

# 2

## Working with Procedure Builder

Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

| Schedule: | Timing     | Topic    |
|-----------|------------|----------|
|           | 30 minutes | Lecture  |
|           | 15 minutes | Practice |
|           | 45 minutes | Total    |

## Objectives

**After completing this lesson, you should be able to do the following:**

- **Describe the features of Procedure Builder**
- **Manage program units using the Object Navigator**
- **Create and compile program units using the Program Unit Editor**
- **Invoke program units using the PL/SQL Interpreter**

### Lesson Aim

In this lesson you learn about the features of the Procedure Builder tool, and how they can be used to create, compile, and invoke subprograms.

## Components of Procedure Builder

| Component                  | Use It to ...                                        |
|----------------------------|------------------------------------------------------|
| Object Navigator           | Manage PL/SQL constructs; perform debug actions      |
| PL/SQL Interpreter         | Debug PL/SQL code; evaluate PL/SQL code in real time |
| Program Unit editor        | Create and edit PL/SQL source code                   |
| Stored Program Unit editor | Create and edit server-side PL/SQL source code.      |
| Database Trigger editor    | Create and edit database triggers                    |

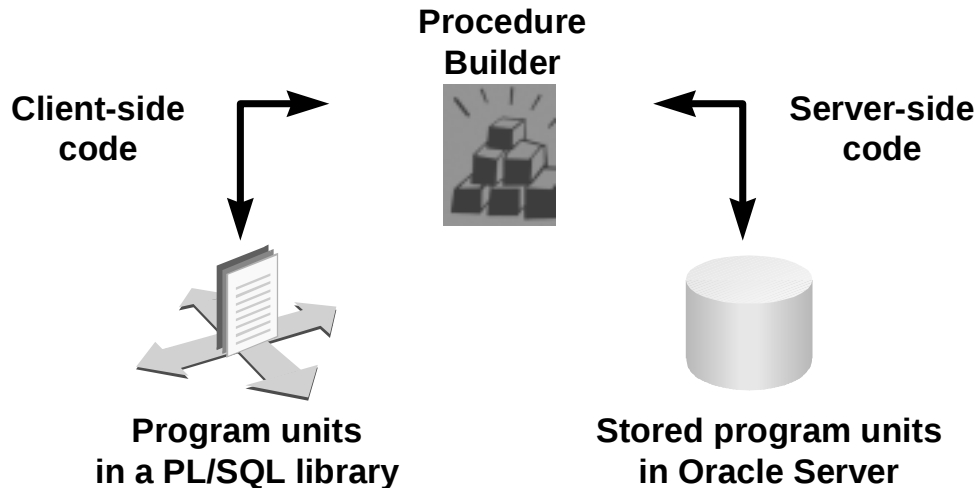
### Components of Procedure Builder

Procedure Builder is an integrated development environment. It enables you to edit, compile, test, and debug client-side and server-side PL/SQL program units within a single tool.

### Instructor Note

This is an overview of the capabilities of Procedure Builder. Do not go into detail about the different components at this stage. This lesson covers the components in detail. The Database Trigger Editor is covered in a later lesson.

# Developing Program Units and Stored Programs Units



2-4

Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

## Program Units and Stored Program Units

Use Procedure Builder to develop PL/SQL subprograms that can be used by client and server applications.

Program units are client-side PL/SQL subprograms that you use with client applications, such as Oracle Developer. Stored program units are server-side PL/SQL subprograms that you use with all applications, client or server.

### Developing PL/SQL Code

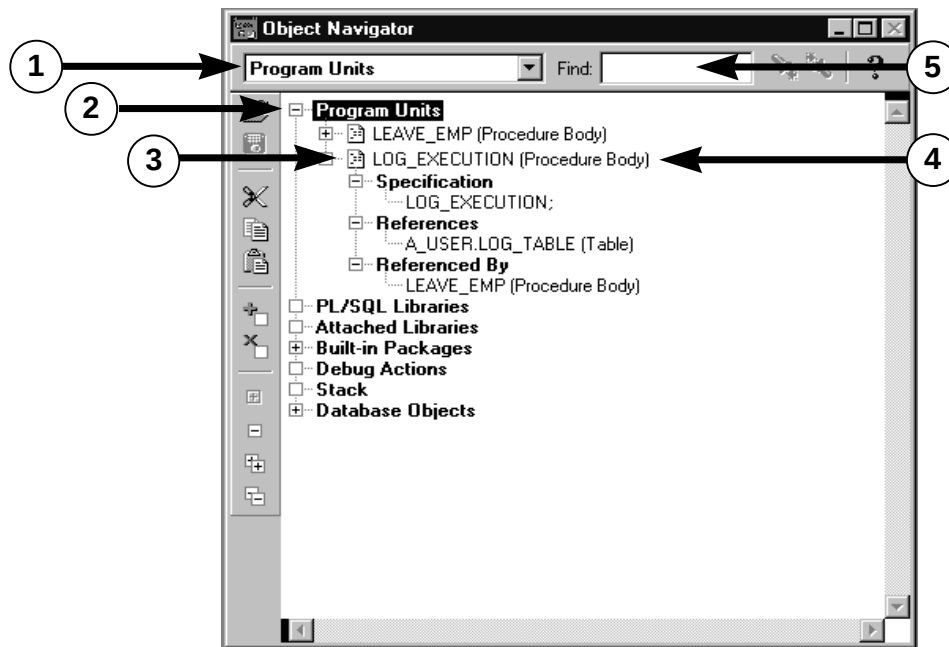
Client-side code:

- Create program units by using the Program Unit Editor
- Drag a server-side subprogram to the client by using the Object Navigator

Server-side code:

- Create stored programs by using the Stored Program Unit Editor
- Drag a client-side program unit to the server by using the Object Navigator

## Procedure Builder Components: The Object Navigator



2-5

Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

### Components of the Object Navigator

The following descriptions correspond to the numbered components on the slide:

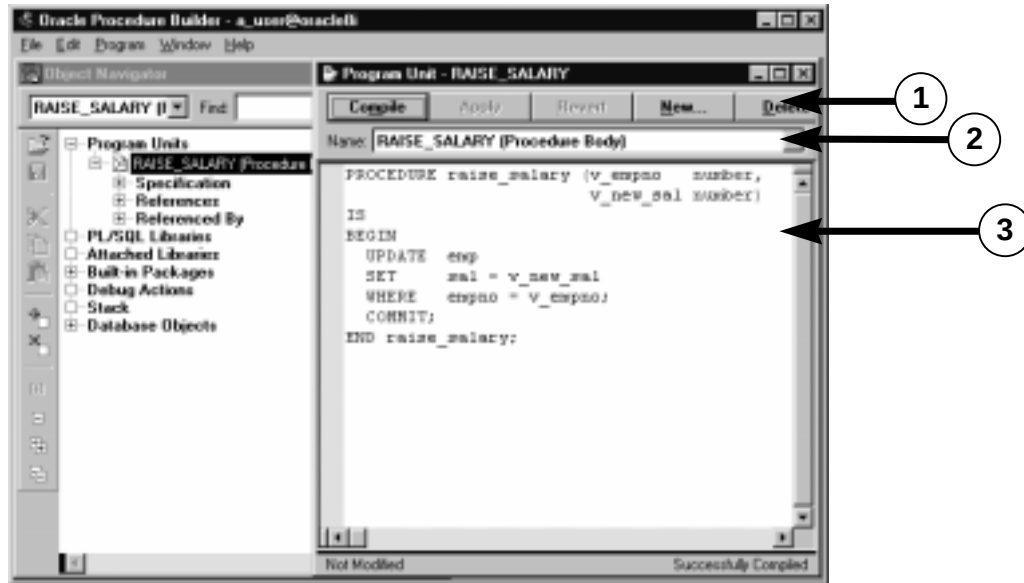
1. Location indicator: Shows your current location in the hierarchy.
2. Subobject indicator: Allows you to expand and collapse nodes to view or hide object information. Different icons represent different classes of objects.
3. Type icon: Indicates the type of object, followed by the name of the object. In the example, the icon indicates that LOG\_EXECUTION is a PL/SQL block. If you double-click the icon, Procedure Builder opens the Program Unit editor, with the code of that object.
4. Object name: Shows you the names of the objects.
5. Find field: Allows you to search for objects.

### Object Navigator

The Object Navigator is Procedure Builder's browser for locating and working with both client and server program units, libraries, and triggers.

The Object Navigator allows you to expand and collapse nodes, cut and paste, search for an object, and drag PL/SQL program units between the client and the server side.

## Procedure Builder Components: The Program Unit Editor



2-6

Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

### Program Unit Editor

The following descriptions correspond to the numbered components on the slide:

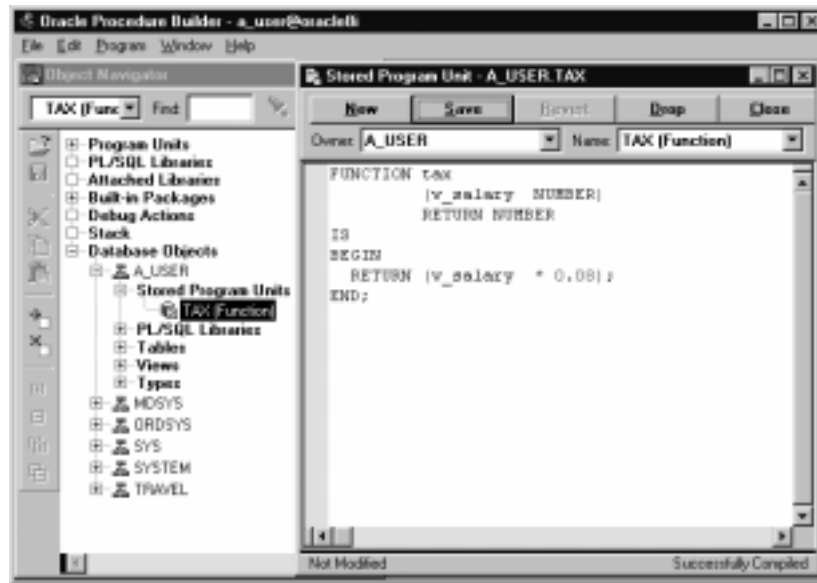
1. Compile, Apply, Revert, New, Delete, Close, and Help buttons
2. Name drop-down list
3. Source text pane

### Program Unit Editor

Use the Program Unit Editor to edit, compile, and browse warning and error messages during development of client-side PL/SQL subprograms.

To bring a subprogram into the source text pane, select an option from the Name field drop-down list. Use the buttons to decide what action to take once you are in the Program Unit Editor.

## Procedure Builder Components: The Stored Program Unit Editor



2-7

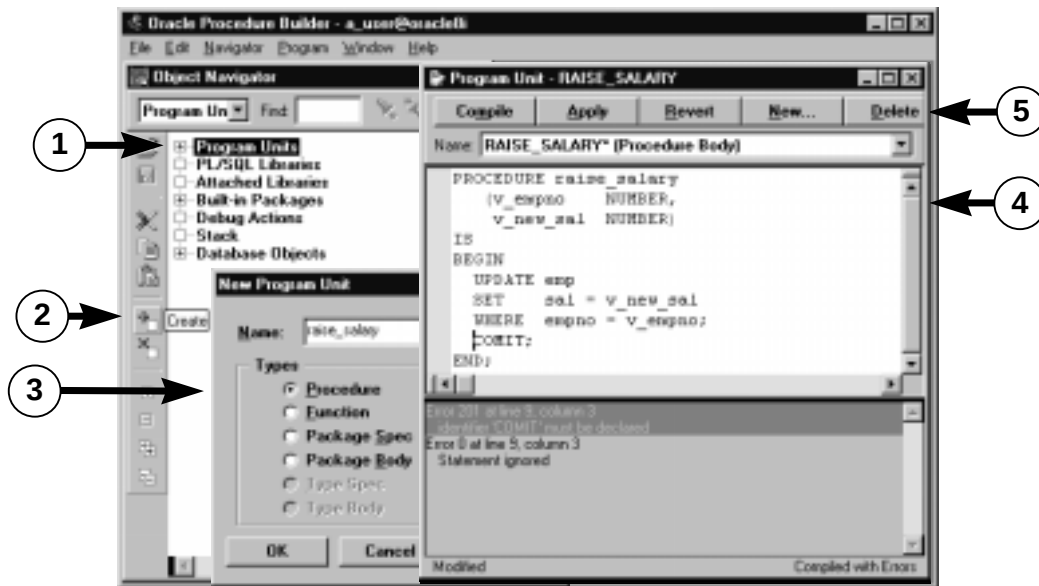
Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

### Stored Program Unit Editor

Use the Stored Program Unit Editor to edit server-side PL/SQL constructs. The Save operation submits the source text to the server-side PL/SQL compiler.

## Creating a Client-Side Program Unit



2-8

Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

### How to Create a Client-Side Program Unit

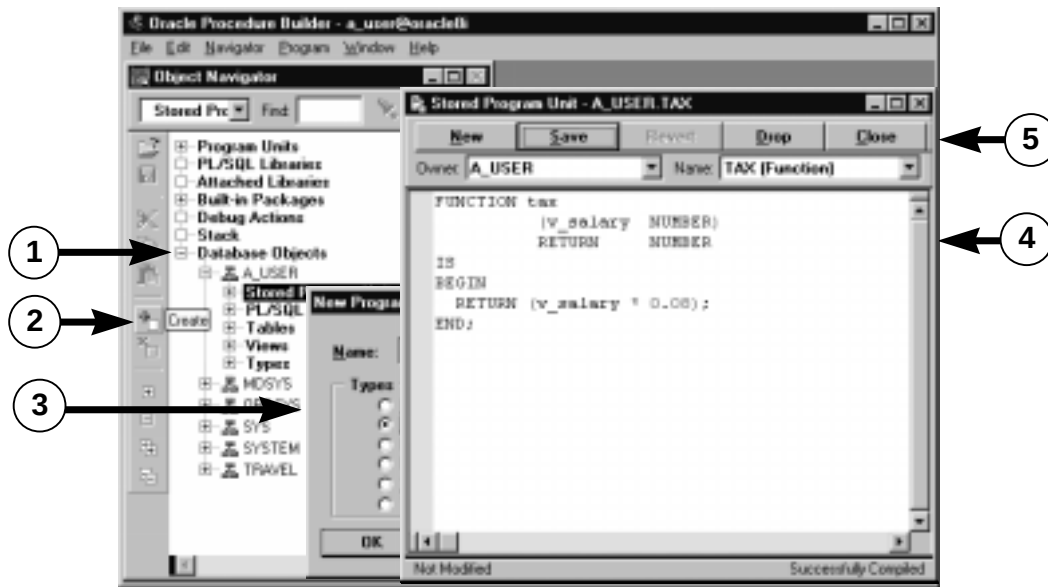
1. Select the Program Units object or subobject.
2. Click Create. The New Program Unit dialog box appears.
3. Enter the name of your subprogram, select the subprogram type, and click OK to accept the entries.
4. The Program Unit editor is displayed. It contains the skeleton for your PL/SQL construct. The cursor is automatically positioned on the line beneath the BEGIN keyword. You can now write the code.
5. When you finish writing the code, click Compile in the Program Unit Editor.

Error messages generated during compilation are displayed in the compilation message pane in the Program Unit window. When you select an error message, the cursor moves to the location of the error in the program screen.

When your PL/SQL code is error free, the compilation message disappears, and the Successfully Compiled message appears in the status line of the Program Unit Editor.

**Note:** Program units that reside in the Program Units node are lost when you exit Procedure Builder. You must export them to a file, save them in a PL/SQL library, or store them in the database.

## Creating a Server-Side Program Unit



2-9

Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

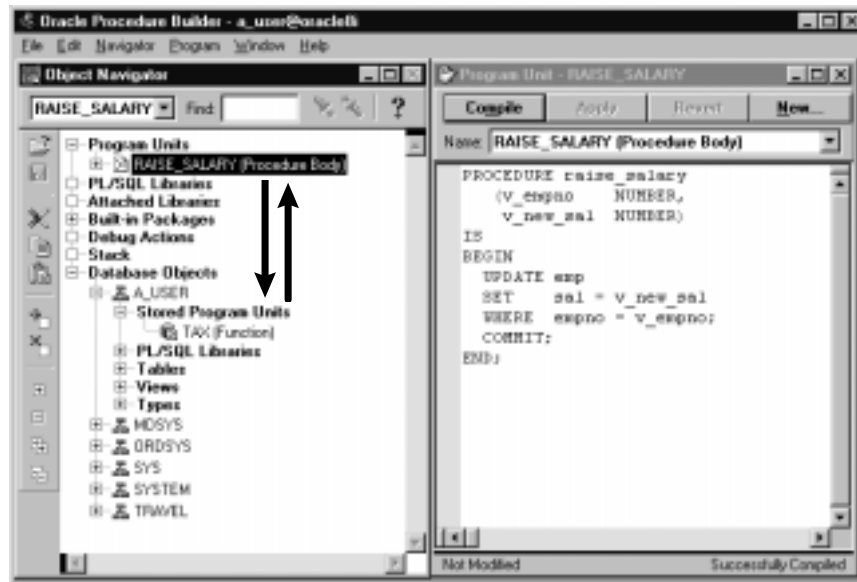
### How to Create a Server-Side Program Unit

1. Select the Database Objects node in the Object Navigator, expand the schema name and click Stored Program Units.
2. Click Create.
3. In the New Program Unit window, enter the name of the subprogram, select the subprogram type, and click OK to accept the entries.
4. The Stored Program Unit editor is displayed. It contains the skeleton for your PL/SQL construct. The cursor is automatically positioned on the line beneath the BEGIN keyword. You can now write the code.
5. When you finish writing the code, click Save in the Stored Program Unit Editor.

Error messages generated during compilation are displayed in a compilation message at the bottom of the window. Click an error message to move to the location of the error.

When the PL/SQL code is error-free, the compilation message does not appear. The Successfully Compiled message appears in the status line at the bottom of the Stored Program Unit Editor window.

## Transferring Program Units Between Client and Server



2-10

Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

### Application Partitioning

Using Procedure Builder you can create PL/SQL program units on both the client and the server. Using Procedure Builder, you can also copy program units created on the client to stored program units on the server (or vice versa). You can do this by dragging the program unit to the destination Stored Program Units node in the appropriate schema.

PL/SQL code that is stored in the server is processed by the server-side PL/SQL engine; therefore any SQL statements contained within the program unit do not have to be transferred between a client application and the server.

Program units on the server are potentially accessible to all applications (subject to user security privileges).

## Procedure Builder Components: The PL/SQL Interpreter



2-11

Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

### Components of the PL/SQL Interpreter

1. Source pane: Displays the PL/SQL code of your program.
2. Navigator pane: Displays the same information as the Object Navigator, but within the PL/SQL Interpreter.
3. Interpreter pane: Allows you to execute subprograms, Procedure Builder commands, and SQL statements.

To execute subprograms, at the PL/SQL prompt, enter the name of your PL/SQL program, provide any parameters, and terminate with a semicolon.

```
PL/SQL> construct_name [parameter1|parameter2,...];
```

To execute SQL statements, enter your SQL statement and terminate with a semicolon.

```
PL/SQL> SELECT      *
      +> FROM        dept;
```

### Instructor Note

Demonstration: Use the menu option View, and then select which of the panes you wish to see at any time. There must always be one pane on display.

## Procedure Builder Built-in Package: TEXT\_IO

- **TEXT\_IO package:**
  - Contains a procedure **PUT\_LINE**, which writes information to the PL/SQL Interpreter window
  - Is used for client-side program units
- **TEXT\_IO.PUT\_LINE:**  
Accepts one parameter

```
PL/SQL> TEXT_IO.PUT_LINE(1);  
1
```

### TEXT\_IO Built-in Package

You can use TEXT\_IO packaged procedures to output values and messages from a client-side procedure or function to the PL/SQL Interpreter window.

TEXT\_IO is a built-in package that is part of Procedure Builder.

Use the Oracle supplied package DBMS\_OUTPUT to debug server-side procedures, and the Procedure Builder built-in, TEXT\_IO to debug client-side procedures.

#### Note

- You cannot use TEXT\_IO to debug server-side procedures. The program will fail to compile successfully because TEXT\_IO is not stored in the database.
- DBMS\_OUTPUT does not display messages in the PL/SQL Interpreter window if you execute a procedure from Procedure Builder.

# Summary

**In this lesson, you should have learned about:**

- **Advantages of using Procedure Builder:**
  - Application partitioning
  - Editors
  - Execution environment
- **Procedure Builder components:**
  - Object Navigator
  - Program Unit editors
  - PL/SQL Interpreter

## Practice Overview

**This practice covers the following topics:**

- **Exploring Procedure Builder menus and windows**
- **Creating a program unit**
- **Executing the program unit**

### **Practice Overview**

In this practice you will create and execute a procedure as a means of becoming familiar with Procedure Builder.

### **Instructor Note**

Practice Timing: 15 minutes

## Practice 2

Connect to the database as directed by your instructor.

1. Load and execute a loop counter.
  - a. Launch Procedure Builder. Your instructor will give you the login information.
  - b. From the menu, load the `p2loop.pls` file.
  - c. Examine the code using the Program Unit Editor.
  - d. Compile the procedure. Correct any compilation errors.
  - e. Execute the procedure from the interpreter pane. Pass a numeric value into the procedure as demonstrated below.

```
PL/SQL> my_loop(4);
```

2. Create a client-side procedure called `MY_MESSAGE` to print the message you provide at design time. The executable part in your procedure should contain the following line of code:

```
text_io.put_line('Hello world');
```

Compile and execute your procedure from the interpreter pane as shown below.

```
PL/SQL> my_message;
```

3. Add a new department to your `DEPT` table using the interpreter pane.

```
PL/SQL> INSERT INTO dept VALUES (50, 'MIS', 'HOUSTON');
```

Verify the insert.

4. Insert a new department using a procedure.
  - a. From the menu, load the `p2ins.pls` file.
  - b. Replace the comment inside the procedure with your `INSERT` statement. Remember to use a different department number than the one above.
  - c. Execute the procedure from the interpreter pane.
  - d. Verify the insert.

