

## Getting Started with Oracle Data Provider for .NET (VB.NET Version)

### Purpose

This tutorial describes how you can build a .NET application that uses the Oracle Data Provider for .NET to access data in an Oracle Database.

### Time to Complete

Approximately 30 minutes

### Overview

In addition to basic Oracle client connectivity software, .NET applications require the use of what is known as a *managed data provider* (where "managed" refers to code managed by the .NET framework). The data provider is the layer between the .NET application code and the Oracle client connectivity software.

The Oracle Data Provider for .NET (ODP.NET) is Oracle's high performance ADO.NET 2.0 compliant data provider that exposes a complete set of Oracle specific features and tuning options including support for Real Application Clusters, XML DB, and advanced security. It is available for free download from the Oracle Technology Network website.

When ODP.NET and any required Oracle client connectivity software is installed, application development using Visual Studio can begin. It is a good idea to confirm client connectivity before starting development. If you can connect to Oracle using SQL\*Plus on the same machine as Visual Studio, then you know that your Oracle client-side software is properly installed and configured.

### Prerequisites

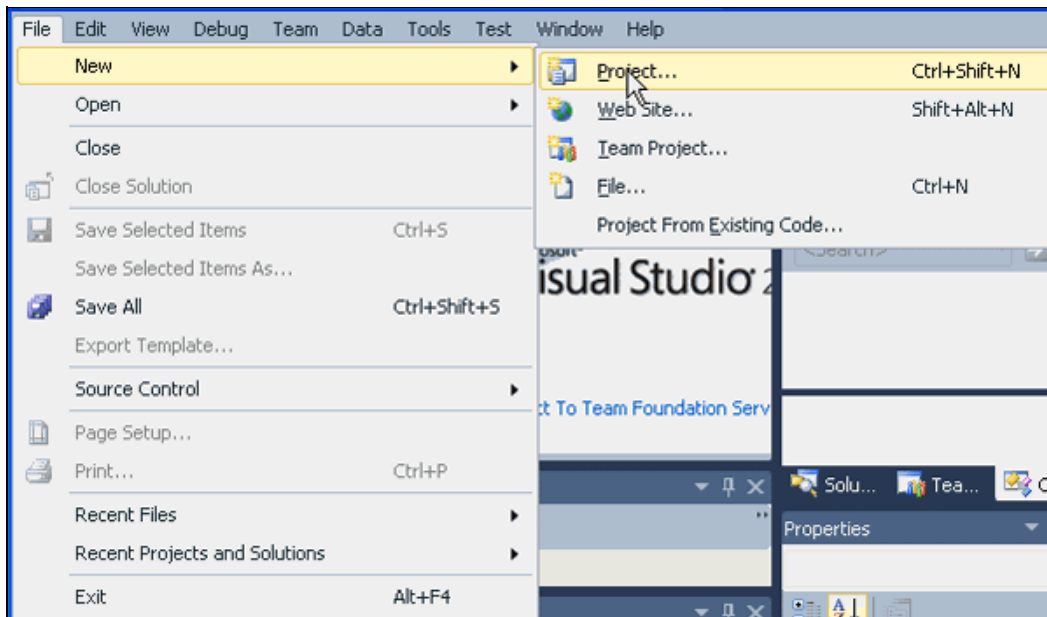
Before starting this tutorial, you should:

1. Install Microsoft Visual Studio 2010
2. Install Oracle Database 9.2 or later or Oracle Database XE
3. Install Oracle 11g Oracle Data Access Components (ODAC) with Oracle Developer Tools for Visual Studio version 11.2.0.1.2 or later from OTN

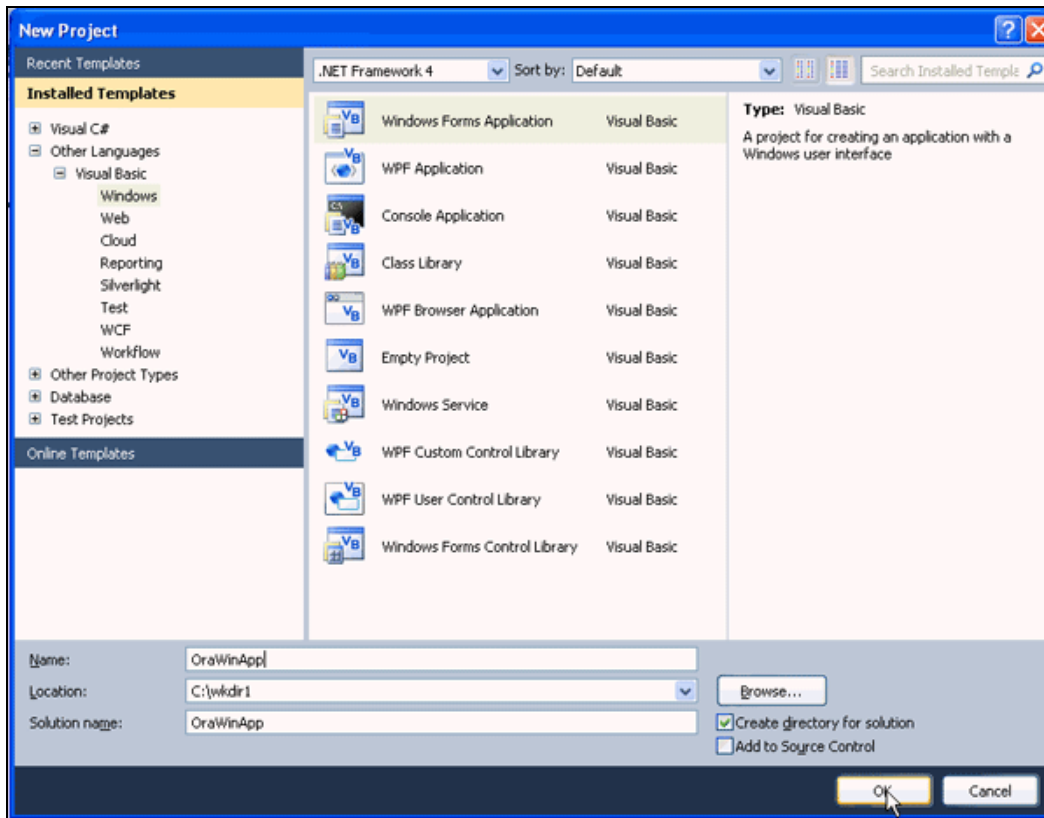
### Creating a Project in Visual Studio

To create a new project in Visual Studio, perform the following steps:

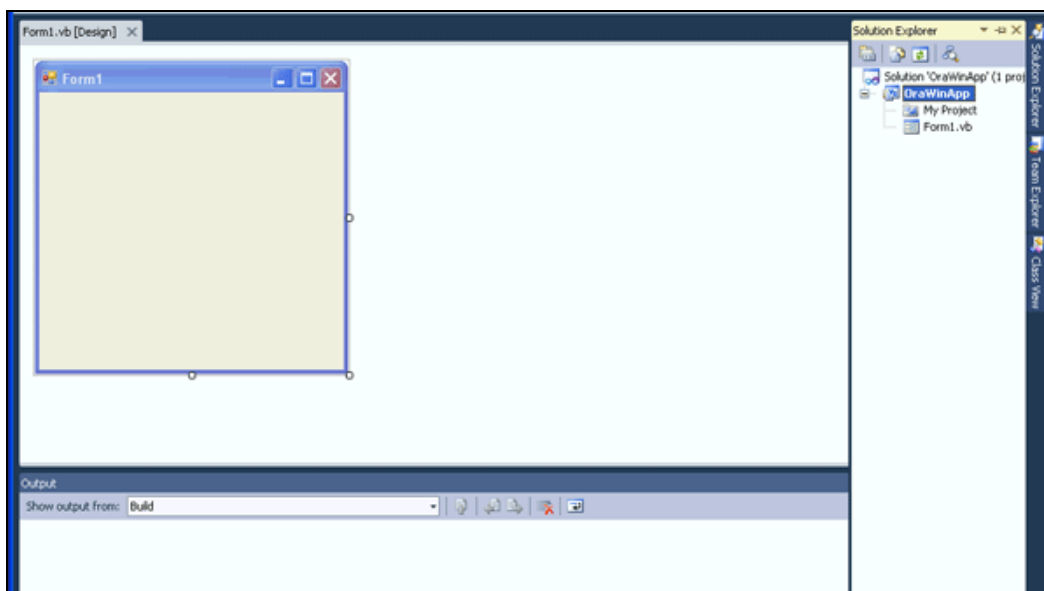
1. From your Windows Program menu, start Visual Studio. From the File menu, select **New Project**.



2. From the Project types list, select **Visual Basic: Windows**. Select the Template **Windows Form Application**. In the Name field, enter **OraWinApp**. In the Location field, enter the directory where you want to save the files. (Note: if the directory does not exist, it is created). Click **OK**



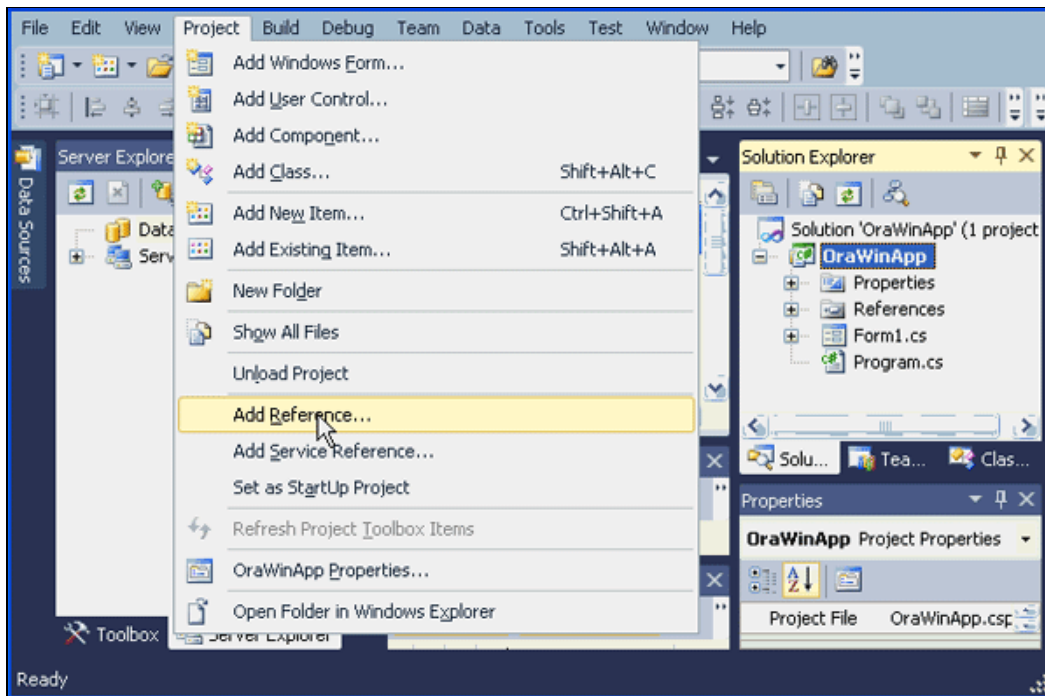
3. The project has been created. From View menu, select **Solution Explorer** to open the Solution Explorer if not already open.



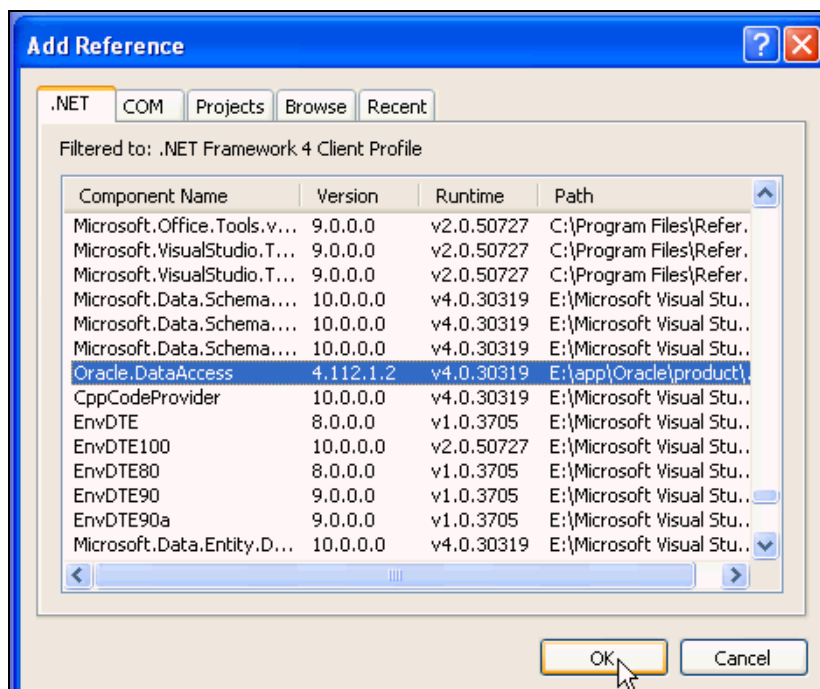
## Adding a Reference

Because your project needs access to an Oracle database, it is necessary to add a reference to the dll containing the data provider. Perform the following steps:

1. From Project menu, select **Add Reference...**



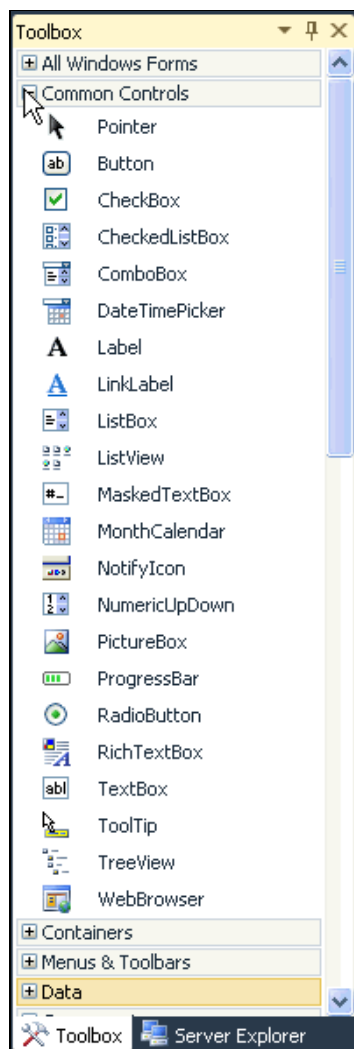
2. Scroll down the list of Component Names and select **Oracle.DataAccess**. Click **OK**.



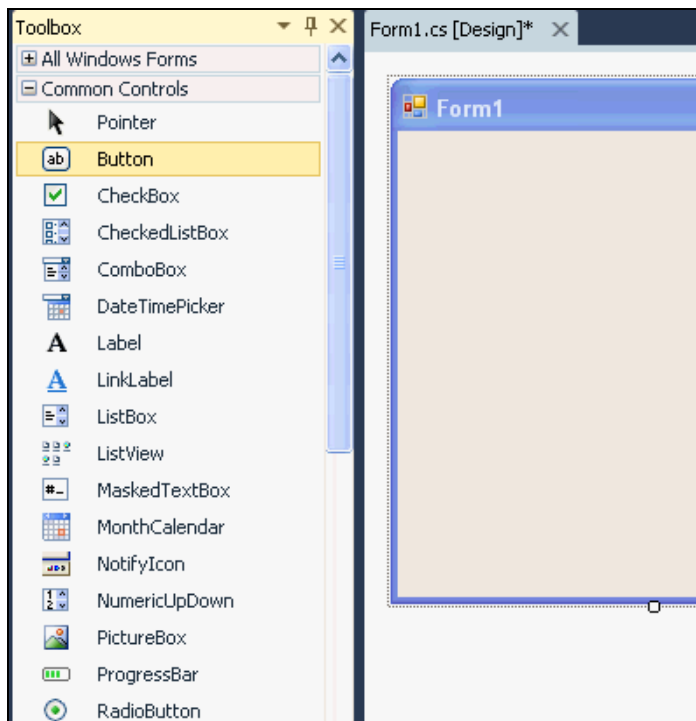
## Retrieving Data from the Database

To retrieve data from the database, you can add some controls to your form. Perform the following steps:

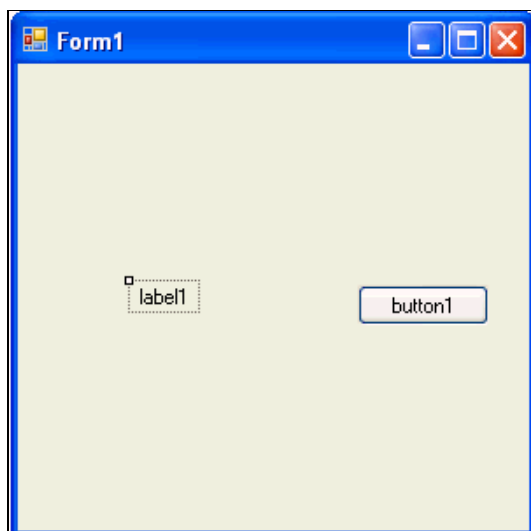
1. From **View** menu, select **Toolbox**. The toolbox appears. Expand **Common Controls**.



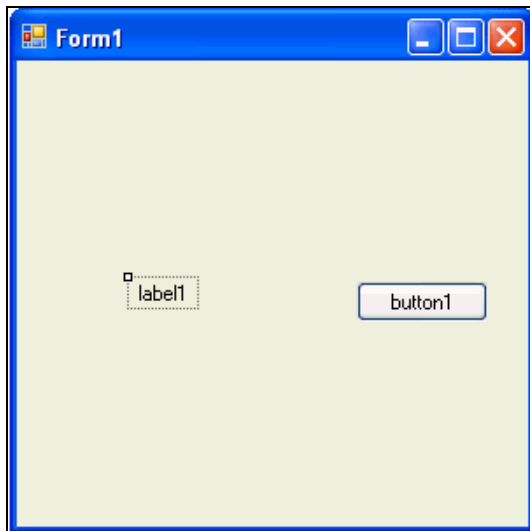
- 2 . Drag the **Button** control to the Form1 canvas.



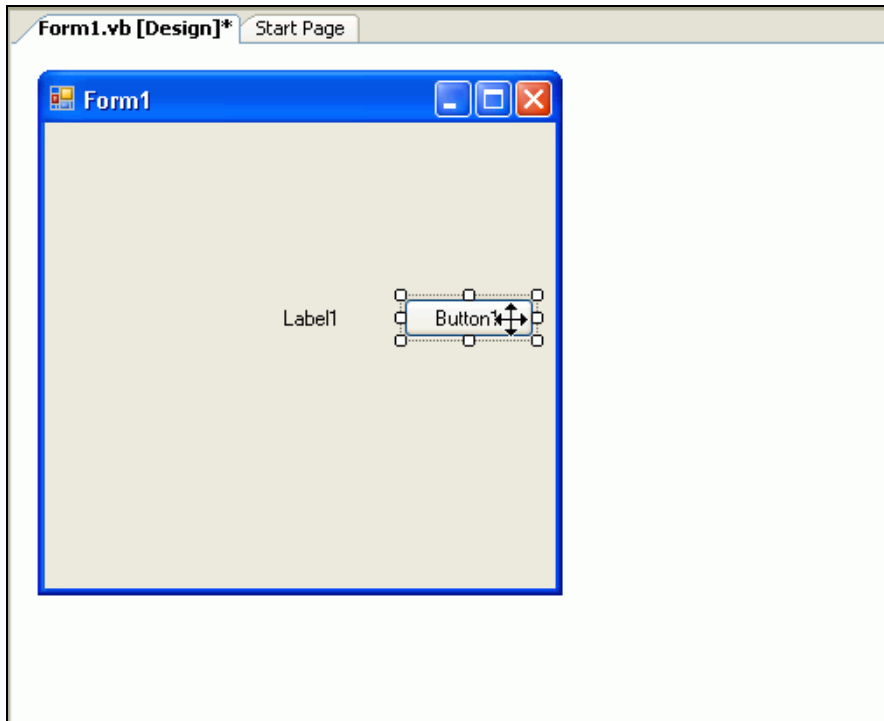
The control is shown on the canvas. You also need to create a Label.



- 3 . From the Toolbox, drag the **Label** control to the Form1 canvas.

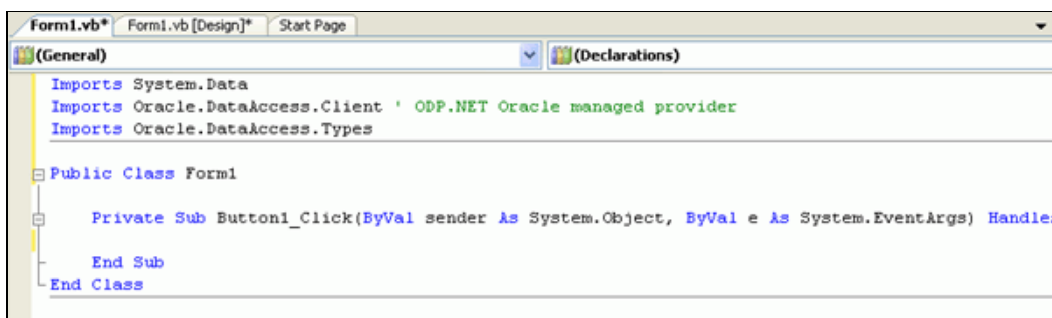


4. Double-click **Button1** to open the code window. Note, Form1.vb code window opens.



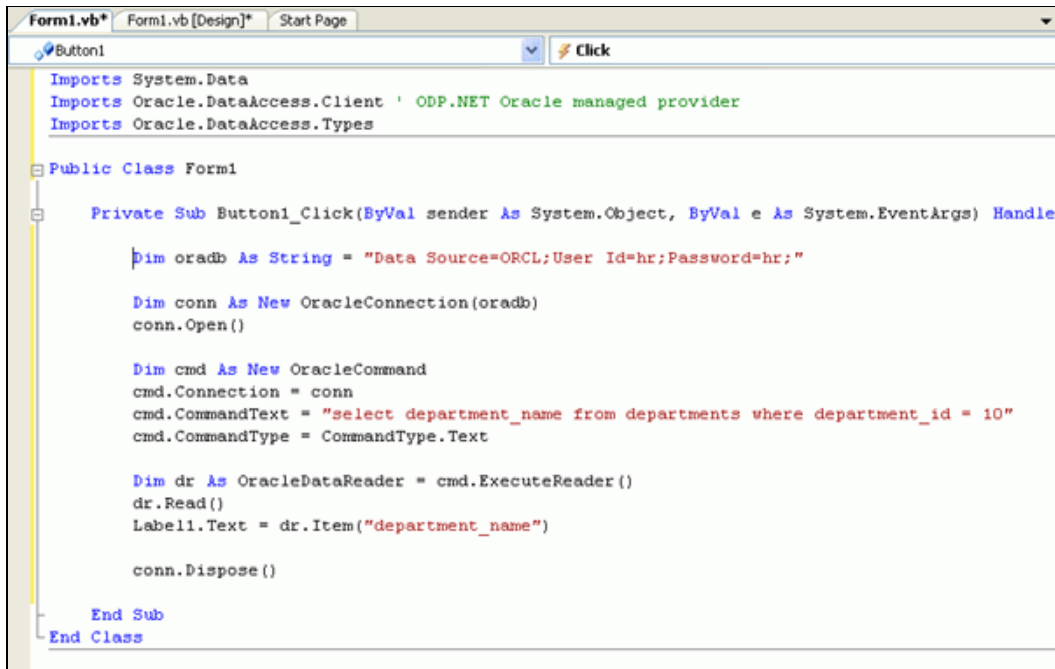
5. Add the following VB.NET Imports statements before the Public Class declaration.

```
Imports System.Data
Imports Oracle.DataAccess.Client ' ODP.NET Oracle managed provider
Imports Oracle.DataAccess.Types
```

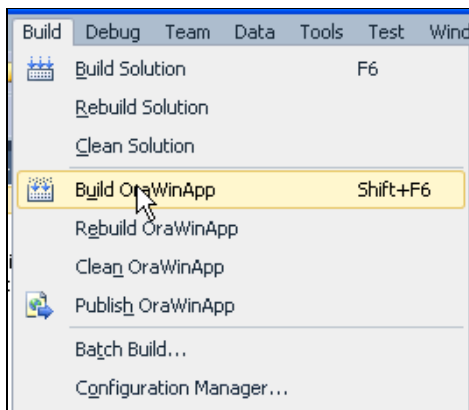


6. Add the following VB.NET code between the Private Sub and End Sub statements.

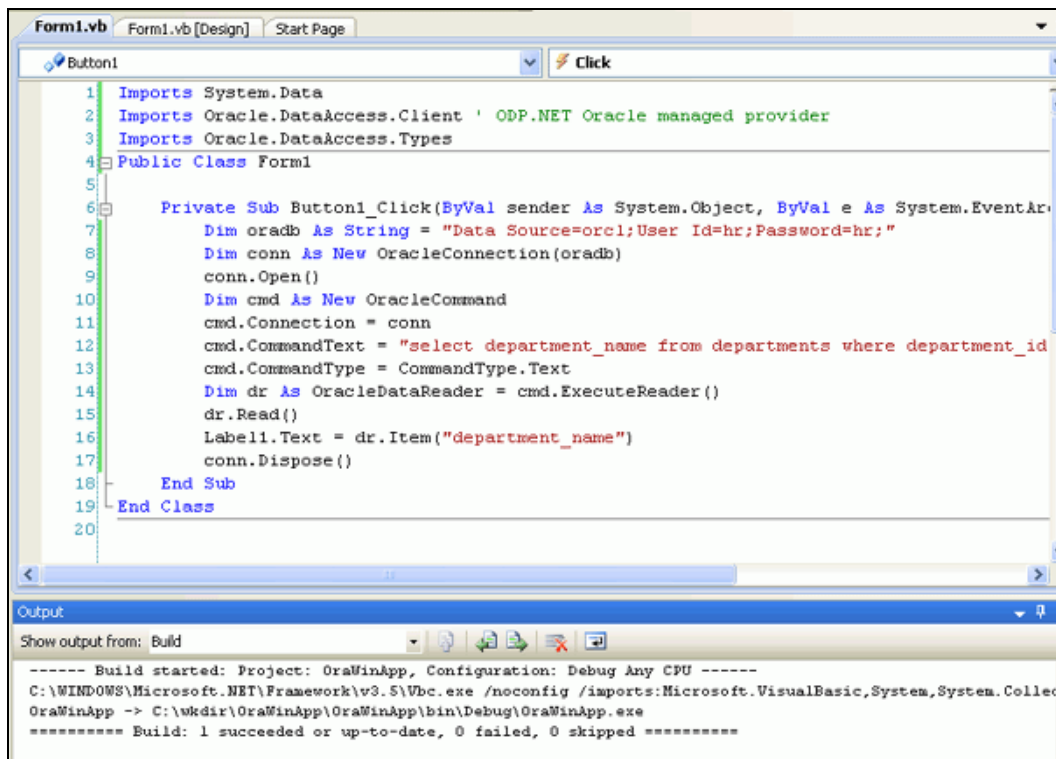
```
Dim oradb As String = "Data Source=orcl;User Id=hr;Password=hr;"
Dim conn As New OracleConnection(oradb)
conn.Open()
Dim cmd As New OracleCommand
cmd.Connection = conn
cmd.CommandText = "select department_name from departments where department_id = 10"
cmd.CommandType = CommandType.Text
Dim dr As OracleDataReader = cmd.ExecuteReader()
dr.Read()
Label1.Text = dr.Item("department_name")
conn.Dispose()
```



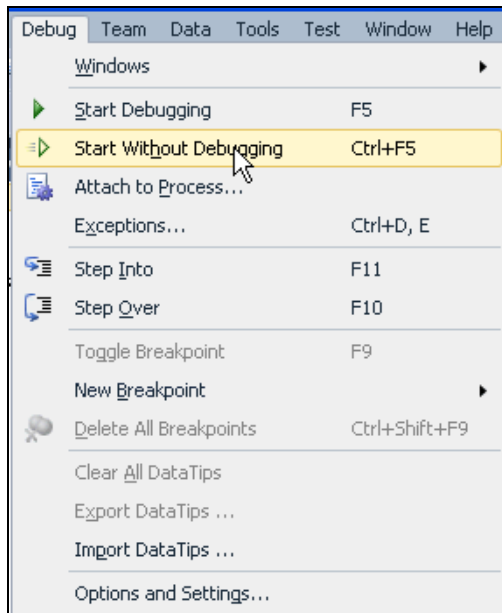
7. Select **Build > Build OraWinApp**.



In the output window, ensure you did not get any errors. Now you can run the form.

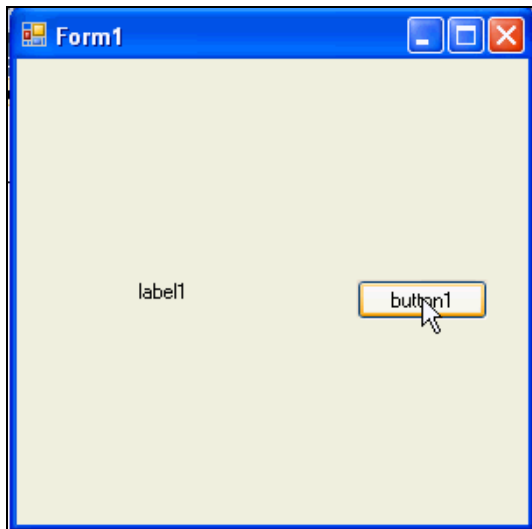


#### 8 . Select **Debug > Start Without Debugging**.



#### 9 . The form appears. Click **button1**.





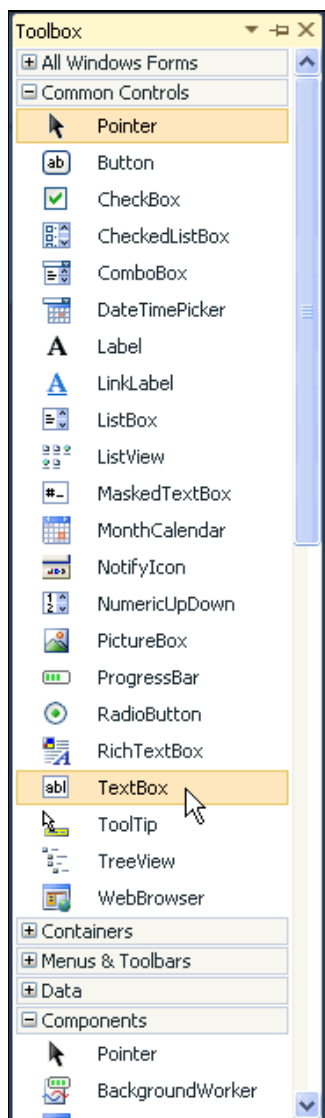
The result is shown in the label. Since the query in the code was to show the department name for department id of 10, the result is the Administration department. Close Form1.



### *Incorporating Interactive Controls*

Instead of running the hard coded query, you can add a textbox control to accept user input for the department id. Perform the following steps:

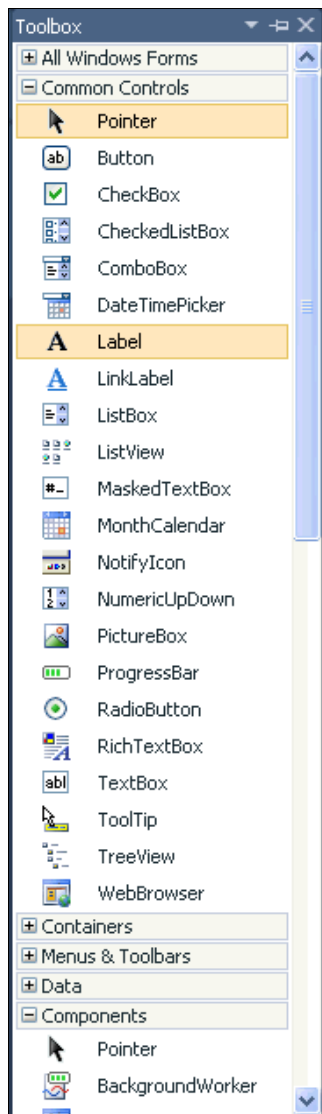
1. Select the **Form1.vb [Design]** tab. From the Toolbox, drag the **TextBox** control to Form1 canvas.



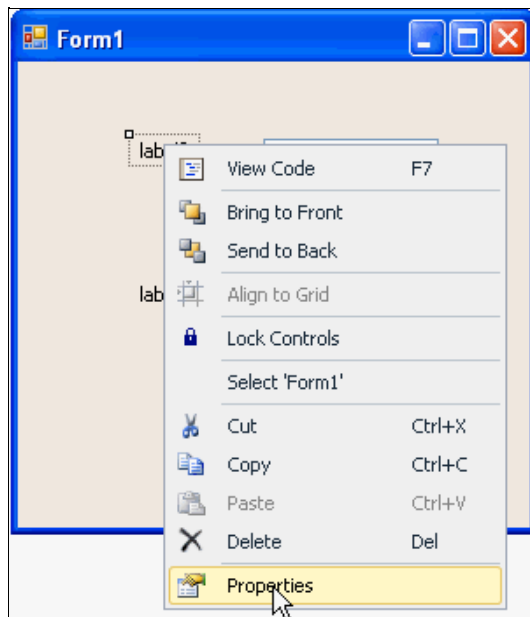
The textbox is shown. You can drag it to the desired location.



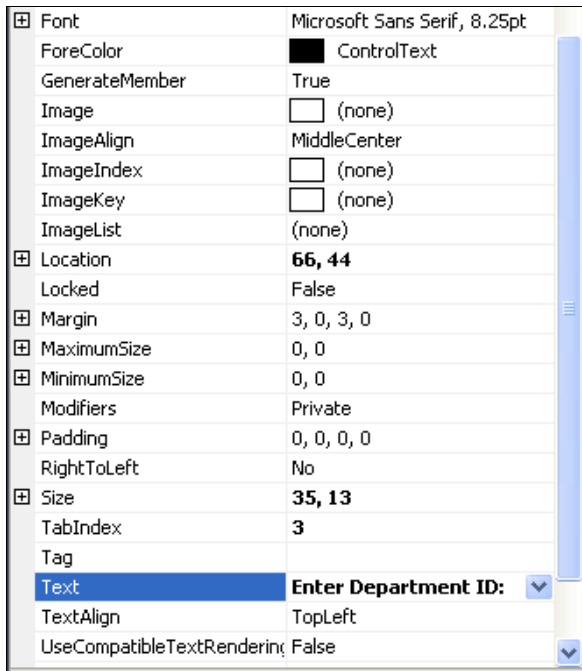
2. Drag a **Label** control on Form1 canvas.



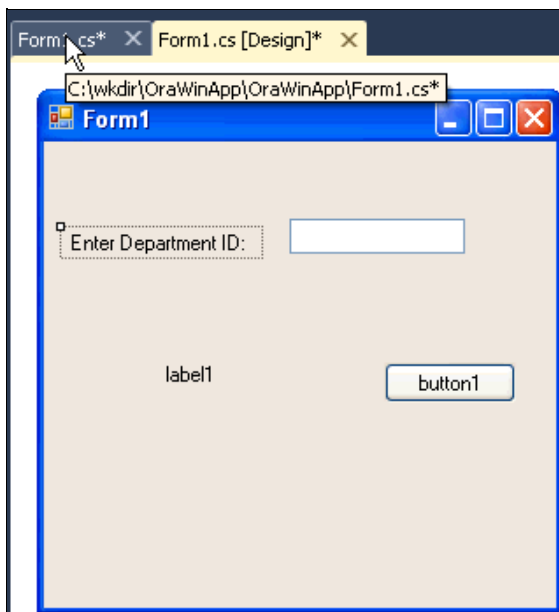
- 3 . The Label is shown. Right-click **Label2** and select **Properties**.



4. In the Properties sheet, change the Text property to **Enter Department ID:**



5. Select the **Form1.vb** tab.



6. You use a bind variable to include the value of the text box as part of the SELECT statement. Bind variables improve performance by preventing the database from having to reparse the SQL statement every time the text box value changes.

Add the following code right after the `conn.Open()` statement:

```
Dim parm As New OracleParameter
parm.OracleDbType = OracleDbType.Decimal
parm.Value = TextBox1.Text
```

```

Dim oradb As String = "Data Source=ORCL;User Id=hr;Password=hr;"

Dim conn As New OracleConnection(oradb)
conn.Open()

Dim parm As New OracleParameter
parm.OracleDbType = OracleDbType.Decimal
parm.Value = TextBox1.Text

Dim cmd As New OracleCommand
cmd.Connection = conn
cmd.CommandText = "select department_name from departments where department_id = 10"
cmd.CommandType = CommandType.Text

Dim dr As OracleDataReader = cmd.ExecuteReader()
dr.Read()
Label1.Text = dr.Item("department_name")

```

7. Add the cmd.Parameters statement right after the cmd.Connection statement:

```
cmd.Parameters.Add(parm)
```

```

Dim oradb As String = "Data Source=ORCL;User Id=hr;Password=hr;"

Dim conn As New OracleConnection(oradb)
conn.Open()

Dim parm As New OracleParameter
parm.OracleDbType = OracleDbType.Decimal
parm.Value = TextBox1.Text

Dim cmd As New OracleCommand
cmd.Connection = conn
cmd.Parameters.Add(parm)
cmd.CommandText = "select department_name from departments where department_id = 10"
cmd.CommandType = CommandType.Text

Dim dr As OracleDataReader = cmd.ExecuteReader()
dr.Read()

```

8. Change the code for the cmd.CommandText statement to the following:

```
cmd.CommandText = "select department_name from departments where department_id = :1"
```

```

Dim conn As New OracleConnection(oradb)
conn.Open()

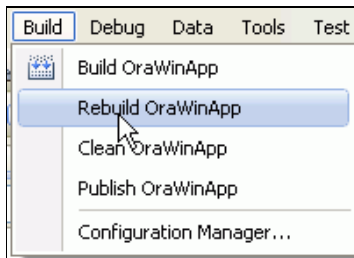
Dim parm As New OracleParameter
parm.OracleDbType = OracleDbType.Decimal
parm.Value = TextBox1.Text

Dim cmd As New OracleCommand
cmd.Connection = conn
cmd.Parameters.Add(parm)
cmd.CommandText = "select department_name from departments where department_id = :1"
cmd.CommandType = CommandType.Text

Dim dr As OracleDataReader = cmd.ExecuteReader()
dr.Read()
Label1.Text = dr.Item("department_name")

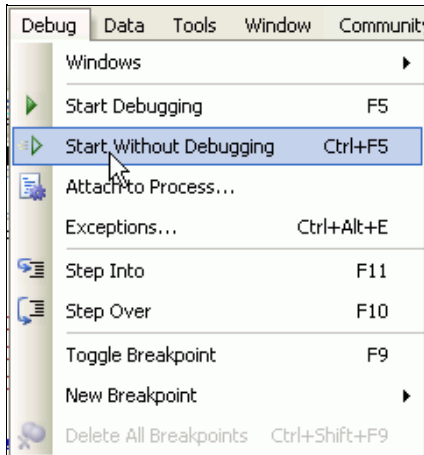
```

9. Select **Build > Rebuild OraWinApp**.

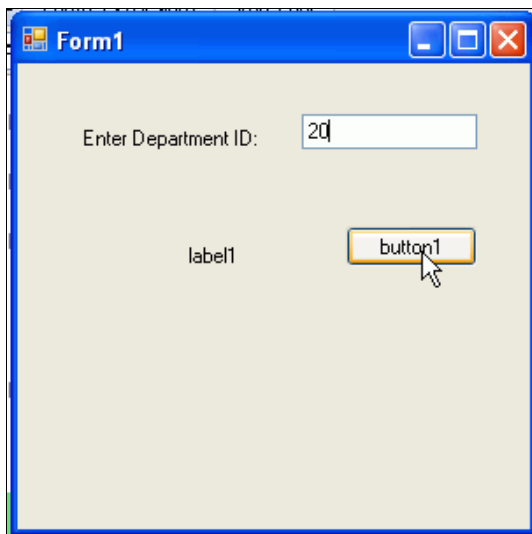


Check in the output window for any errors reported. If no errors or warnings, proceed with the next step.

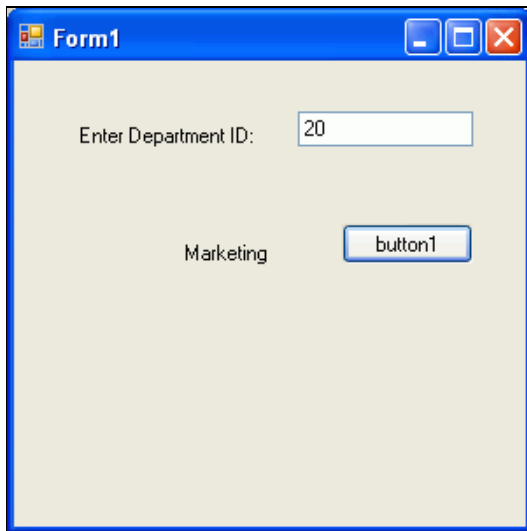
**10 . Select **Debug > Start Without Debugging**.**



**11 . Enter **20** for the Department ID and click **button1**.**



**12 . Notice that the Label1 changes to the name of the department. This value depends on what is entered in the Department ID field.**

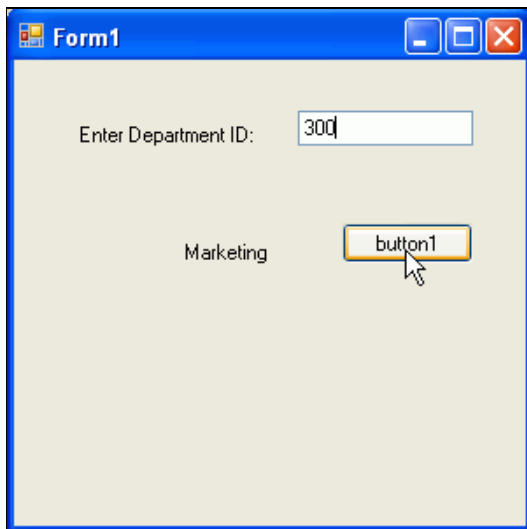


Form1

Enter Department ID: 20

Marketing button1

13 . Enter **300** for the Department ID and click **button1**.

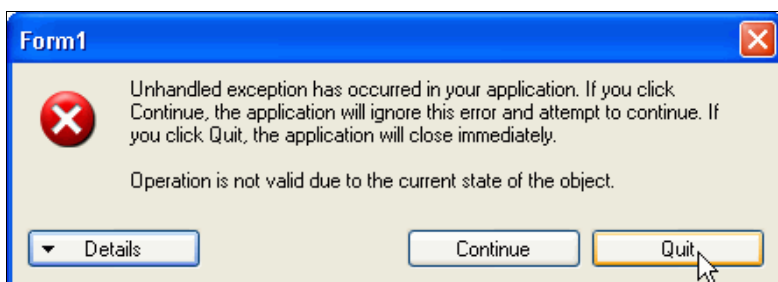


Form1

Enter Department ID: 300

Marketing button1

14 . This time you receive an error because there is no department with the id of 300. Click **Quit**. In the next topic you add some error handling so you know exactly why you received the error.



Form1

Unhandled exception has occurred in your application. If you click Continue, the application will ignore this error and attempt to continue. If you click Quit, the application will close immediately.

Operation is not valid due to the current state of the object.

Details Continue Quit

## Adding Error Handling

Error handling is needed to instruct the user why something doesn't work properly. In this topic, you modify your code to prevent an error when an invalid Department ID is provided. In this tutorial, you add some very simple error handling code. For more robust error handling, Structured Exception Handling should be used. For more information on Structured Exception Handling with ODP.NET please refer to the [ODP.NET online documentation](http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/dotnet/GettingStartedVBVersion/GettingStartedNET_VBVersion.htm). Perform the following steps:

- 1 . In Form1.vb window, change the following code from:



```

dr.Read()
label1.Text = dr.Item("department_name");

to

If dr.Read() Then
    Label1.Text = dr.Item("department_name")
Else
    Label1.Text = "Department ID not found"
End If

```

```

Dim conn As New OracleConnection(oracle)
conn.Open()

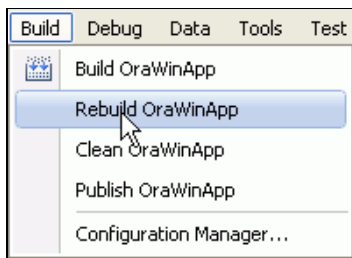
Dim parm As New OracleParameter
parm.OracleDbType = OracleDbType.Decimal
parm.Value = TextBox1.Text

Dim cmd As New OracleCommand
cmd.Connection = conn
cmd.Parameters.Add(parm)
cmd.CommandText = "select department_name from departments where department_id = :1"
cmd.CommandType = CommandType.Text

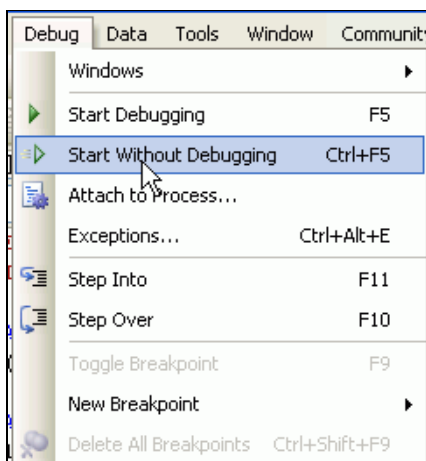
Dim dr As OracleDataReader = cmd.ExecuteReader()
If dr.Read() Then
    Label1.Text = dr.Item("department_name")
Else
    Label1.Text = "Department ID not found"
End If

```

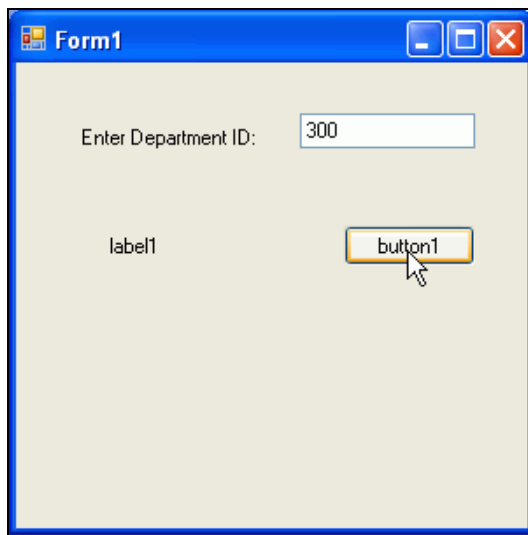
2. Select **Build > Rebuild OraWinApp**. Ensure that there are no errors reported in the output window.



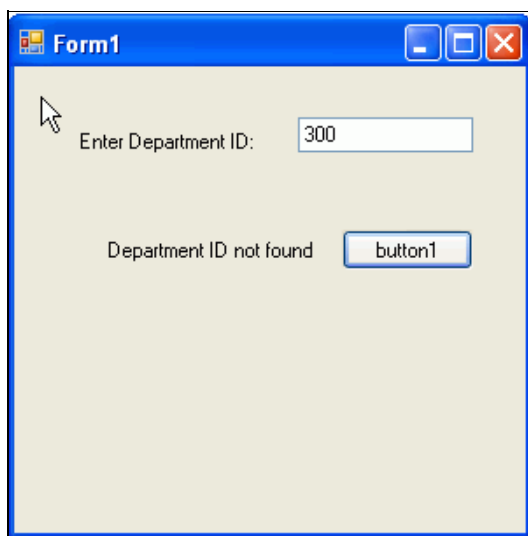
3. Select **Debug > Start Without Debugging**.



4. Enter **300** again for the Department ID and click **button1**.



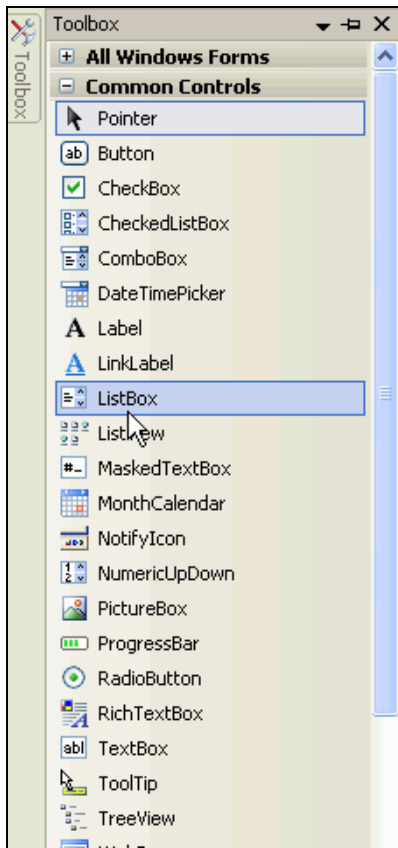
- 5 . This time, the message **Department ID not found** is displayed. Close Form1.



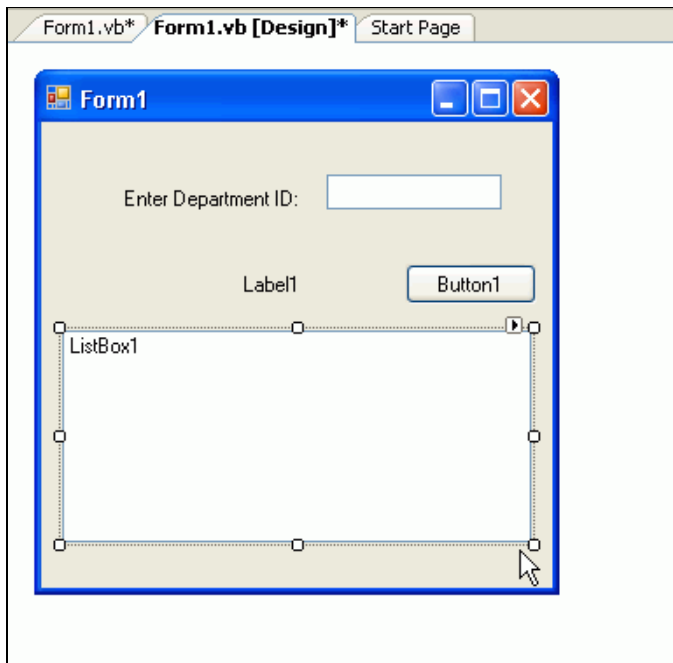
### ***Retrieving Multiple Columns and Rows***

In this topic, you want to retrieve more than one set of information using a listbox. Perform the following steps:

- 1 . Click the **Form1.vb [Design]** tab. From the Toolbox, drag the **Listbox** control to the Form 1 canvas.



2. The new listbox appears on the form. You can position and re-size the listbox appropriately. Click **Form1.vb** tab.



3. Delete the `cmd.Parameters.Add(parm);` statement.

```

Dim conn As New OracleConnection(oradb)
conn.Open()

Dim parm As New OracleParameter
parm.OracleDbType = OracleDbType.Decimal
parm.Value = TextBox1.Text

Dim cmd As New OracleCommand
cmd.Connection = conn
cmd.CommandText = "select department_name from departments where department_id = :1"
cmd.CommandType = CommandType.Text

Dim dr As OracleDataReader = cmd.ExecuteReader()
If dr.Read() Then
    Label1.Text = dr.Item("department_name")
Else
    Label1.Text = "Department ID not found"
End If

```

4. Change the cmd.CommandText code from:

```
cmd.CommandText = "select department_name from departments where department_id = :1";
```

to

```

cmd.CommandText = "select department_id, department_name, city"
+ " from departments d, locations l"
+ " where d.location_id = l.location_id";

```

```

Dim conn As New OracleConnection(oradb)
conn.Open()

Dim parm As New OracleParameter
parm.OracleDbType = OracleDbType.Decimal
parm.Value = TextBox1.Text

Dim cmd As New OracleCommand
cmd.Connection = conn
cmd.CommandText = "select department_id, department_name, city" _
    + " from departments d, locations l" _
    + " where d.location_id = l.location_id"
cmd.CommandType = CommandType.Text

Dim dr As OracleDataReader = cmd.ExecuteReader()
If dr.Read() Then
    Label1.Text = dr.Item("department_name")
Else
    Label1.Text = "Department ID not found"
End If

```

5. Also change the following lines of code from:

```

If dr.Read() Then
    Label1.Text = dr.Item("department_name")
Else
    Label1.Text = "Department ID not found"
End If

```

to

```

While dr.Read()
    ListBox1.Items.Add("The " + dr.Item(1) _
        + " department is in " + dr.Item("city"))
End While

```

```

Dim conn As New OracleConnection(oradb)
conn.Open()

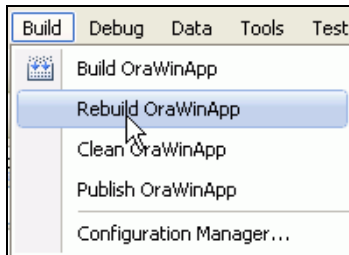
Dim parm As New OracleParameter
parm.OracleDbType = OracleDbType.Decimal
parm.Value = TextBox1.Text

Dim cmd As New OracleCommand
cmd.Connection = conn
cmd.CommandText = "select department_id, department_name, city" _
    + " from departments d, locations l" _
    + " where d.location_id = l.location_id"
cmd.CommandType = CommandType.Text

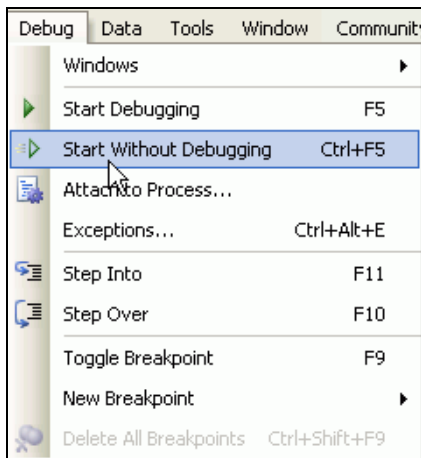
Dim dr As OracleDataReader = cmd.ExecuteReader()
While dr.Read()
    ListBox1.Items.Add("The " + dr.Item(1) _
        + " department is in " + dr.Item("city"))
End While

```

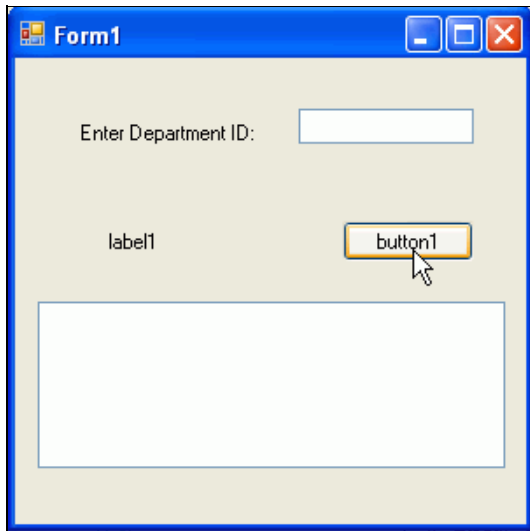
6. Select **Build > Rebuild OraWinApp**.



7. Select **Debug > Start Without Debugging**.



8 . Click **button1**.



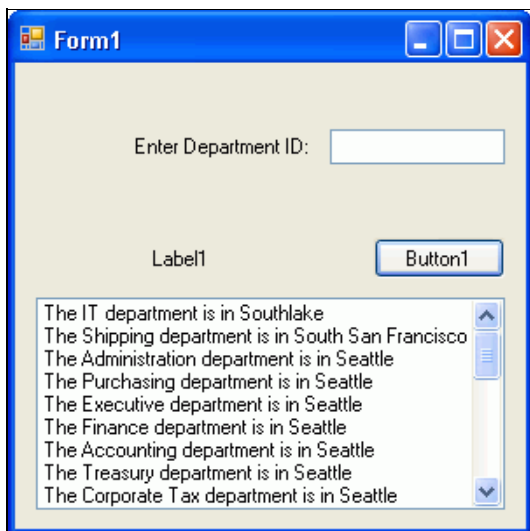
Form1

Enter Department ID:

label1

button1

9 . The list of all the departments and their location is shown. Close Form1.



Form1

Enter Department ID:

Label1

Button1

The IT department is in Southlake  
The Shipping department is in South San Francisco  
The Administration department is in Seattle  
The Purchasing department is in Seattle  
The Executive department is in Seattle  
The Finance department is in Seattle  
The Accounting department is in Seattle  
The Treasury department is in Seattle  
The Corporate Tax department is in Seattle

## Summary

In this tutorial, you have learned how to:

- Create a project and add a reference
- Retrieve data from the database
- Incorporate interactive controls and error handling
- Retrieve Multiple Columns and Rows