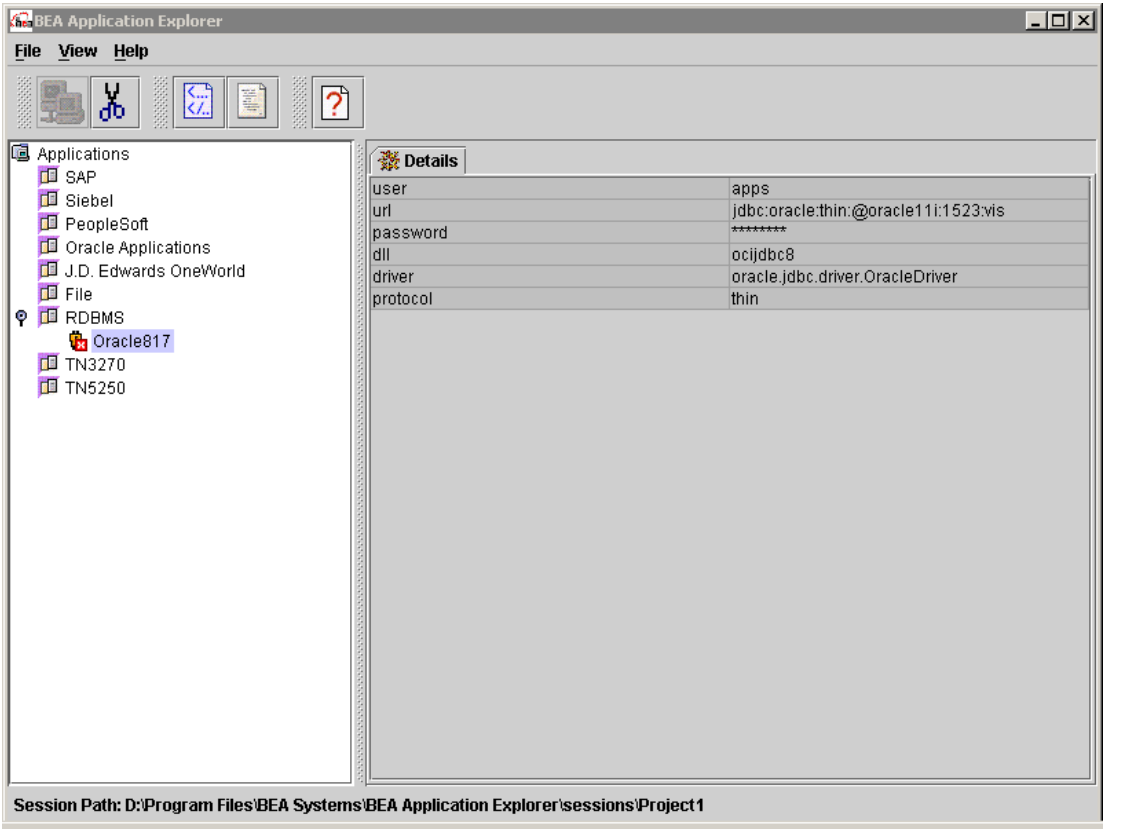
2. When the BEA Application Explorer open, right-click the connection node and select Disconnect.

Figure 3-14 BEA Application Explorer - Disconnect Node Option



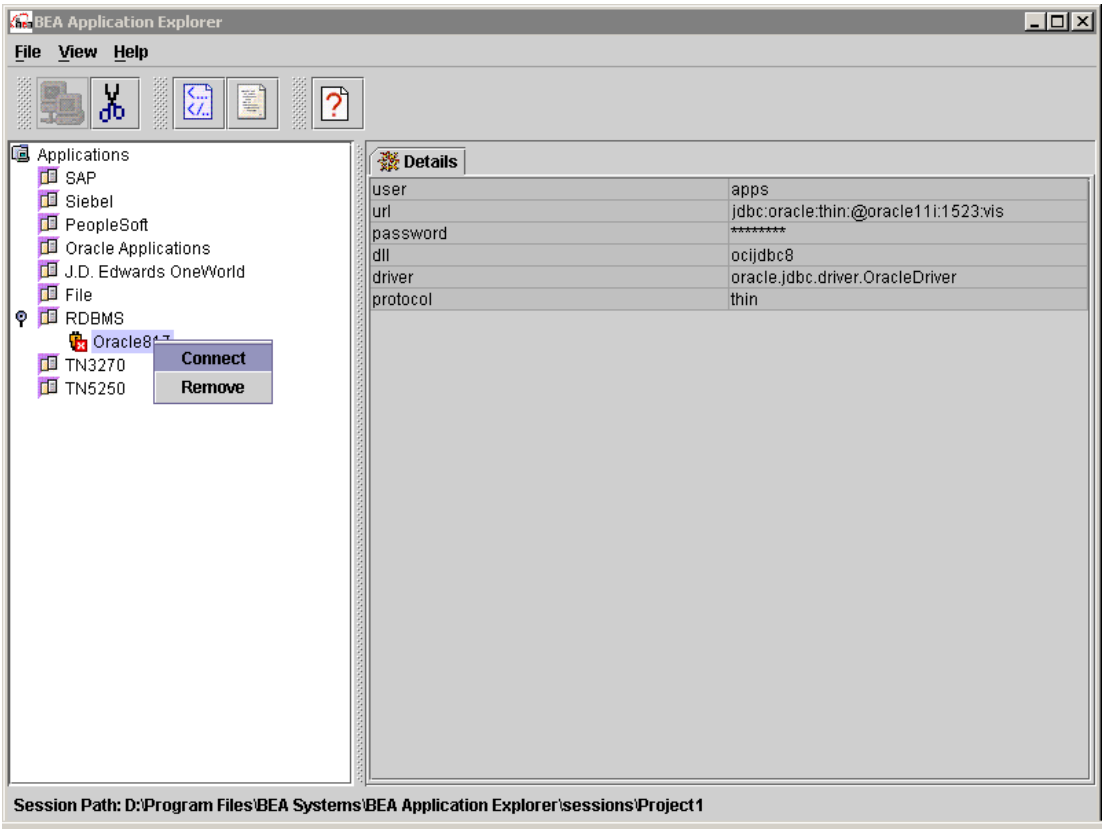
Disconnecting from the RDBMS drops the connection with the RDBMS, but the node remains. Note that the disconnected node has a different icon, , indicating that it is disconnected, as shown in the following figure.

Figure 3-15 BEA Application Explorer - Disconnected Node



3. To re-establish the connection, right-click the disconnected node and select Connect, as shown.

Figure 3-16 BEA Application Explorer - Node Reconnection

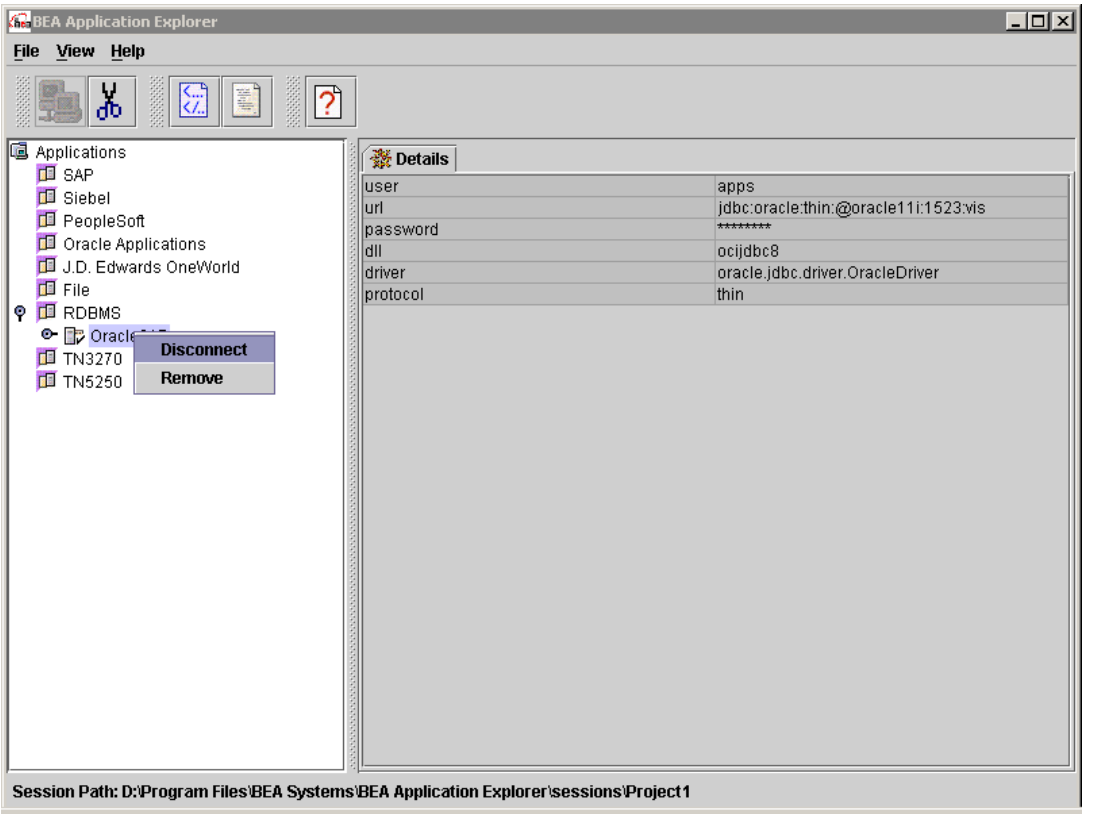


Removing a Connection

To remove a connection from the existing connections list:

1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.
2. When the BEA Application Explorer opens, right-click the desired connection and select Remove.

Figure 3-17 BEA Application Explorer - Connection Removal



The connection is closed, if open. The entry is removed from the available connection list.

Viewing Table-Based Metadata

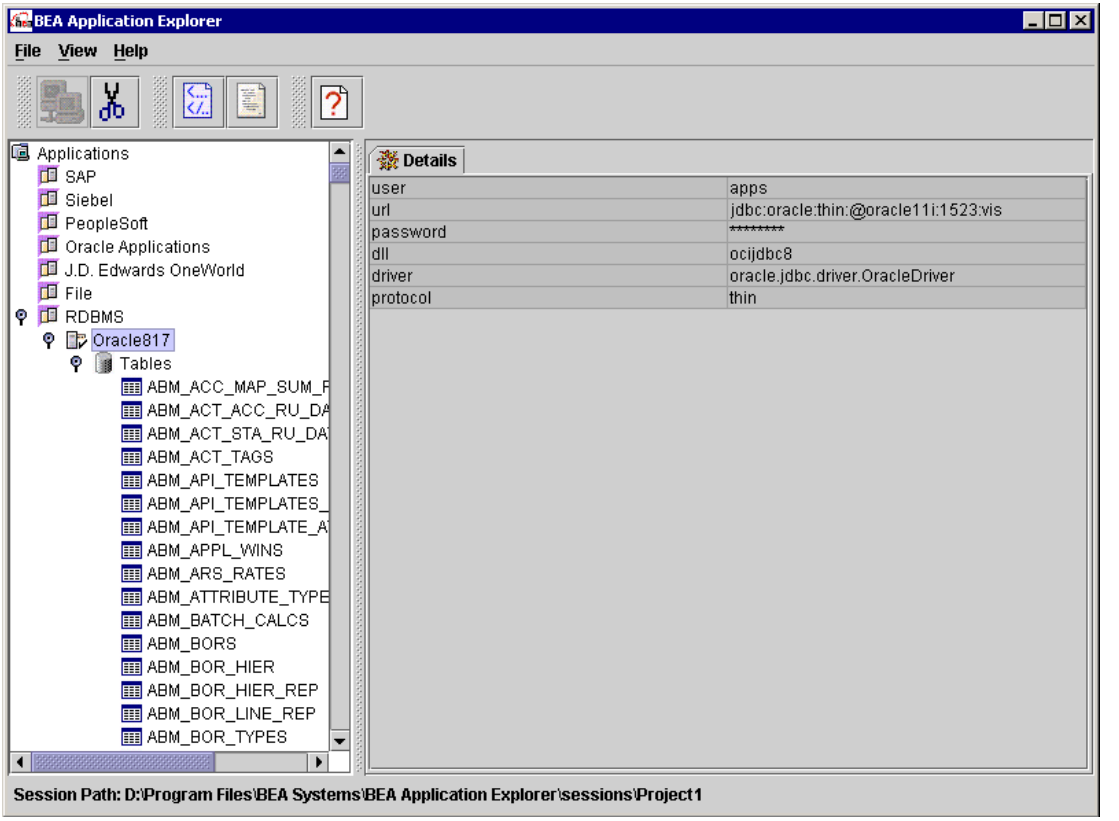
Viewing metadata is useful when creating schemas. For more information, see "Generating Event Schemas" on page 3-24 and "Generating Service Schemas" on page 3-31.

The BEA Application Explorer enables you to view the tables available in the RDBMS. To view the available tables:

1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.

2. When the BEA Application Explorer opens, expand the table node under the desired connection.

Figure 3-18 BEA Application Explorer - Expanded Table Node

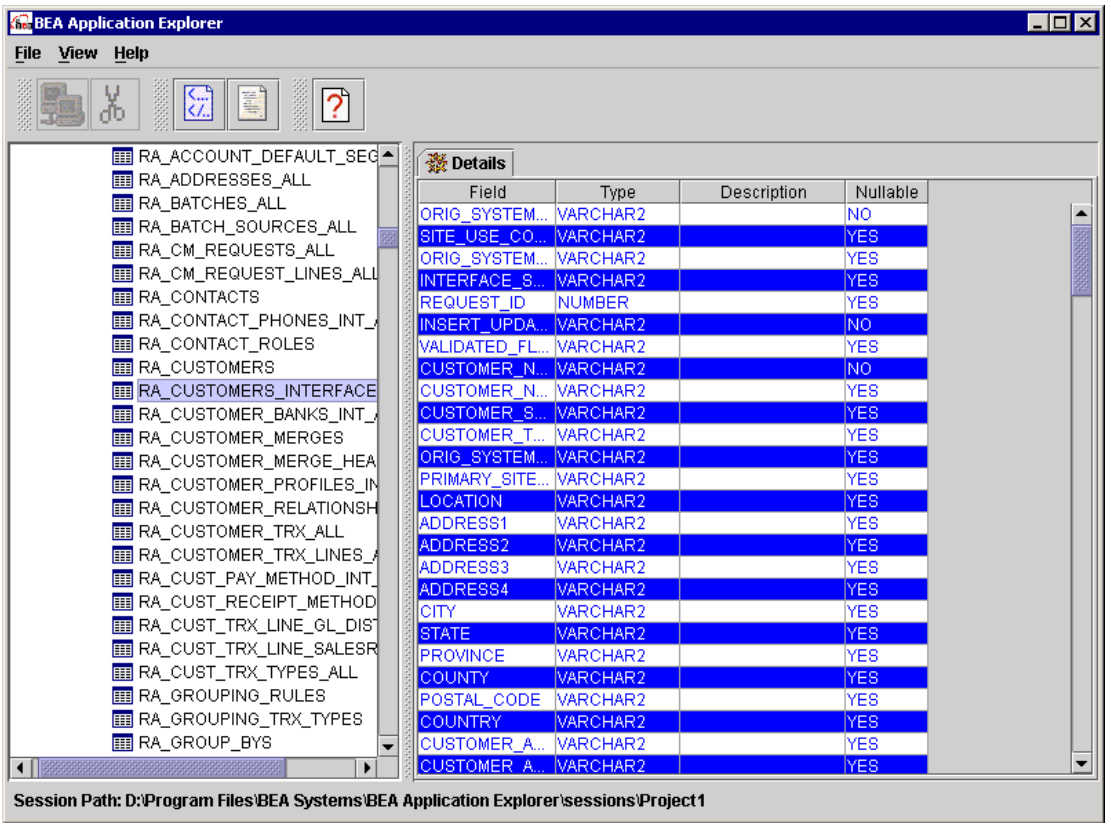


3. Scroll down and select the specific table to review.

Note: The list of tables includes all tables on the RDBMS. It is possible that the user ID used for the connection does not have access to the specified table. If this is the case, the creation of schemas fails.

When you select a specific table by clicking it, the table metadata appears in the right pane, as shown in the following figure. This information can be used to determine the table (or tables) and fields to use when creating the schema.

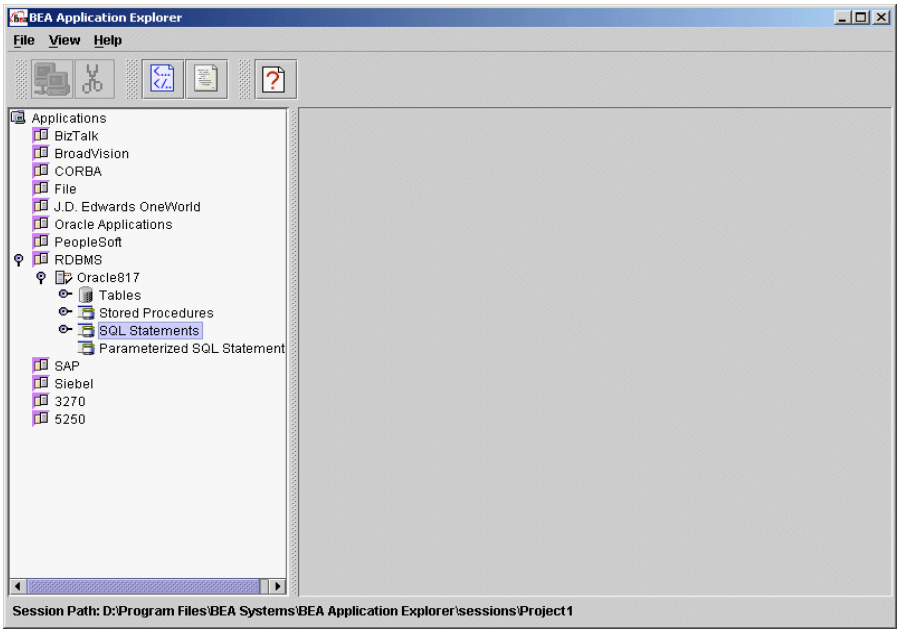
Figure 3-19 BEA Application Explorer - Table Metadata



Viewing Stored Procedures

Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer. When you first start the BEA Application Explorer, in the left pane you see a list of all BEA WebLogic Adapters supported by the version of the BEA Application Explorer you are using.

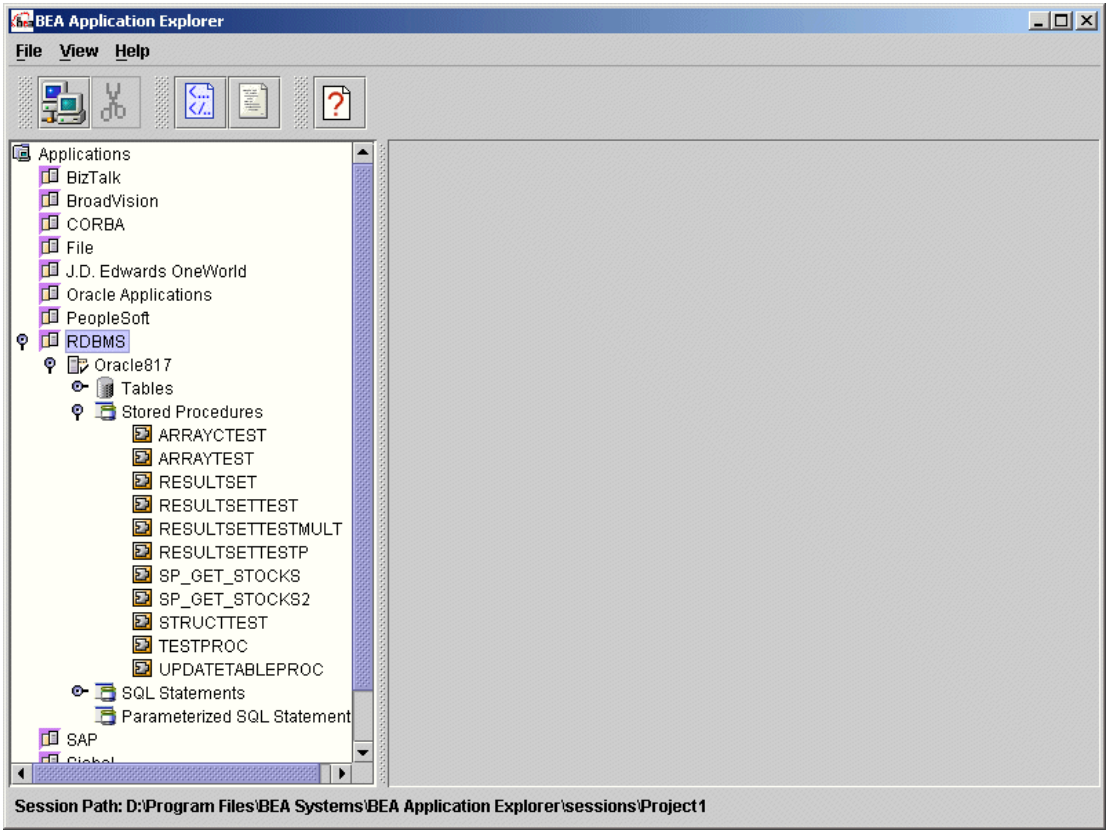
Figure 3-20 BEA Application Explorer



1. From the BEA Application Explorer main menu, expand the RDBMS branch.
2. Connect to a back-end Oracle system. For more information on connecting to back-end systems, see "Connecting to an RDBMS" on page 3-2 or "Connecting to an RDBMS Using an Existing Connection" on page 3-9.

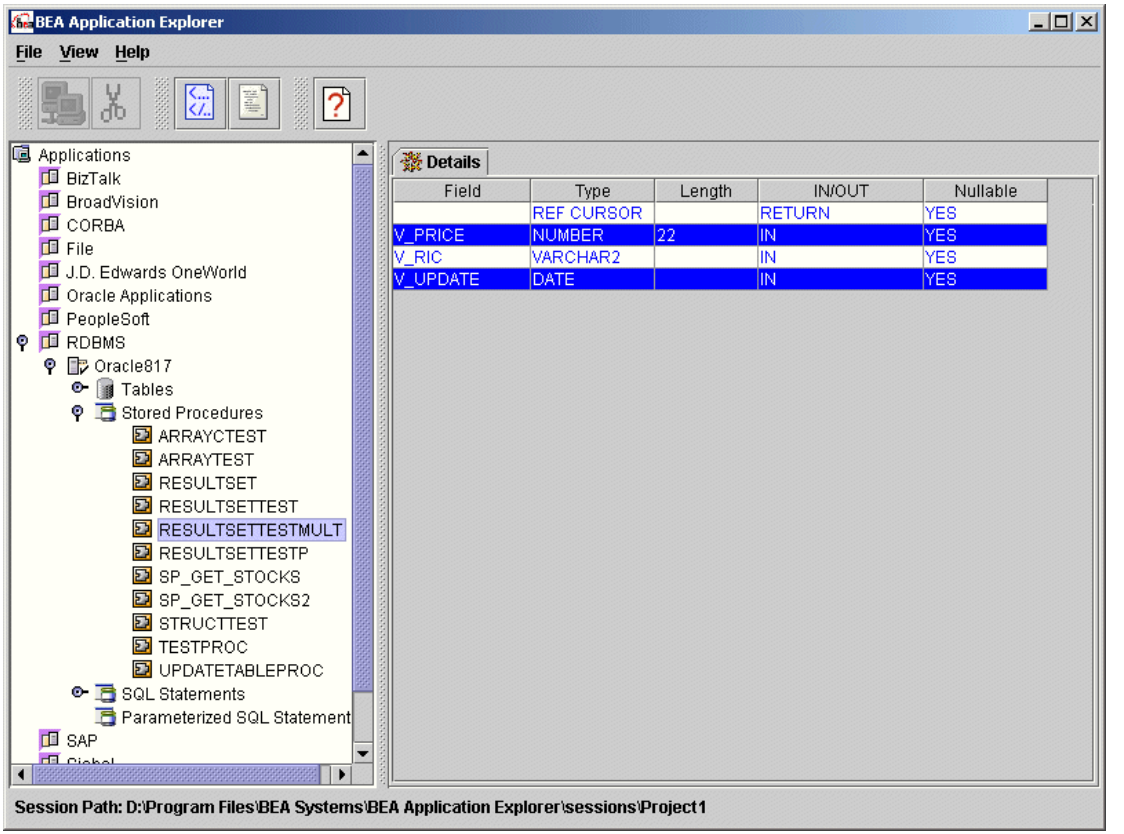
3. Under the Connection name, expand the Stored Procedures branch by clicking the icon next to Stored Procedures. All stored procedures available under the schema you specified appear, as shown in the following figure.

Figure 3-21 BEA Application Explorer - Expanded Stored Procedures List



4. Highlight a stored procedure to view its parameters, as shown in the following figure.

Figure 3-22 BEA Application Explorer - Stored Procedure Parameters



The RESULTSETTESTMULT Stored Procedure contains three input parameters, V_PRICE, V_RIC, and V_UPDATE, listed in the Field column.

Note: Output parameters are displayed without names in the field column.

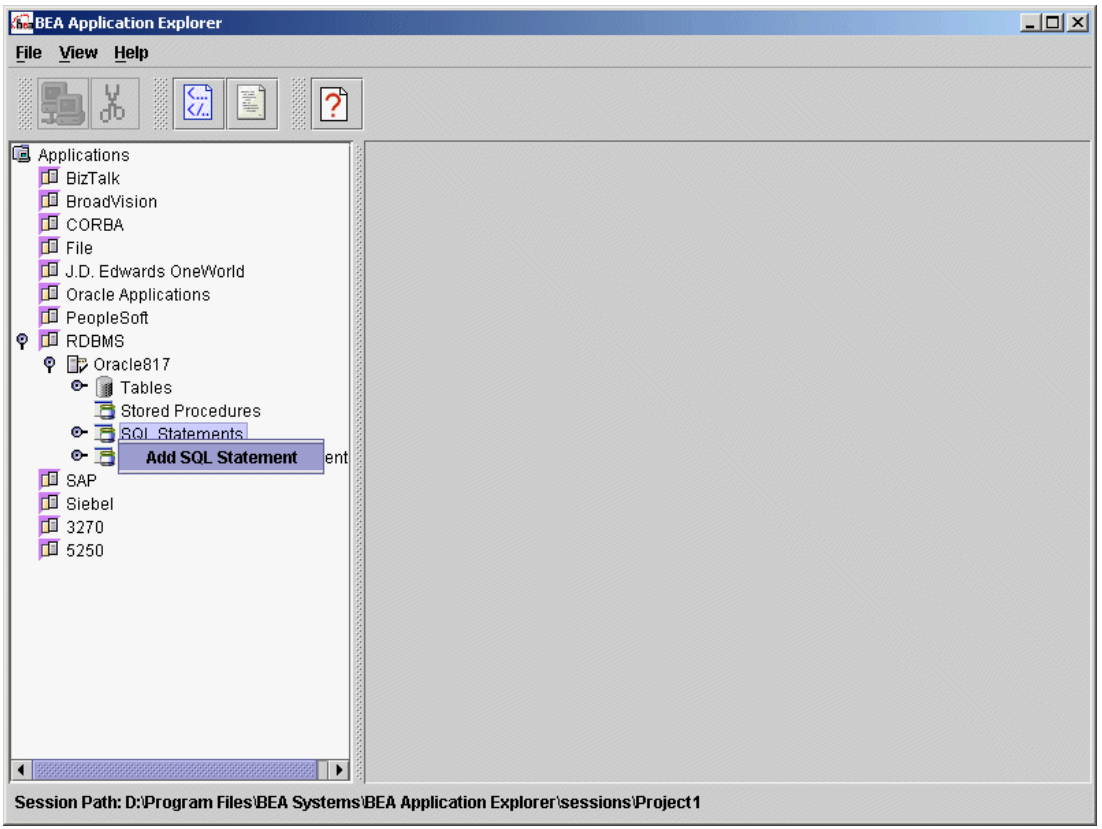
For Oracle Stored Procedures returning result sets, the first parameter is designated as a type REF_CURSOR. This is actually the returning result set.

Generating Event Schemas

The generation of schemas is handled under the SQL statement node. To start the process of generating an event schema:

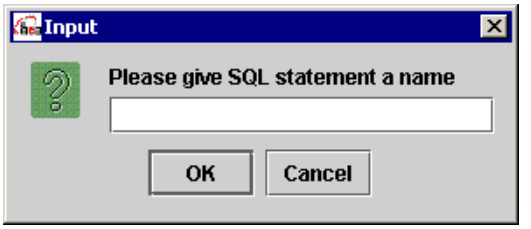
1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.
2. When the BEA Application Explorer opens, right-click the SQL statement node and select Add SQL Statement, as shown in the following figure.

Figure 3-23 BEA Application Explorer - Add SQL Statement



The SQL statement input box opens.

Figure 3-24 SQL Statement Name Input Box



3. Enter a name for the schema group being generated.

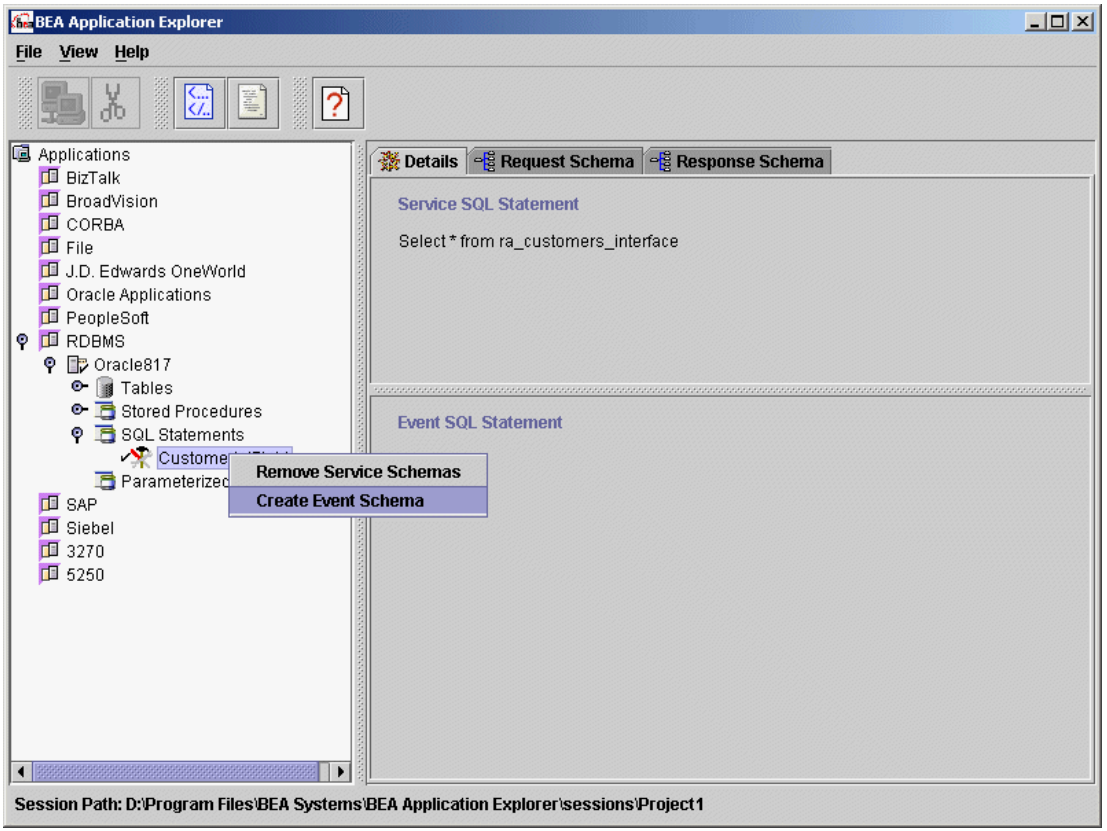
   It is good practice to specify a name that describes the service. For example, a name of CustomerInt would represent an event on the Customer Interface table returning a Field format response document.

4. Click OK.

   After the SQL statement node is added, you are ready to build schemas.

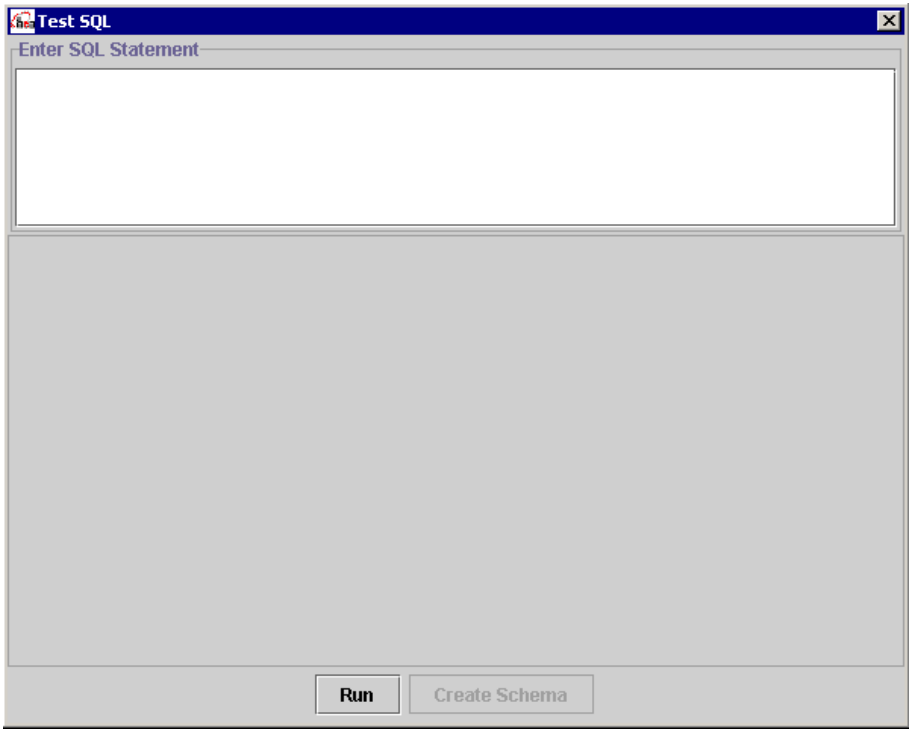
5. To generate the schemas, right-click the SQL Statement and select the Create Event Schema option, as shown in the following figure.

Figure 3-25 BEA Application Explorer - Create Event Schema



A Test SQL window opens.

Figure 3-26 Test SQL Statement



6. In the top pane, type the SQL statement to be used by the application view service.

   The RDBMS listener allows the creation of events that utilize complex SQL statements, including JOINs, WHEREs, and other verbs.

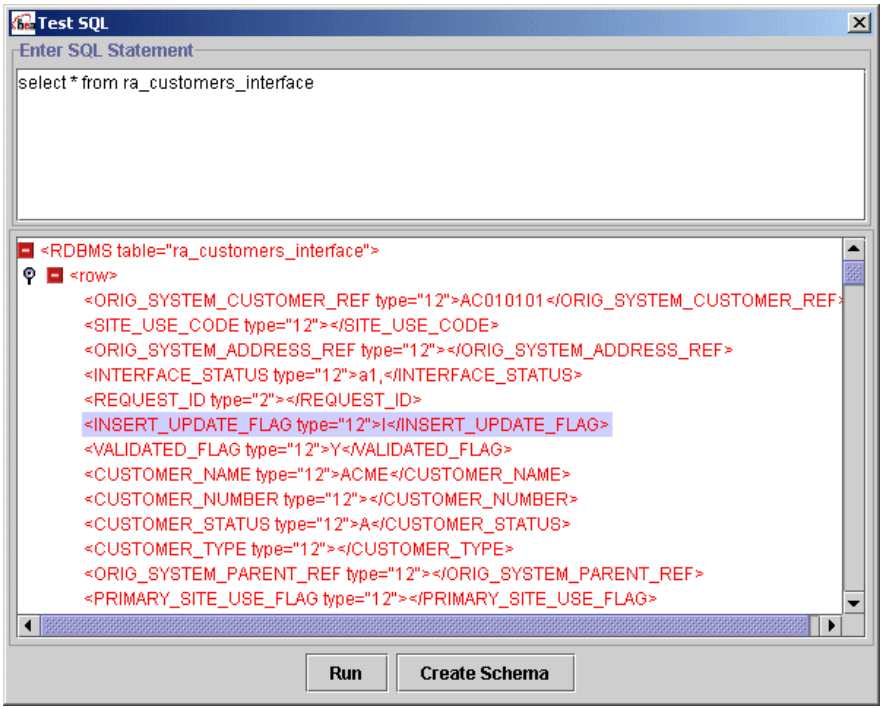
   When entering the SQL statement, ensure you use the SQL generating the input record for the event.

7. Click Run.

   If there is a problem executing the SQL statement, the error message from the RDBMS appears in an error window.

   When the SQL statement is correct, the following window appears.

Figure 3-27 Test SQL Results

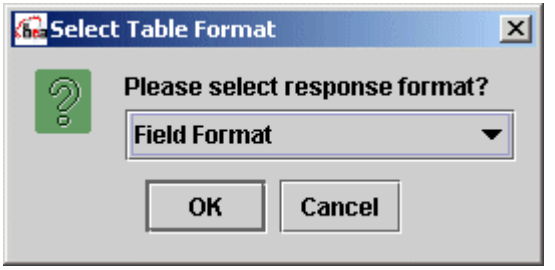


The SQL can be modified and the statement re-run, or the Create Schema button can be selected to produce the schema.

8. Click Create Schema.

The Select Table Format window is displayed.

Figure 3-28 Select Table Format Window



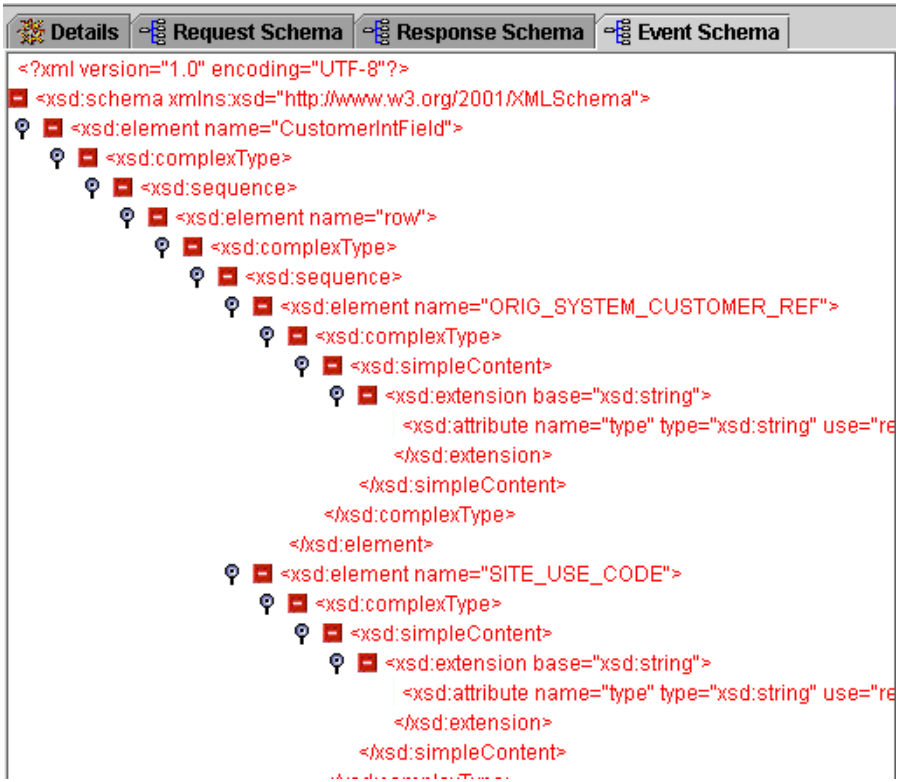
9. Select a formatting option from the field format drop-down list.

   The value you select must match the format used to generate the event adapter application view.

10. Click OK to create the schema.

The right pane of the BEA Application Explorer displays the SQL statement used, in addition to the Event, Request, and Response Schema tabs, as shown in the following figure.

Figure 3-29 Schema Display



The schema is now generated and ready to use.

Generating Service Schemas

You can generate service schemas for:

- SQL statements
- Parameterized SQL statements

There are significant differences between request schemas generated by an SQL statement and a parameterized SQL statement.

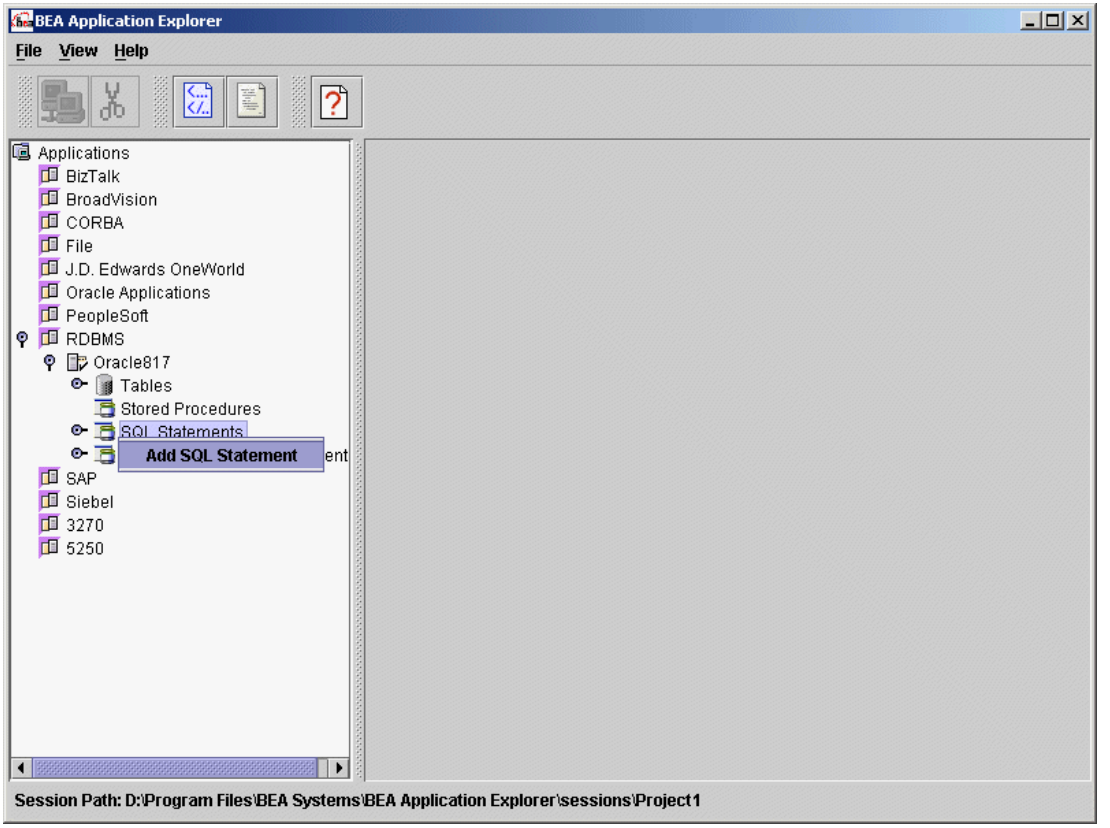
Generating Service Schemas Under the SQL Statement Node

Generating service schemas under the SQL statement node produces the required request and response schemas for building a service adapter application view. To generate a service schema:

1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.

The BEA Application Explorer opens.

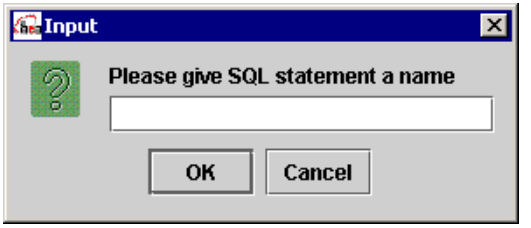
Figure 3-30 BEA Application Explorer - Add SQL Statement



2. Right-click the SQL Statements node and select Add SQL Statement.

The SQL statement name input box appears.

Figure 3-31 SQL Statement Name Input Box



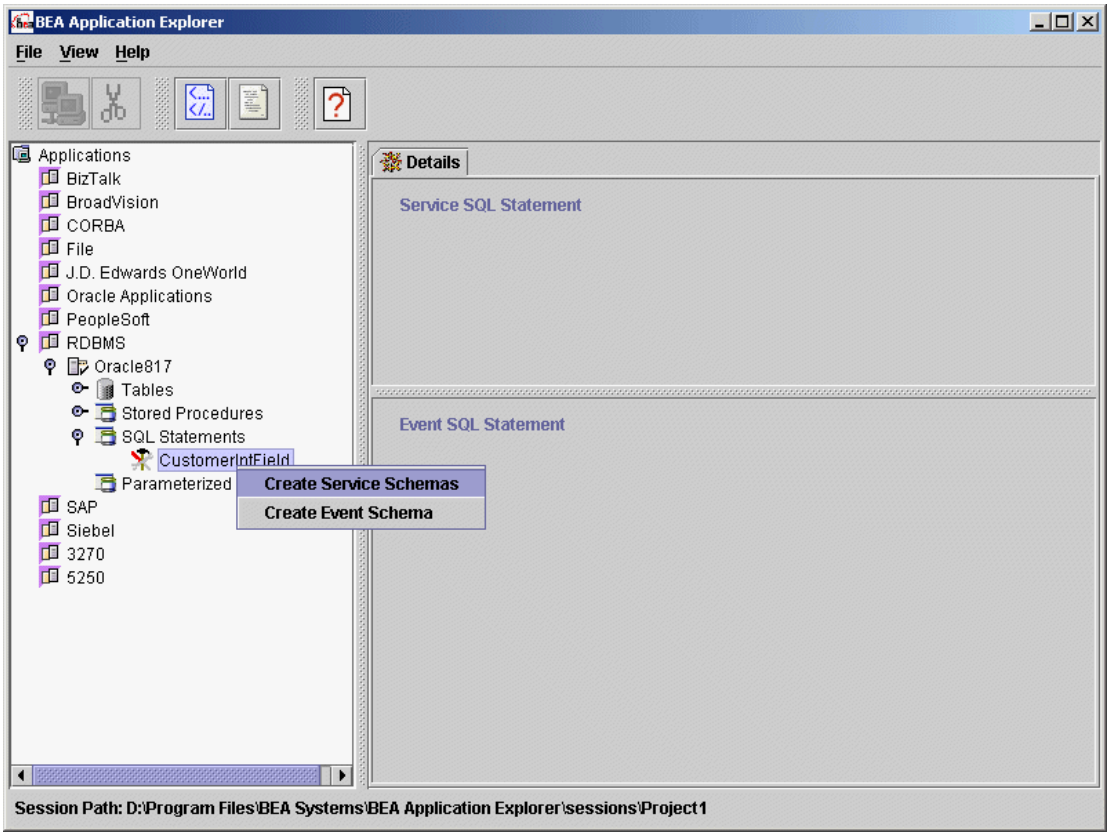
3. Provide a name for the schema group being generated.

   It is good practice to specify a name that describes the service. For example, a name of CustomerIntField would represent a request against the Customer Interface table returning a Field format response document.

4. Click OK.

   After the SQL statement node is built, you are ready to build schemas.

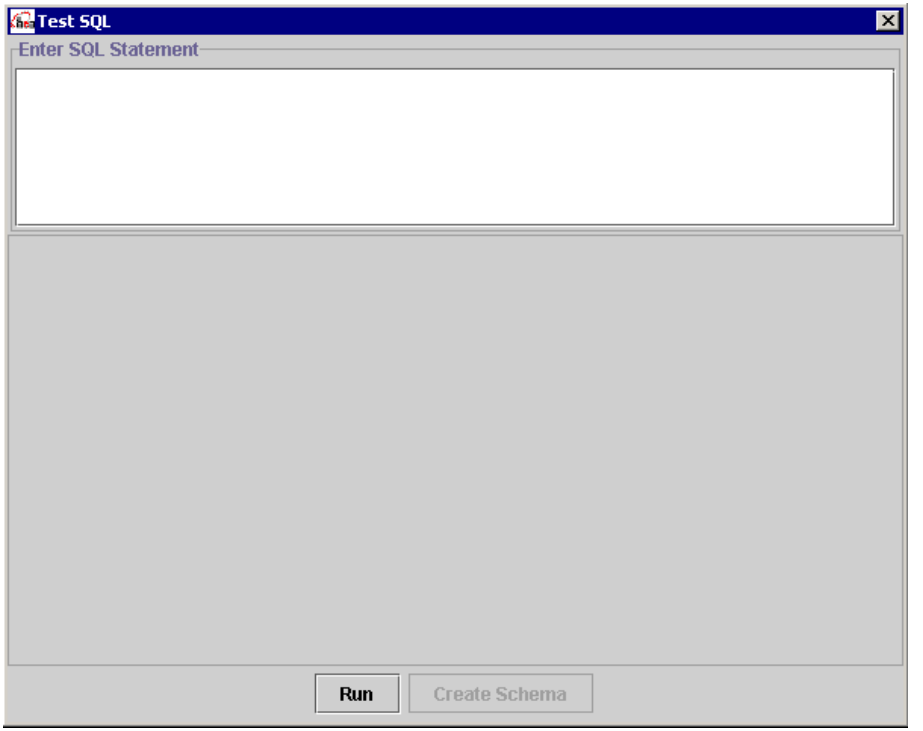
Figure 3-32 BEA Application Explorer - Create Service Schemas



5. To generate the schemas, right-click the SQL statement and select Create Service Schemas.

A Test SQL window opens.

Figure 3-33 Test SQL Statement

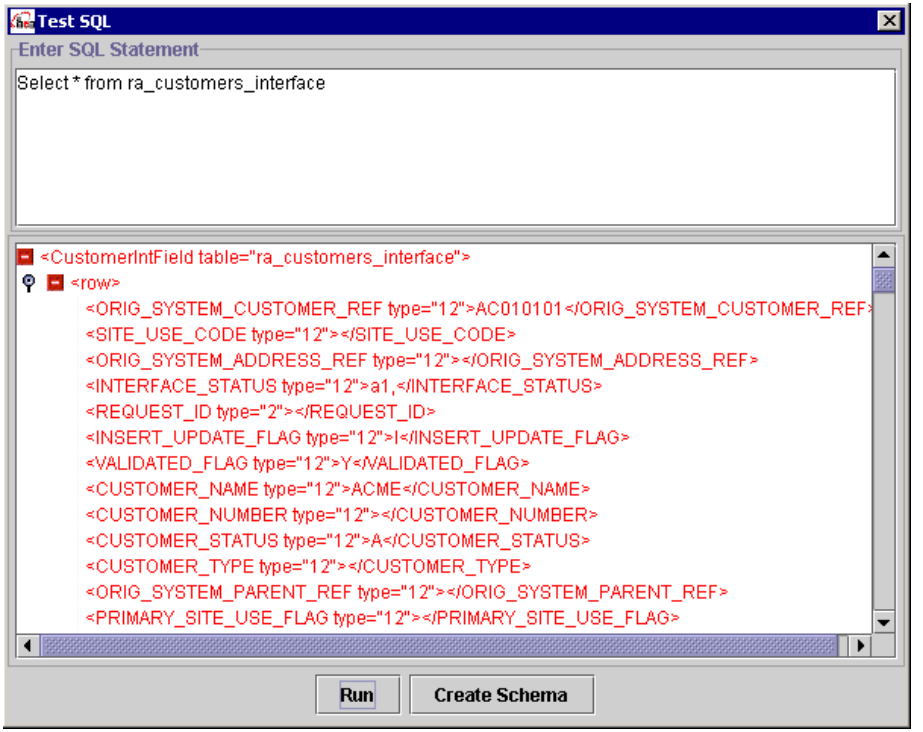


6. In the top pane, type the SQL statement to be used by the application view service.
7. Click Run.

   If there is a problem executing the SQL statement, the error message from the RDBMS appears in an error window.

When the SQL statement is correct, the following window appears.

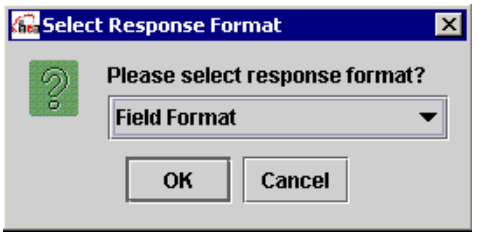
Figure 3-34 Test SQL Results



The SQL can be modified and the statement re-run, or the Create Schema button can be selected to produce the schemas.

Before the schemas are created, the systems prompts you for the format of the response.

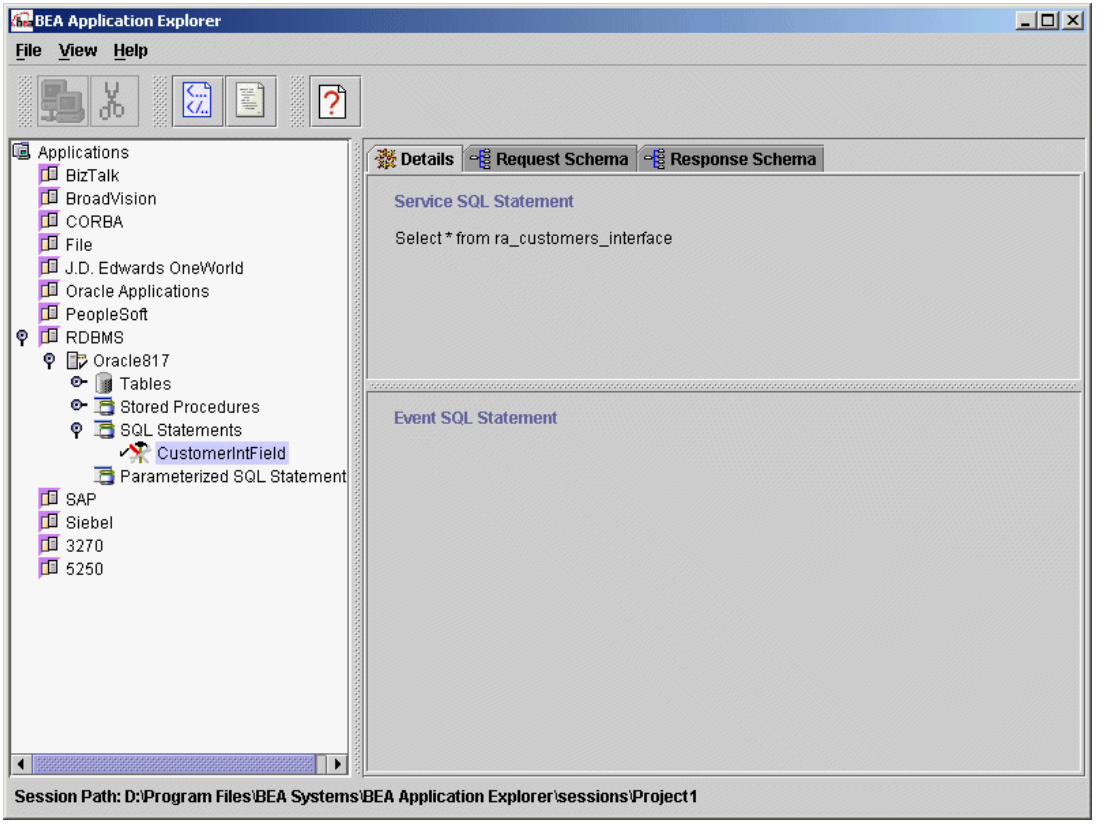
Figure 3-35 Select Response Format



8. Select a formatting option from the response format drop-down list. The value selected must match the format value used to generate the application view service, or the schema of the response document will not validate.
9. After a format is selected, click OK to produce the schemas.

The right pane displays the SQL statement used, in addition to the Request and Response Schema tabs.

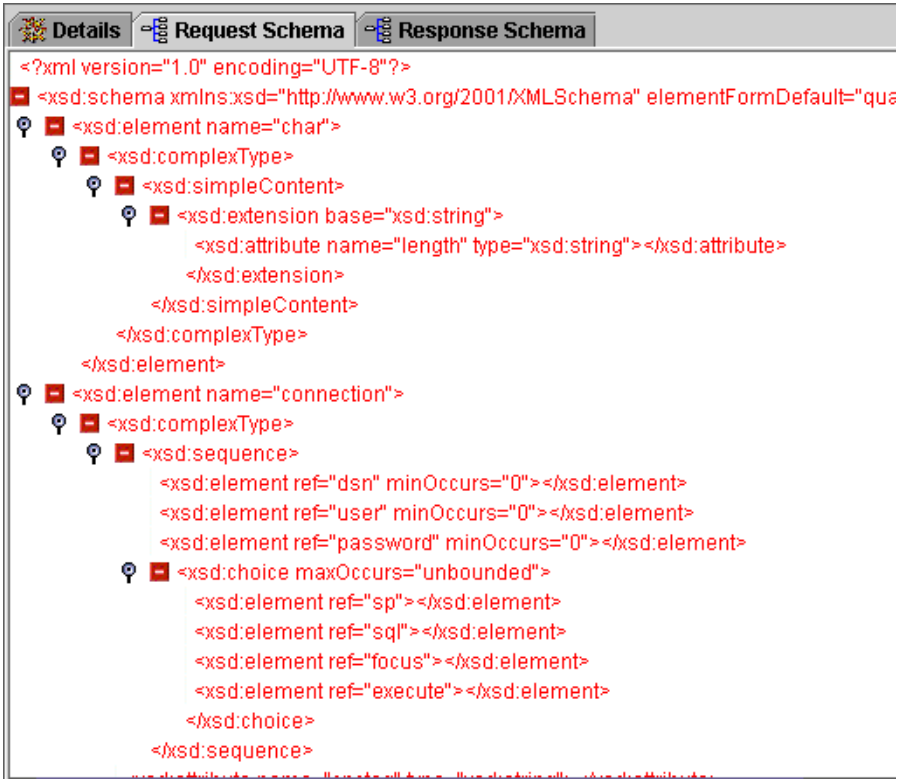
Figure 3-36 BEA Application Explorer - Schema Display



10. To view the schema, click the desired tab at the top of the right pane.

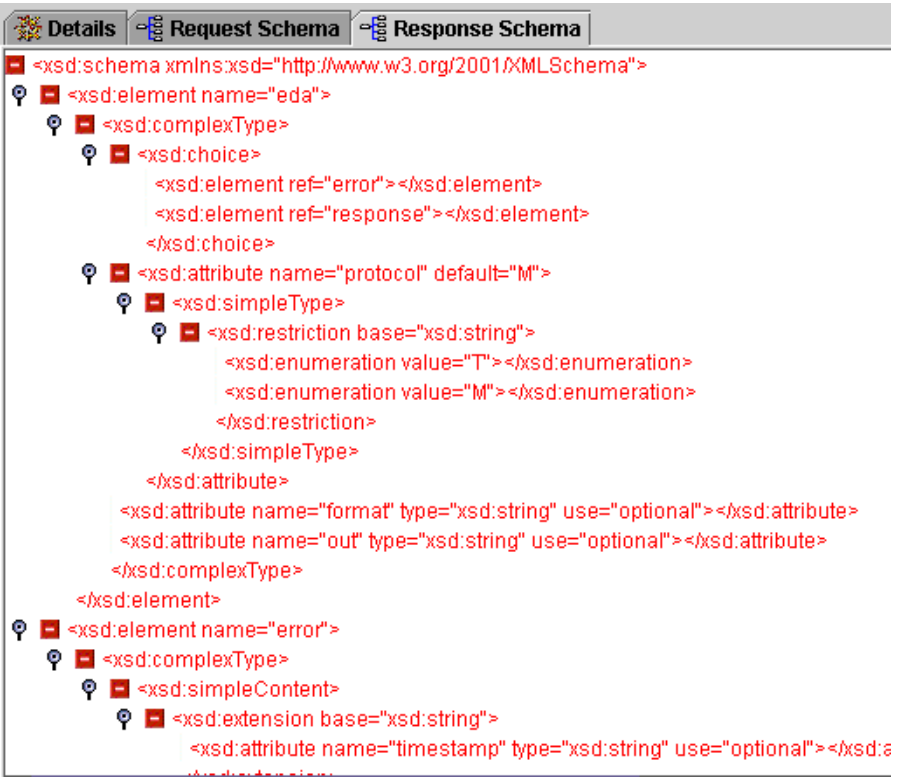
The following figure shows the request schema.

Figure 3-37 Request Schema



The following figure shows the response schema.

Figure 3-38 Response Schema



The schemas are now generated and ready to use.

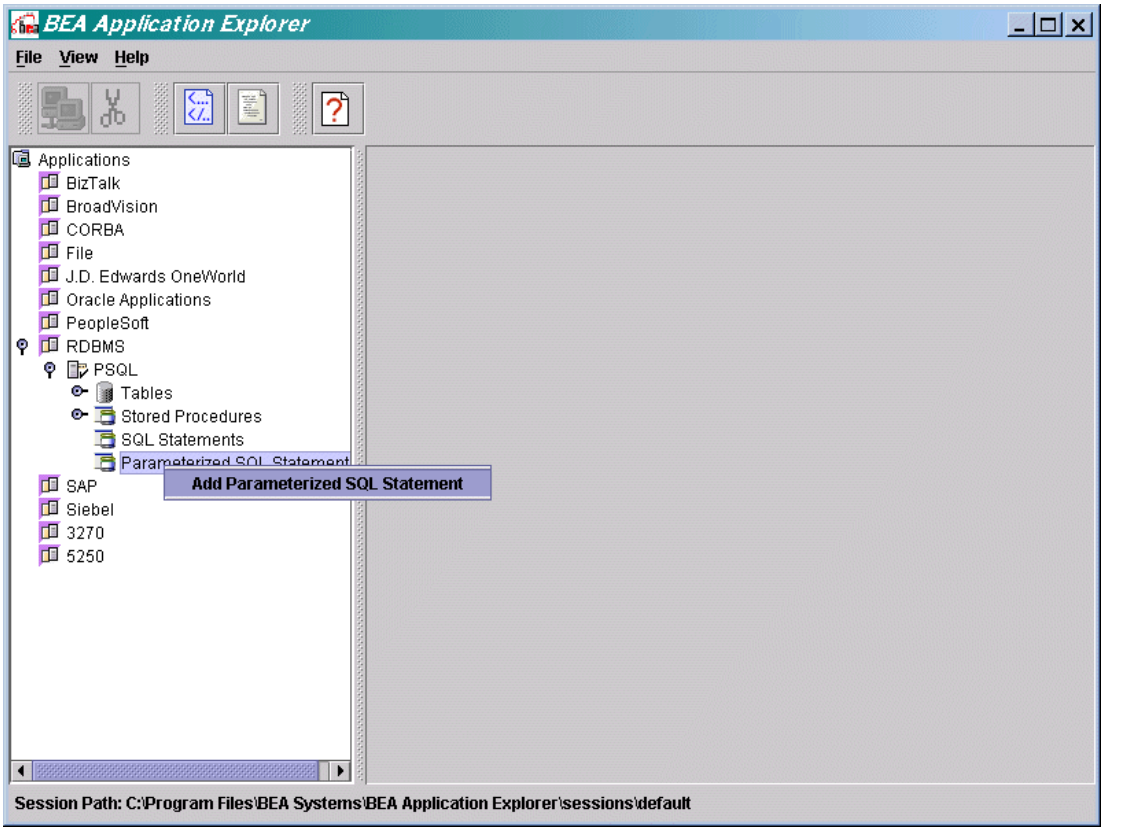
Generating Service Schemas Under the Parameterized SQL Statement Node

Generating schemas under the Parameterized SQL statement node produces the required request and response schemas for building a service adapter application view.

To generate service schemas:

1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.

Figure 3-39 BEA Application Explorer - Add Parameterized SQL Statement

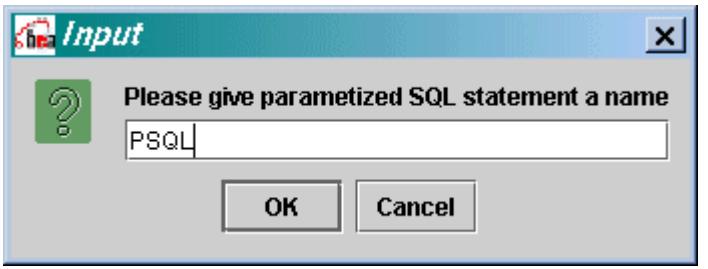


2. When the BEA Application Explorer opens, open and expand an existing connection.
3. Right-click the Parameterized SQL Statements node and choose Add Parameterized SQL Statement.

   For more information on how to create and configure a connection, see "Connecting to an RDBMS" on page 3-2.

   An input box prompts you for the name of the parameterized SQL statement.

Figure 3-40 Parameterized SQL Statement Name Input Box



   a. Enter a name for the schema group that you are generating.

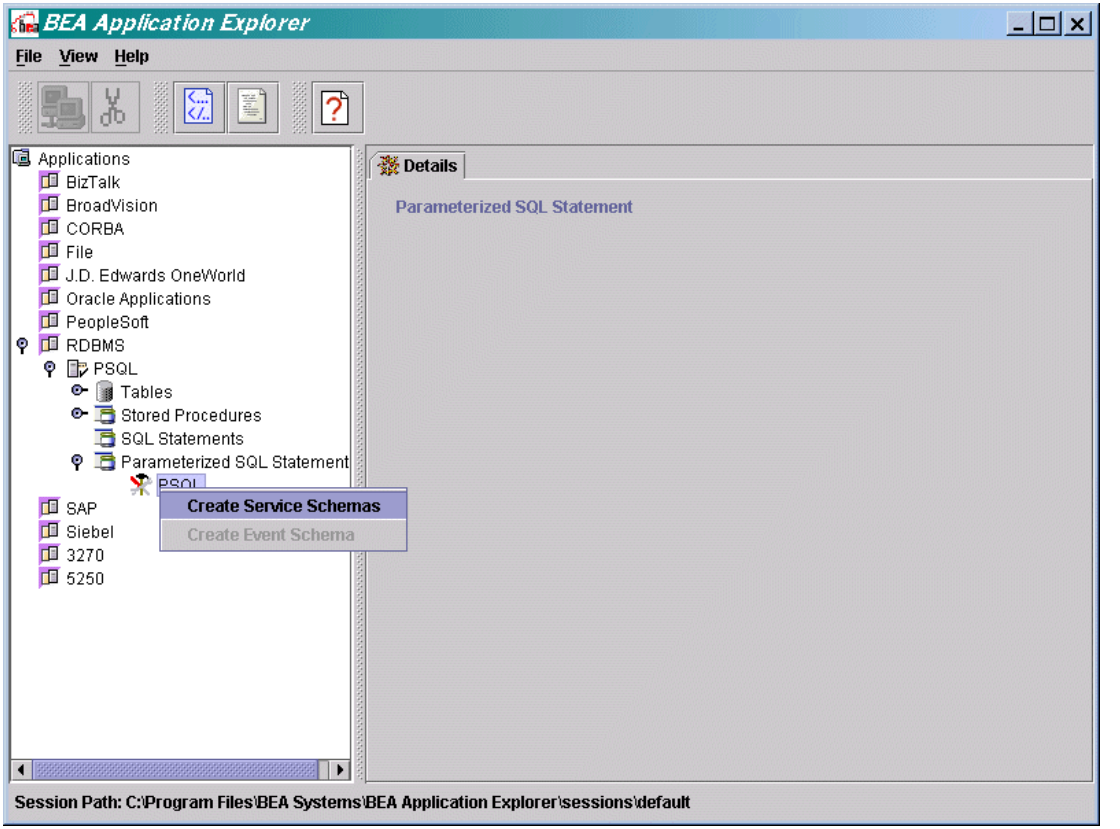
      Note: It is good practice to specify a name that describes the service. For example, the name `PSQL` might represent a parameterized request against the `PSQL` table returning a field format response document.

   b. Click OK.

   You have created the parameterized SQL statement node, and you can generate schemas.

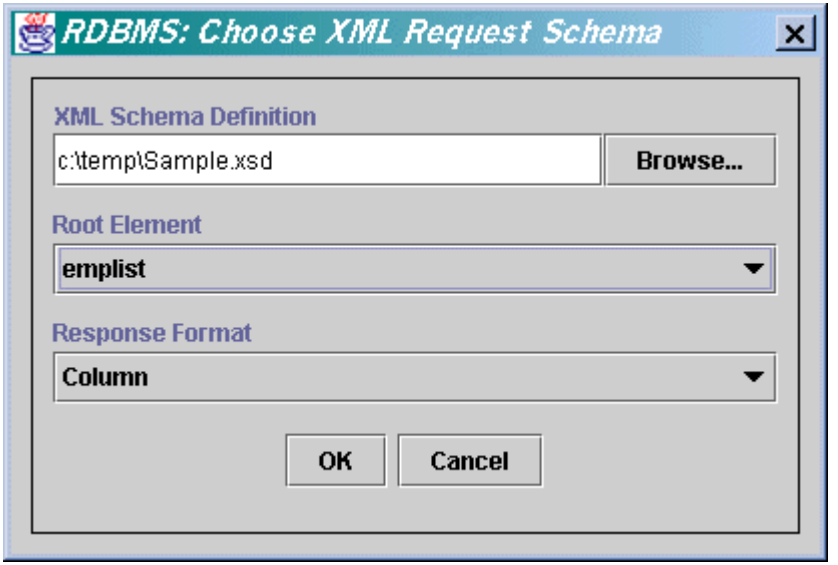
4. Right-click the new SQL Parameterized statement node and choose Create Service Schemas.

Figure 3-41 BEA Application Explorer - Creating Service Schemas



The Choose XML Request Schema dialog box appears.

Figure 3-42 Choose XML Request Schema Dialog Box



For this example, the following input XML is used (the corresponding schema is not displayed).

Listing 3-1 Input XML Sample

```
<emplist>
<emp>
       <name age='32'>Bess Armstrong</name>
       <address>211 Chadwick Rd</address>
       <citystatezip>New York, NY 10009</citystatezip>
</emp>
<emp>
       <name age='40'>Merle Pensk</name>
       <address>41 Church Rd</address>
       <citystatezip>Marlboro, NJ 07601</citystatezip>
</emp>
</emplist>
```

a. Click Browse and select the XML Input Schema Definition for the input XML document.

   The input XML document is the XML that contains the values that will be used to substitute for the parameters.

b. Click the arrow for the Root Element drop-down list to select the node in the XML document that corresponds to the root node of the input XML document.

c. Click the the arrow for the Response Format drop-down list to select the response format.

   The choices are Column, Row, and Field, as defined in the following table.

Table 3-1 Format Definitions

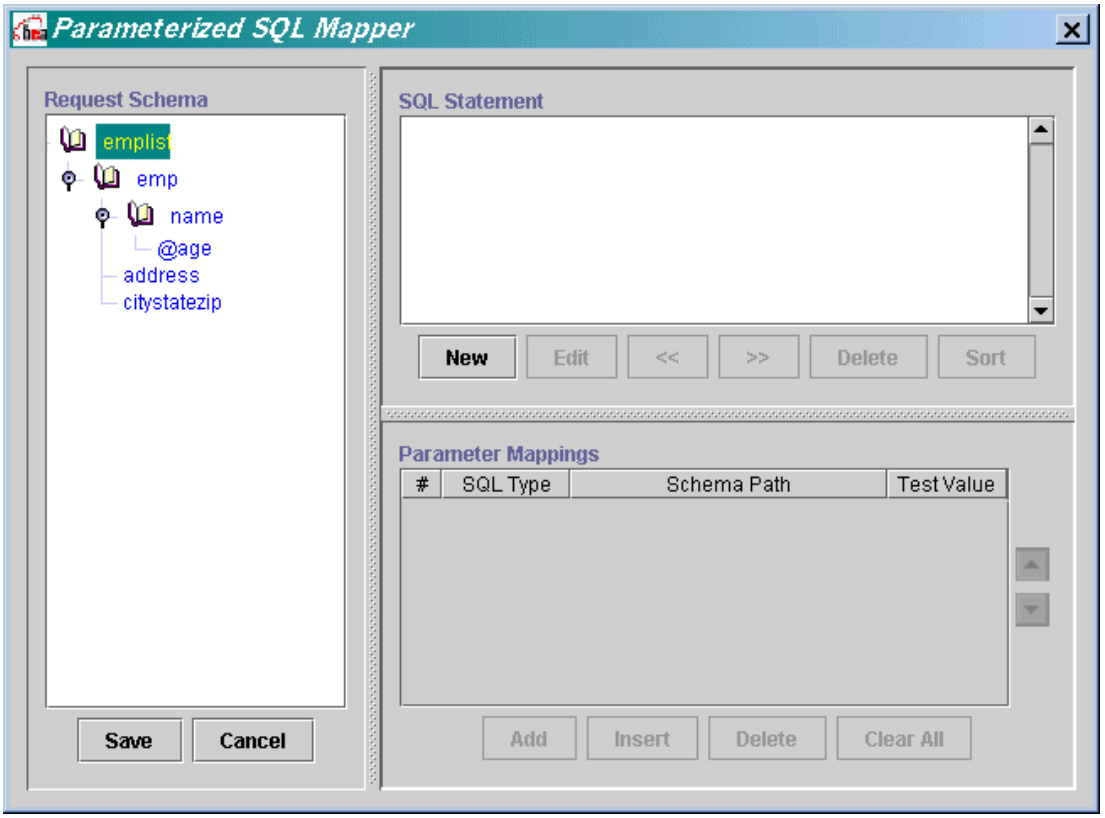
format*
(*Required)
	Choose one of the following:
	■ **row.** The data that is produced is returned on a single line (per record) enclosed in `<row>` tags.
	■ **column.** The data produced is returned field by field, and each field is enclosed in <column> tags. The column tag has an attribute whose value is the name of the field; for example,
	`<row>`
	`<column name="ID">1000</column>`
	`<column name="First_Name">Scott</column>`
	`</row>`
	■ **field.** The data produced is returned field by field, and each field is enclosed in a tag that bears the field name; for example,
	`<row>`
	`<ID>1000></column>`
	`<FIRST_NAME>Scott</column>`
	`</row>`

Note: Due to the flexibility of parameterized SQL statement processing, the input document can be in virtually any valid XML format, from a standards-based format such as an OAGIS Business Object Document or xCBL document, to a non-standards-based XML document, to the output of a previous task. To support this flexibility, you must provide the predefined `.xsd` file and the root name for the request document.

d. Click OK.

The Parameterized SQL Mapper window opens.

Figure 3-43 Parameterized SQL Mapper Window

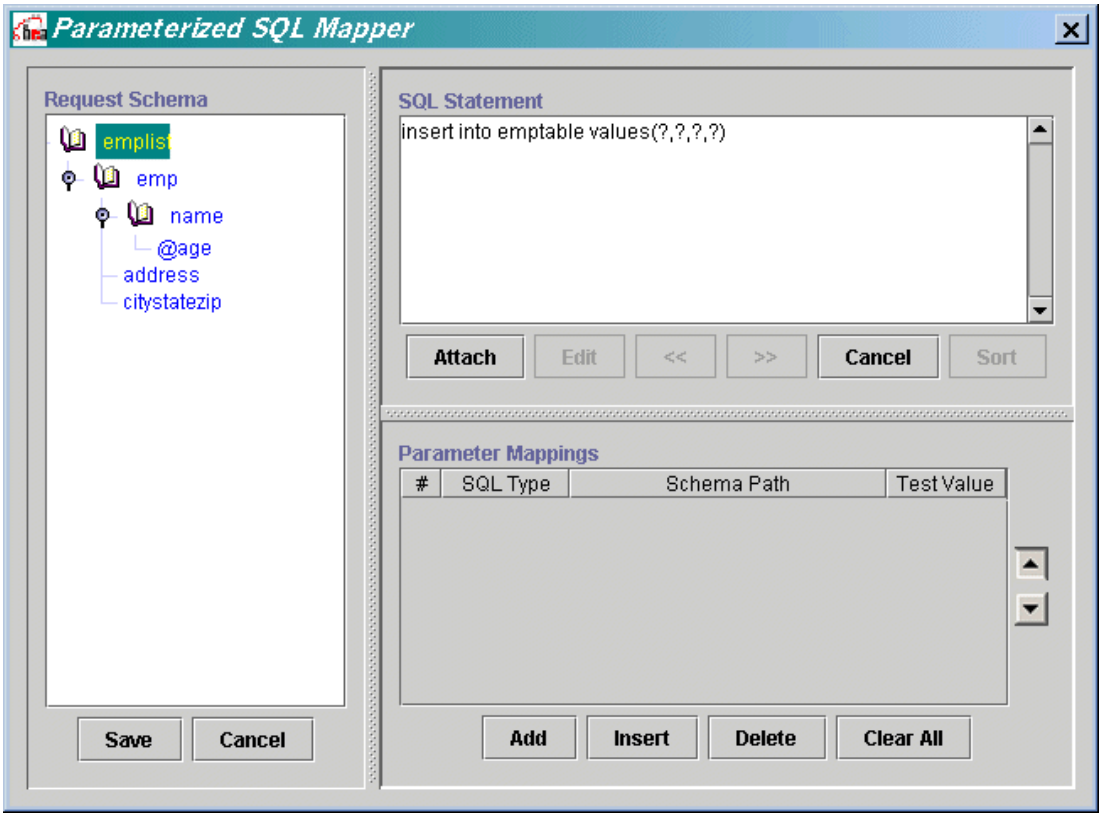


This mapping tool can be used to map fields from an input XML document to the parameterized SQL you will enter.

a. Click New.

The SQL Statement test box becomes active.

Figure 3-44 Parameterized SQL Mapper - Active Test Box



b. In the active test box, enter a parameterized SQL statement.

   Substitute question marks for parameters.

   In this example, four fields (corresponding with four parameters) are inserted into the table called emptable (that is, the table field names: Name, Address, CityStateZip, and Age).

Note: For combining stored procedures with parameterized SQL, issue a CALL statement in the SQL statement box.

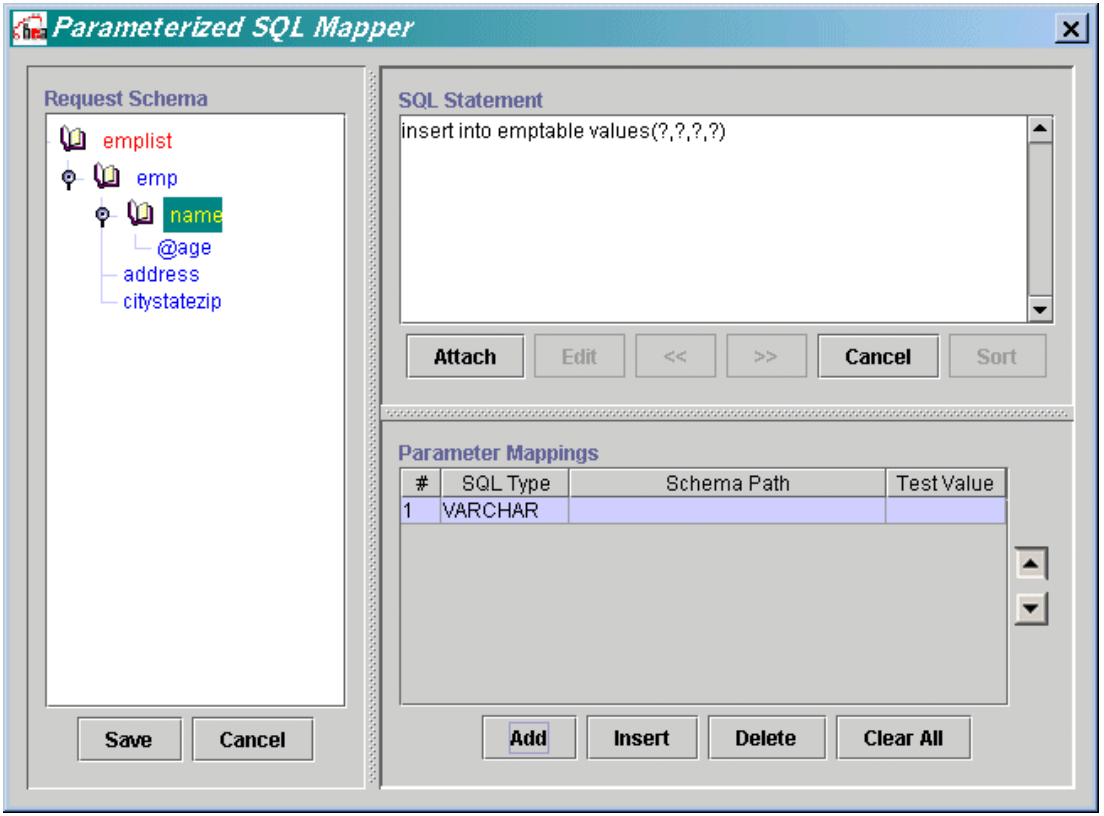
Syntax: CALL Stored_Procedure_Name(?,?,? and so forth). For example, the following is a stored procedure named MyStoredProcedure that takes three parameters:

```
Call MyStoredProcedure(?,?,?)
```

c. Click Add.

The Parameter Mappings Dialog box becomes active.

Figure 3-45 Parameterized SQL Mapper - Active Parameter Mappings

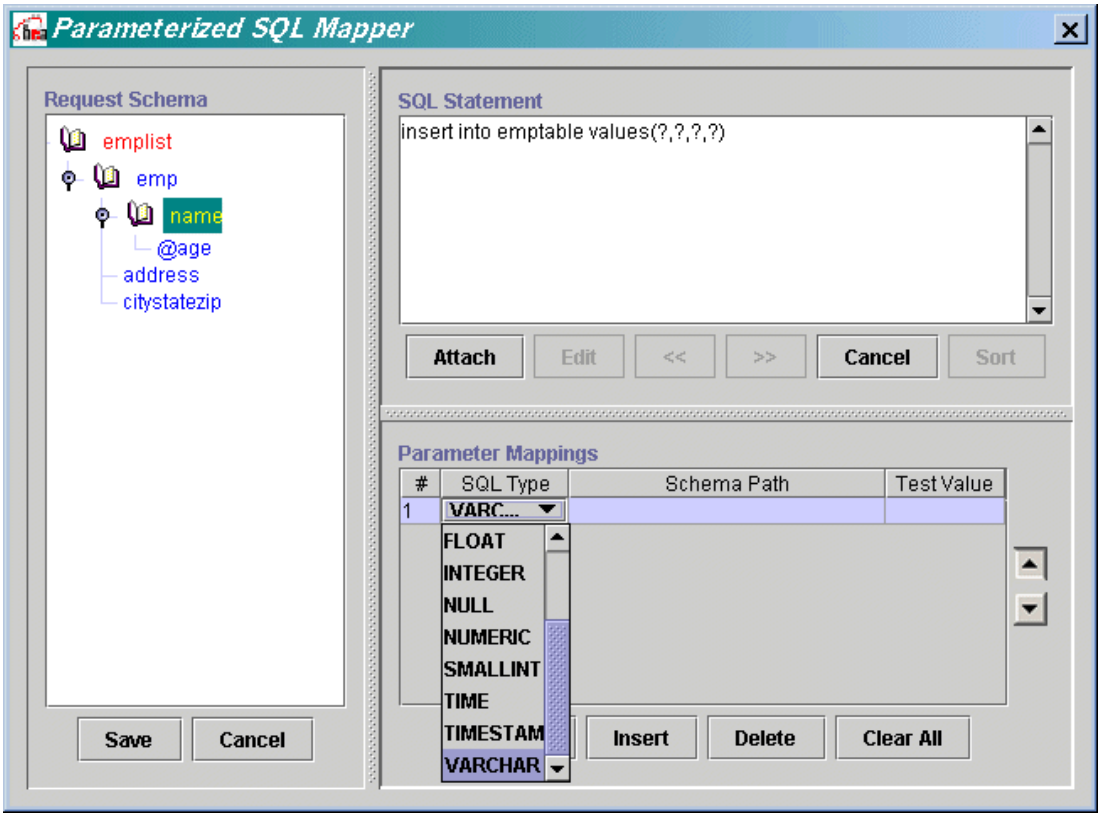


You now can map values taken from the sample XML to the four parameters specified in the parameterized SQL. For each parameter value there must be one entry in the Parameter Mappings dialog box.

d. Click SQL Type in the Parameter Mappings box.

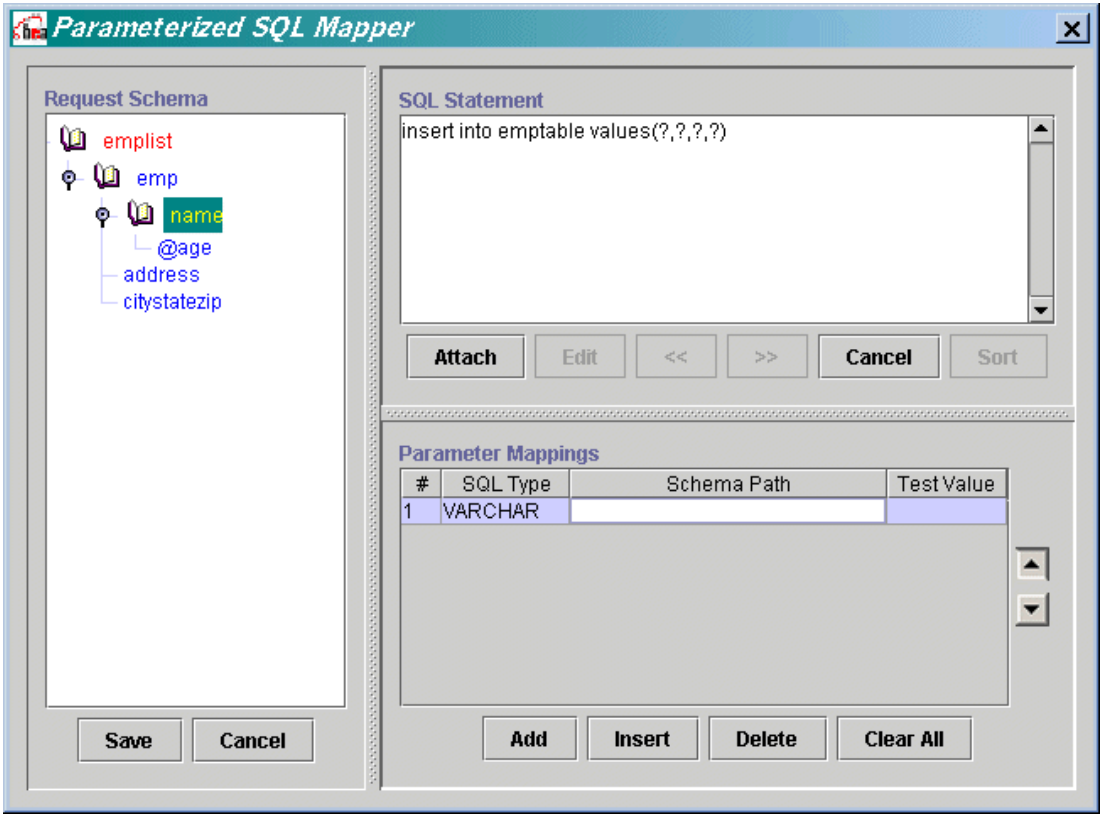
A drop-down list opens.

Figure 3-46 Parameter Mappings Drop-Down



e. Select the SQL Type of the first parameter specified positionally from the Parameterized SQL Statement.

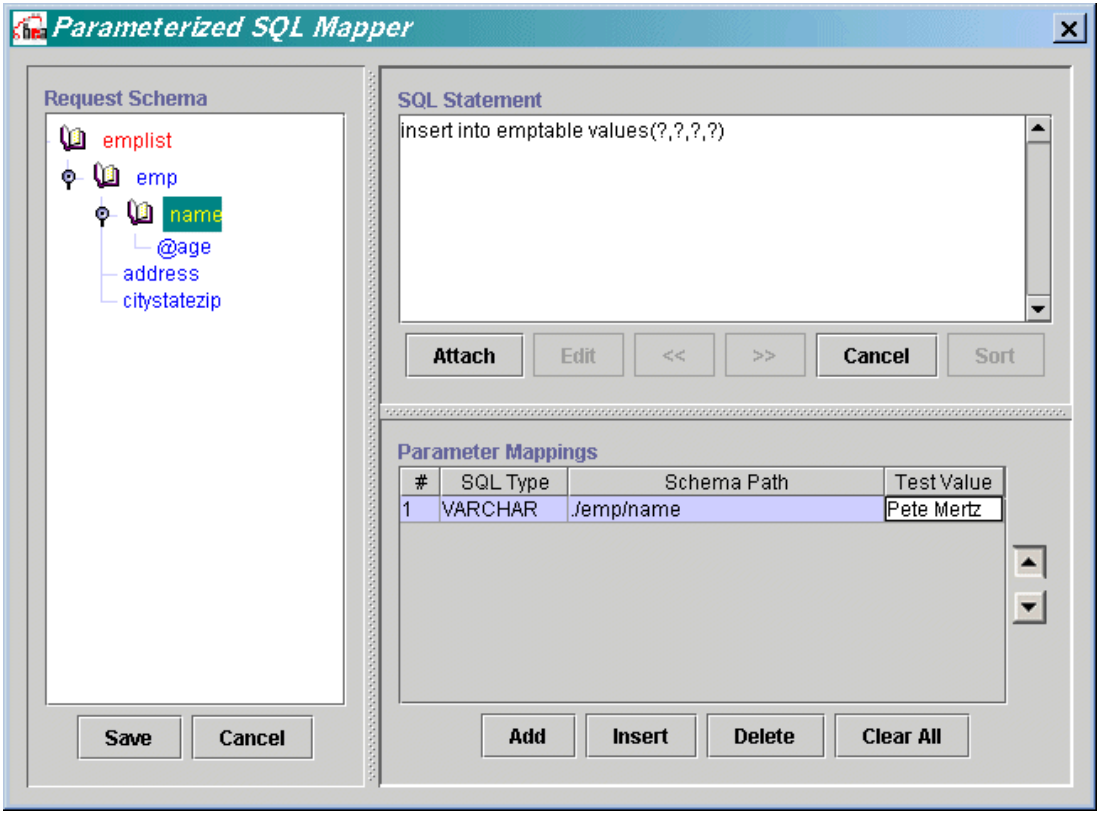
Figure 3-47 Parameterized SQL Mapper - Selecting Name Element



f. Double-click the value in the Request Schema tree that corresponds to the element name to which this parameter will map, in this example, the Name element.

The Schema Path field is now updated with the element selected relative to the root node of the document, as shown in the following figure.

Figure 3-48 Parameterized SQL Mapper with Updated Element

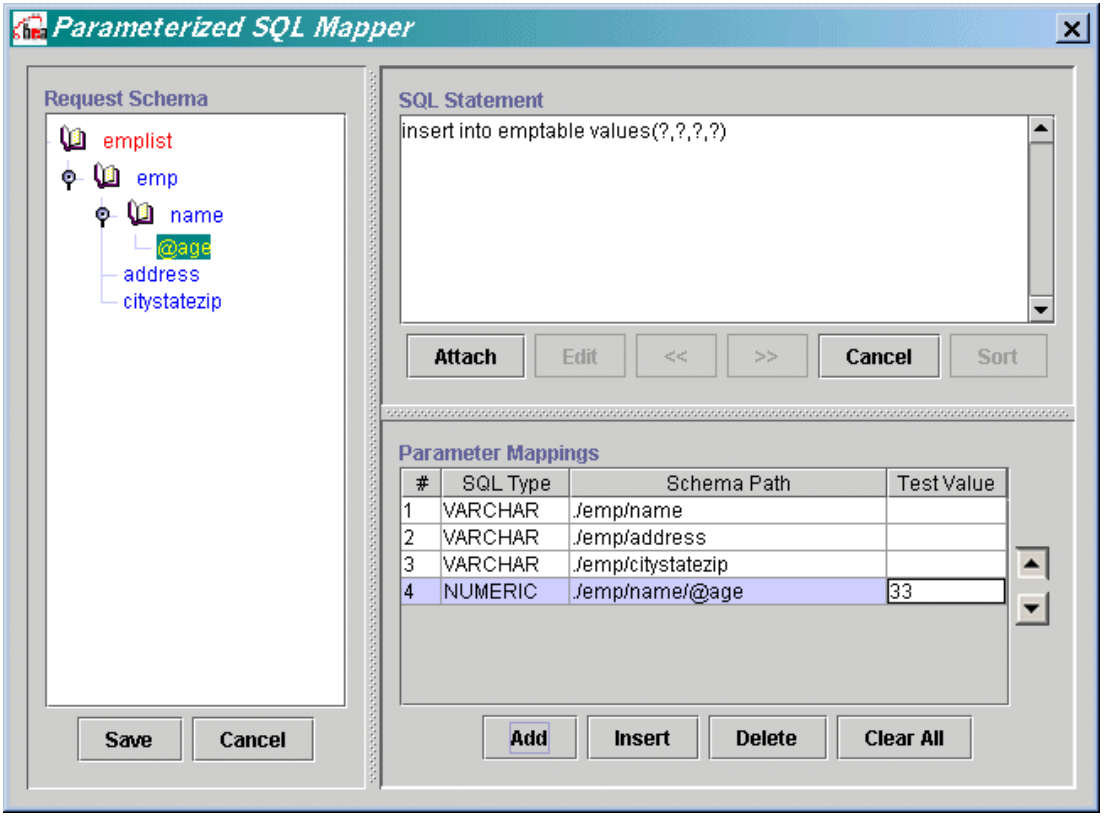


g. Click Test Value and enter a sample value used to verify this type.

h. Click Add to map the next parameter. Repeat this step until all parameters are mapped (in this case, four).

Note: In this example, it is possible to map a parameter to an attribute by selecting the @age entry from the Request Schema tree.

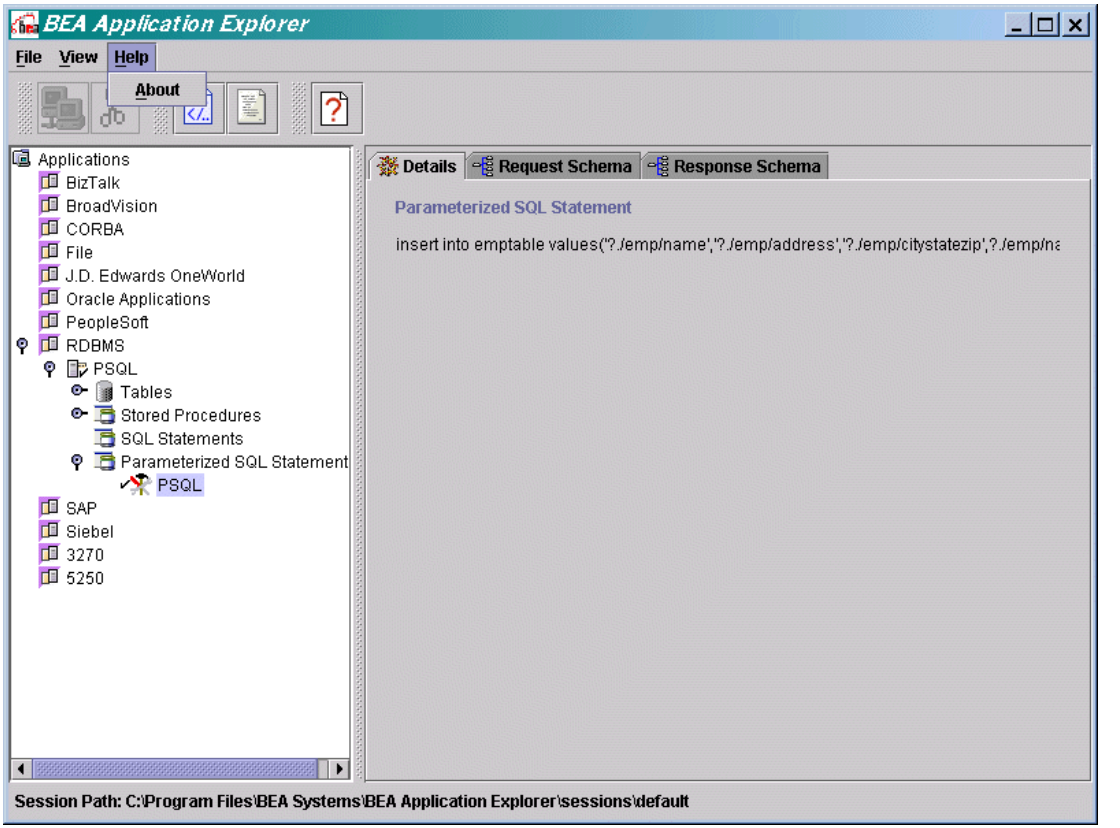
Figure 3-49 Parameterized SQL Mapper with @Age Entry Selected



5. After all the parameters are mapped, click Attach.

A Details tab appears.

Figure 3-50 BEA Application Explorer - Details Tab



6. Click the Request Schema tab or the Response Schema tab to view the Request or Response schema, respectively.

Combining Parameterized SQL Feature with Stored Procedures

It is possible to combine the parameterized SQL feature and stored procedures. In effect, this is done to set up a parameterized SQL statement that calls a stored procedure.

For more information on how to combine the parameterized SQL feature and stored procedures, see "Generating Service Schemas Under the Parameterized SQL Statement Node" on page 3-41.

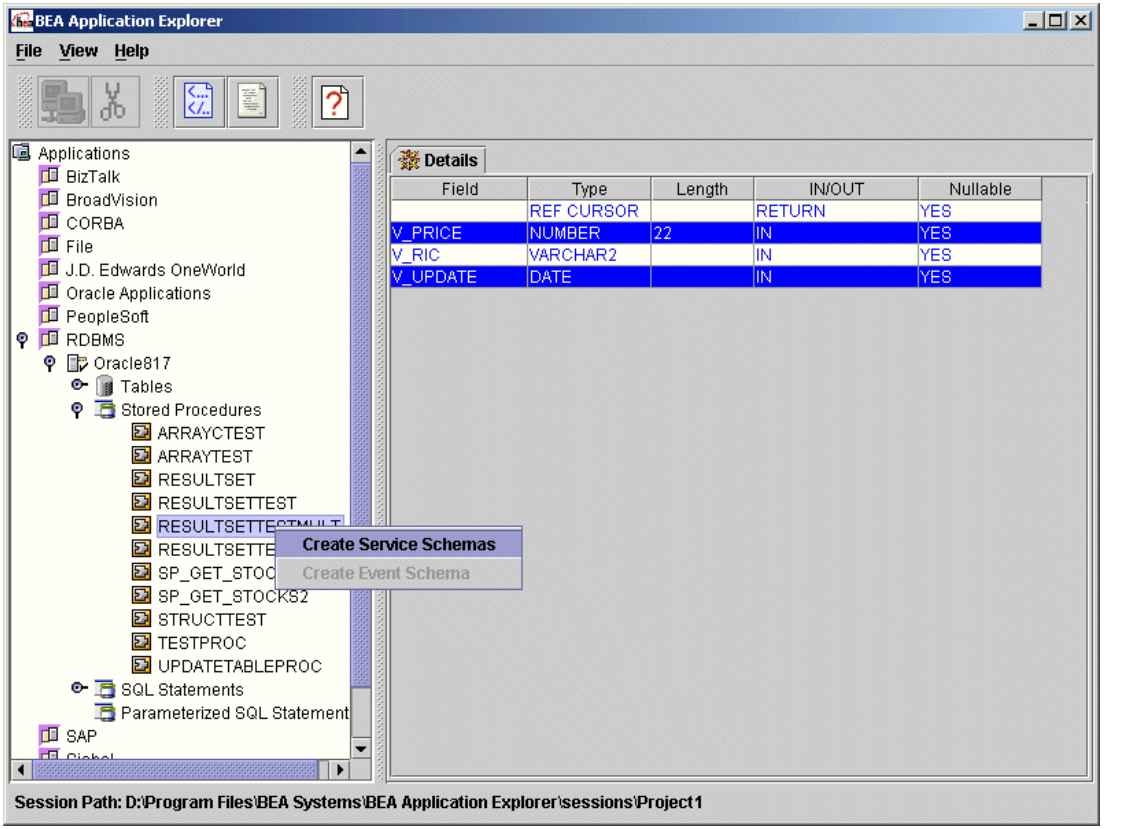
Generating Schemas for Stored Procedures

To create a service schema for a stored procedure for use with the BEA WebLogic Adapter for RDBMS, perform the following steps:

1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.

The BEA Application Explorer opens.

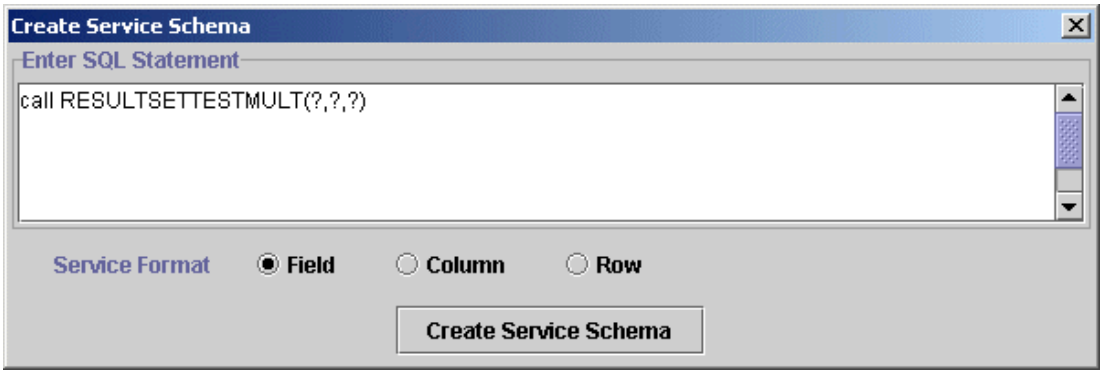
Figure 3-51 BEA Application Explorer - Create Service Schemas



2. Right-click the stored procedure. Note that Create Event Schemas is not available.
3. Select Create Service Schemas.

The Create Service Schema window opens.

Figure 3-52 Create Service Schema Window



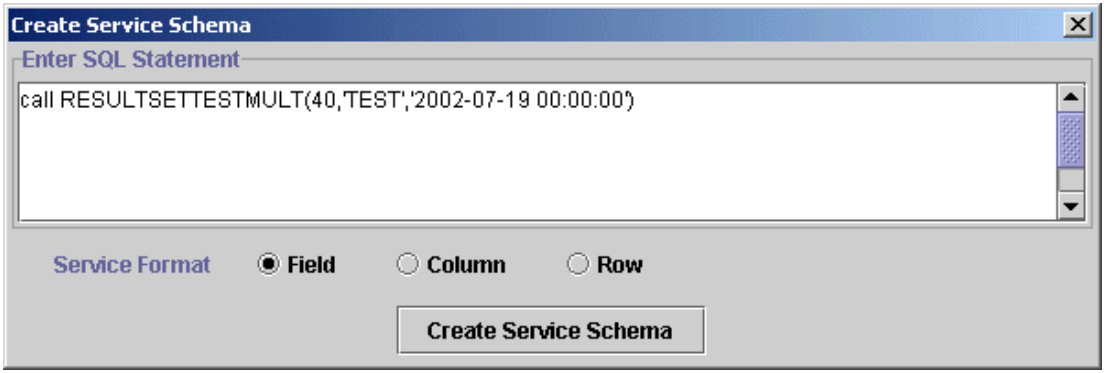
The Create Service Schema window shows all parameter markers denoted by a "?".

a. Replace each ? associated with an IN parameter with the value you want to call the stored procedure.

   Ignore all parameter markers (?'s) that represent OUT parameters.

   For example you can enter 40, 'TEST', '2002-07-19', as shown in the following figure.

Figure 3-53 Create Service Schema - Test SQL Value



b. Select Field, Column, or Row.

   This determines the service response format for this stored procedure.

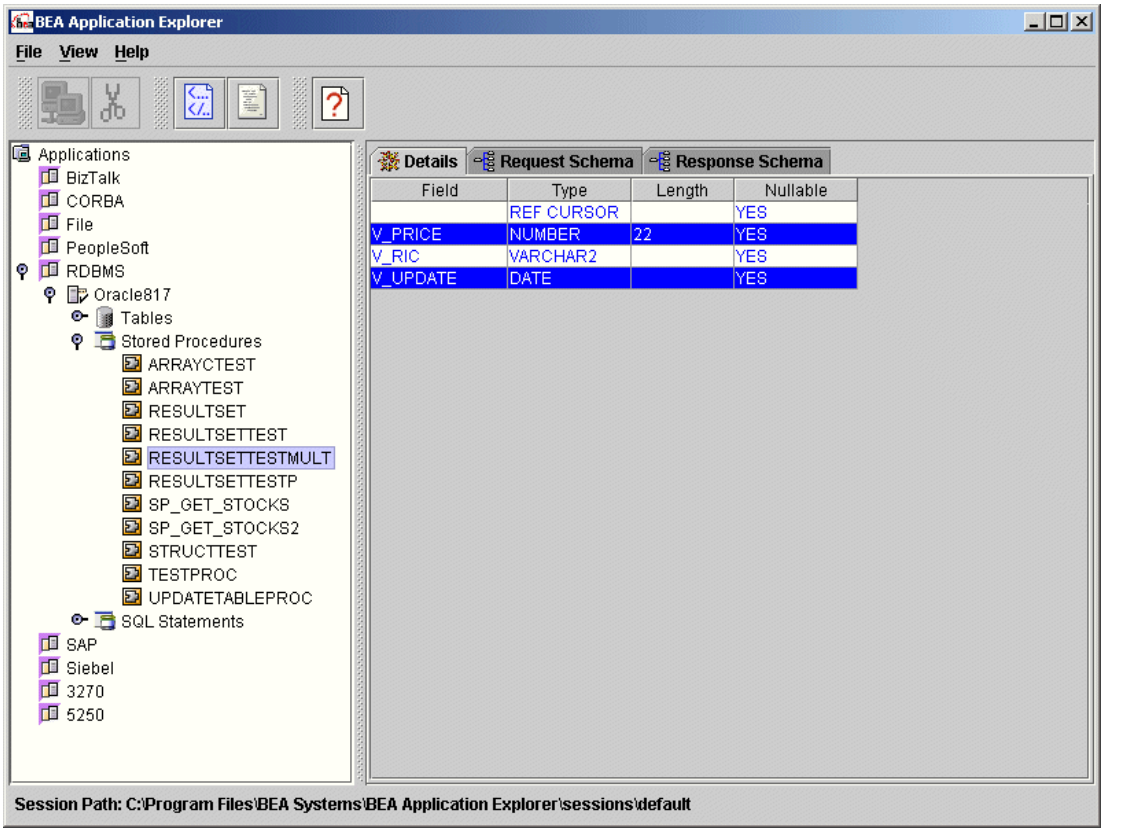
   The value selected must match the format value used to generate the application view service, or the schema of the request document does not validate.

c. To execute the stored procedure, click Create Service Schema.

   The stored procedure is run with the parameters entered.

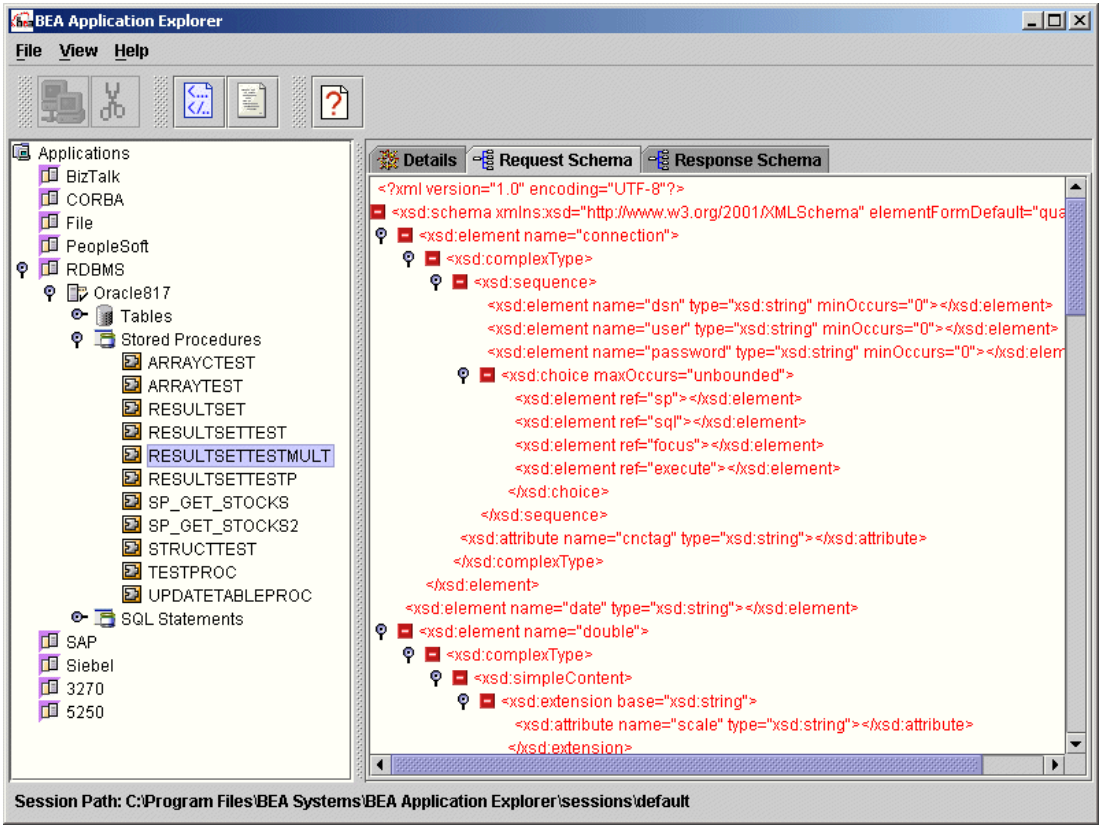
   The right pane of the Application Explorer displays the Details, Request Schema, and Response Schema tabs as shown in the following figure.

Figure 3-54 BEA Application Explorer - Schema Display



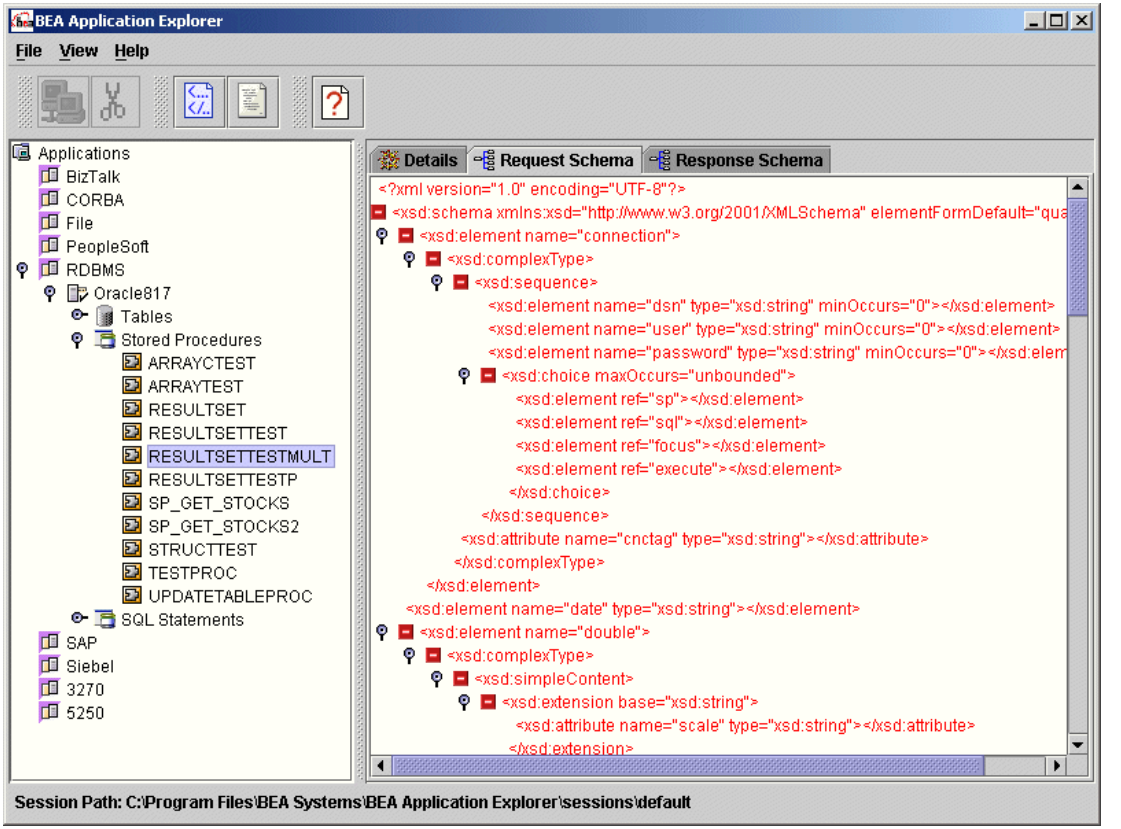
d. Click the Request Schema tab to see the request schema, as shown in the following figure.

Figure 3-55 BEA Application Explorer - Request Schema Tab



e. Click the Response Schema tab to see the response schema, as shown in the following figure.

Figure 3-56 BEA Application Explorer - Response Tab



The schemas are now ready to use.

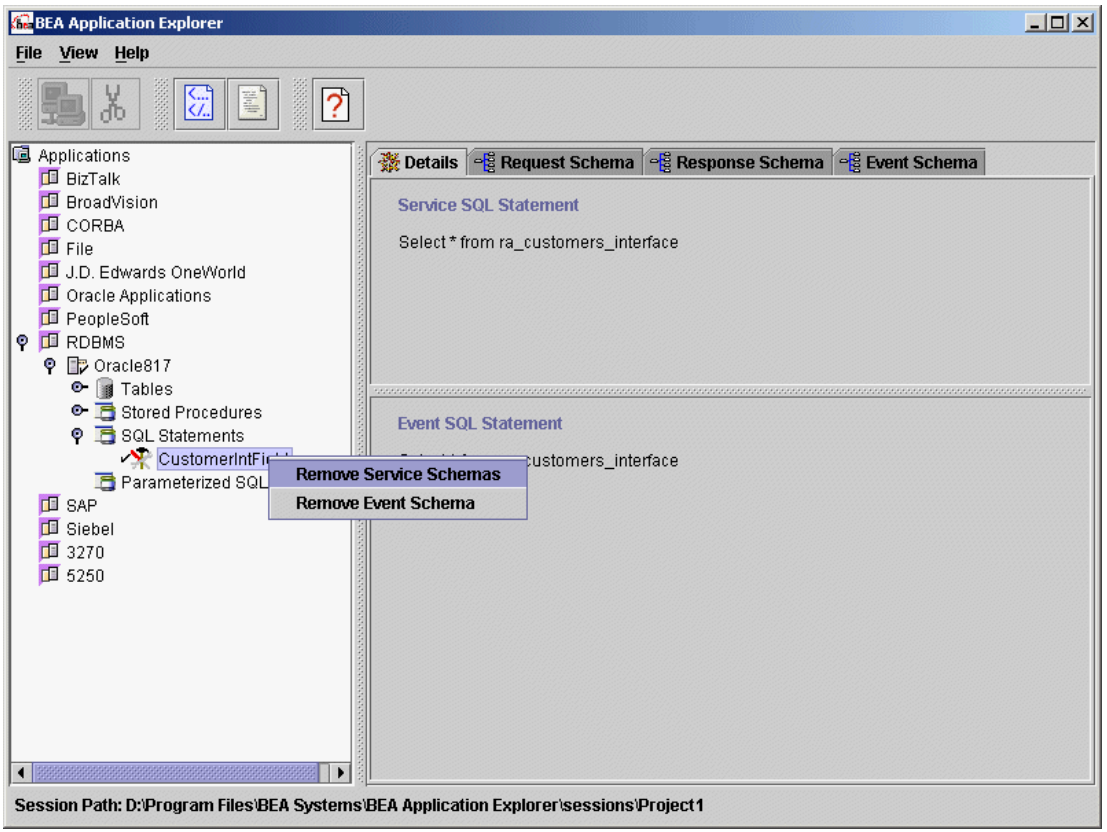
Removing Schemas

To remove a schema from an SQL Statement:

1. Start the BEA Application Explorer by choosing Start→Programs→BEA Application Explorer.

   The BEA Application Explorer opens.

Figure 3-57 BEA Application Explorer - Remove Schema Option



2. Right-click the desired SQL Statement.

3. Select a remove schema option.

   The schema(s) (service or event) is removed, and the `manifest.xml file` is updated.

   The right pane no longer displays the Request and Response Schema tabs for the schema after you select Remove Service Schemas as shown in the following figure.

Figure 3-58 BEA Application Explorer - Schema Removed