

Getting Started with Oracle Data Provider for .NET (C# 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

Topics

This tutorial covers the following topics:

- ☐ [Overview](#)
- ☐ [Prerequisites](#)
- ☐ [Creating a Project in Visual Studio](#)
- ☐ [Adding a Reference](#)
- ☐ [Retrieving Data from the Database](#)
- ☐ [Incorporating Interactive Controls](#)
- ☐ [Adding Error Handling](#)
- ☐ [Retrieving Multiple Columns and Rows](#)
- ☐ [Summary](#)

Viewing Screenshots

 **Place the cursor over this icon to load and view all the screenshots for this tutorial. (Caution: This action loads all screenshots simultaneously, so response time may be slow depending on your Internet connection.)**

Note: Alternatively, you can place the cursor over an individual icon in the following steps to load and view only the screenshot associated with that step. You can hide an individual screenshot by clicking it.

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.

[Back to Topic List](#)

Prerequisites

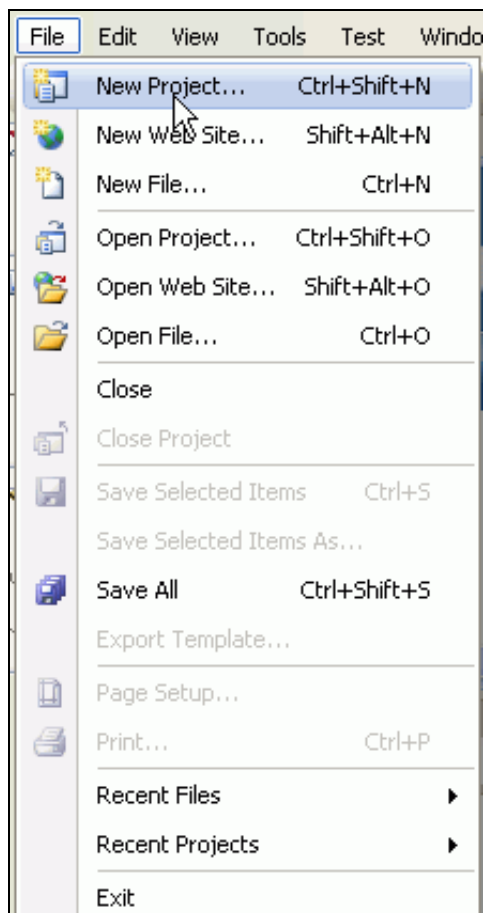
Before you perform this tutorial, you should:

1. Install Microsoft Visual Studio 2008
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 11.1.0.6.20 or later from [OTN](#)

Creating a Project in Visual Studio

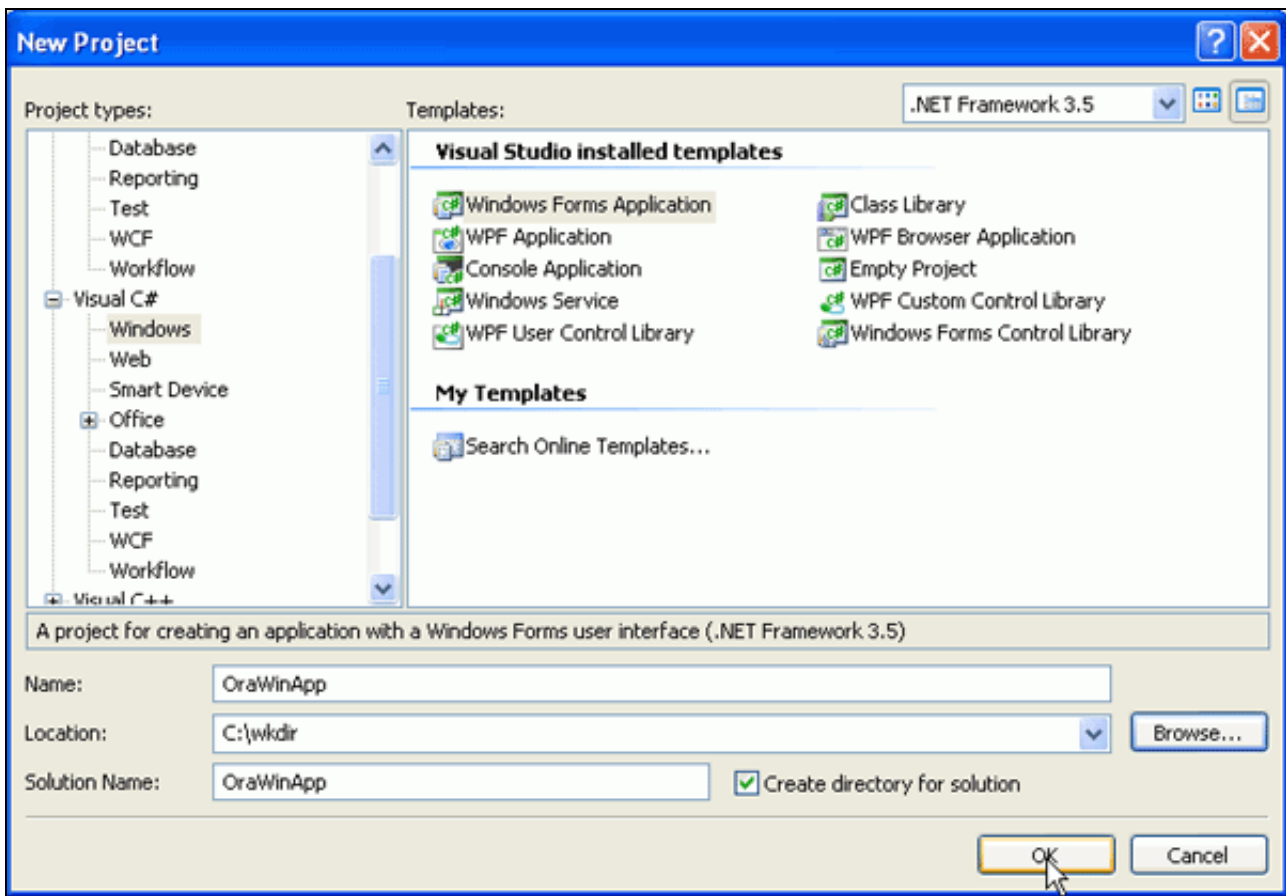
To create a new project in Visual Studio, perform the following steps:

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

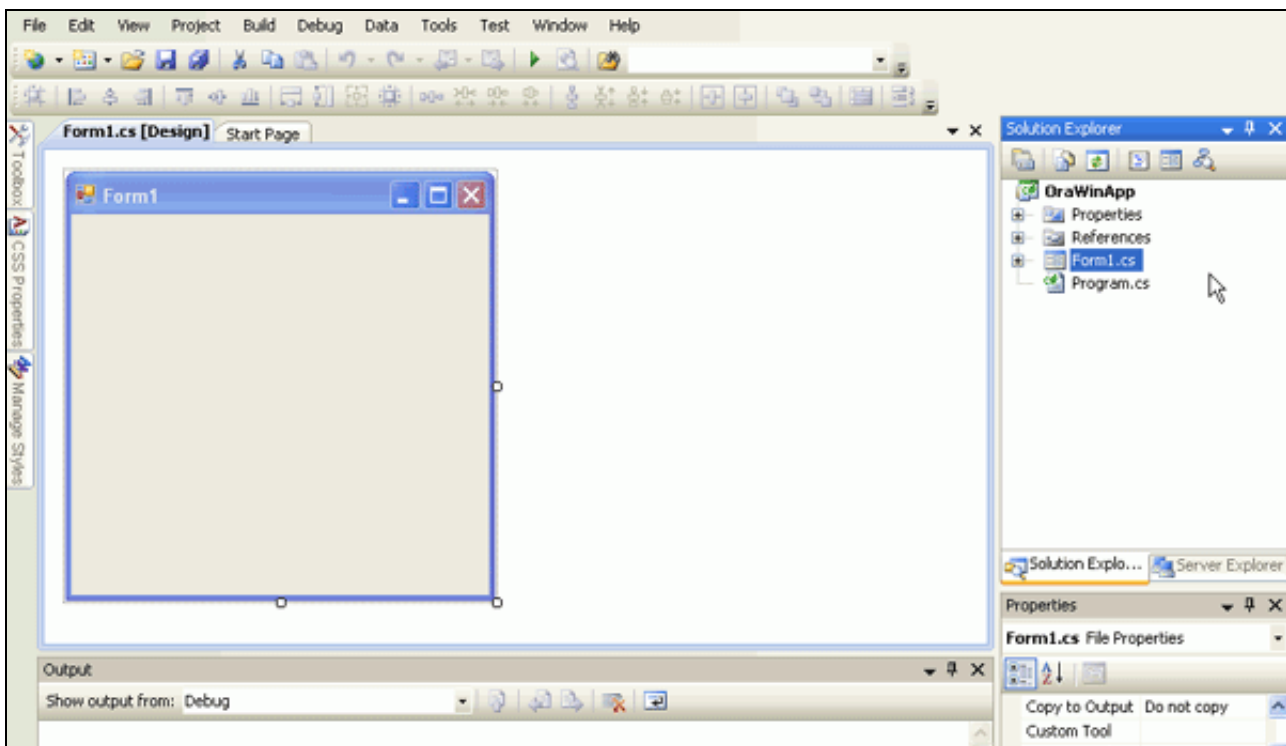


A New Project window appears.

2. From the Project types list, select **Visual C#: 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**.



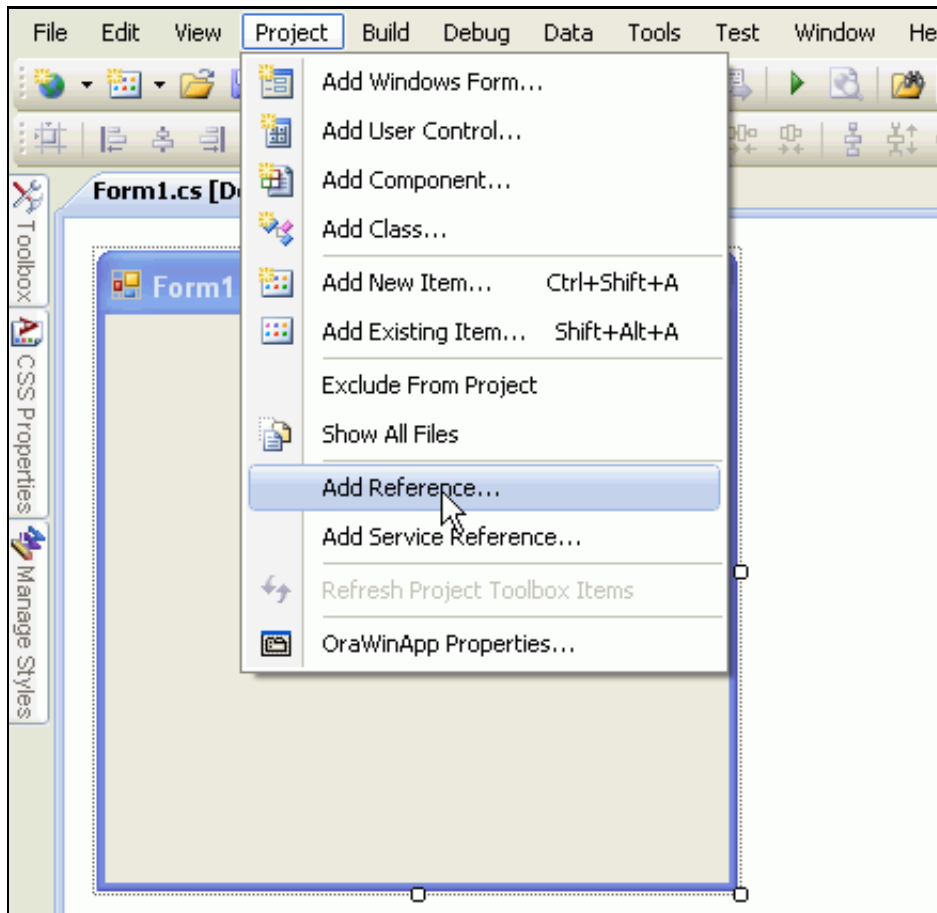
3. The project has been created. From View menu, select **Solution Explorer** to open the Solution Explorer if not already open.



Adding a Reference

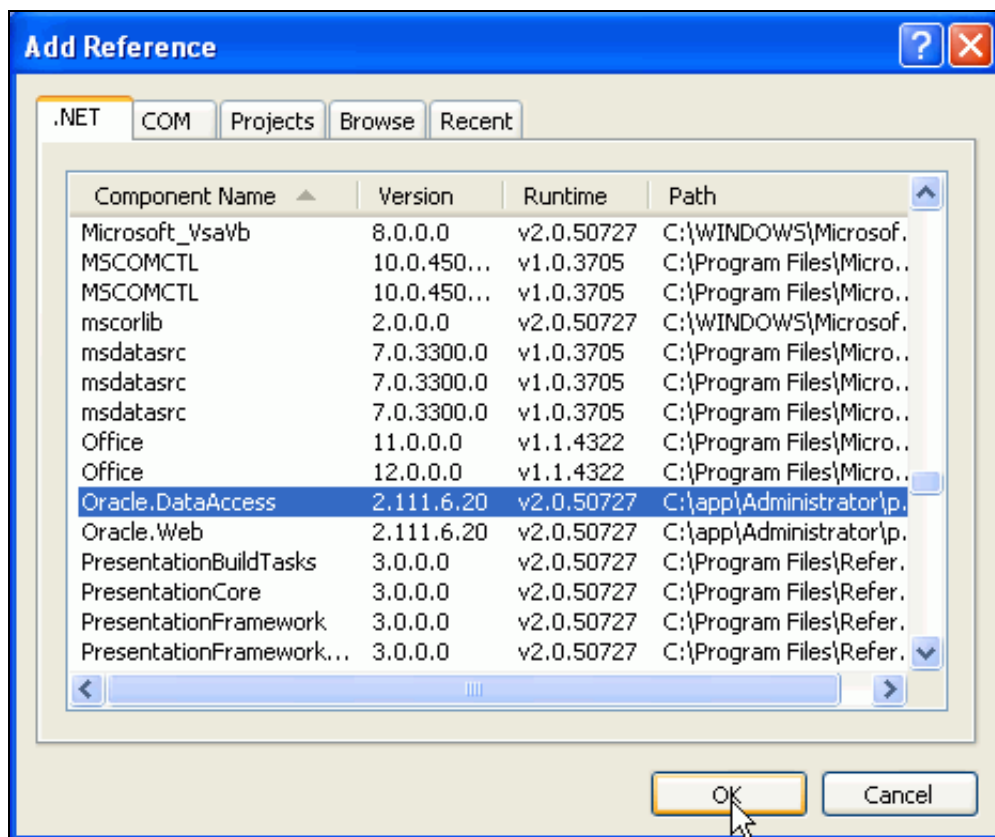
Because your project needs access to an Oracle database, it is necessary to add a reference to the dll containing the data provider. Perform the following steps:

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



An Add Reference dialog box appears.

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

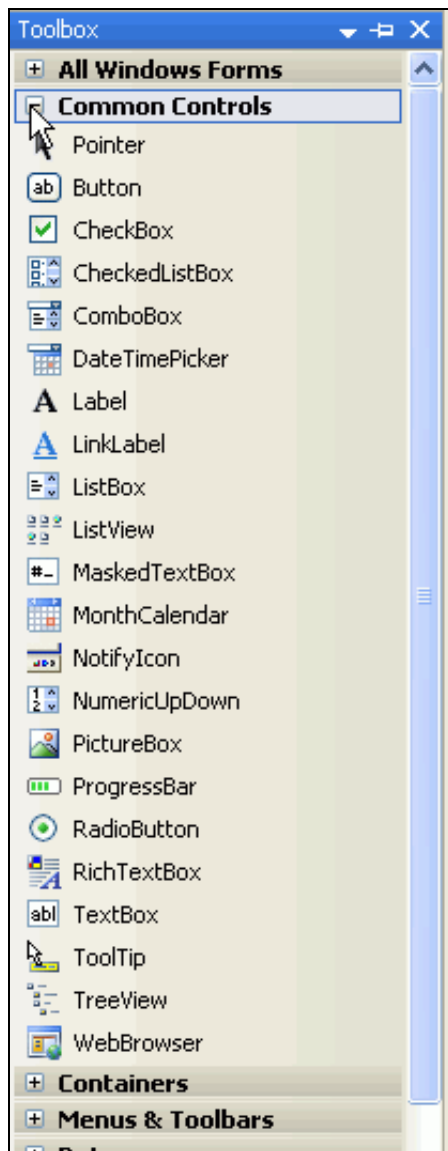


[Back to Topic List](#)

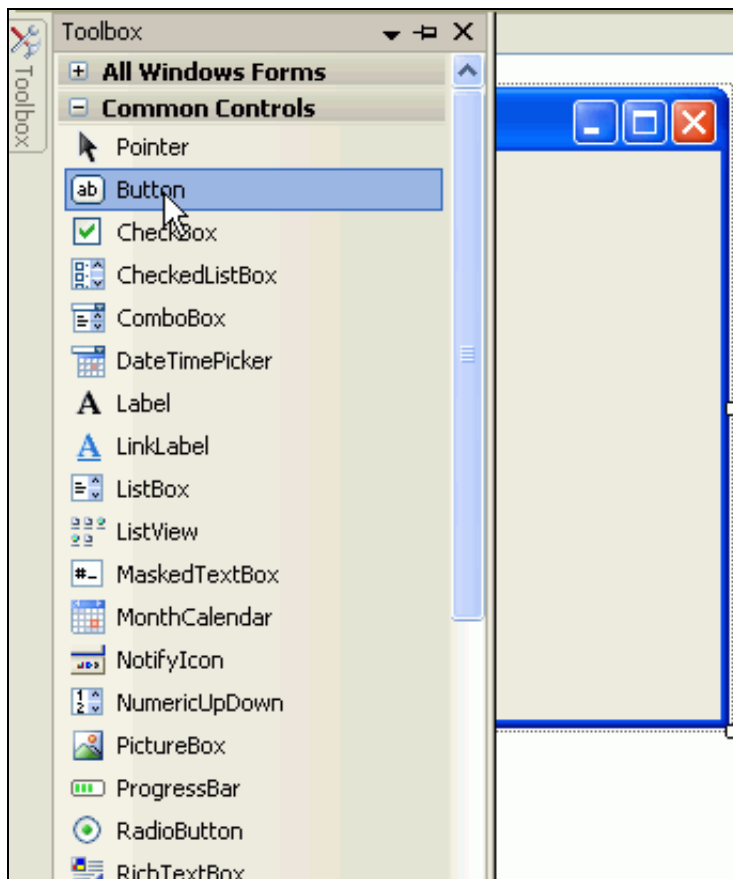
Retrieving Data from the Database

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

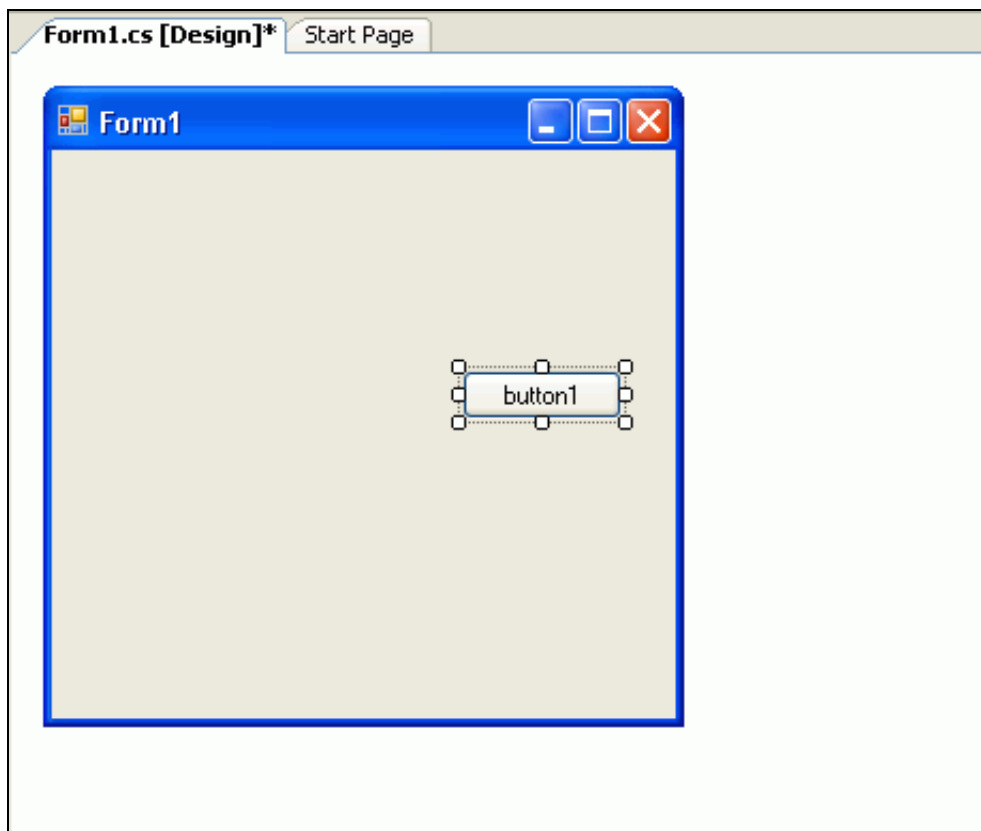
1. From **View** menu, select **Toolbox**. The toolbox appears. Expand **Common Controls**.



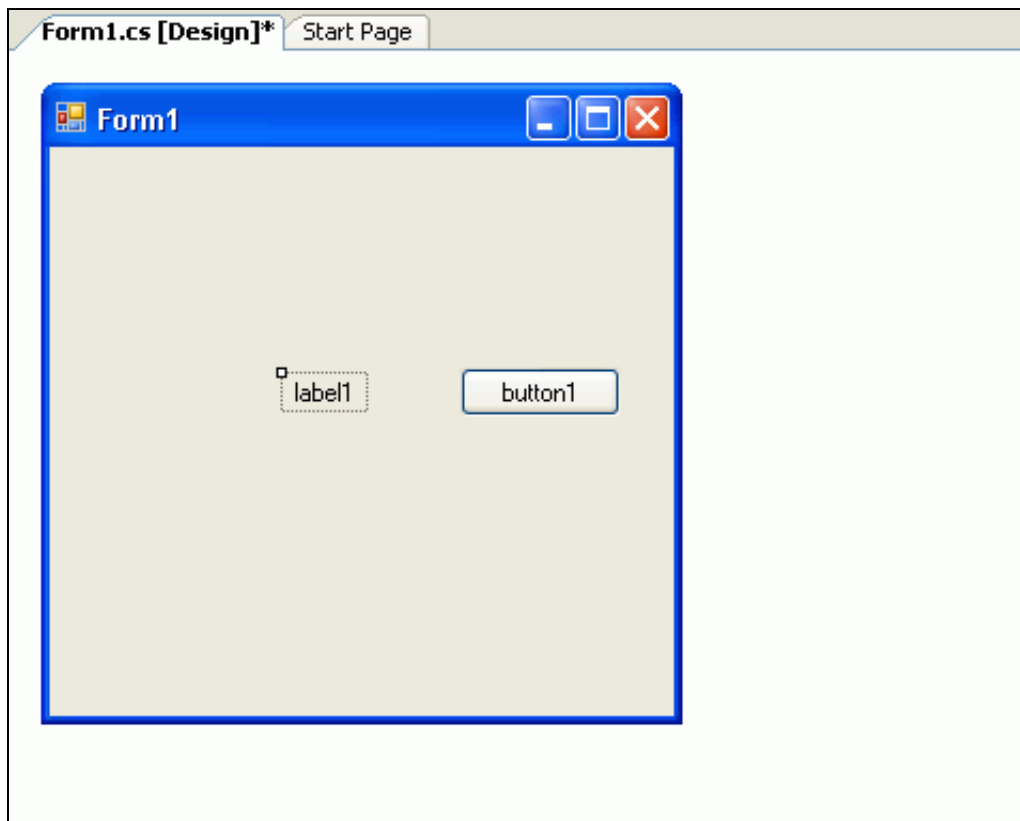
2. Drag the **Button** control to the Form1 canvas.



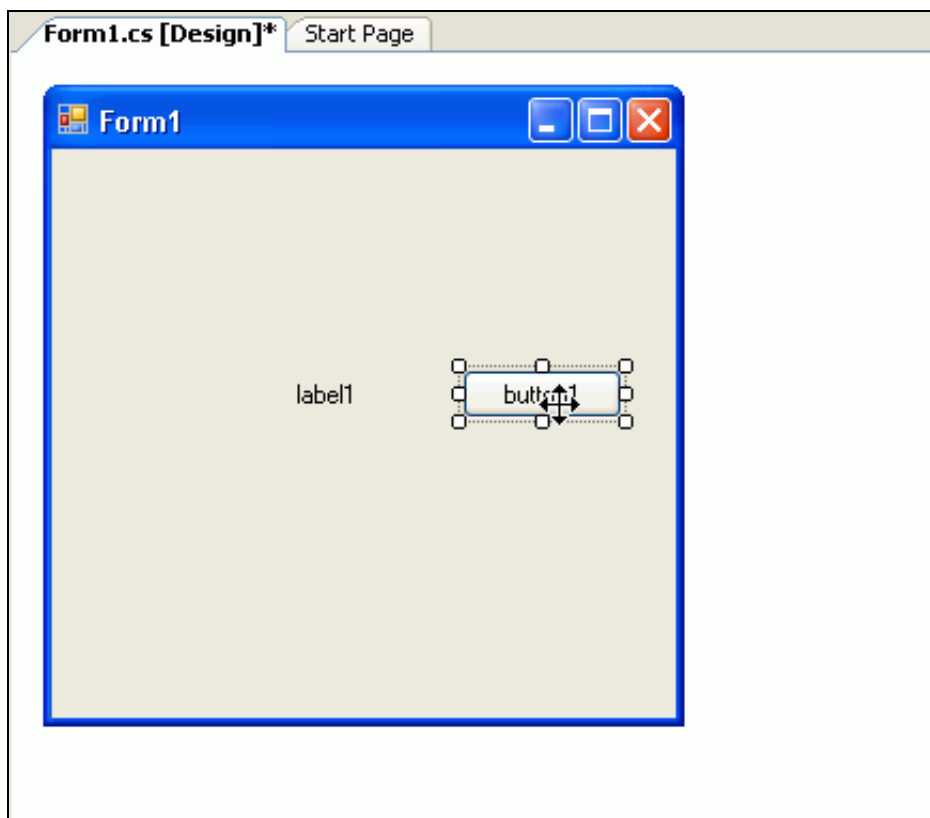
The control is shown on the canvas. You also need to create a Label.



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

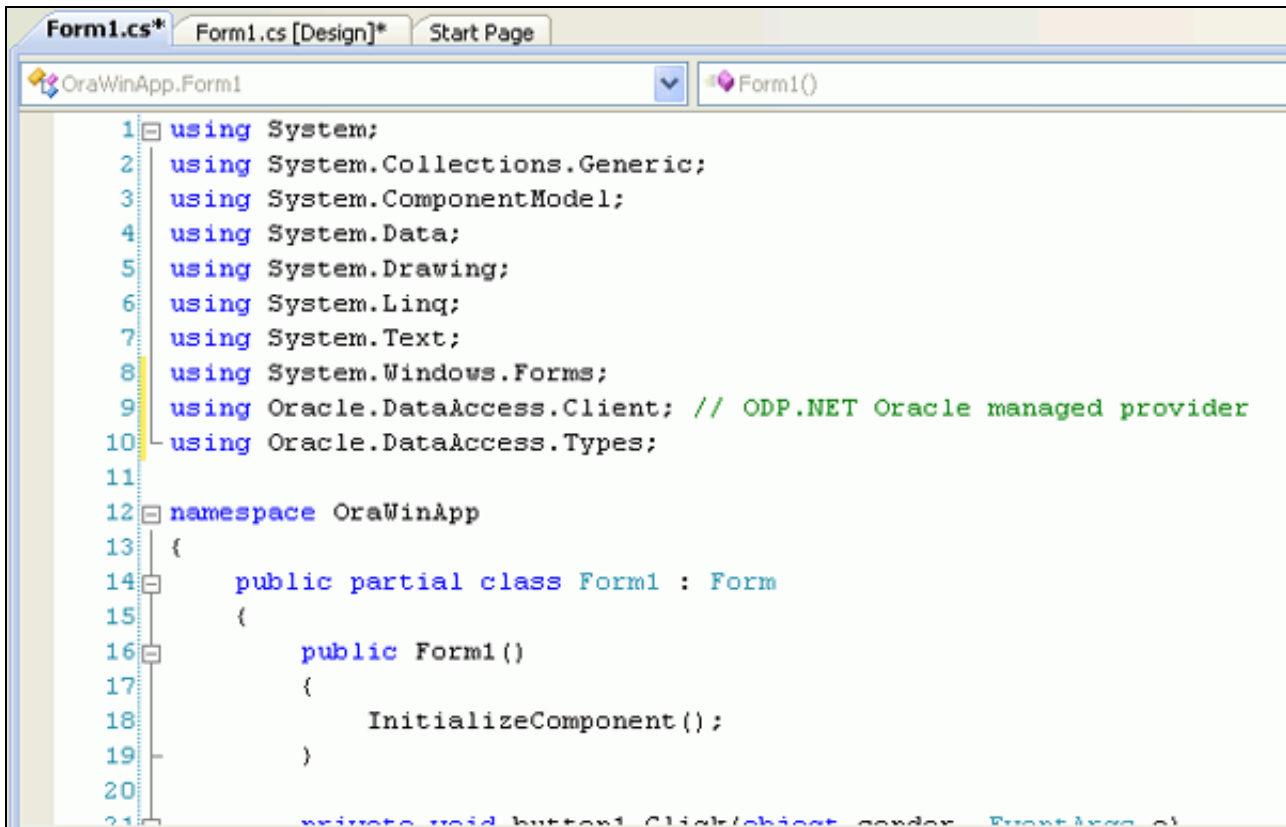


4. Double-click **button1** to open the code window. Note, Form1.cs code window opens.



5. Add the following C# using statements before the Public Class declaration.

```
using Oracle.DataAccess.Client; // ODP.NET Oracle managed provider
using Oracle.DataAccess.Types;
```

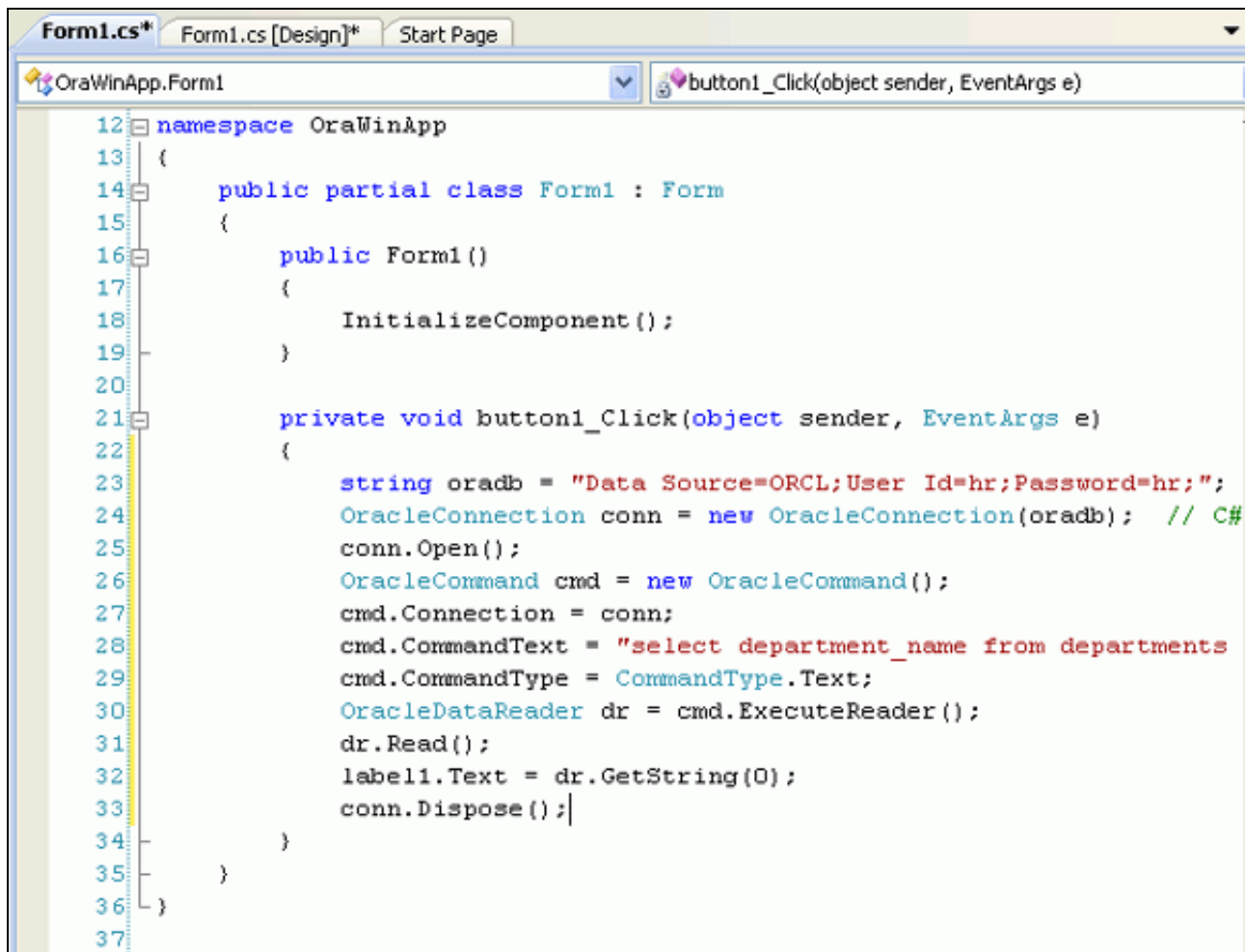



6. Add the following C# code in between the private void button1_Click(object sender, EventArgs e) { and } statements.

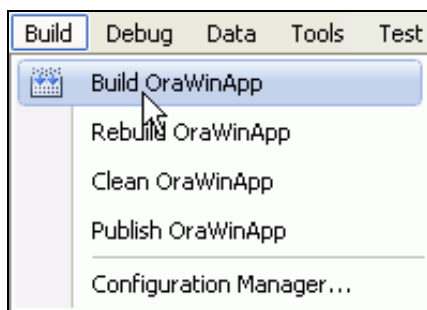
```

string oradb = "Data Source=ORCL;User Id=hr;Password=hr;";
OracleConnection conn = new OracleConnection(oradb); // C#
conn.Open();
OracleCommand cmd = new OracleCommand();
cmd.Connection = conn;
cmd.CommandText = "select department_name from departments where department_id = 10";
cmd.CommandType = CommandType.Text;
OracleDataReader dr = cmd.ExecuteReader();
dr.Read();
label1.Text = dr.GetString(0);
conn.Dispose();

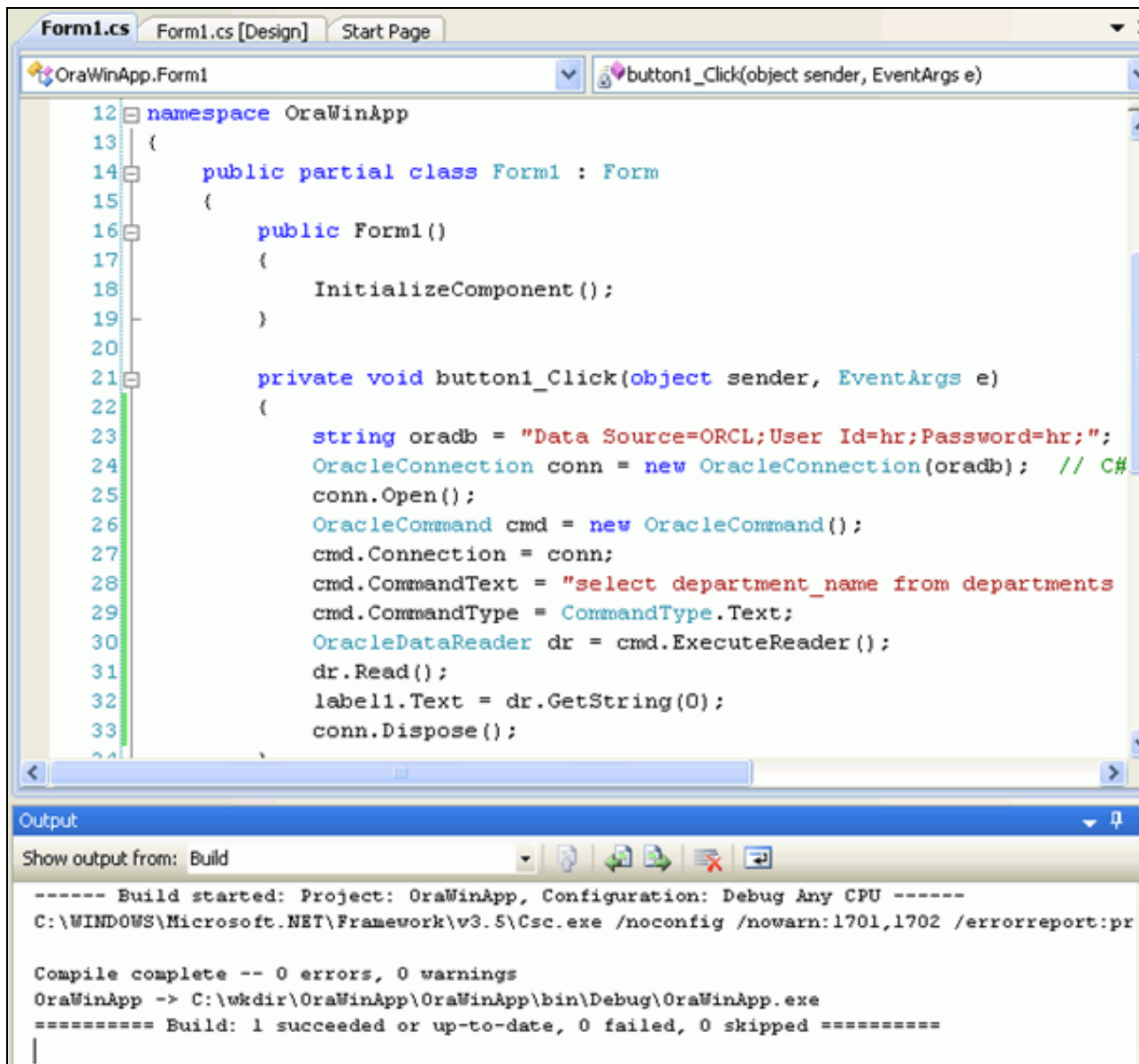
```



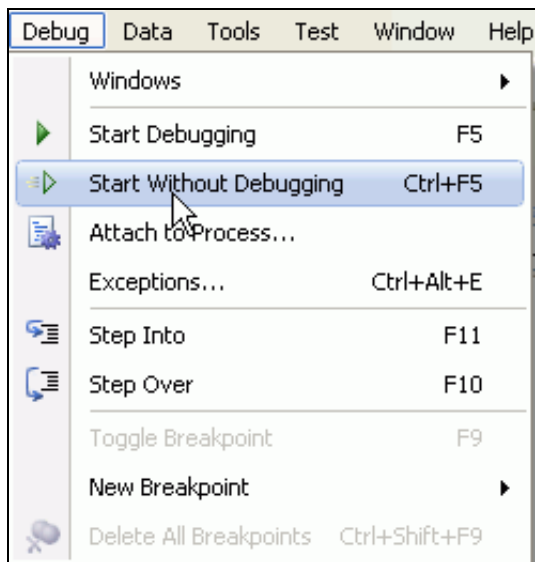
7. Select **Build > Build OraWinApp**.



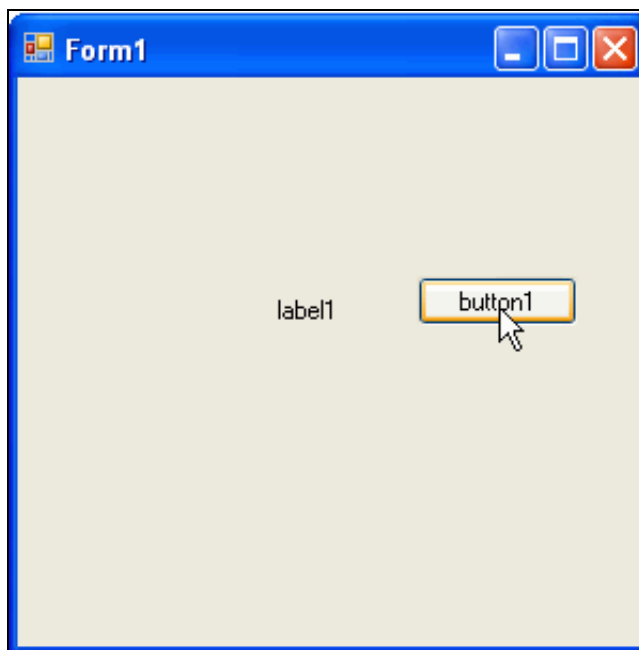
In the output window, ensure you did not get any errors. Now you can run the form.



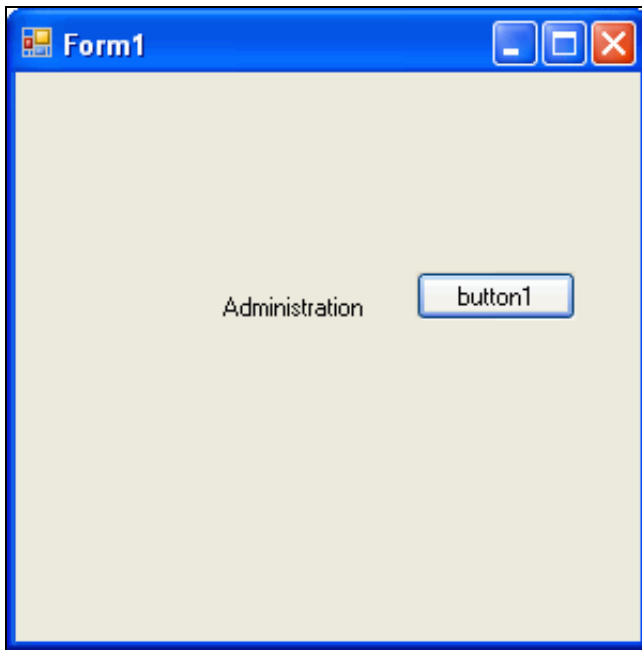
8. Select **Debug > Start Without Debugging**.



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.

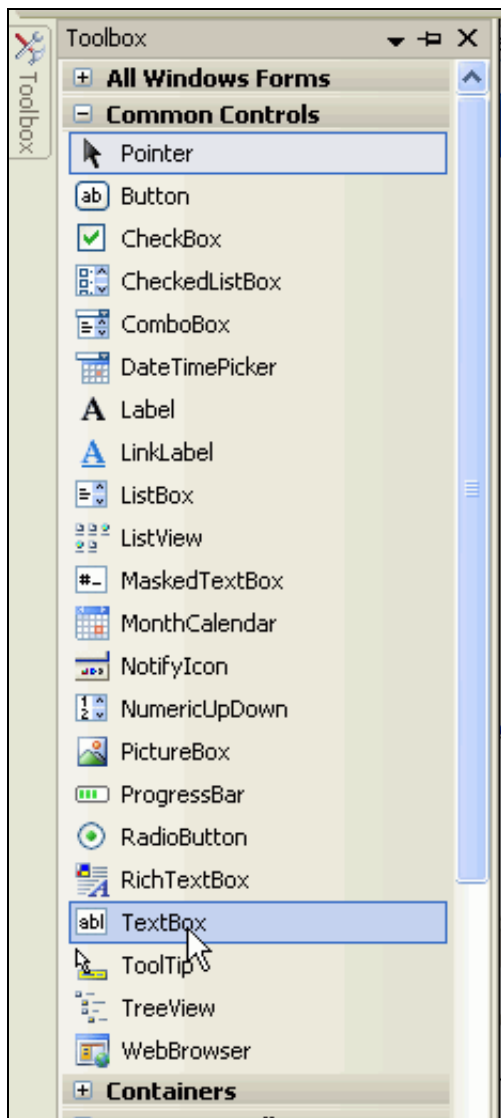


[Back to Topic List](#)

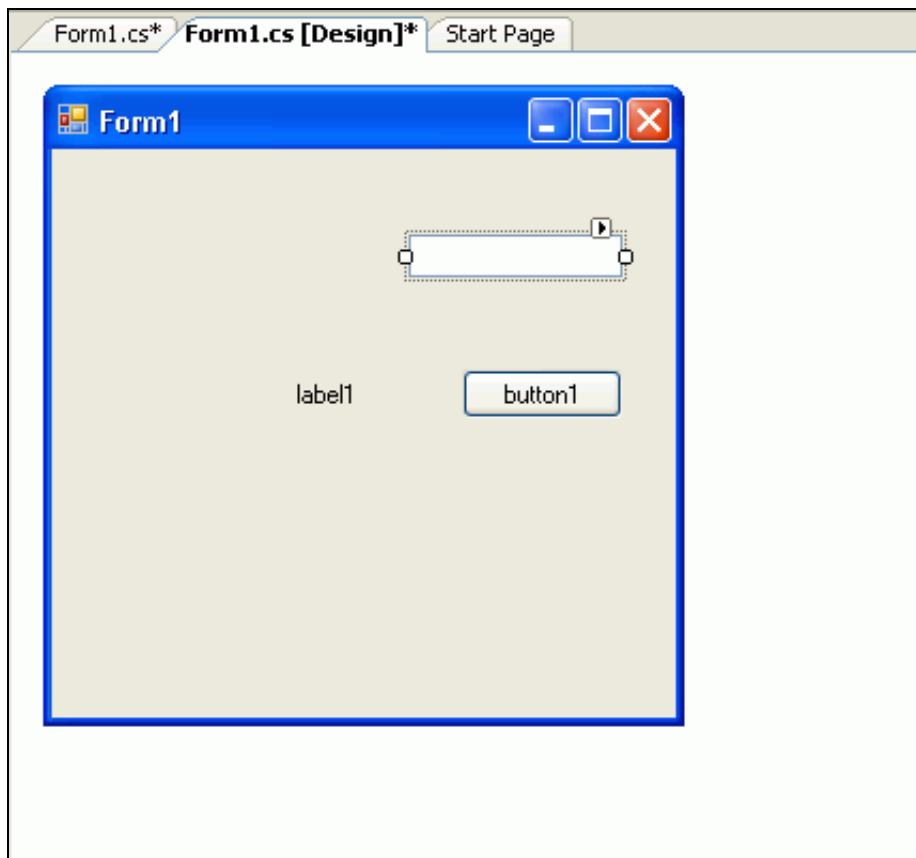
Incorporating Interactive Controls

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:

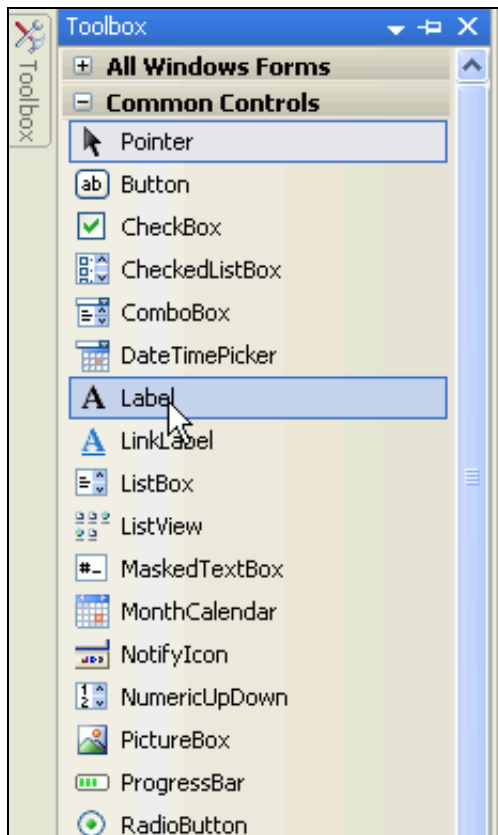
1. Select the **Form1.cs [Design]** tab. From the Toolbox, drag the **TextBox** control to Form1 canvas.



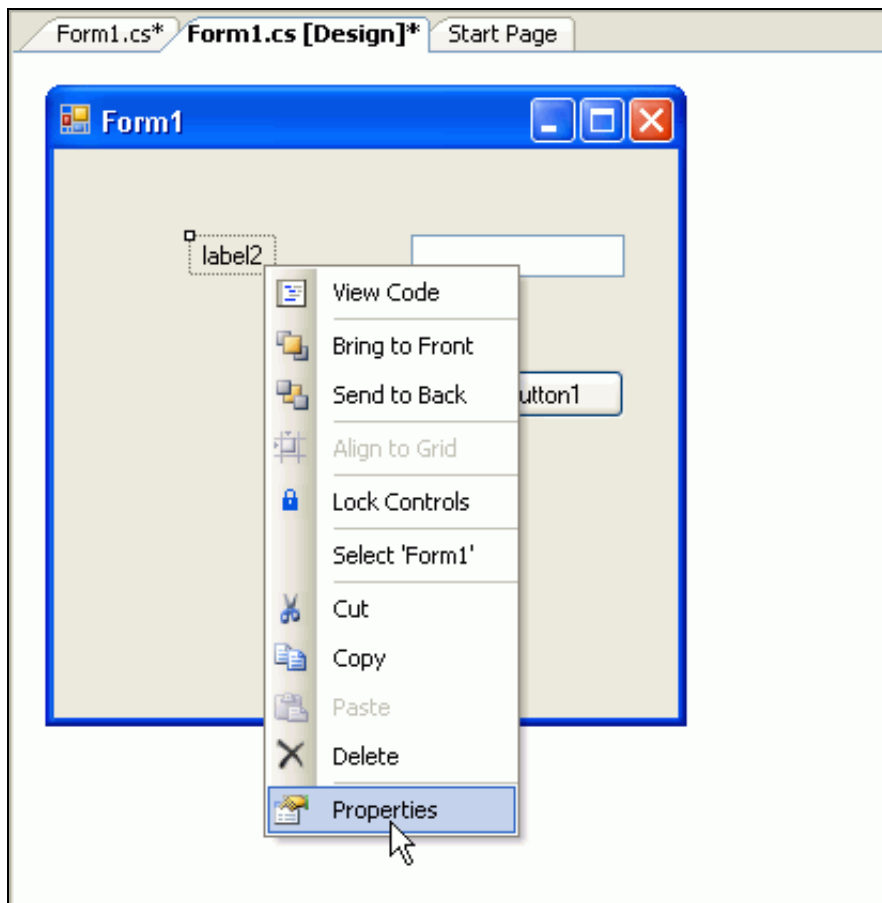
The textbox is shown. You can drag it to the desired location.



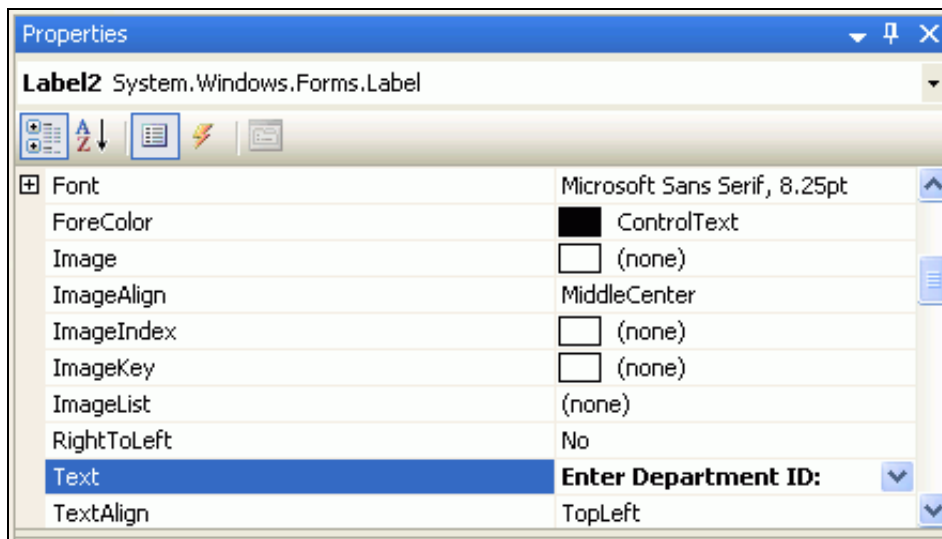
2. Drag a **Label** control on Form1 canvas.



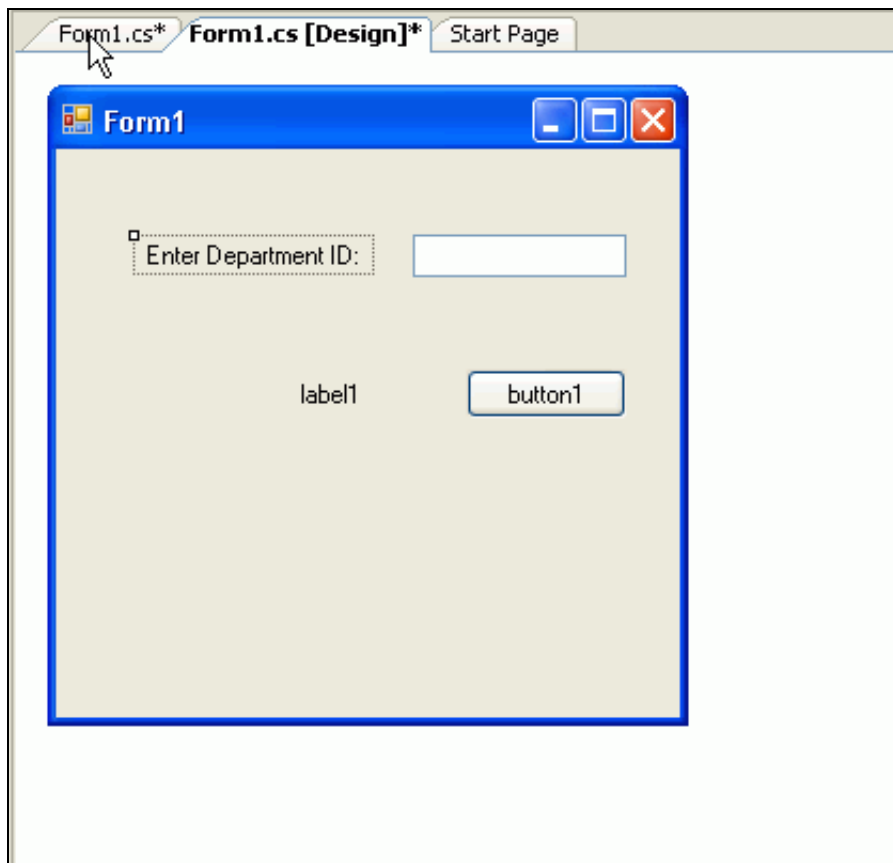
3. The Label is shown. Right-click **Label2** and select **Properties**.



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



5. Select the **Form1.cs** 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:

```
OracleParameter parm = new OracleParameter();  
parm.OracleDbType = OracleDbType.Decimal;  
parm.Value = textBox1.Text;
```

```
private void button1_Click(object sender, EventArgs e)
{
    string oradb = "Data Source=ORCL;User Id=hr;Password=hr;";

    OracleConnection conn = new OracleConnection(oradb); // C#
    conn.Open();

    OracleParameter parm = new OracleParameter();
    parm.OracleDbType = OracleDbType.Decimal;
    parm.Value = textBox1.Text;

    OracleCommand cmd = new OracleCommand();
    cmd.Connection = conn;
    cmd.CommandText = "select department_name from departments where department_id = 10";
    cmd.CommandType = CommandType.Text;

    OracleDataReader dr = cmd.ExecuteReader();
    dr.Read();
    label1.Text = dr.GetString(0);

    conn.Dispose();
}
```

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

`cmd.Parameters.Add(parm);`

```
private void button1_Click(object sender, EventArgs e)
{
    string oradb = "Data Source=ORCL;User Id=hr;Password=hr;";

    OracleConnection conn = new OracleConnection(oradb); // C#
    conn.Open();

    OracleParameter parm = new OracleParameter();
    parm.OracleDbType = OracleDbType.Decimal;
    parm.Value = textBox1.Text;

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

    OracleDataReader dr = cmd.ExecuteReader();
    dr.Read();
    label1.Text = dr.GetString(0);

    conn.Dispose();
}
```

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

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

```
private void button1_Click(object sender, EventArgs e)
{
    string oradb = "Data Source=ORCL;User Id=hr;Password=hr;";

    OracleConnection conn = new OracleConnection(oradb); // C#
    conn.Open();

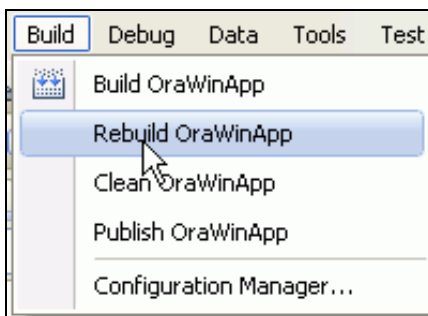
    OracleParameter parm = new OracleParameter();
    parm.OracleDbType = OracleDbType.Decimal;
    parm.Value = textBox1.Text;

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

    OracleDataReader dr = cmd.ExecuteReader();
    dr.Read();
    label1.Text = dr.GetString(0);

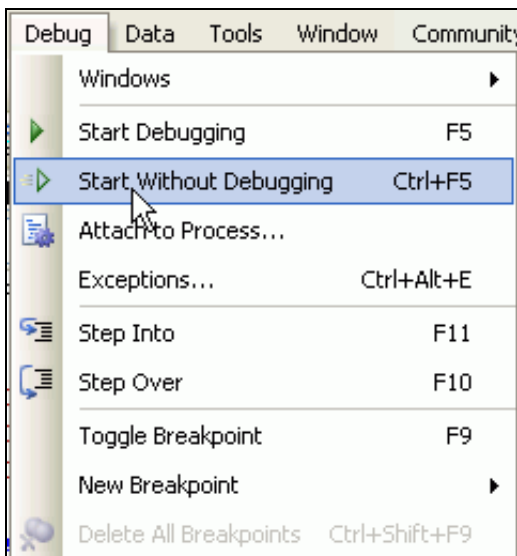
    conn.Dispose();
}
```

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

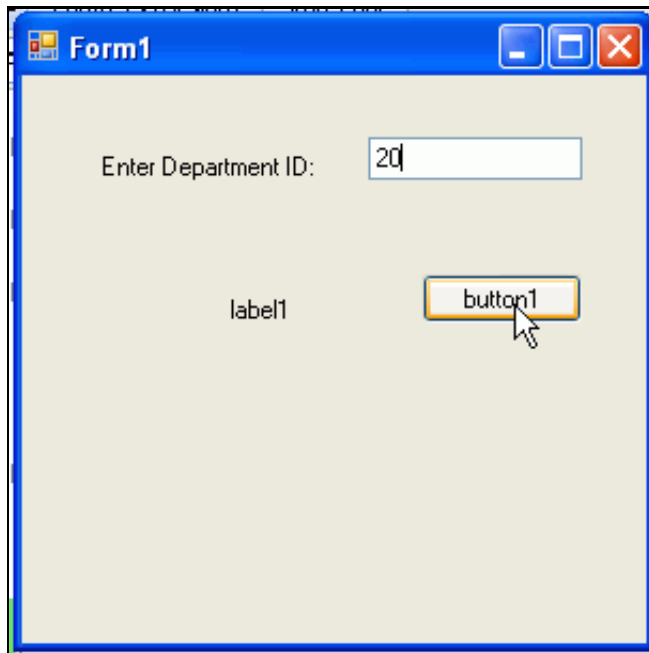


Check in the output window for any errors reported. If no errors or warnings, proceed with the next step.

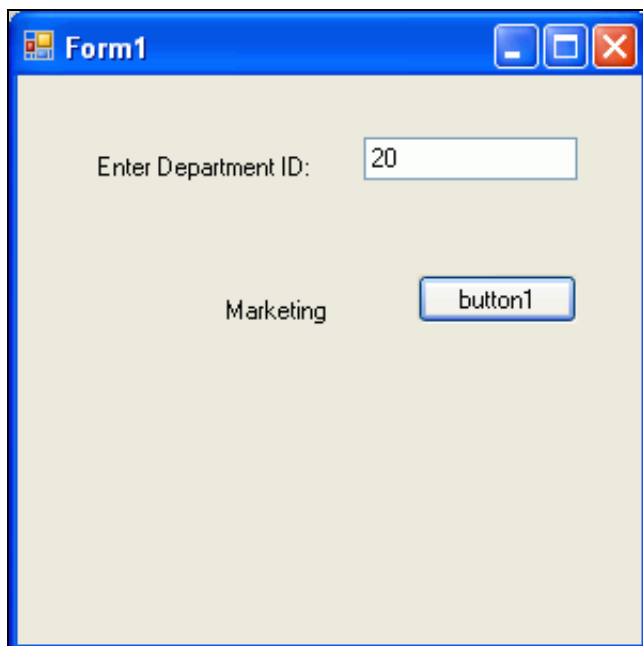
10. Select **Debug > Start Without Debugging**.



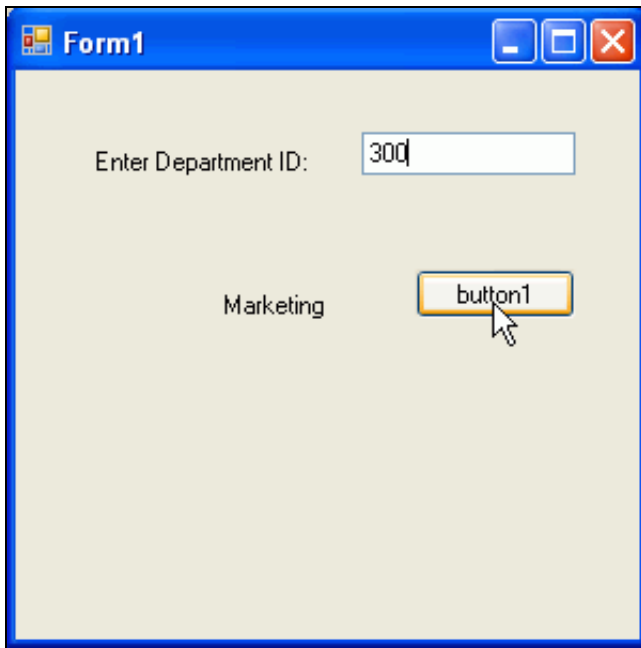
11. Enter **20** for the Department ID and click **button1**.

A screenshot of a Windows application window titled "Form1". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main area of the form is light gray. It contains a label "Enter Department ID:" followed by a text input field containing the number "20". Below the input field, the text "label1" is displayed. To the right of "label1" is a button labeled "button1". A mouse cursor is hovering over the button.

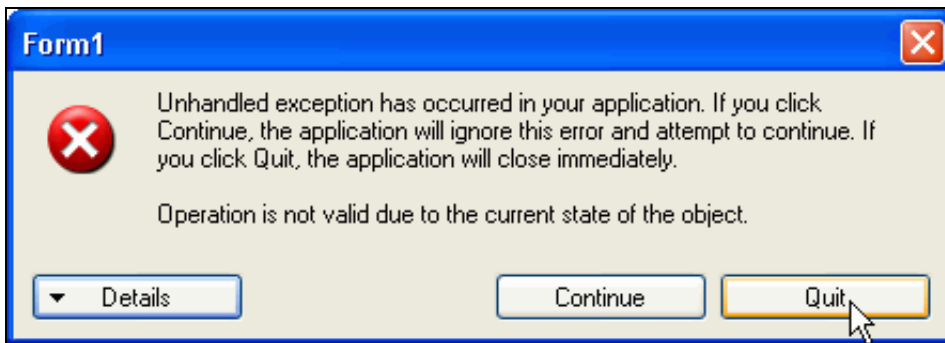
12. Notice that the Label1 changes to the name of the department. This value depends on what is entered in the Department ID field.

A screenshot of the same "Form1" window after a button click. The text input field still contains "20". The text "label1" has been replaced with the word "Marketing". The button "button1" is still present and visible.

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



[Back to Topic List](#)

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](#). Perform the following steps:

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

```
dr.Read();  
label1.Text = dr.GetString(0);  
  
to  
  
if (dr.Read())  
{  
    label1.Text = dr.GetString(0);  
}  
else  
{  
    label1.Text = "Department ID not found";  
};
```

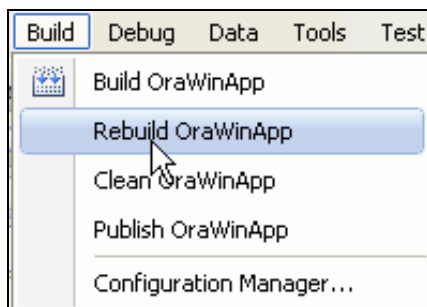
```
OracleConnection conn = new OracleConnection(oradb); // C#
conn.Open();

OracleParameter parm = new OracleParameter();
parm.OracleDbType = OracleDbType.Decimal;
parm.Value = textBox1.Text;

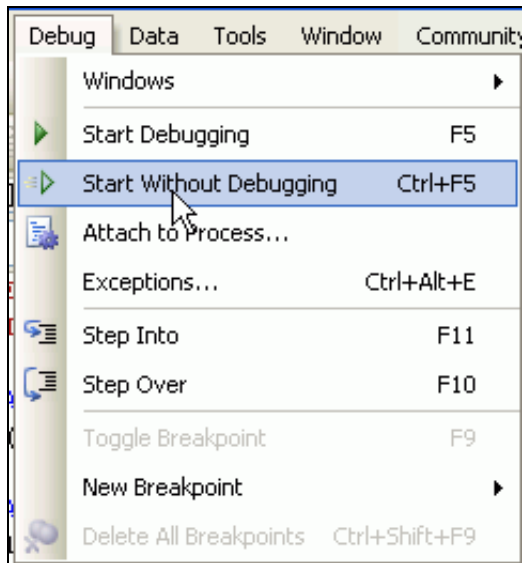
OracleCommand cmd = new OracleCommand();
cmd.Connection = conn;
cmd.Parameters.Add(parm);
cmd.CommandText = "select department_name from departments where department_id = :1";
cmd.CommandType = CommandType.Text;

OracleDataReader dr = cmd.ExecuteReader();
if (dr.Read())
{
    label1.Text = dr.GetString(0);
}
else
{
    label1.Text = "Department ID not found";
};|
```

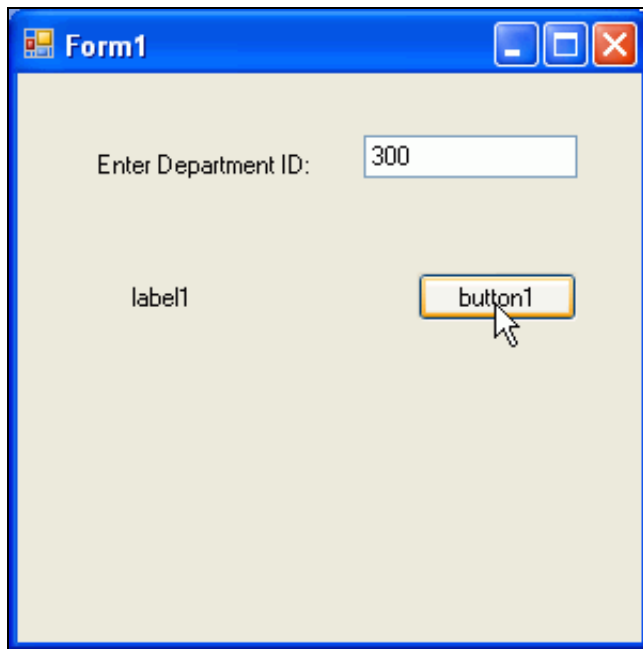
2. Select **Build > Rebuild OraWinApp**. Ensure that there are no errors reported in the output window.



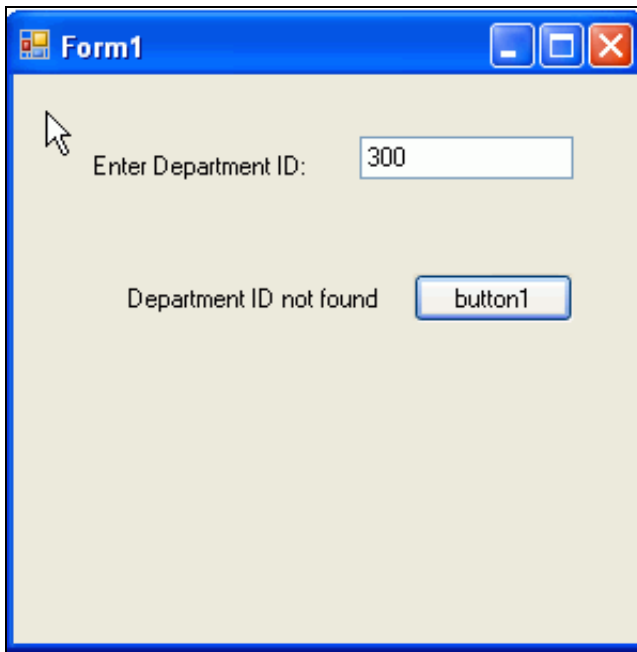
3. Select **Debug > Start Without Debugging**.



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



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

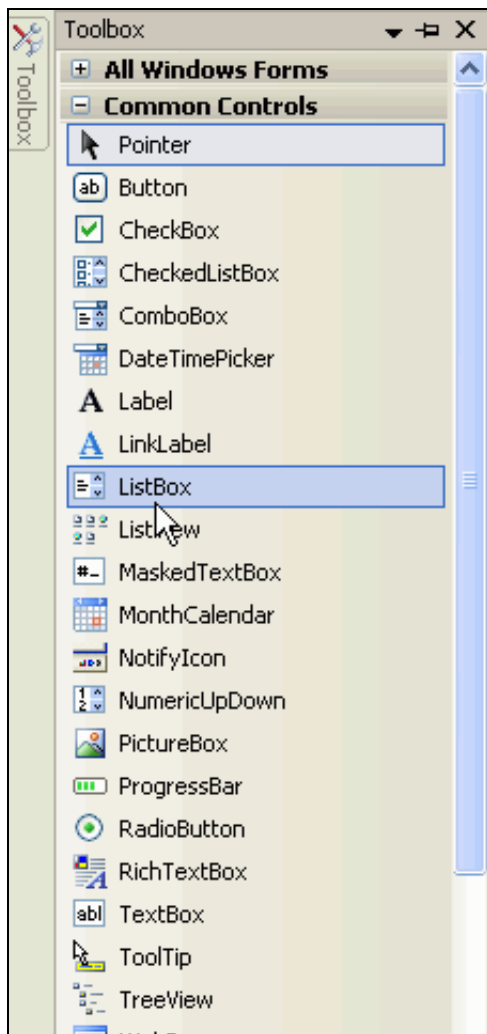


[Back to Topic List](#)

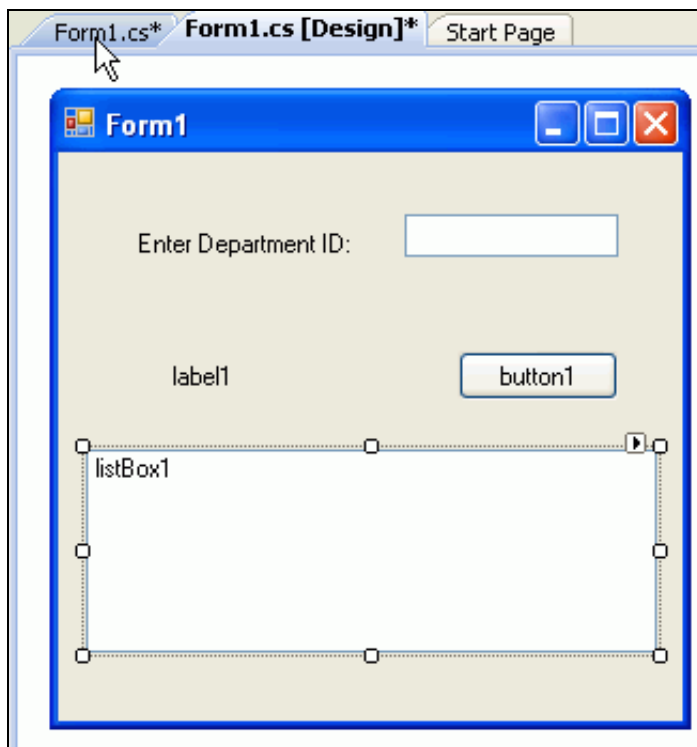
Retrieving Multiple Columns and Rows

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

1. Click the **Form1.cs [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.cs** tab.



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

```
OracleConnection conn = new OracleConnection(oradb); // C#
conn.Open();

OracleParameter parm = new OracleParameter();
parm.OracleDbType = OracleDbType.Decimal;
parm.Value = textBox1.Text;

OracleCommand cmd = new OracleCommand();
cmd.Connection = conn;
cmd.CommandText = "select department_name from departments where department_id = :1";
cmd.CommandType = CommandType.Text;

OracleDataReader dr = cmd.ExecuteReader();
if (dr.Read())
{
    label1.Text = dr.GetString(0);
}
else
{
    label1.Text = "Department ID not found";
};
```

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";
```

```

OracleConnection conn = new OracleConnection(oradb); // C#
conn.Open();

OracleParameter parm = new OracleParameter();
parm.OracleDbType = OracleDbType.Decimal;
parm.Value = textBox1.Text;

OracleCommand cmd = 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;

OracleDataReader dr = cmd.ExecuteReader();
if (dr.Read())
{
    label1.Text = dr.GetString(0);
}
else
{
    label1.Text = "Department ID not found";
}

```

5. Also change the following lines of code from:

```

if (dr.Read())
{
    label1.Text = dr.GetString(0);
}
else
{
    label1.Text = "Department ID not found";
};

to

while (dr.Read())
{
    listBox1.Items.Add("The " + dr.GetString(1)
        + " department is in " + dr.GetString(2));
}

```

```
OracleConnection conn = new OracleConnection(oradb); // C#
conn.Open();

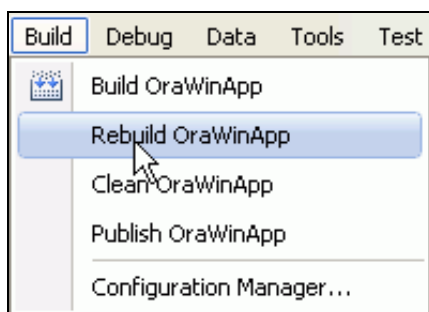
OracleParameter parm = new OracleParameter();
parm.OracleDbType = OracleDbType.Decimal;
parm.Value = textBox1.Text;

OracleCommand cmd = 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;

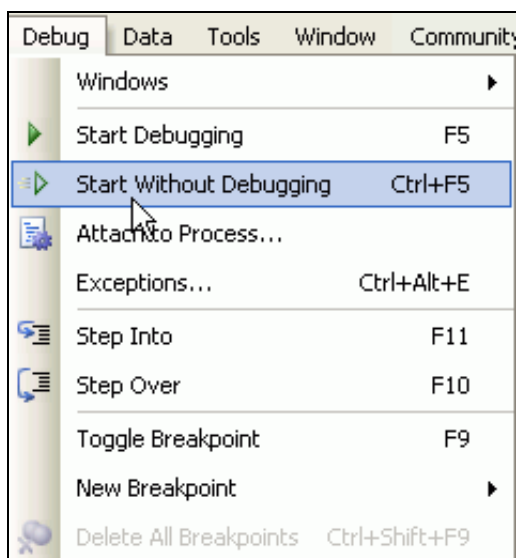
OracleDataReader dr = cmd.ExecuteReader();
while (dr.Read())
{
    listBox1.Items.Add("The " + dr.GetString(1) +
        " department is in " + dr.GetString(2));
}

conn.Dispose();
```

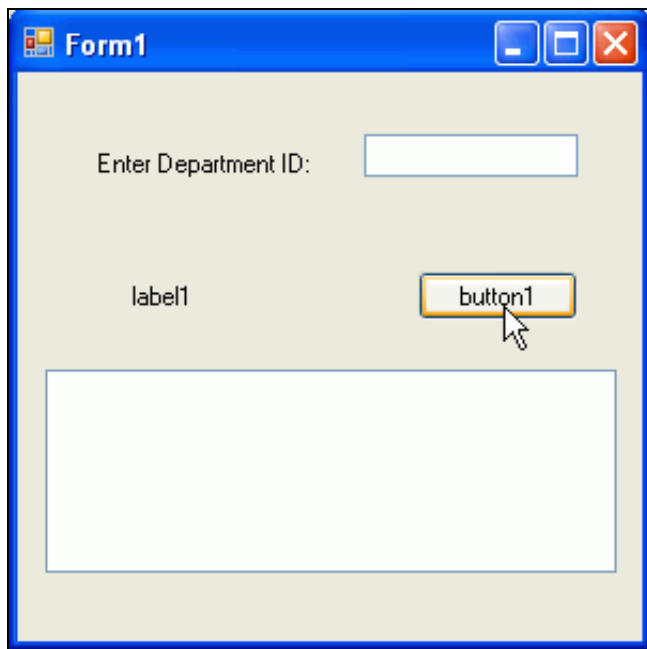
6. Select **Build > Rebuild OraWinApp**.



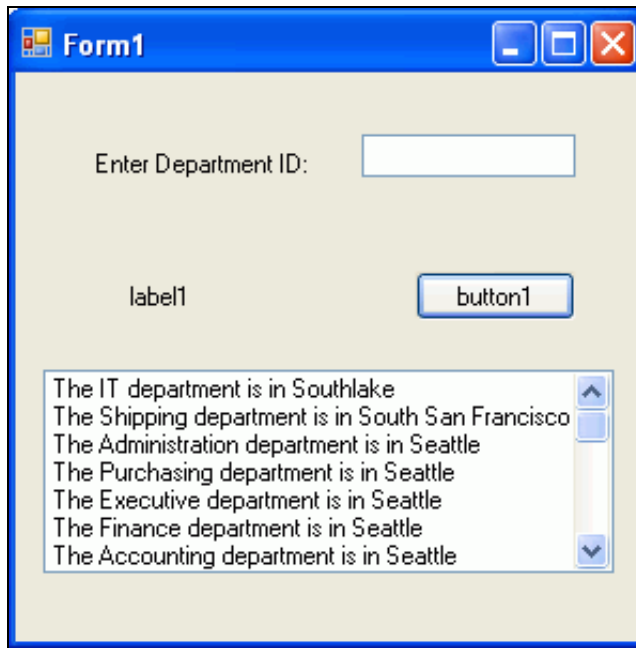
7. Select **Debug > Start Without Debugging**.



8. Click **button1**.



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

[Back to Topic List](#)

Summary

In this tutorial, you 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

[Back to Topic List](#)

 **Move your mouse over this icon to hide all screenshots.**