#### Oracle® Rdb Developer Tools for Visual Studio Developer's Guide V7.3.2.1 May 2011

Oracle Rdb Developer Tools for Visual Studio Developer's Guide, Release 7.3-21 Copyright © 2011 Oracle Corporation Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the Programs on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, back up, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Rdb, Oracle RMU and Oracle SQL/Services are trademark or registered trademarks of Oracle Corporation.

All other company or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

# Contents

Preface		4
Chapter 1		7
Introducing (	Oracle Rdb Developer Tools for Visual Studio	7
Chapter 2	Installing Oracle Rdb .NET Products	7
Chapter 3	Using Oracle Rdb Developer Tools for Visual Studio	8
3.1 Using	Oracle Rdb Developer Tools	8
3.2 Conne	cting to the Oracle Rdb Database	8
Chapter 4	Building a Simple .NET Application Using ORDP.NET	
4.1 Creatir	ng a New Project	13
4.2 Adding	a Reference	16
4.3 Adding	y Namespace Directives	17
4.4 Design	ning the User Interface	19
4.5 Writing	g the Connection Code	23
4.6 Compi	ling and Running the Application	26
	Handling	
	ing Try-Catch-Finally Block Structure andling General Errors	
	Retrieving and Updating with ORDP.NET	
_	the Command Object	
-	ving Data: a Simple Query	
	ving Data: Bind Variables	
	ving Data: Multiple Values	
5.5 Using	the DataSet Class with Oracle Rdb Data Provider for .NET	
5.6 Enablii	ng Updates to the Database	
Chapter 6	ORDP.NET and Visual Studio Wizards	45
•	a New application	
	onnection to RDB database in MS server explorer	

6.3 Create	a datasource for the employees table	45
6.4 Design	the windows application	47
Chapter 7	ORDP.NET and drag and drop	50
7.1 Create	a New application	50
7.2 Add col	nnection to RDB database in MS server explorer	50
7.3 Add a i	new Typed DataSet	
7.4 Designi	ing the windows application	
	g the windows application	
Chapter 8	Configuration wizard and generating queries	54
8.1 Using t	he Query Configuration Wizard	54
8.2 Using t	he Query Builder	56
8.3 Method	ds Generation	
Chapter 9	Copying a Form	61
Chapter 10	Oracle Rdb Entity Framework Provider	64
10.1 Creatin	ng new Windows Form Application	64
	ding an Entity Data Model	
	Iding a new Data Source	
	ing the Data Source in a Windows Form Iding Code to Populate the Data Grid View	
	a new Entity	
10.3 Creatin	ng Complex Types	75
10.3.1 Us	ing the Complex Type created	77
10.4 Genera	ting Database Schema from Model	79
Chapter 11	Unsupported features	82

# <u>Glossary</u>

# **Send Us Your Comments**

#### Oracle Rdb Developer Tools for Visual Studio Developer's Guide, V7.3

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: nedc-doc\_us@oracle.com
- FAX 603-897-3825 Attn: Oracle Rdb
- Postal service:
   Oracle Corporation
   Oracle Rdb Documentation
   One Oracle Drive
   Nashua, NH 03062-2804
   USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

# Preface

This document is your primary source of introductory and usage information for Oracle Rdb Developer Tools for Visual Studio.

This preface contains these topics:

- Audience
- Organization
- Related Documentation
- Conventions

### Audience

*Oracle Rdb Developer Tools for Visual Studio Developer's Guide* is intended for developers who are developing applications to access an Oracle Rdb database using Oracle Rdb Data Provider for .NET. This documentation is also valuable to systems analysts, project managers, and others interested in the development of database applications.

To use this document, you must be familiar with Microsoft .NET Framework classes and ADO.NET and have a working knowledge of application programming using Microsoft C#, Visual Basic, or C++.

Users should also be familiar with the use of Structured Query Language (SQL) to access information in relational database systems.

## Organization

This document contains:

- Chapter 1, Introducing Oracle Rdb Developer Tools for Visual Studio.
  Provides an overview of Oracle Rdb Developer Tools for Visual Studio.
- <u>Chapter 2, Installing Oracle Rdb .NET Products</u>
   Describes how to install Oracle Rdb Data Provider for .NET and provides system requirements. Read this chapter *before* installing or using Oracle Rdb Developer Tools for Visual Studio.
- Chapter 3, Using Oracle Rdb Developer Tools for Visual Studio
- Describes how to create a database connection using ORDP.NET within Visual Studio.
   Chapter 4, Building a Simple .NET Application Using ORDP.NET
   Provides an example on how you may build a simple application with database access using
- ORDP.NET.
   Chapter 5, Retrieving and Updating with ORDP.NET Provides an example of data retrieval and update using the features of Visual Studio and data access using ORDP.NET.
- <u>Chapter 6, ORDP.NET and Visual Studio Wizards</u> Provides an example on using standard Visual Studio wizards to automatically generate the code to access data from your Rdb database.
- <u>Chapter 7, ORDP.NET and drag and drop</u> Provides an example of using the drag and drop features of Visual Studio to generate code for your application.
- Chapter 8, Configuration wizard and generating queries Provides an example of using Table Adapter Query Configuration Wizard to generate queries on your database.
- <u>Chapter 9, Copying a Form</u> Describes how to copy a form as required when following the examples provided in this document.
- Chapter 10. Oracle Rdb Entity Framework Provider Describes how to use ORDP.NET within the Entity Framework using standard Entity Framework tools and features of Visual Studio.
- <u>Chapter 11. Unsupported features</u> Specifies what Visual Studio features are currently not supported by ORDT.
- <u>Glossary</u> Defines terms used in this document.

# **Related Documentation**

For more information, see these Oracle Rdb resources:

- Oracle Rdb7 Guide to Database Design and Definition
- Oracle Rdb7 Guide to Database Performance and Tuning
- Oracle Rdb Introduction to SQL
- Oracle Rdb 7.2 SQL Reference Manual
- Oracle Rdb Guide to SQL Programming
- Oracle SQL/Services Server Configuration Guide
- Guide to Using the Oracle Rdb7 Oracle SQL/Services (tm) Client API
- Oracle Rdb JDBC Driver Users Guide

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Rdb web site:

#### http://www.oracle.com/technetwork/database/rdb

For additional information, see: http://msdn.microsoft.com/netframework

### Conventions

Oracle Rdb Data Provider for .NET is often referred to as ORDP.NET.

Oracle Rdb is often referred to as Rdb.

Hewlett-Packard Company is often referred to as HP.

The following conventions are used in this document:

word	A lowercase word in a format example indicates a syntax element that you supply.
[]	Brackets enclose optional clauses from which you can choose one or none.
{ }	Braces enclose clauses from which you must choose one alternative.
	A horizontal ellipsis means you can repeat the previous item
•	A vertical ellipsis in an example means that information not directly related to the example has been omitted.

#### **Conventions in Code Examples**

Code examples illustrate SQL or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

SELECT last name FROM employees WHERE last name = 'TOLIVER';

Show Navigation

# Chapter 1 Introducing Oracle Rdb Developer Tools for Visual Studio.

Oracle Rdb Developer Tools for Visual Studio (ORDT) is a set of application tools that integrate with the Visual Studio environment. These tools provide graphical user interface access to Oracle Rdb functionality, enable the user to perform a wide range of application development tasks, and improve development productivity and ease of use.

ORDT includes:

• Oracle Rdb Data Provider for .NET ( ORDP.NET).

A standard .NET Data Provider that allows access to Oracle Rdb databases from the .NET environment

• Oracle Rdb DDEX Provider.

A standard .NET DDEX provider that integrates ORDP.NET functionality into Visual Studio.

• Oracle Rdb Entity Framework Provider.

A standard .NET Entity Framework provider that integrates ORDP.NET functionality into Entity Framework.

# Chapter 2 Installing Oracle Rdb .NET Products

See the Oracle Rdb Developer Tools for Visual Studio Release Notes for information on how to install Oracle Rdb .NET products.

Show Navigation

# Chapter 3 Using Oracle Rdb Developer Tools for Visual Studio

This chapter contains:

- Using Oracle Rdb Developer Tools
- <u>Connecting to the Oracle Rdb Database</u>

## 3.1 Using Oracle Rdb Developer Tools

Oracle Rdb Developer Tools for Visual Studio (ORDT) is a tightly integrated Add-in for Visual Studio. Using enhancements that ORDT brings to the Server Explorer, you can automatically create tables, indexes, constraints, data connections and other database schema objects. Additionally you can automatically generate application code.

## 3.2 Connecting to the Oracle Rdb Database

This section shows you how to use the Server Explorer to connect to the Oracle Rdb Database for the purpose of automatically creating or modifying database schema objects.

### To connect to the database:

- **Step1:** From the View menu, select **Server Explorer**.
- **Step2:** In Server Explorer, right-click **Data Connections**.
- Step3: Select Add Connection.

Server Explorer			
🛃 🗵 💐 🛄			
🕀 👘 Data Conne	ections		1
🗄 🛛 📇 Servers	\$	Refresh	
	$\times$	Delete	
		Add Connection 💦	
		Create New SQL Server Database	
	Ŀ.	Properties	

**Step4:** When the **Add Connection** window appears, determine if the Data source says Oracle Rdb Database (RdbClient).

If it does, skip to Step  $\underline{6}$ .

Add Connection	? 🗙
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.	
Data source:	
Char	nge

If Data source does not say Oracle Rdb Database (RdbClient), select **Change**.

The Change Data Source window appears.

Change Data Source	? 🛛
Data source: Microsoft Access Database File Microsoft ODBC Data Source Microsoft SQL Server 2005 Compact Edition Microsoft SQL Server Database File Oracle Database Oracle Rdb Database <other> Data provider: .NET Framework Data Provider For Oracle</other>	Description .NET Framework Data Provider For Oracle Rdb
Always use this selection	OK Cancel

- **Step5:** Choose Oracle Rdb Database and then select .NET Framework Data Provider For Oracle Rdb.
- **Step6:** On the Connection Details tab, in the Add Connection window, enter the following information:

### To connect to a database using a SQL/Services universal service:

For **Server**, enter the node address and a valid SQL/Services service that will be running on the selected node. Remember to separate the node address and service name with a colon (:). In this example we assume that a universal service called GENERIC is running on the selected node.

The node address must be valid TCP/IP node specification.

For **User name**, enter a valid username for the database you will select.

For **Password**, enter the password for that user

To save the password for future sessions, check the **Save password** box.

For **Database name**, Enter a valid database file specification. Depending on the setup of SQL/Services on the server node, this database specification may include device and directory information as well as the database filename and may use OpenVMS logical names.

The **Connection name** should be generated automatically from the **Database name**.

Add Connection	? 🛛
	o connect to the selected data source or oose a different data source and/or
Oracle Rdb Databa	se (RdbClient)
S <u>e</u> rver (name: <por< td=""><td>t or service&gt;):</td></por<>	t or service>):
my_sqs_server:GE	NERIC
Use <u>T</u> hin Server	
<u>U</u> ser name:	rdbnet_user
Password:	•••••
	✓ <u>S</u> ave my password
<u>D</u> atabase name:	
disk1:[MY_TEST_D	ATABASES]MF_PERSONNEL
	Ad <u>v</u> anced
Test Connection	OK Cancel

#### To connect to a database using a SQL/Services database service:

For **Server**, enter the node address and a valid SQL/Services database service that will be running on the selected node. Remember to separate the node address and service name with a colon (:).

In this example we assume that a database service called MF\_PERS is running on the selected node.

The node address must be valid TCP/IP node specification.

For **User name**, enter a valid username for the database you will select.

For **Password**, enter the password for that user

To save the password for future sessions, check the **Save** password box.

Leave the **Database name** blank as this is not required when a database service is used.

The Connection name should be generated automatically from the Database name.

Add Connection	? 🛛
	connect to the selected data source or oose a different data source and/or se (RdbClient) Change
Server (name: <port my sgs_server:MF</port 	· · · · · · · · · · · · · · · · · · ·
Use Thin Server	
User name:	rdbnet_user
Password:	
	Save my password
Database name:	
	Advanced
Test Connection	OK Cancel

### **Step7:** Click Test connection.

Microso	ft Visual Studio	X
(į)	Test connection succeeded.	
	ОК	

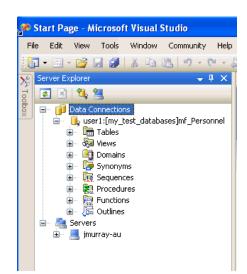
The test should succeed. Click OK.

If the test fails, it may be due to one or more of the following issues that you must address before proceeding with further steps:

- The database listener or server is not started.
- The database connectivity is not properly configured.
- You do not have the correct user name, or password.

**Step8:** In the Add Connection window, click OK.

**Step9:** In the Server Explorer, expand the database connection to show the contents of the database. You should see Tables, Views, Procedures, Functions, Packages, Synonyms, Sequences, and so on.



Show Navigation

# Chapter 4 Building a Simple .NET Application Using ORDP.NET

This chapter contains:

- <u>Creating a New Project</u>
- Adding a Reference
- <u>Adding Namespace Directives</u>
- Designing the User Interface
- <u>Writing the Connection Code</u>
- <u>Compiling and Running the Application</u>
- Error Handling

## 4.1 Creating a New Project

Visual Studio groups all development code that you create into containers known as projects. Simpler projects often contain only one file. In this section, you will learn how to create a new development project.

The application you build in this chapter serves as a starting point for work in subsequent chapters, so it is important to follow the order of this guide.

Note:

When necessary, instructions specify **Visual C#** or **Visual Basic**.

#### To start a new project:

**Step1:** Start Visual Studio.

Open the **Start menu**, select **All Programs**, and then select **Microsoft Visual Studio** 2005.



The Microsoft Visual Studio IDE environment appears.

**Step2:** In the Start Page, under the Recent Projects heading, click **Create: Project**.

Alternatively, from the File menu, select New, and then select Project.

A New Project dialog box appears.

**Step3:** In Project Types, select the type of project you are creating:

#### Visual C#:

Visual C#: Windows

### **Visual Basic:**

Other Languages: Visual Basic: Windows

- **Step4:** In Templates, select **Windows Forms Application**.
- **Step5:** In the Name field, enter the appropriate name.

### Visual C#:

PERS Connect CS

### **Visual Basic:**

PERS Connect VB

The abbreviation **cs** indicates C# projects and VB indicates Visual Basic projects.

**Step6:** In Location, enter the directory where you want to save the files.

For this guide, enter this directory C:\PERS Projects.

**Step7:** In Solution Name, the appropriate name, **PERS\_Connect\_CS** or **PERS\_Connect\_VB** should appear.

A solution can contain several projects; when it contains only one project, you can use the same name for both.

#### Step8: Check Create directory for solution. Step9: Click OK.

New Project				? 🛛
Project types:		Templates:		
Visual C#     Visual C#     Visual Sum     Visual Stud     Visual Stud	; ; fypes Deployment Y	Visual Studio installed tem,         Image: Studio Studio         Image: Studio Studio Studio         Image: Studio Studio Studio         Image: Studio Studio Studio Studio         Image: Studio St	plates [편] Class Library (편] Web Control Library (편] Windows Service (편] Crystal Reports Application	
A project for creati	ng an application with	n a Windows user interface		
Name:	PERS_Connect_CS			]
Location:	C:\PERS_		×	Browse
Solution Name:	PERS_Connect_CS		Create directory for solution	
			ОК	Cancel

The project is created.

The main window now displays a new title, either **PERS\_Connect\_CS** - **Microsoft Visual Studio** or **PERS\_Connect\_CS** - **Microsoft Visual Studio**, depending on the language, and contains Form1 shown below.

It is important to remember that many projects automatically name the first form Form1. This is the name of the form control. Do not confuse this with the actual name given to the code file, which is typically <code>Form1.cs</code> or <code>Form1.vb</code>.

Both Form1 and Form1.xx can be renamed. For the purposes of this guide, we will rename Form1.xx several times.

Form1.cs [Design] Start Page	▼ X
Form1.cs [Design] Start Page	• ×

# 4.2 Adding a Reference

This section shows you how to add a reference to the Oracle.DataAccess.Rdb.dll file, which contains the data provider, Oracle Rdb Data Provider for .NET.

### To add a reference:

**Step1:** From the **Project** menu, select **Add Reference**.

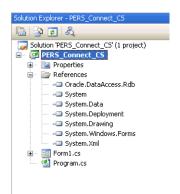
Proj	ect Build Debug Data Format
1	Add Windows Form
1	Add User Control
1	Add Component
₹\$	Add Class Shift+Alt+C
-	Add New Item Ctrl+Shift+A
:::	Add Existing Item Shift+Alt+A
	Exclude From Project
ð	Show All Files
	Add Reference
	Add Web Reference
	Set as StartUp Project
	Project Dependencies
	Project Build Order
c	PERS_Connect_CS Properties

The Add Reference windows appears.

**Step2:** In the Add Reference window, under the .NET tab, select **Oracle.DataAccess.Rdb** Click OK.

ET COM Projects Br	owse Recen	t	
Component Name 🔺	Version	Runtime	Path
Microsoft.Windows.Desi	3.5.0.0	v2.0.50727	C:\Program Files\Micro.
Microsoft.Windows.Desi	3.5.0.0	v2.0.50727	C:\Program Files\Micro.
Microsoft_VsaVb	8.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
MSCOMCTL	10.0.450	v1.0.3705	C:\Program Files\Micro.
MSCOMCTL	10.0.450	v1.0.3705	C:\Program Files\Micro.
mscorlib	2.0.0.0	v2.0.50727	C:\WINDOWS\Microsof.
msdatasrc	7.0.3300.0	v1.0.3705	C:\Program Files\Micro.
msdatasrc	7.0.3300.0	v1.0.3705	C:\Program Files\Micro.
msdatasrc	7.0.3300.0	v1.0.3705	C:\Program Files\Micro.
Office	11.0.0.0	v1.1.4322	C:\Program Files\Micro.
Office	12.0.0.0	v1.1.4322	C:\Program Files\Micro.
Oracl DataAccess.Rdb	7.3.1.0	v2.0.50727	C:\Program Files\Oracl.
Oracle Rdb.VsDevTools	1.0.0.0	v2.0.50727	C:\Program Files\Oracl.
PresentationBuildTasks	3.0.0.0	v2.0.50727	C:\Program Files\Refer.
PresentationCore	3.0.0.0	v2.0.50727	C:\Program Files\Refer.
<			>

Note that the new reference appears in the Solution Explorer.



# 4.3 Adding Namespace Directives

You can add Oracle namespace directives that allow you to indicate an assembly's namespaces within the module. To do this, add C# using statements or Visual Basic Imports statements, at or near the top of a code file.

Note:

Adding a reference makes the namespace available within the application. Adding a namespace directive within the application code makes the namespace more visible and allows for additional scoping.

#### To add Oracle Rdb namespace directives:

**Step1:** With Form1 active, from the **View** menu select **Code**.

Alternatively, you can use the **F7** keyboard shortcut.

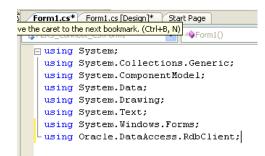
Vie	View Project		Build	Debug
F	Code			F7
-8	De	esigner	Sł	nift+F7
<b>e</b>	Se	erver Explo	orer Ctr	1+W, L
~~	So	olution Exp	lorer Ctr	l+w, s
<u> </u>	c	ass View	Ctr	l+₩, C

**Step2:** Add the following statements to the list of declarations depending on the language you are using.

#### Visual C#:

Add with other using statements, before the namespace.

using Oracle.DataAccess.RdbClient;



### **Visual Basic:**

Add to the top of the file, in the declarations section.

Imports Oracle.DataAccess.RdbClient

Form1.vb* Form1.vb [Design]	* Form1.cs*
(General)	💙 🎬 (Declarations)
Imports Oracle.Data	Access.RdbClient
⊡ Public Class Form1	
- End Class	

**Step3:** Save the changes by selecting **Save** from the **File** menu, or using the **Ctrl+S** keyboard shortcut.

# 4.4 Designing the User Interface

You can create a user interface by adding the toolbox controls to the design form. This interface accepts connection information from the user. To add toolbox controls:

**Step1:** From the **View** menu, select **Designer**.

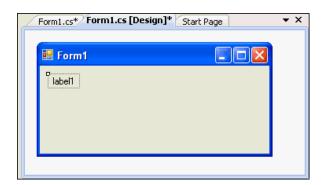
This opens Form1, in design view, if it is not already open.

You will toggle between Code and Designer a lot. The keyboard shortcuts are **F7** and **shift- F7** respectively.

- **Step2:** From the **View** menu, select **Toolbox**.
- **Step3:** In the Toolbox, expand **Common Controls**.

Tool	x →	×
•	All Windows For	^
Ξ (	Common Contr	
R.	Pointer	
ab	Button	
	CheckBox	
	CheckedListBox	
=	ComboBox	
	DateTimePicker	
Α	Label	
A	LinkLabel	
= 0	ListBox	
222	ListView	
#_	MaskedTextBox	
	MonthCalendar	

**Step4:** In the Toolbox, select **Label**, and drag it onto the Form1.



**Step5:** On Form1, right-click **label1**.

🔡 Form'	]	
label1		
	View Code	
<b>G</b>	Bring to Front	
	Send to Back	
卓	Align to Grid	
8	Lock Controls	
	Select 'Form1'	
*	Cut	
	Сору	
<b>11</b>	Paste	
×	Delete	
2	Properties	

**Step6:** From the menu, select **Properties**, if the Properties Window is not already visible.

The Properties Window appears.

**Step7:** In the Properties Window, change the Text property from **label1** to **User ID**.

Pr	operties	<b>→</b> ₽ ×			
la	label1 System.Windows.Fr 🗸				
	2↓ 🗉 🖋 🖻				
	Tag	~			
	Text	User ID 🔽			
	TextAlign	TopLeft			
	UseCompa	False			
	UseMnema	True			
	UseWaitCu	False 📃			
	Visible	True 💌			
Text The text associated with the control.					

**Step8:** Repeat steps <u>4</u> through <u>7</u> three times, placing three more labels on Form 1 and changing their text properties to **Password**, **Server** and **Database**.

🔜 Form1				
l, ar ID	Password	Server	Database	
ů				

**Step9:** In the Toolbox, select **TextBox**, and drag it onto the Form1, under the User ID label.

🔜 Form1				
User ID	Password	Server	Database	
k				

**Step10:** In the Properties Window, change the Name property to userID.

Pr	operties	→ ¶ ×			
te	textBox1 System.Window -				
0	A I	9 6			
Ŧ	(Applicatio	~			
Ð	(DataBindi				
	(Name)	userID			
	AcceptsRe	False			
	AcceptsTa	False			
	Accessible				
	Accessible	~			

**Step11:** Repeat steps <u>9</u> and <u>10</u> three times, positioning three more text boxes under the existing labels, and setting the Name property to **password**, **server** and **dataSource**.

🖶 Form1				
User ID	Password	Server	Database	]

**Step12:** Select the text box under the Password label. In the Properties Window, scroll to the PasswordChar property and set it to an asterisk (\*).

This masks the password during entry.

Pr	operties	→ ‡ ×
pa	assword Syste	m.Winc 👻
0	. <u></u> <b>2</b> ↓ 💷 🥠	
Ŧ	MaximumSize	0,0 🔺
	MaxLength	32767
Ð	MinimumSize	0, 0
	Modifiers	Privab
	Multiline	False 💻
	PasswordChar	*
	ReadOnly	False 🔽

**Step13:** From the Toolbox, select **Button** and drag it onto Form1.

In the Properties Window, change the Text property of the button from **button1** to **Connect**, and change the **Name** property to **connect**.

🔡 Form1				
User ID	Password	Server	Database	
			Connect	

# Step14: Save.

**Step15:** Close the Toolbox

Show Navigation

# 4.5 Writing the Connection Code

Now we write the code that takes the information provided to the user interface and connects to the database.

To connect to the database, you must create a connection object.

### To write code that connects to the database:

These steps enable your application to connect to the database based on data that the user enters into the Form1 control. See <u>Compiling and Running the</u> <u>Application</u>.

From the View menu, select Code.

**Step1:** Add the code indicated to instantiate a database connection string.

**Visual C#:** Add the class variable conn to the Form1 class right after the public Form1 () block with this code.

private RdbConnection conn = new RdbConnection();

Form1.cs* Form1.cs [Design]* Start Page	×
PERS_Connect_CS.Form1	*
<pre>using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Drawing; using System.Text; using System.Windows.Forms; using Oracle.Data&amp;ccess.RdbClient</pre>	
<pre>namespace PERS_Connect_CS {     public partial class Form1 : Form     {         </pre>	
<pre>public Form1() {     InitializeComponent();     private RdbConnection conn = new RdbConnection(); } </pre>	

**Visual Basic:** Add the conn class variable in the Form1 class declaration, using this code.

```
Public Class Form1
Dim conn As New RdbConnection
```

- **Step2:** Save your changes.
- **Step3:** Change to Designer view by clicking on the **View** menu and selecting **Designer**.
- **Step4:** Double-click the **Connect** button on Form1 to open the code window to the connect Click() method.

Insert the code indicated into the connect Click() method.

### Visual C#:

```
conn.ConnectionString = "User Id=" + userID.Text +
  ";pwd=" + password.Text +
  ";server=" + server.Text +
  ";database=" + database.Text +
```

```
";type=Thin;";
```

conn.Open();

### **Visual Basic:**

```
conn.ConnectionString = "User Id=" + userID.Text & _
    ";pwd=" + password.Text & _
    ";database=" + database.Text & _
    ";type=Thin;"
conn.Open()
```

Visual Basic Imports statements, at or near the top of a code file.

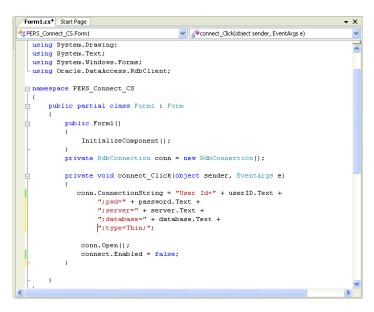
Before a connection can be opened, it must be built from user input for the User Id, pwd, server, and database. The Open() method makes the actual connection.

**Step5:** Set the Enabled attribute of the button to false by inserting the indicated code at the end of the connect Click() method.

This disables the Connect button, which is a good practice once a connection is successfully made.

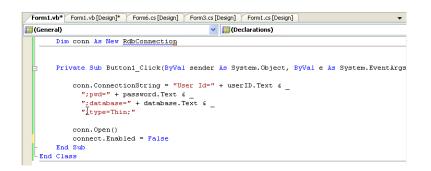
### Visual C#:

connect.Enabled = false;



### **Visual Basic:**

connect.Enabled = false



You have now finished writing an application that can connect to the Oracle Rdb database.

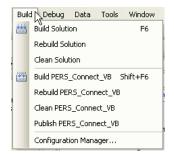
The following sections show how to use it.

## 4.6 Compiling and Running the Application

This section shows how to compile and run the application you created in the previous sections.

To compile and run the application:

Step1: From the Build menu, select Build Solution.



**Step2:** Ensure that there are no errors reported in the output window, available from the **View** menu.

The following graphics shows a typical output result.

- **Step3:** If there are any errors indicated, from the **View** menu, select **Error List** and fix the errors.
- **Step4:** From the **Debug** menu, select **Start Without Debugging** to run the application.

Deb	ug Data	Tools	Window	Community		
	Windows					
	Start Debugging F5					
≡⊳	Start Without Debugging Ctrl+F5					
	Attach to Process					
	Exceptions.		C	Ctrl+D, E		
<b>S</b> I	Step Into			F11		
Ç≣	Step Over	F10				
	Toggle Breakpoint F9					
	New Breakpoint					
Q.	Delete All B	reakpoint	s Ctrl+S	hift+F9		

**Step5:** In the Form1 application, enter the User ID, Password , Server, and Database.

### Click Connect.

Once the connection is opened, the Connect button is disabled. You have succeeded in implementing a connection to an Oracle Rdb Database.

🔜 Form1				
User ID Rdbnst_user	Password xxxxxxxxxxxx	Server SQSServer:GENERIC	Database user1:[dbs]mf_personnel Connect	

# 4.7 Error Handling

Applications must be able to handle run-time errors gracefully. For example, if you try to log in using an incorrect password, the application you developed so far cannot establish a connection to the database, and exits with the following unhandled exception

%RDB-E-AUTH\_FAIL: authentication failed for user

You must reselect Start Without Debugging to try this with a different password.

Error handling manages occurrences of conditions that change the normal flow of program execution. Oracle Rdb Data Provider for .NET contains a single classes for error handling and support:

• The RdbException class represents an exception that is thrown when the Oracle Rdb Data Provider for .NET encounters an error. Each RdbException object contains a description and an error number that describes the error or warning.

### 4.7.1 Using Try-Catch-Finally Block Structure

.NET languages use Try-Catch-Finally block structure for error handling. With this structure, the Try code is the main code, the goal that the application wants to accomplish. The Catch code catches errors of various types, as shown in the next two section. The Finally block comes last and always executes.

The Finally block frequently contains the Dispose method, which closes and disposes of the connection. Having the Dispose method in the Finally block ensures that the database connection is always closed after the Try-Catch-Finally block completes. Closing database connections after the application no longer requires database access is important for many reasons, especially data security.

Attempting to close a closed database connection does not cause an error. The attempt is irrelevant. Nonetheless, placing Dispose() in the Finally code block guarantees that the connection is closed.

The next section shows how to use Try-Catch-Finally block structure with general errors, and the section after that, with Oracle Rdb errors.

### 4.7.2 Handling General Errors

This section shows how to handle general errors using a Try-Catch-Finally block.

### To handle general errors:

**Step1:** Change the code of the connect\_Click() method in Form1 by adding an implementation of the Try-Catch-Finally syntax.

New code is in bold font.

### Visual C#:

```
private void connect_Click(object sender, EventArgs e)
  conn.ConnectionString = "User Id=" + userID.Text +
   ";pwd=" + password.Text +
   ";server=" + server.Text +
   ";database=" + database.Text +
   ";type=Thin;";conn.Open();
  try
  {
    conn.Open();
    connect.Enabled = false;
  }
  catch (Exception ex)
    MessageBox.Show(ex.Message.ToString());
  finally
  {
    conn.Dispose();
  }
}
```

Alternatively, you can use C# syntax that disposes of a connection when it goes out of scope, with the using keyword, as follows:

```
using (RdbConnection conn = new RdbConnection())
{
    conn.Open();
    // application code
    ...
}
```

### Visual Basic:

```
Try
  conn.Open()
  connect.Enabled = false
Catch ex As Exception
  MessageBox.Show(ex.Message.ToString())
Finally
  conn.Dispose()
End Try
```

Step2: From the Build menu, select Rebuild Solution.

Ensure that there are no errors.

- **Step3:** From the Debug menu, select Start Without Debugging.
- **Step4:** Run the application again, as described in section <u>Compiling and</u> <u>Running the Application</u>, and attempt to connect using an incorrect password.

This time, the application catches the error and displays it in a pop-up window, %RDB-E-AUTH\_FAIL: authentication failed for user.

Failed to connect : in <rdbjdbcsrv:connect failure=""> %RDB-E-AUTH_FAIL, authentication failed for user murray @rdb.Client.<idle>. OK</idle></rdbjdbcsrv:connect>

Show Navigation

# Chapter 5 Retrieving and Updating with ORDP.NET

This chapter contains:

- Using the Command Object
- Retrieving Data: a Simple Query
- <u>Retrieving Data: Bind Variables</u>
- <u>Retrieving Data: Multiple Values</u>
- Using the DataSet Class with Oracle Rdb Data Provider for .NET
- Enabling Updates to the Database

# 5.1 Using the Command Object

To view, edit, insert or delete data in a database, you must encapsulate a request in an RdbCommand object specifying a SQL command, stored procedure, or table name. The RdbCommand object creates the request, sends it to the database, and returns the result.

### To use the command object:

**Step1:** Make two copies of Form1.xx, from application PERS\_Connect\_xx in <u>Building a Simple .NET Application Using ORDP.NET</u>. To make copies, see the instructions in <u>Copying a Form</u>.

Name the copies Form2.cs or Form2.vb and Form3.cs or Form3.vb. The first copy is for the first part of the chapter, and the second copy for the second part of the chapter

Step2: Open Form2.cs or Form2.vb.

Note that the actual form in the designer still says Form1, as you renamed code files but not the actual form controls within the project.

**Step3:** Create a string that represents the SQL query and add to the body of the try statement.

The new code is in bold typeface.

### Visual C#:

### **Visual Basic:**

```
Try

conn.Open()

connect.Enabled = False

Dim sql As String = "select department_name from departments" &

"where department_code = 'ENG'"
```

**Step4:** Use the new sql variable to create the **RdbCommand** object, and set the **CommandType** property to run a text command.

#### Visual C#:

```
try
{
    conn.Open();
    connect.Enabled = false;
    // SQL Statement
    string sql = "select department_name from departments"
        + " where department_code = 'ENG'";
    RdbCommand cmd = new RdbCommand(sql, conn);
    cmd.CommandType = CommandType.Text;
}
```

#### **Visual Basic:**

```
Try
conn.Open()
connect.Enabled = False
Dim sql As String = "select department_name from departments" & _
    "where department_code = 'ENG'"
Dim cmd As New RdbCommand(sql, conn)
cmd.CommandType = CommandType.Text
```

Step5: Save your work.

# 5.2 Retrieving Data: a Simple Query

This section demonstrates retrieving data from the database.

The ExecuteReader() method of an RdbCommand object returns an RdbDataReader object, which can be accessed to display the result on the form. The application uses a **ListBox** to display the results.

#### To retrieve data:

**Step1:** Create an RdbDataReader object, by adding the code indicated to the bottom of the Try block of the connect Click() method.

This enables you to read the result of the query.

The new code is in bold typeface

### Visual C#:

```
try {
    conn.Open();
    connect.Enabled = false;
    // SQL Statement
    string sql = "select department_name from departments"
        + " where department_code = 'ENG'";
    RdbCommand cmd = new RdbCommand(sql, conn);
    cmd.CommandType = CommandType.Text;
    RdbDataReader dr = cmd.ExecuteReader();
    dr.Read();
}
```

#### **Visual Basic:**

Step2: Open Form1 in Design view. From the View menu, select Designer.Step3: From the View menu, select Toolbox.

- **Step4:** From the **Toolbox**, select a **Label** and drag it onto Form1.
- **Step5:** From the **View** menu, select **Properties Window**.
- **Step6:** In the **Properties** window, change the **Text** of the label to Department.
- **Step7:** From the Toolbox, under Window forms, select a **ListBox** and drag it onto Form1.
- **Step8:** In the **Properties** window, under Design, change the **Name** to departments.

🔜 Form1				
User ID	Password	Server	Database	
Department			Connect	
departments				

**Step9:** Add accessor type methods for retrieving data from the query result.

Double-click the **connect** button to edit the **connect\_click**() method, and add the code indicated to the bottom of the try block.

#### Visual C#:

departments.Items.Add(dr.GetString(0));

#### **Visual Basic:**

departments.Items.Add(dr.GetString(0))

Typed accessors, such as GetString, return native .NET data types and native Oracle Rdb data types. Zero-based ordinals passed to the accessors specify which column in the result set to return.

The try block should look like this now:

### Visual C#:

```
try
{
    conn.Open();
    connect.Enabled = false;
```

```
// SQL Statement
string sql = "select department_name from departments"
    + " where department_code = 'ENG'";
RdbCommand cmd = new RdbCommand(sql, conn);
cmd.CommandType = CommandType.Text;
RdbDataReader dr = cmd.ExecuteReader();
dr.Read();
departments.Items.Add(dr.GetString(0));
```

#### **Visual Basic:**

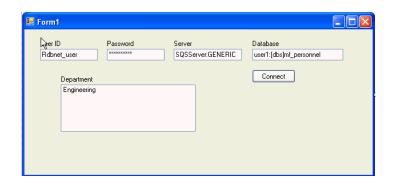
}

```
Try
conn.Open()
connect.Enabled = False
Dim sql As String = "select department_name from departments" & _
    "where department_code = 'ENG'"
Dim cmd As New RdbCommand(sql, conn)
cmd.CommandType = CommandType.Text
Dim dr As RdbDataReader = cmd.ExecuteReader()
dr.Read()
departments.Items.Add(dr.GetString(0))
```

**Step10:** Build and save the application.

Step11: Run the application. Enter the login server and database information.

After you connect, the departments list box shows Engineering, the correct name for department with code 'ENG' in the Departments table, as requested by the SELECT statement.



# **5.3 Retrieving Data: Bind Variables**

Bind variables are placeholders inside a SQL statement.

The following code shows a typical SELECT statement that does not use bind variables, with the value 10 specified in the WHERE clause of the statement.

SELECT department\_name FROM departments WHERE department\_code = 'ENG'

The following code replaces the numerical value with a bind variable :department\_id. A bind variable identifier always begins with a single colon ( :).

```
SELECT department_name FROM departments WHERE department_code =
:department_id
```

Note that bind variables can also be used with UPDATE, INSERT, and DELETE statements, and also with stored procedures. The following code illustrates how to use bind variables in an UPDATE statement:

```
UPDATE departments SET department_name = :department_name
    WHERE departname_code = : department_id
```

You can use the RdbParameter class to represent each bind variable in your .NET code. The RdbParameterCollection class contains the RdbParameter objects associated with the RdbCommand object for each statement. The RdbCommand class passes your SQL statement to the database and returns the results to your application.

### To retrieve data using bind variables:

- Step1: Move the ListBox named Departments to the right.
- **Step2:** From the **View** menu, select **Toolbox**.
- **Step3:** From the Toolbox, select a **TextBox** and drag it onto Form1, under the label that says **Department**.
- **Step4:** From the **View** menu, select **Properties** Window.
- **Step5:** In the Properties window, change **Name** to departmentID.

**Step6:** Change the SELECT statement to use the bind variable by adding the code indicated to the Try block of the connect Click() method.

Changed or new code is in bold typeface.

Visual C#:

```
string sql = "select department_name from departments "+
    "where department_code = :department_id";
RdbCommand cmd = new RdbCommand(sql, conn);
cmd.CommandType = CommandType.Text;
RdbParameter p_department_id = new RdbParameter("department_id",
DbType.String);
p_department_id.Value = departmentID.Text;
cmd.Parameters.Add(p_department_id);
RdbDataReader dr = cmd.ExecuteReader();
dr.Read();
departments.Items.Add(dr.GetString(0));
Visual Basic:
Dim sql As String = "select department_name from departments " & _
    "where department_code = :department_id"
Dim cmd As RdbCommand = New RdbCommand(sql, conn)
```

```
RdbParameter("department_id", DbType.String)
p_department_id.Value = departmentID.Text
cmd.Parameters.Add(p_department_id)
Dim dr As RdbDataReader = cmd.ExecuteReader()
dr.Read()
departments.Items.Add(dr.GetString(0))
For this code, the parameter object sets the DbType property, but there is no
```

need to set the Direction property because it uses the default value, Input. There is no need to set the Size property because the object is an input parameter, and the data provider can determine the size from the value.

**Step7:** Save and run the application.

cmd.CommandType = CommandType.Text

Dim p\_department\_id as RdbParameter = new

**Step8:** Enter the login information, and a typical department code, such as "MNFG", from the MF\_PERSONNEL database.

### Step9: Click Connect.

The application returns the name of the department that corresponds to the department ID.

🖶 Form1				
User ID Rdbnet_user	Password ******	Server SQSServer:GENERIC	Database user1:[dbs]mf_Personnel	
Department MNFG	Manufacturing		Connect	

# 5.4 Retrieving Data: Multiple Values

You frequently need to retrieve more than just one value from the database. A DataReader object can retrieve values for multiple columns and multiple rows. Consider the multiple column, multiple row query in the following example:

```
SELECT department_code, department_name, manager_id FROM
departments
WHERE department_code < 'ENG'</pre>
```

Processing multiple rows from the DataReader object requires a looping construct. Also, a control that can display multiple rows is useful. Because the RdbDataReader object is a forward-only, read-only cursor, it cannot be bound to an updatable or backward scrollable control such as Windows Forms DataGrid control. An RdbDataReader object is, however, compatible with a ListBox control.

## To retrieve multiple values:

**Step1:** In the try block of the connect\_Click() method, change the SQL query to return a multiple row result set and add a while loop to enclose the read method that displays the department names.

## Visual C#:

```
try
{
...
string sql = "select department_name from departments " +
   "where department_code < :department_id";
...
while (dr.Read())
{
   departments.Items.Add(dr.GetString(0));
  }
}</pre>
```

## **Visual Basic:**

**Step2:** Save and run the application.

**Step3:** Enter the login information and enter "**ENG**" for the department.

### Step4: Click Connect.

The application returns the name of the departments that correspond to the query.

💀 Form1				
User ID Rdbnet_user	Password	Server SQSServer:GENERIC	Database user1:[dbs]mf_Personnel	
Department ENG	Corporate Administrat Electronics Engineeri Large Systems Engin Mechanical Engineer	ng eering	Connect	

# 5.5 Using the DataSet Class with Oracle Rdb Data Provider for .NET

The DataSet class provides a memory-resident copy of database data. It consists of one or more tables that store relational or XML data. Unlike an RdbDataReader object, a DataSet is updatable and backward scrollable.

## To use the DataSet class:

- **Step1:** If you have not done so before, make another copy of the Form1 that you completed in Chapter 3, and name it Form3.vb or .cs, as described in Copying a Form
- **Step2:** If Form1.xx does not appear in the Solution Explorer, from the Project menu, select **Show All Files**.
- **Step3:** From the View menu, select **Designer view**.
- **Step4:** From the View menu, select **Toolbox**.
- **Step5:** From the Toolbox, select a **DataGridView** and drag it onto Form1.

**Step6:** From the View menu, select **Properties Window**.

**Step7:** In the Properties window, change the Name of the data grid view to departments.

Form1			
User ID	Password	Server	Database
			Connect
			Lonnect

**Step8:** From the View menu, select **Code.** 

**Step9:** Immediately after the conn declaration in the code, add variable declarations to the class variables, as indicated.

### Visual C#:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
    private RdbConnection conn = new RdbConnection();
    private RdbCommand cmd;
    private RdbDataAdapter da;
    private RdbCommandBuilder cb;
    private DataSet ds;
...
```

Visual Basic:

Public Class Form1 Dim conn As New RdbConnection Private cmd As RdbCommand Private da As RdbDataAdapter Private cb As RdbCommandBuilder Private ds As DataSet

Step10: Within the connect Click() method try block, add code to:

- Query the database
- Fill the DataSet with the result of the command query
- Bind the DataSet to the data grid (departments)

Visual C#:

```
conn.Open();
connect.Enabled = false;
string sql = "select * from departments where "+
   "department_id < 'ENG'";
cmd = new RdbCommand(sql, conn);
cmd.CommandType = CommandType.Text;</pre>
da = new RdbDataAdapter(cmd);
cb = new RdbCommandBuilder(da);
ds = new DataSet();
da.Fill(ds);
departments.DataSource = ds.Tables[0];
Visual Basic:
conn.Open()
connect.Enabled = False
Dim sql As String = "select * from departments where " + _
 "department_id < 'ENG'"
cmd = New RdbCommand(sql, conn)
cmd.CommandType = CommandType.Text
da = New RdbDataAdapter(cmd)
cb = New RdbCommandBuilder(da)
ds = New DataSet()
da.Fill(ds)
departments.DataSource = ds.Tables(0)
```

# Step11: Build and save the application.Step12: Run the application, entering the login and data source.

After you successfully connect to the database, the data grid is populated with the results of the query.

User ID Rdbnet_user		Passw ******			rver QSServer:GENERIC	Database user1:[dbs]mf_Personnel	
	DEPARTME	VT	DEPARTMENT		MANAGER_ID	Connect	
•	ADMN		Corporate Admi		00225		
	ELEL		Electronics Eng	gin	00188		
	ELGS		Large Systems	; E	00369		
	ELMC		Mechanical Eng	gi	00190		

# 5.6 Enabling Updates to the Database

At this point, the DataSet contains a client copy of the database data. In this section, you will add a button that enables client data changes to be saved back to the database. The following section will show you how to test updating, inserting, and deleting the data.

To enable saving data from the DataSet to the database:

- **Step1:** From the Toolbox, drag and drop a **Button** onto **Form1**.
- **Step2:** In the Properties window, change the Name of the button to save.

Change the Text property to Save.

**Step3:** At the top of the Properties Window, click Events (the lightning bolt). In the list of events, select the click event. In the second column, enter the event name, save Click.

Form1			
User ID	Password	Server	Database
			Connect
			Save

**Step4:** From the View menu, select **Code**.

**Step5:** Add code that updates the data, to the body of the save\_Click() method, as indicated.

Visual C#:

da.Update(ds.Tables[0]);

Visual Basic:

da.Update(ds.Tables(0))

You may see some errors show up in the Error List. These will disappear after you add the code in the next step.

Step6: Within the Form() method or Form1\_Load method, add the code
indicated.

```
Visual C#:
```

```
public Form1()
{
    InitializeComponent();
    save.Enabled = false;
}
```

Visual Basic:

```
Private Sub Form1_Load(ByVal sender As System.Object, & _
ByVal e As System.EventArgs) Handles MyBase.Load
save.Enabled = false
```

**Step7:** Within the connect\_Click() method try block, add code to enable the Save button as indicated:

Visual C#:

```
conn.Open();
departments.DataSource = ds.Tables[0];
save.Enabled = true;
Visual Basic:
```

```
conn.Open()
...
departments.DataSource = ds.Tables(0)
save.Enabled = True
```

**Step8:** Remove the conn.Dispose() call from the finally block in the connect Click() method.

**Note:** In the previous code used in this example, this method was necessary to dispose or close the connection. However, with these changes to the code, it is necessary to keep the connection open after the query result returns, so that data changes made by the end user are propagated to the database. A general override call, components.Dispose(), is already part of the definition of Form1.

**Step9:** Build and save the application.

Step10: Run the application, entering the login and data source.

After you successfully connect to the database, the data grid is populated with the results of the query.

er ID		Passw			erver	Database	
dbnet	_user	XXXXXXXX	***		QSServer:GENERIC	user1:[dbs]mf_Personnel	
	DEPARTMEN	T	DEPARTMENT		MANAGER_ID	Connect	
•	ADMN		Corporate Ad	mi	00225		
	ELEL	Electronics Eng Large Systems		gin	00188		
	ELGS			s E	00369		
	ELMC		Mechanical En	gi	00190	Save	

Show Navigation

# Chapter 6 ORDP.NET and Visual Studio Wizards

In this section you will use the Visual Studio integrated development environment (IDE), along with the standard Visual Studio wizards to automatically generate the code to access data from your Rdb database

This chapter contains:

- Create a New application
- Add connection to RDB database in MS server explorer
- <u>Create a datasource for the employees table</u>
- <u>Design the windows application</u>

# 6.1 Create a New application

Create a new windows application using the **File**  $\rightarrow$ **New**  $\rightarrow$ **project** menu. Call it empapp.

See Creating a New Project for the steps to create a new project.

# 6.2 Add connection to RDB database in MS server explorer

See <u>Connecting to the Oracle Rdb Database</u> for steps to connect to your database.

# 6.3 Create a datasource for the employees table

### To create a new Data Source:

**Step1:** Using the top-level **menu Data** → **Add New** Data Source.

🏁 EmpApp - Microso	oft Visual St	tudio					
File Edit View Pro	oject Build	Debug D	Data Tools	Test V	Vindow	Commur	nity Help
i 🛅 🕶 🔛 🕶 🔛 i	🗗 🐰 🖬	B 9	🚺 Show Data	Sources	Shift+/	Alt+D	<ul> <li>Any CPU</li> </ul>
		a in F <mark>l</mark>	👌 🛛 Add New 🛛	ata Sourc	:e		* @ @ @
		×	New Query	y			
Server Explorer			Close Conr	nection			
🖉 🗐 Data Conne			Modify Cor	nnection			
🖃 🕂 📑 user 1: [r	nurray]mf_per	sonne					

- Step2: In the wizard, keep the default selection for "Database." Click Next
- **Step3:** From the list of connection, select the connection to the sample that was created that was created earlier, select the option to include the sensitive data in connection string. Click **Next**
- **Step4:** Check to save the connection string to the Application Configuration File Click **Next**

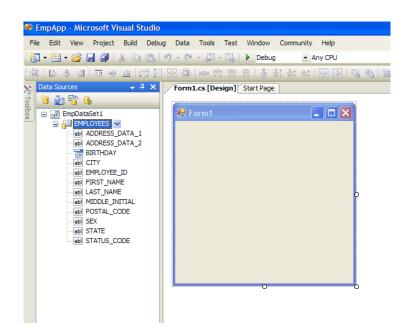
Data Source Configuration Wizard	? 🛛
Choose Your Data Connection	
Which data connection should your application use to connect to the data	base?
user1:[murray]mf_personnel	New Connection
This connection string appears to contain sensitive data (for example, a password), wil connect to the database. However, storing sensitive data in the connection string can you want to include this sensitive data in the connection string?	
O No, exclude sensitive data from the connection string. I will set this information	n in my application code.
Yes, include sensitive data in the connection string.	
+ Connection string	
< Previous Next > Fin	ish Cancel

**Step5:** A tree of available table will be displayed, select the employees table from the list of tables by checking the box next to it. Click **finish** 

Data Source Configuration Wizard	? 🛛
Choose Your Database Objects	
Which database objects do you want in your dataset?	
DataSet name:	
EmpDataSet1	
< Previous Next > Finish C	ancel

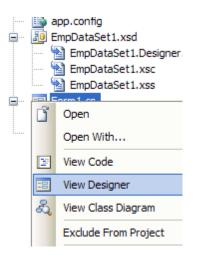
**Step6:** A data source for employees will be added in the "Data Sources" window.

You can make the data source window visible using the top level menu **Data** → **Show Data sources** 



# 6.4 Design the windows application

**Step1:** Ensure that **From1.c**s is open in **design** mode.



**Step2:** Drag and Drop the **Employees** data source from the data sources window onto the form.

	<sup>F</sup> orm'										
M	4	0	of {0}	► M	🕆 🗙						
		ADDRESS_	DATA_	ADDRE	SS_DATA_	BIRTHDAY	CITY	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MIDDLE_INITI
*											
<											>

**Step3:** Run the windows application

🖻 empDataSet1 😽 eMPLOYEESBindingSource 🔯 eMPLOYEESTableAdapter 💡 eMPLOYEESBindingNavigator

1	🖣 📔 1 of 102 📔 🕨 🗍 🕂 🛟	< 🚽					
	ADDRESS_DATA_ ADDRESS_DATA	BIRTHDAY	CITY	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MIDDLE_IN
•	146 Pamell Place	3/28/1947	Chocorua	00164	Alvin	Toliver	A
	120 Tenby Dr	5/15/1954	Chocorua	00165	Teny	Smith	D
	19 Union Square	3/20/1954	Boscawen	00166	Rick	Dietrich	
	143 Pine St	3/5/1937	Marlow	00167	Janet	Kilpatrick	
	87 West Rd	10/23/1932	Meadows	00168	Noman	Nash	
	51 Maple St	8/13/1938	Etna	00169	Susan	Gray	0
	140 Searles Rd	6/3/1957	Jefferson	00170	Brian	Wood	
	67 Underhill St	1/29/1932	Sandown	00171	Aruwa	D'Amico	

#### Show Navigation

# Chapter 7 ORDP.NET and drag and drop

In this section you will use the Visual Studio integrated development environment (IDE), along with drag and drop to automatically generate the code to access data from your Rdb database.

This chapter contains:

- <u>Create a New application</u>
- Add connection to RDB database in MS server explorer
- <u>Add a new Typed DataSet</u>
- Designing the windows application
- Running the windows application

# 7.1 Create a New application

Create a new windows application using the **File**  $\rightarrow$ **New**  $\rightarrow$ **project** menu. Call it **empapp**.

See <u>Creating a New Project</u> for the steps to create a new project.

# 7.2 Add connection to RDB database in MS server explorer

See <u>Connecting to the Oracle Rdb Database</u> for steps to connect to your database if you do not already have a connection.

# 7.3 Add a new Typed DataSet

### To add a new Typed DataSet to your project:

**Step1:** Run the windows application

Using the Microsoft Solution Explorer, right click on the **EmpApp**, and select **Add** → **New Item** 

<b>→</b> X	Solution	Explo	orer - EmpApp	<b>→</b> # X
		) 4		_
	😡 Sol	ject)		
			Build	
	<b>.</b>		Rebuild	
			Clean	
			Publish	
			Run Code Ana	ysis
New Item			Add	•
Evicting Item			Add Reference	

Step2: Select DataSet. Click Add. The Microsoft Dataset Designer will be opened.

Add New Item -	ЕтрАрр			? 🛛
Templates:				
Visual Studio i	installed templat	es		^
Class Windows For Component C ML File MTML Page Bitmap File Crystal Repo Installer Class Windows Scri Resources Fil Schout Box	m Class rt s ipt Host	Litterface User Control SQL Database XML Schema Cursor File Corsor File Corsor File Script File Settings File Settings File Debugger Visualizer	Code File Custom Control Custom Configuration File Custom File Custom Custom Custom Custom File Custom File Custom Custom Custom File Custom Custom Custom File Custom C	
A DataSet for usin	g data in your appli	cation		
Name:	DataSet1.xsd			
			Add	Cancel

From the Microsoft Server Explorer, drag and drop "Employees" table onto the open designer.

This will create a datasource and an associated **EMPLOYEESTableAdapter** that can you use in your application to access the employees table.

Ľ.	EMPLOYEES 🔝	
	ADDRESS_DATA_1	
	ADDRESS_DATA_2	
	BIRTHDAY	I
	CITY	
	EMPLOYEE_ID	I
8	FIRST_NAME	I
	LAST_NAME	
8	MIDDLE_INITIAL	I
8	POSTAL_CODE	
8	SEX	I
	STATE	I
8	STATUS_CODE	
10	EMPLOYEESTableAdapter 🔝	
SQL	Fill,GetData ()	1

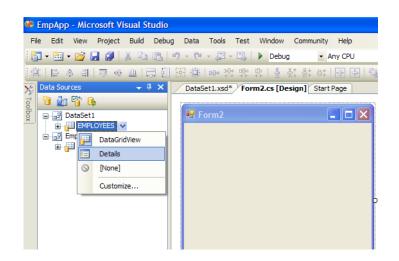
**Step3:** Ensure the password is added to the connection string. Select **EMPLOYEESTableAdapter**, and right click to view its properties. Append the password in the below format **Password=yourpassword** 

Pr	operties	
EI	IPLOYEESTableAdapter TableAdapter	
•	2↓   □	
Ξ	Code Generation	
	BaseClass	System.ComponentModel.Componer
	ConnectionModifier	Internal
	Modifier	Public
	Name	EMPLOYEESTableAdapter
Ξ	Data	
Ξ	Connection	ConnectionString (Settings)
	Name	ConnectionString (Settings)
	Modifier	Internal
	ConnectionString	server=tezuka.us.oracle.com;ı
	DeleteCommand	(None)

# 7.4 Designing the windows application

## To design the application:

- **Step1:** Ensure the Form1.cs is open in design mode.
- **Step2:** Using the Microsoft data sources window, use the smart menu on the Employees data source, and select **Details**



**Step3:** Drag and drop the Employees data source from the data source window onto the form.

🔜 Form2	
. <b> </b> 4	🕨 🕅   🕂 🗙 🔚
ADDRESS DATA 1:	
ADDRESS DATA 2:	
BIRTHDAY:	Tuesday , October 21, 2008
CITY:	
EMPLOYEE ID:	
FIRST NAME:	
LAST NAME:	
MIDDLE INITIAL:	
POSTAL CODE:	
SEX:	
STATE:	
STATUS CODE:	

A prototype form to access the employees data will be created for you.

# 7.5 Running the windows application

Run the application using the top-level menu. You can insert, delete rows and change the existing data.

4 7 of 10	2   🕨 🕅   🕂 🗙 🔚	
ADDRESS DATA 1:	140 Searles Rd.	
ADDRESS DATA 2:		
BIRTHDAY:	Monday , June 03, 1957 💌	
CITY:	Jefferson	
EMPLOYEE ID:	00170	
FIRST NAME:	Brian	
LAST NAME:	Wood	
MIDDLE INITIAL:		
POSTAL CODE:	03583	
SEX:	M	
STATE:	NH	
STATUS CODE:	1	

Show Navigation

# Chapter 8 Configuration wizard and generating queries

Wizards make it easy to perform many operations within Visual studio. The Table Adapter Query Configuration Wizard simplifies the creation of queries you may execute on your Rdb database.

This chapter contains:

- Using the Query Configuration Wizard
- Using the Query Builder
- <u>Methods Generation</u>

# 8.1 Using the Query Configuration Wizard

### To use the Query Configuration Wizard:

**Step1:** Right click dataset and click **add→Query** 

I.	EMPLOYEES			<b></b>	
	ADDRESS_DATA_1		Add 🕨		Query
	ADDRESS_DATA_2		Configure	ξ¤.	Relation
	BIRTHDAY	v	Cut	-	Key
	CITY	ŝ	Cut	9	
	EMPLOYEE_ID		Сору		Column
P	FIRST_NAME	12	Paste	-	

The Table Adapter Query Configuration Wizard will be shown.

TableAdapter Query Configuration Wizard
Choose a Command Type TableAdapter query uses SQL statements or a stored procedure.
How should the TableAdapter query access the database?
• Use SQL statements
Specify a SELECT statement to load data.
O Create new stored procedure
Specify a SELECT statement, and the wizard will generate a new stored procedure to select records.
O Use existing stored procedure
Choose an existing stored procedure.
< Previous Next > Finish Cancel

### Step2: Select Use SQL Statement

Clicking **Next** will bring up the query Type window. This window allows you to choose from several types of standard SQL statements.

TableAdapter Query Configuration Wizard	? 🔀
Choose a Query Type Choose the type of query to be generated	- San and a san a sa Tan a san
What type of SQL query would you like to use?         • SELECT which returns rows         Returns one or many rows or columns.         • SELECT which returns a single value         Returns a single value (for example, Sum, Count, or any other aggregate function).         • UPDATE         Changes existing data in a table.         • DELETE         Removes rows from a table.         • INSERT         Adds a new row to a table.	
< Previous Next > Finish	Cancel

## **Step3:** Select the query type **SELECT which returns rows.**

**Step4:** Click **Next** to bring up the query specification window. This is where you may enter a SQL query statement.. Use the ":" character pre-pended to parameter names to specify parameters you wish to use.

TableAdapter Query Configuration Wizard	? 🗙
Specify a SQL SELECT statement The SELECT statement will be used by the query.	- The second sec
Type your SQL statement or use the Query Builder to construct it. What data should be loaded in What data should the table load?	nto the table?
SELECT CITY, FIRST_NAME FROM EMPLOYEES WHERE EMPLOYEE_ID=@EMPLOYEE_ID	×
Q	uery Builder
< Previous Next > Finish	Cancel

For example to return details of an employee giving the **employee\_id**:

SELECT CITY, FIRST\_NAME FROM EMPLOYEES WHERE EMPLOYEE\_ID=:EMPLOYEE\_ID

**Step5:** Click **Finish**. We have just added a select functionality to the typed dataset using **TableAdapter** without writing a single line of code.

Repeat the above steps to produce other statements such as  $\ensuremath{\textbf{UPDATE}}$  and  $\ensuremath{\textbf{DELETE}}$  .

# 8.2 Using the Query Builder

Visual Studio provides the **Query Builder** option within the configuration Wizard that allows quick and easy creation of syntactically correct SQL queries on your datasets.

## Using Query Builder to generate queries:

Step1: Click Query Builder to bring up the builder window .

Add Table	?×
Tables Views	
BIGTABLE CANDIDATES CBM_RESULT COLLEGES DATE_TIME	
DEGREES DEPARTMENTS DOI_ORDER_ITEM DYP_IN_PROCESS_PALLET DZR_DORMANT_IN_PROCESS_REEL EMP	
EMPINFO EMPLOYEES FLD65	~
Refresh Add Cla	ose

**Step2:** Select **EMPLOYEE** table and click **ADD**. This will bring up a window allowing you to choose columns for your query .

Qı	uery B	uilder						?	X
		EMPLOYEES     (All Columns)     ADDRESS_DATA     ADDRESS_DATA     BIRTHDAY     CITY	_1						
	<							>	~
		Column	Alias	Table	Output	Sort Type	Sort Order	Filter	^
	•								
	<							>	~
	SELECT	EMPLOYEES							
	14 4	0 of0	▶ <b>▶</b> ] ▶≈						ĺ
(	Execut	e Query				(	ок	Cancel	

**Step3:** Click **"Close"** on ADD Table window to close this window.

**Step4:** Within the **EMPLOYEES** table check the columns name and Parameters to the query (Filter) as shown below.

uery Bi	uilder						?
	EMPLOYEES     ADDRESS_DATA     BIRTHDAY     CITY     EMPLOYEE_ID     FIRST_NAME						
							>
	Colu		Sort Type	Sort Order	Filter	Or	^
	CITY	<b>V</b>					
	FIRST_NAME	<b>V</b>					
•	EMPLOYEE_ID				= '@EMPLOY		
<							>
ELECT ROM /HERE	CITY, FIRST_NAME EMPLOYEES EMPLOYEE_ID =@E		-				
	0 of 0						)
Execute	e Query				ОК	Cano	:el

**Step5:** Click **Execute Query** to execute the generated query. The results of the operation will be displayed in the results window.

Query B	uilder						- [?	JX
	EMPLOYEES     ADDRESS_DATA     BIRTHDAY     CITY     EMPLOYEE_ID     FIRST_NAME						>	<
	Colu	Output Sor	t Type	Sort Order	Filter	Or		~
	CITY		c type	boreorder	T HOLE	- Orini	_	
	FIRST_NAME							
F	EMPLOYEE_ID				= 00164			
								~
<							>	
FROM								
	CITY	FIRST_NAME	EMPLO	YEE_ID				^
	Chocorua	Alvin	00164					<b>*</b>
*	NULL	NULL	NULL					~
M 4	1 of 1	▶ ▶ ▶ ▶ ■   (						
Execut	te Query				OK	: c	ancel	

**Step6:** Click "**OK**" to accept the query .

TableAdapter Query Configuration Wizard	? 🛛
Specify a SQL SELECT statement The SELECT statement will be used by the query.	Transfer Stransf
Type your SQL statement or use the Query Builder to construct it. What data should be loaded in What data should the table load?	to the table?
SELECT CITY, FIRST_NAME, EMPLOYEE_ID FROM EMPLOYEES WHERE EMPLOYEE_ID = @ EMPLOYEE_ID	K
	Jery Builder
< Previous Next > Finish	Cancel

**Step7:** Click "**Finish**". Your query is now ready to use. The Query Configuration Wizard will now display the Methods Generation window.

# 8.3 Methods Generation

Once you have generated a query, this query may be used within the TableAdapter . The next step is to choose what methods should be available within your TableAdapter.

How to choose methods:

TableAdapter	Query Configuration Wizard
0110000011100	hods to Generate dapter methods load and save data between your application and the
Which method	ds do you want to add to the TableAdapter? able
	of that takes a DataTable or DataSet as a parameter and executes the SQL statement or rocedure entered on the previous page.
Method name:	FillBy
🗹 Return a D	ataTable
	od that returns a new DataTable filled with the results of the SQL statement or SELECT stored ed on the previous page.
Method name:	GetDataBy_EMPID
	< Previous Next > Finish Cancel

Step1: Select the Fill A DataTable and Return a DataTable .

# Step2: Change the Return a DataTable method name to GetDataBy\_EMPID

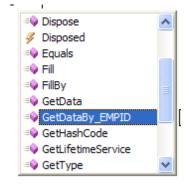
The final dataset in the designer show look like the following:

	ADDRESS_DATA_1	
	ADDRESS_DATA_2	
	BIRTHDAY	
	CITY	
	EMPLOYEE_ID	
3	FIRST_NAME	
	LAST_NAME	
3	MIDDLE_INITIAL	
3	POSTAL_CODE	
3	SEX	
	STATE	
3	STATUS_CODE	
5	EMPLOYEESTableAdapter	8
QL	Fill,GetData ()	
QL	FillBy,GetDataBy EMPID (@EMPLOYEE ID)	

**Step3:** To execute the above functionality in your code, instantiate the table adapter as follows:

```
DataSet1TableAdapters.EMPLOYEESTableAdapter adapt = new
DataSet1TableAdapters.EMPLOYEESTableAdapter();
```

This new select function can now also be seen in the TableAdapter intellisense.



Step4: The methods provide by the TableAdapter may be used within your code

// Get Employee details by employee\_id

dataGridView1.DataSource=adapt.GetDataBy\_EMPID(00164);



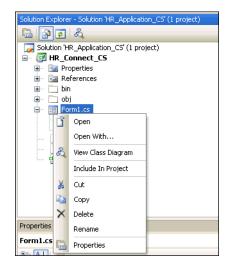
# **Chapter 9** Copying a Form

Because you will be using this application to learn about various aspects of application development with Oracle Rdb, you should make copies of your form for reuse.

### To create a copy of an existing form:

**Step1:** In the Solution Explorer, right-click on Form1.xx or any other file you need to copy. Select **Copy**.

If Form1.xx does not appear in the Solution Explorer, from the Project menu, select **Show All Files**.

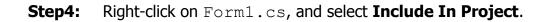


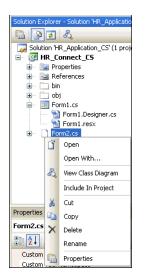
**Step2:** Right-click **HR\_Connect\_CS** or other project. Select **Paste**.

Solution Exp	olorer - HR_Connect_CS
	\$
	on 'HR_Application_CS' (1 pro;
	Build
Œ	Rebuild
±	Clean
Ē	Publish
	Add 🕨
	Add Reference
	Add Web Reference
ಪ್	View Class Diagram
	Set as StartUp Project
	Debug +
Proper 👗	Cut
HR C	Paste
	Remove
Poli	Rename
Poli	Unload Project
Pro 🛅	Properties

**Step3:** Right-click **Copy of Form1.cs**. Select **Rename**. Change the name of the form to Form2.cs.

Solution Explorer - Sol	ution 'HR_Application_CS' (					
Solution 'HR_Application_CS' (1 project)						
Image: Barrier Ba						
😥 🔤 Properties						
😥 🔤 References						
	i bin					
💼 🖳 obj						
Copy of Form1.cs						
Open						
	Open With					
- 🗋 HR_ 🖧	View Class Diagram					
	Include In Project					
*	Cut					
Сору						
×	Delete					
Properties	Rename					
Copy of Form1.	Properties					





Step5: Right-click on Form1.cs, and select Exclude From Project.

You can include and exclude forms from the project just by reversing these steps.

Note:

This process generally works smoothly. If you encounter a problem, try running Rebuild Solution from the Build menu.

Show Navigation

# Chapter 10 Oracle Rdb Entity Framework Provider

The Entity Framework is a new part of ADO.NET that allows you to build your applications against conceptual data models. It provides a greater level of abstraction and supports code that is independent of any particular relational database.

It provides an Entity Data Model (EDM) for defining data at the database and conceptual level and mapping between the two. It includes a nice set of tools that can be used to generate the EDM and corresponding objects that represent the database. This eliminates much of the required boilerplate data access code, and makes it a snap to create data-centric applications.

The chapter also provides some simple examples on how to carry out Model-First and Database-First application development using the Oracle Rdb Entity Framework Provider.

Model-first is the ability to start with a conceptual model and create the database from it. Database-first is the ability to start with the database and build your EDM from the database objects.

This chapter contains:

- Creating new Windows Form Application
- Create a new Entity
- <u>Creating Complex Types</u>
- Generating Database Schema from Model

# **10.1 Creating new Windows Form Application**

The first step is to create a new Windows Forms application:

**Step1:** In Visual Studio IDE, select **File→New→Project** from the main menu.

Choose the Windows Forms Application installed template. Click **OK**. The solution is created.

Step2: Right-click on Form1.cs and select Exclude From Project.

You can now proceed to generate a Entity Data Model from an existing database schema.

## **10.1.1 Adding an Entity Data Model**

### How to add an Entity Data Model:

Step1: In the Solution Explorer, right-click your application and select Add→ New Item. This opens Visual Studio installed templates screen.

Step2: From Visual Studio installed templates, select ADO.NET Entity Data Model. Click Add.

nstalled Templates	Sort by:	Default 💌 🧾 🧰		Search Installed Templates
Visual C# Items Code Data	C#	Class	Visual C# Items	Type: Visual C# Items A project item for creating an ADO.NET Entity
General Web	s <b>c</b> ≱	Interface	Visual ⊂# Items	Data Model.
Windows Forms WPF		Windows Form	Visual C# Items	
Reporting Workflow	The second se	User Control	Visual C# Items	
nline Templates	- 8	Component Class	Visual C# Items	
	•	User Control (WPF)	Visual C# Items	
	5	About Box	Visual C# Items	
	03	ADO.NET Entity Data Model	Visual C# Items	
	93	ADO.NET EntityObject Generator	Visual C# Items	
	<i>Q</i> 2	ADO.NET Self-Tracking Entity Generator	Visual C# Items	
		Application Configuration File	Visual C# Items	
	1	Application Manifest File	Visual ⊂# Items	
	C#	Assembly Information File	Visual C# Items	
	1	Bitmap File	Visual C# Items	
Jame: Model1.edm	<b>D</b>	Close Nicercen	Unual C# Boome	

You will now see the **Entity Data Model Wizard**. This wizard is used to generate the Entity Data Model from the database specified in the database connection string ( in next step).

Step3: Select the icon Generate from database. Click Next.

Entity Date	a Model Wizard	?×
þ	Choose Model Content <i>s</i>	
What she	ould the model contain?	
	品	
Generate databa	From Empty model	
	s the model from a database. Classes are generated from the model when the project This wizard also lets you specify the database connection and database objects to inc del.	
	< Previous Next > Finish Can	cel

**Step4:** You can now select the connection if you have made connection to the test database earlier or create a new connection:

Entity Data Model Wizard						
Choose Your Data Connection						
Which data connection should your application use to connect to the database?						
e_regtest_mf_personnelConnectionString (Settings)						
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?						
No, exclude sensitive data from the connection string. I will set it in my application code.						
Yes, include the sensitive data in the connection string.						
Entity connection string:						
metadata=res://*/Model2.csdl res://*/Model2.ssdl res://*/Model2.msl;provider=Oracle.DataAcce ss.RdbClient;provider connection string="server=localhost:1701;database=e:\regtest\mf_personnel;server type=THIN;buffer size=5000;user id=murray"						
Save entity connection settings in App.Config as:						
Entities						
< Previous Next > Einish Cancel						

If you have not already done so, you can create the new connection at this time by clicking **New Connection**. See <u>Connecting to the Oracle Rdb</u> <u>Database</u> for details on each fields shown in **New Connection** Form.

### Step5: Click Next

The Entity Data Model Wizard connects to the database. You are then presented with a tree structure of the database.

Here you can select the object you would like to include in your model.

Here we have choosen to select all tables, just as a part of example.

But, You have a option to select any specific table.

**Step6:** Check the box against **Tables** node.

Entity Data Model Wizard	?×
Choose Your Database Objects	
Which database objects do you want to include in your model?	
Image: Tables         Image: Tables <td< td=""><td></td></td<>	
Pluralize or singularize generated object names	
Include foreign key columns in the model	
Model Namespace:	
Model	
<pre></pre>	cel

**Step7:** Click **Finish** to create the model and exit the wizard.

Visual Studio will generate the model then display it.

Tie Edit Vew Project Bult Debug Tean Data Tools Tex Window Help ModelLednx X FormLos (Desp) ModelLednx X FormLos (Desp)
Prodell.edmx × FormL-cs (Lesup)  Con Properties Properties CustomersDD Properties Properties Con ConVerse ConVe
Customerstb () Properties Customerid Cu

# 10.1.2 Adding a new Data Source

You will now add a new Data Source to your project and see how it can be used to read and write to the database.

### To add a new Data source:

Step1: From the Visual Studio Main Menu, select Data→ Add New Data Source.

You will be presented with the Data Source Configuration Wizard.

Data Source C	onfiguration	Wizard				? 🗙
L Ch	oose a Data	a Source Ty	ре			
Where will th	e application	get data fror	n?			
			6			
Database	Service	Object	SharePoint			
Lets you choos	e objects that c	an later be use	ed to generate d	ata-bound contro	ls.	
			< Previous	Next >	Finish	Cancel

## **Step2:** Select the **Object** icon. Click **Next**.

**Step3:** You will now select the Object you wish to bind to. Expand the tree.

In this tutorial we have choosen **Con** table as a example. Once the **Con** table has been selected click **Finish**.

Data Source Configuration Wizard	? 🛛
Select the Data Objects	
Expand the referenced assemblies and namespaces to select your objects. If an object referenced assembly, cancel the wizard and rebuild the project that contains the object What objects do you want to bind to?	
Image: State Stat	Add Reference
< Previous Next > Fini	sh Cancel

The **Con** object will be displayed in the Data Sources panel. If the Data Sources panel is not displayed, select **Data→Show Data** Sources from the Visual Studio Main Menu.

The docked panel will then be displayed.

<b>-</b>	Data Sources	▼ -= ×
	10 💩 Pi 📭	
Source Evolution 🥀 Toolbox 🏧 Data Sources	CustomersDb V CustomersDb V CustomersDb V Firstname Pysn Surname Cons CustomersDbs	

10.1.3 Using the Data Source in a Windows Form

## To use the Data Source:

**Step1:** In the Data Sources panel, select the **Data Source** you just created and **drag and drop** it onto the Form Designer.

By default the Data Source object will be added as a **Data Grid View control**.

**Note:** the Data Grid View control is bound to the **consBindingSource** and the Navigator control is bound to **consBindingNavigator**.

🙀 Data Sources 🛛 🔻 🖙 🗸	Form1.cs [Design] ×				
j 🔞 💩 😚 📭					 
Image: Section 1         Image: Section 1 <tr< td=""><td>▶ ₩ ♦ X</td><td>ConflictId</td><td>ConsType</td><td>ConsYear ►</td><td></td></tr<>	▶ ₩ ♦ X	ConflictId	ConsType	ConsYear ►	

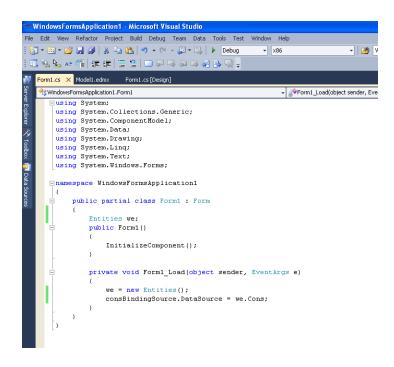
Step2: Save and rebuild the solution before continuing.

**10.1.4** Adding Code to Populate the Data Grid View

You are now ready to add code to ensure that the Data Grid View control will be populated with data from the **Con** database table.

## How to add code to Populate the view:

- **Step1:** Double-click the form to access its code.
- **Step2:** Add code to instantiate the Entity Data Model's EntityContainer object and retrieve data from the database to populate the control.



- Step3: Save and rebuild the solution.
- **Step4: Run** the solution. Ensure the grid is populated and you can navigate through the data.

🖶 Form1					
∢ ∢   1	of 3   🕨 🔰   🕂	×			
			ConHistId	ConsType	ConsYear
		▶	21	p1	1230
			22	p1	1230
			23	p1	1230
		*			
		<	III		>

# 10.2 Create a new Entity

A new Entity Data model can be created from scratch by using the **Entity Data Model Designer.** 

## How to create an Entity Data Model:

- Step1: In the Solution Explorer, right-click your application and select Add→ New Item.
- **Step2:** From Visual Studio installed templates, select **ADO.NET Entity Data Model**. Click **Add**.

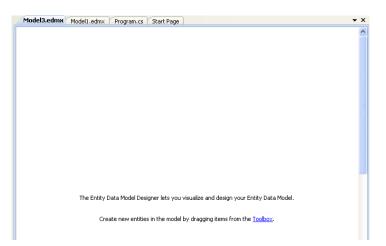
nstalled Templates	Sort by:	Default		Search Installed Templates
Visual C# Items Code	¢.	Class	Visual C# Items	Type: Visual C# Items A project item for creating an ADO.NET Entity
Data General Web	SC#	Interface	Visual C# Items	Data Model.
Windows Forms WPF		Windows Form	Visual C# Items	
Reporting Workflow	翻	User Control	Visual C# Items	
nline Templates		Component Class	Visual C# Items	
		User Control (WPF)	Visual C# Items	
	*	About Box	Visual C# Items	
	94	ADO.NET Entity Data Model	Visual C# Items	
	<u></u>	ADO.NET EntityObject Generator	Visual C# Items	
	<u>_</u>	ADO.NET Self-Tracking Entity Generator	Visual C# Items	
		Application Configuration File	Visual C# Items	
	-	Application Manifest File	Visual C# Items	
	4	Assembly Information File	Visual ⊂# Items	
	<b>*</b>	Bitmap File	Visual C# Items	
ame: Model1.e		Close Diseases	Secol C# Bone	

You will now see the **Entity Data Model Wizard**. This wizard is used to generate the Entity Data Model from the database specified in the database connection string

Step3: Select the icon Entity model. Click Finish.

niny bata	Model Wizard	<u>?</u> ×
	Choose Model Contents	
What shou	ld the model contain?	
Generate fr database		
	empty model as a starting point for visually designing a conceptual model from the	
LOOIDOX, Cie	sses are generated from the model when the project is compiled. You can specify a	
database co	sses are generated from the model when the project is compiled. You can specify a nnection later to map the conceptual model to the storage model.	
database co	isses are generated from the model when the project is compiled. You can specify a nnection later to map the conceptual model to the storage model.	
database co	isses are generated from the model when the project is compiled. You can specify a innection later to map the conceptual model to the storage model.	
database co	sses are generated from the model when the project is compiled. You can specify a nnection later to map the conceptual model to the storage model.	
database co	isses are generated from the model when the project is compiled. You can specify a nnection later to map the conceptual model to the storage model.	
database co	sses are generated from the model when the project is compiled. You can specify a nnection later to map the conceptual model to the storage model.	
database co	sses are generated from the model when the project is compiled. You can specify a nnection later to map the conceptual model to the storage model.	el

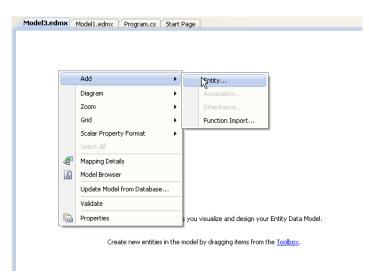
An empty Entity Data Model Designer Form should now be displayed.



We will now show how to design a following sample entity:



Step4: Add a new Entity by right-clicking in the Model Designer and selecting Add→ Entity.



The Add Entity form will be displayed.

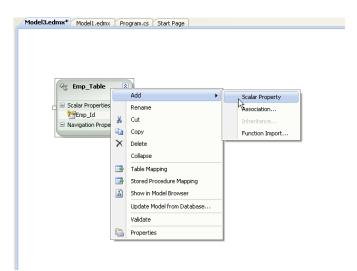
dd Entity	? 🛛
Properties	
Entity name:	
Entity1	
Base type:	
(None)	<b>~</b>
Entity Set:	
Entity1Set	
Key Property Create key property Property name:	
ld	
Property type:	
Int32	✓
	<u>D</u> K Cancel

**Step5:** Change the **Entity name** to Emp\_Table and the Key **Property name** to Emp id. The Entity Set will be automatically changed to Emp TableSet

Key Property	
Create key property	
Property name:	
Emp_)d	
Property type:	
Int32	×
	OK Cancel

### Step6: press OK.

Step7: Right-click the new entity and use Add →Scalar Property to add three more String properties Last\_name, First\_name and Title



Remember to change the **Properties**  $\rightarrow$ **Type** for each of the new properties to **String** by selecting the respective scalar properties.

	Pr	operties			🗕 🛨 🖞	×
	м	Model3.Emp_Table.Last_Name Property			-	
🗞 Emp_Table 🛛 🖄						
		Concurrency Mode	•	None		
Scalar Properties		Default Value				
	Ξ	Documentation				
🚰 Emp_Id		Entity Key		False		
St_name		Getter		Public		
		Name		Last_Name		
Navigation Properties		Nullable	N	True		
		Setter	43	Public		
		Туре		String		~

## **10.3 Creating Complex Types**

Complex types are non-scalar properties of entity types which enable scalar properties to be organized within entities. This makes easier to work with objects, allowing the grouping of related properties in Entities.

As an example, let us create the Complex type Employee\_Name instead of the Last Name and First Name properties.

To create a Complex type:

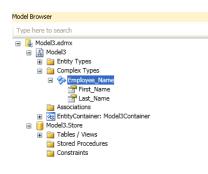
- **Step1:** Hold the **CRTL** key and **Select** the properties in the data model that you want to group together
- Step2: Right click in the context menu, select Refactor into New Complex Type

🔏 Emp_Table	۲		
Properties			
Pro_Id			
PLast_Name			
Title		Refactor into New Complex Type	
Navigation Propertie	¥	Cut	Ctrl+X
		Сору	Ctrl+C
	8	Paste	Ctrl+V
	$\times$	Delete	Del
		Update Model from Database	
		Generate Database from Model	
		Add Code Generation Item	
		Validate	
	<b>P</b>	Properties	Alt+Enter

The model browser will display a new Complex Type.



**Step3:** Now let's give a more meaningful name. If you expand the **Complex Type1**, you will see the properties that we previously selected. Change the name from ComplexType1 to Employee Name.



If you look at your data model you will see that **ComplexProperty** has been added to the list.



**Step4:** Change the name from ComplexProperty to Emp Name.



If you look at the properties, you will see that the type of **Emp\_Name** is the complex type name that is **Employee\_Name** which we just created.

Pro	perties		γų×			
м	Model3.Emp_Table.Emp_Name Property					
•	2↓ □					
Ξ	Code Generation					
	Getter	Public				
	Setter	Public				
Ξ	General					
	Concurrency Mode	None				
Ŧ	Documentation					
	Name	Emp_Name				
	Туре	Employee_Name				

## 10.3.1 Using the Complex Type created

The Complex type you created may be used as follows:

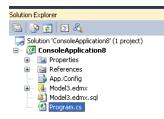
**Step1:** In Visual Studio, select **File→New→Project** from the main menu.

Choose the Console Application installed template. Click **OK**. The solution is created.

**Step2:** Design a new Entity with Complex Types. See <u>Create a new Entity</u>, <u>Creating Complex Types</u>

**Step3:** Generate the Database schema from the data model you have designed. See <u>Generating Database Schema from Model</u>

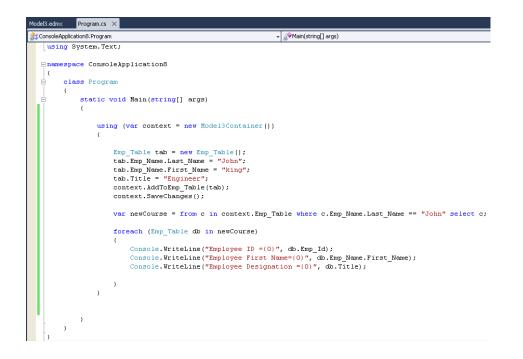
The Console Application's solution explorer looks as follows:



Step4: Open Program.cs file by double clicking on it.
Step5: Write the code to use the complex type just created.

ConsoleApplication8.	Program		→  args)
<b>⊡using</b> Syste	em;		1-
using Syste	em.Collections	s.Generic;	
using Syste	em.Linq;		
using Syste	em.Text;		
namespace C	ConsoleApplica	ation8	
(			
🚊 class F	rogram		
{			
	atic void Main	n(string[] args)	
•			12/2-04-04-04-04-04-04-04-04-04-04-04-04-04-
•	using (var (	context = new Mode.	isconcainer())
	ι.		
	Emn Tab	le tab = new Emp Ta	able():
		Name.Last Name =	
		Name.   "king";	
•	tab.Tit:		
	context	. AddTo	
	context	. SaveC 💊 GetHashCode	
		=♦ GetType	
_		Tast Name	
	}	PropertyChan	her
		PropertyChan	-
· .		Fropertychan	30.12
}		Tooding	
}			

**Step6:** The Complete code looks as follows.



**Step7: Save** and **rebuild** the solution. See <u>Compiling and Running the</u> <u>Application</u>

**Step8: Run** the solution and we get following output on Console.

C: WINDOWS\system32\cmd.exe Employee ID =321 Employee First Name=king Employee Designation =Engineer Press any key to continue		

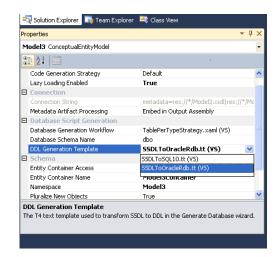
## **10.4 Generating Database Schema from Model**

Visual Studio 2010 with Entity Framework V2.0 enables you to generate a database schema from an existing Data Entity Model.

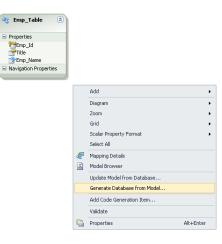
This section will take you through the steps required to generate a fresh database schema definition from a model.

### How to generate a schema from an Entity Data Model:

Step1: From the Properties window of your selected data model choose SSDLToOracleRdb.tt in the DDLGeneration Template property as shown below:



Step2: Right click on entity designer and select Generate Database from model



**Step3:** Add a Connection to the database when prompted. **Step4:** Click **Next**. A new SQL script will be generated .

Generate Database Wizard	?×
Summary and Settings	
Save DDL As: Model3.edmx.sql	
DDL	
Creating table "Emp_Table" CREATE TABLE "Emp_Table" ( "Emp_Id" AUTOMATIC INSERT AS CAST(Emp_Table_IDENTITY.NEXTVAL AS INTEGER) PRIMARY KEY, "Title" NCHAR VARYING(2000) , "ComplexProperty_First_Name" NCHAR VARYING(2000) , "ComplexProperty_Last_Name" NCHAR VARYING(2000) ) );	
	<b>~</b>
< Previous Next > Finish Cano	:el

You can now copy the SQL script create and execute on your Rdb database using Interactive SQL.

Show Navigation

# **Chapter 11 Unsupported features**

- Foreign Key support
- Open Table definition
- Alter Procedure
- Data View Designer
- Import/export table data as XML.

Show Navigation

# Glossary

#### assembly

Assembly is Microsoft's term for the module that is created when a DLL or .EXE is complied by a .NET compiler.

#### **Binary Large Object (BLOB)**

A large object datatype who's content consists of binary data. Additionally, this data is considered raw as its structure is not recognized by the database.

#### Character Large Object (CLOB)

The LOB datatype whose value is composed of character data corresponding to the database character set.

#### data provider

As the term is used with Rdb Data Provider for .NET, a data provider is the connected component in the ADO.NET model and transfers data between a data source and the DataSet.

#### dirty writes

Dirty writes means writing uncommitted or dirty data.

#### DDL

DDL refers to data definition language, which includes statements defining or changing data structure.

#### DOM

Document Object Model (DOM) is an application program interface (API) for HTML and XML documents. It defines the logical structure of documents and the way that a document is accessed and manipulated.

#### flush

Flush or flushing refers to recording changes (that is, sending modified data) to the database.

#### instantiate

A term used in object-based languages such as C# to refer to the creation of an object of a specific class.

#### Large Object (LOB)

The class of SQL datatype that is further divided into internal LOBs and external LOBs. Internal LOBs include BLOBs, CLOBs, and NCLOBs while external LOBs include BFILES.

#### **Microsoft .NET Framework Class Library**

The Microsoft .NET Framework Class Library provides the classes for the .NET framework model.

#### namespace

.NET:

A namespace is naming device for grouping related types. More than one namespace can be contained in an assembly.

• XML Documents:

A namespace describes a set of related element names or attributes within an XML document.

#### National Character Large Object (NCLOB)

The LOB datatype whose value is composed of character data corresponding to the database national character set.

#### octet

An 8-bit unit, usually referred to as BYTE

#### RdbDataReader

An RdbDataReader is a read-only, forward-only result set.

#### primary key

The column or set of columns included in the definition of a table's PRIMARY KEY constraint.

#### reference semantics

Reference semantics indicates that assignment is to a reference (an address such as a pointer) rather than to a value. See **value semantics**.

#### result set

The output of a SQL query, consisting of one or more rows of data.

#### savepoint

A point in the workspace to which operations can be rolled back.

#### stored procedure

A stored procedure is a block of SQL code that Rdb stores in the database and can be executed from an application.

#### Unicode

Unicode is a universal encoded character set that enables information from any language to be stored using a single character set.

#### URL

URL (Universal Resource Locator).

#### value semantics

Value semantics indicates that assignment copies the value, not the reference or address (such as a pointer). See **reference semantics**.

Show Navigation