#### Oracle® Rdb Developer Tools for Visual Studio

Developer's Guide Release 7.3.6.0.0 March 2018

Oracle Rdb Data Provider for .NET Developer's Guide, Release 7.3.6.0.0.

Copyright © 2011, 2018 Oracle and/or its affiliates. All rights reserved.

Primary Author: Jim Murray.

Contributing Author:

Contributor:

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it 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 computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Contents

Preface		5
Chapter 1		8
Introducing C	Oracle Rdb Developer Tools for Visual Studio	8
Chapter 2	Installing Oracle Rdb .NET Products	9
Chapter 3	Using Oracle Rdb Developer Tools for Visual Studio	10
3.1 Using O	racle Rdb Developer Tools	10
3.2 Connect	ing to the Oracle Rdb Database	10
Chapter 4	Building a Simple .NET Application Using ORDP.NET	16
4.1 Creating	g a New Project	16
4.2 Adding d	a Reference	20
4.3 Adding I	Namespace Directives	22
4.4 Designir	ng the User Interface	23
4.5 Writing	the Connection Code	28
4.6 Compili	ng and Running the Application	31
	andling	
	ng Try-Catch-Finally Block Structure dling General Errors	
Chapter 5	Retrieving and Updating with ORDP.NET	
5.1 Using th	e Command Object	37
5.2 Retrievii	ng Data: a Simple Query	39
5.3 Retrievii	ng Data: Bind Variables	42
5.4 Retrievii	ng Data: Multiple Values	44
5.5 Using th	e DataSet Class with Oracle Rdb Data Provider for .NET	46
5.6 Enabling	g Updates to the Database	48
Chapter 6	ORDP.NET and Visual Studio Wizards	52
6.1 Create a	New application	52

6.2 Add con	nnection to Rdb database in Server Explorer	
6.3 Create d	a datasource for the employees table	
	the windows application	
Chapter 7	ORDP.NET and drag and drop	58
7.1 Create d	a New application	58
7.2 Add con	nnection to Rdb database in Server Explorer	58
7.3 Add a n	ew Typed DataSet	58
7.4 Designi	ng the windows application	61
7.5 Running	g the windows application	62
Chapter 8	Configuration wizard and generating queries	64
8.1 Using th	he Query Configuration Wizard	64
8.2 Using th	he Query Builder	67
8.3 Method.	s Generation	72
Chapter 9	Copying a Form	75
Chapter 10	Oracle Rdb Entity Framework Provider	78
10.1 Creati	ng new Windows Form Application	79
10.2 Refere	nce the Oracle Rdb EntityFramework Provider	79
10.3 Addin	g an Entity Data Model	
10.4 Adding	g a new Data Source	85
	sing the Data Source in a Windows Form	
10 1 0 1	I din a Carda ta Davarlata tha Data Carid Viana	00
	dding Code to Populate the Data Grid View	
10.5 Create	e a new Entity	90
10.5 Create 10.6 Creatia		90 93
10.5 Create 10.6 Creatie 10.6.1 Us	a new Entity ng Complex Types	
10.5 Create 10.6 Creatin 10.6.1 Us 10.7 Genero	a new Entity ng Complex Types sing the Complex Type created	
10.5 Create 10.6 Creatin 10.6.1 Us 10.7 Genero 10.8 Using 10.8.1 Da	e a new Entity ng Complex Types sing the Complex Type created ating Database Schema from Model Fluent API atabase Creation and Removal	
10.5 Create 10.6 Creatin 10.6.1 Us 10.7 Genero 10.8 Using 10.8.1 Da	e a new Entity ng Complex Types sing the Complex Type created ating Database Schema from Model Fluent API	

### **Glossary**

# **Send Us Your Comments**

#### Oracle Rdb Developer Tools for Visual Studio Developer's Guide

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
- Access to Oracle Support
- 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.

## Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <u>http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info</u> or visit <u>http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs</u> if you are hearing impaired.

## 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.
- **Chapter 2, Installing Oracle Rdb .NET Products** 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.
- <u>Chapter 3, Using Oracle Rdb Developer Tools for Visual Studio</u>
   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.
- <u>Chapter 5, Retrieving and Updating with ORDP.NET</u> 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</u> and drag and drop
   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.
- <u>Chapter 10. Oracle Rdb Entity Framework Provider</u> 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 SQL Reference Manual
- Oracle Rdb Guide to SQL Programming
- Oracle SQL/Services Server Configuration Guide
- Guide to Using the Oracle Rdb 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: <u>http://msdn.microsoft.com/netframework</u>

### 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';
```

### **Example Screen Shots**

Unless otherwise stated, examples and screen shots found within this document relate to Visual Studio 2012.

As the user interface for Visual Studio may change between versions, if you are using another version of Visual Studio, please consult your Visual Studio documentation to determine how to carry out the same operations described in the provided example.

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

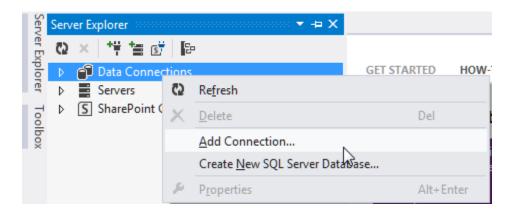
Oracle Rdb Developer Tools for Visual Studio (ORDT) is a tightly integrated Add-in for Visual Studio. Using enhancements that ORDT brings to the Server Explorer, you can automatically create tables, indexes, constraints, data connections and other database schema objects. Additionally you can automatically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

### 3.2 Connecting to the Oracle Rdb Database

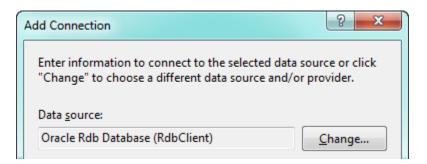
This section shows you how to use the Server Explorer to connect to the Oracle Rdb Database for the purpose of automatically creating or modifying database schema objects.

### 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.



Step4: If you have selected any Data Source in the past, the Add Connection window appears, determine if the Data source says "Oracle Rdb Database (RdbClient)". If it does, skip to Step <u>6</u>.



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

If you have not already selected a Data source previously, or you have selected **Change**, the Choose Data Source window will appear.

Choose Data Source	8 <b>x</b>
Data <u>s</u> ource: Microsoft Access Database File Microsoft ODBC Data Source Microsoft SQL Server Microsoft SQL Server Compact 4.0 Microsoft SQL Server Database File Oracle Database Oracle Rdb Database <other></other>	Description .NET Framework Data Provider For Oracle Rdb
Data provider:	
.NET Framework Data Provider For Orac 👻	
Always <u>u</u> se this selection	Continue 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				? X
Enter information to "Change" to choose				
Data <u>s</u> ource:				
Oracle Rdb Databas	e (RdbClient)		<u> </u>	nange
S <u>e</u> rver (name: <port< td=""><td>or service&gt;):</td><td></td><td></td><th></th></port<>	or service>):			
MY_SQS_SERVER:G	ENERIC			
📄 Use <u>T</u> hin Server				
<u>U</u> ser name:	RDBNET_USE	R		
Password:	•••••	•		
	Save my pa	assword		
Database name:				
DISK1:[MY_TEST_D	ATABASES]MF	PERSONNEL		
		(	Ad <u>v</u>	anced
Test Connection		ОК	•	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				?	x
Enter information t "Change" to choos					lick
Data <u>s</u> ource:					
Oracle Rdb Databa	se (RdbClient)		C	hange	·
S <u>e</u> rver (name: <port< td=""><td>or service&gt;):</td><td></td><td></td><th></th><th></th></port<>	or service>):				
MY_SQS_SERVER:N	IF_PERS				
📄 Use <u>T</u> hin Server					
<u>U</u> ser name:	RDBNET_USE	R			
Password:	•••••	•			
_	<mark>▼ S</mark> ave my pa	assword			
<u>D</u> atabase name:					
			Ady	<u>v</u> ance	:
Iest Connection		ОК		Cance	<b>.</b>

### To connect to a database using a JDBC Thin Server:

For **Server**, enter the node address and port for a valid JDBC server that will be running on the selected node. Remember to separate the node address and port with a colon (:). The node address must be valid TCP/IP node specification.

Click the **Use Thin Server** box.

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

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

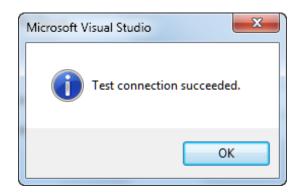
To save the password for future sessions, check the **Save password** box.

For **Database name**, Enter a valid database file specification. 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			? <mark>×</mark>
Enter information to "Change" to choose Data <u>s</u> ource:			
Oracle Rdb Databa	se (RdbClient)		Change
S <u>e</u> rver (name: <port< td=""><td>or service&gt;):</td><td></td><td></td></port<>	or service>):		
MY_TEST_NODE:17	01		
☑ Use <u>T</u> hin Server			
<u>U</u> ser name:	RDBNET_USE	R	
Password:	•••••		
	<mark>▼ S</mark> ave my pa	issword	
<u>D</u> atabase name:			
DISK1:[MY_TEST_D	ATABASES]MF	PERSONNEL	
		[	Ad <u>v</u> anced
Test Connection		ОК	Cancel

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



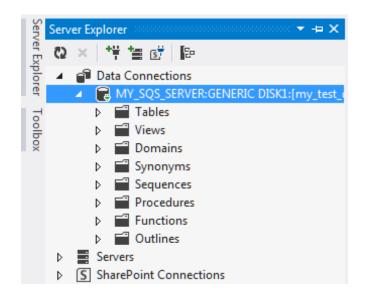
The test should succeed. Click OK.

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.
- $_{\odot}$   $\,$  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.



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

This chapter contains:

- Creating a New Project
- 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** 2012.

The Microsoft Visual Studio IDE environment appears.

Start Page - Microsoft Visual Studio FILE EDIT VIEW DEBUG TEAM SQL TOOLS ○ • ○ 1 10 • 11 11 11 11 11 11 11 11 11 11 11 11 1		ALYZE WINDOW HELP		•
Premium 2012	GET STARTED HOW-TO V		est news ing videos:	
Start New Project Open Project Connect to Team Foundation Server	闭 Improve unit testing workflow with these Test Explorer improvements	Rew testing tools for manual testers	密 Enhanced features for testing software with Visual Studio	
Recent WindowsFormsApplication1	邵 Improving quality with unit tests	How to multi-task with My Work	译 Using Code Review to improve quality	
V73000 rdbjdbc Oracle.DataAccess.Rdb.NET20 rdb_cosi EFDemo	Easily reproducing issues through manual testing	➢ Easily testing user interfaces with Coded UI tests	ৰ্দ্ধি Finding and managing cloned code	
<ul> <li>Close page after project load</li> </ul>	體 Coordinate your team with agile project management	Hanaging lab environments for testing		
Show page on startup	G			

**Step2:** In the Start Page, under the Start heading, click **New Project...**.

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

A **New Project** dialog box appears.

**Step3:** In the Installed Templates tree at the left of the screen, select the type of project you are creating:

### Visual C#:

Visual C#: Windows

Select Windows Forms Application.

New Project			? ×
▷ Recent	.NET Framework 4.5 * Sort by: Default	- # E	Search Installed Templates (Ctrl+E) 🛛 🔎 🗸
▲ Installed	Windows Forms Application	Visual C#	Type: Visual C#
✓ Templates ✓ Visual C# Windows	WPF Application	Visual C#	A project for creating an application with a Windows Forms user interface
Web Extensibility	Console Application	Visual C#	

**Visual Basic:** 

Other Languages: Visual Basic: Windows

Select Windows Forms Application.

ew Project	and the second sec	2
Recent	.NET Framework 4.5	Search Installed Templates (Ctrl+E)
Installed ▲ Templates	Windows Forms Application	Visual Basic Type: Visual Basic
<ul> <li>Femplates</li> <li>Visual C#</li> <li>Windows Installer XML</li> </ul>	WPF Application	A project for creating an application wit Visual Basic Windows user interface
LightSwitch Other Languages	Console Application	Visual Basic
▶ Visual Basic ▶ Visual C++	ASP.NET Web Forms Application	Visual Basic
SQL Server	Class Library	Visual Racio

**Step4:** 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.

**Step5:** In Location, enter the directory where you want to save the files.

For this guide, enter this directory

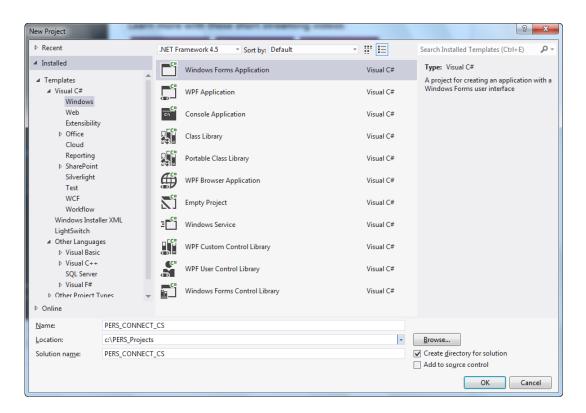
C:\PERS\_Projects

**Step6:** 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.

Step7: Check Create directory for solution.

Step8: Click OK.



The project is created.

The main window now displays a new title, either **PERS\_Connect\_CS** - **Microsoft Visual Studio** or **PERS\_Connect\_VB** - **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.

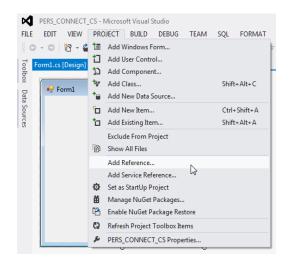
FILE EDIT VI	IECT_CS - Microsoft Visual Studio EW <u>P</u> ROJECT <u>B</u> UILD <u>D</u> EBUG TEA <u>M</u> S <u>(</u> → 🎱 💾 🗳 🦻 → 🦿 → Start → Debug
Form1.cs [Des	ign] + ×
Data Sources	

## 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.

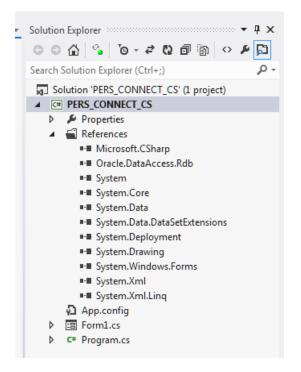


The Add Reference windows appears.

**Step2:** In the **Reference Manager** window, in the **Search Assemblies** field, Enter "Oracle." And select **Oracle.DataAccess.Rdb** that is displayed by clicking the box to the left of the name. Click OK.

Assemblies				Oracle.	×
Framework		Name	Version	Name:	
Extensions	2	Oracle.DataAccess.Rdb	7.3.2.1	Oracle.DataAccess.Rdb	
Search Results				Created by:	
Recent				Oracle Corporation	
				Version: 7.3.2.1	
Solution				File Version:	
COM				7.3.2.1	
Browse					
browse					

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	w Project	Build	Debug
1	Code		F7
-8	Designer	Sł	nift+F7
<b>.</b>	Server Explo	orer Ctr	1+W, L
~7	Solution Exp	lorer Ctr	l+w, s
<u> </u>	Class View	Ctr	I+W, 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;

Form1.cs*     >     X     Form1.cs [Design]*       Image: Second Seco	
PERS_CONNECT_CS.Form1	
⊡using System;	
using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Drawing;	
using System.ComponentModel;	
<pre>g using System.Data;</pre>	
using System.Linq;	
using System.Text;	
<pre>using System.Threading.Tasks; using System.Windows.Forms;</pre>	
using Oracle.DataAccess.RdbClient;	
using of acte.bacaAccess.Rubcilenc,	
□ namespace PERS CONNECT CS	
{	
public partial class Form1 : Form	
{	
public Form1()	
{	
<pre>InitializeComponent();</pre>	
}	
}	
L }	

### **Visual Basic:**

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

```
Imports Oracle.DataAccess.RdbClient
```

Form1.vb* Form1.vb [Design	n]* Form1.cs*
jj (General)	🔽 🎬 (Declarations)
Imports Oracle.Dat	~~~~~~~~
-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**.

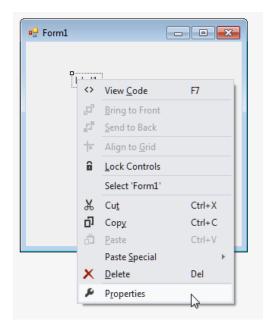
**Step3:** In the Toolbox, expand **Common Controls**.

Toolbo	Toolbox	× ⇔- ▼ ∾ - ۹
		<i>p</i> ·
	/indows Forms	
	mon Controls Pointer	
<b>►</b>	Button	
_		
_	CheckBox	
_	CheckedListBox	
_	ComboBox	
	DateTimePicker	
A	Label	
A	LinkLabel	
闘		
	ListView	
ω.	MaskedTextBox	
	MonthCalendar	
-	NotifyIcon	
₿÷	NumericUpDown	
~	PictureBox	
	ProgressBar	
o	RadioButton	
	RichTextBox	
abl	TextBox	
'n	ToolTip	
Te.	TreeView	
⊕_	WebBrowser	



Form1.cs* Form1.cs [[	)esign]* +⊨ X
- Form1	
label1	

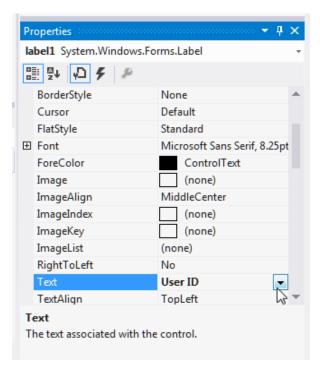
**Step5:** On Form1, right-click **label1**.



**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**.



**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**.

User ID Password Server Database	Form1.cs*	F	orm1.cs [D	esign]* ⊣	×	
User ID Password Server Database	🖳 For	m1				
		User ID	Password	Server	Database	

**Step9:** You may want to now expand the form by dragging the left margin to allow for reasonably sized fields to be accommodated. In the Toolbox, select **TextBox**, and drag it onto the Form1, under the User ID label.

ſ	🖳 Form1				- • •
	User ID	Password	Server	Database	

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

Properties	•	Ψ×
userID System.Windows.Fo	orms.TextBox	-
🔡 💱 🔎 🗲 🎤		
		-
Tag		
🗆 Design		
(Name)	userID	
GenerateMember	True	
Locked	False	
Modifiers	Private	
E Focus		•
CausesValidation	True	3

**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 **database**.

🕂 Form1					
User	r 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.

Properties	<b>-</b> ₽ ×
password System.Windo	ows.Forms.TextBox 🔹
🔡 💱 🔎 🗲 🔎	
ContextMenuStrip	(none) 🔺
Enabled	True
HideSelection	True
ImeMode	NoControl
MaxLength	32767
Multiline	False
PasswordChar	*
ReadOnly	False
ShortcutsEnabled	True
TabIndex	5
TabStop	True

**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					
Us	er ID	Password	Server	Database	
					Connect

#### Step14: Save.

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	х
Image: PERS_Connect_CS.Form1     Image: 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.DataAccess.RdbClient; namespace PERS_Connect_CS {</pre>	
public partial class Form1 : Form {	
<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

Form1.vb* Form1.vb[Design]* Start Page

Form1

Imports Oracle.DataAccess.RdbClient

Public Class Form1

Dim conn As New RdbConnection

End Class
```

- **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 & _
";server=" + server.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.

Note that in this example we are making a connection to a JDBC Thin Server as designated by the ";type=Thin;" portion of the connection string we are building. If you wish to make a connection to a SQL/Services service, omit this part of the connection string.

**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.

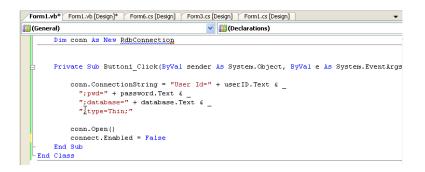
#### Visual C#:

connect.Enabled = false;

A PERS_CON	INECT_CS.Form1	*	ଡି <sub>ଜ</sub> connect_Click(object se
	System.Text;		
	System.Threading.Tasks;		
	System.Windows.Forms;		
using	Oracle.DataAccess.RdbClient;		
⊡names	Dace PERS_CONNECT_CS		
{			
🔅 pi	ublic partial class Form1 : Form		
{			
Ē	public Form1()		
	{ 		
	InitializeComponent();		
-	<pre>private RdbConnection conn = new RdbConnect</pre>	tion():	
	privace habconnección com - neu habconnec	())	
Ė.	<pre>private void connect_Click(object sender, 6</pre>	EventArgs	e)
	{		
	conn.ConnectionString = "User Id=" + us	serID.Text	+
	owd=" + password.Text +		
	server=" + server.Text +		
	database=" + database.Text + type=Thin;";		
· ·	type=inin; ;		
	conn.Open();		
	connect.Enabled = false;		
	}		
[ }			
}			

### **Visual Basic:**

connect.Enabled = false



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

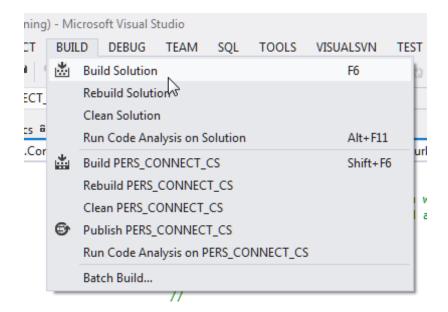
The following sections show how to use it.

### 4.6 Compiling and Running the Application

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

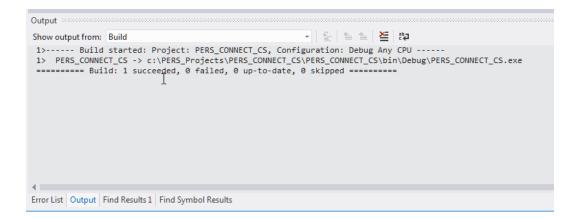
### To compile and run the application:

### **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	TEAM	SQL	TOOLS	VISUAL	SVN	TEST	A	NA
		Win	dows						۲	13
		Gra	phics						Þ	
	•	Star	t Debuggi	ng			F5			jec
1	⊳	Star	t Without	Debugg	jing 🕟		Ctrl+F	5		F
i		Star	t Performa	ince An	5	Alt+F2				
i I	卻	Star	t Performa	ince An	alysis Paus	ed	Ctrl+/	Alt+F2		
		Java	Script Ana	alysis					Þ	
	ø	Atta	ch to Prod	ess						
-		Exce	eptions				Ctrl+[	D, E		
	ς.	Step	Into				F11			
	G,	Step	o Over				F10			
		_								

**Step5:** In the Form1 application, enter the User ID, Password , Server, and Database. If you used the connection string exactly as shown in **Step 4** above, you will be making a connection to a JDBC Thin Server. Please ensure that the data you enter in the **Server** field does contain a valid JDBC server specified in the form "host:port", for example "my server node:1701"

If you omit the ";type=Thin; " part of the connection string, then you are requesting for a SQL/Services service connection, in which case ensure the data entered in the **Server** field is in the form of "host:service" for example, "my sqs server:GENERIC".

### Step6:

### Click Connect.

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

## 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 <code>Dispose()</code> 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
```

```
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.

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

Show Navigation

# Chapter 5 Retrieving and Updating with ORDP.NET

This chapter contains:

- Using the Command Object
- <u>Retrieving Data: a Simple Query</u>
- <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#:

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

### **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:**

```
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()
```

- 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 departments			Connect

**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.

🖶 Form1			
Rdbnet_user Department Engineering	Password Redecorder	Server SQSServer:GENERIC	Database user1:[dbs]mf_personnel Connect

# 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)
cmd.CommandType = CommandType.Text
Dim p_department_id as RdbParameter = new 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.

**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 xxxxxxxxx	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
•			_

**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;
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()</pre>
```

```
da.Fill(ds)
```

**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.

👥 F	orm1							
	Jser ID Rdbnet_u	user	Passw *******		_	rver QSServer:GENERIC	Database user1:[dbs]mf_Personnel	
		DEPARTME	NT	DEPARTMENT		MANAGER_ID	Connect	
	•	ADMN		Corporate Adm	i	00225		
		ELEL		Electronics Eng	in	00188		
		ELGS		Large Systems	E	00369		
		ELMC		Mechanical Eng	ļi	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.

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.

		Password			Database user1:[dbs]mf_Personnel
	DEPARTMENT	DEPARTME	ENT	MANAGER_ID	Connect
•	ADMN	Corporate	Admi	00225	
	ELEL	Electronics	Engin	00188	
	ELGS	Large Syst	ems E	00369	
	ELMC	Mechanical	Engi	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 forms 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 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 **Project** → **Add New** Data Source.

empapp - Micro		sual Stu JECT	udio BUILD	DEBUG	TEAM	SQL	TOOLS
G         -         ○         III         -         -           See         Form1.cs*         +         ×	11 รา	Add U	Vindows F Jser Contr Componer	ol			
Form1.cs* + X cs* empapp.Form using Sy using Sy	5	Add C Add N	lass Iew Data	Source	G	Shift+	Alt+C
using Sy using Sy using Sy	5		lew Item. xisting Ite		-0	Ctrl+S Shift+	
		Exclud	de From P	roject			

- Step2: In the wizard, keep the default selection for "Database". Click Next
- **Step3:** In **Choose a Database Model** keep the default selection for "Dataset". Click **Next**
- **Step4:** 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**

ata Source Configuration Wizard	? ×
Choose Your Data Connection	
Which data connection should your application use to connect to the database?	
MY_SQS_SERVER:GENERIC DISK1:[my_test_databases]mf_personnel	New Connection
This connection string appears to contain sensitive data (for example, a password), which is re database. However, storing sensitive data in the connection string can be a security risk. Do ye sensitive data in the connection string?	
$\odot~$ No, exclude sensitive data from the connection string. I will set this information in my	application code.
Yes, include sensitive data in the connection string.	
Connection string that you will save in the application (expand to see details)	
< Previous Next > Finish	Cancel

**Step5:** Check to save the connection string to the Application Configuration File Click **Next** 

Data Source Configuration Wizard	? ×
Save the Connection String to the Application Configuration File	
Storing connection strings in your application configuration file eases maintenance and deployment. To s connection string in the application configuration file, enter a name in the box and then click Next. <b>Do you want to save the connection string to the application configuration file?</b>	save the
< <u>Previous</u> <u>Next</u> > <u>Finish</u>	Cancel

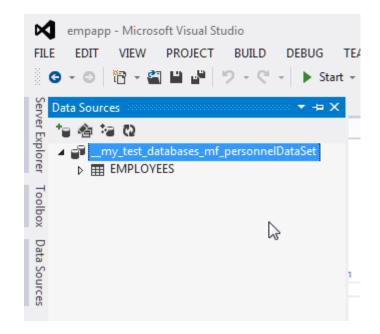
**Step6:** A tree of available table will be displayed, select the employees table from the list of tables by checking the box next to it. Click **finish** 

Data Source	Configuration Wizard	8 ×
Ĩ	Choose Your Database Objects	
Which da	tabase objects do you want in your dataset?	
	Tables         Image: CANDIDATES         Image: COLLEGES         Image: DEGREES         Image: DEPARTMENTS         Image: DEPARTMENTS <td></td>	
<u>D</u> ataSet n		
my_test	_databases_mf_personnelDataSet	
	< <u>P</u> revious <u>N</u> ext > <u>Finish</u>	Cancel

**Step7:** A data source for employees will be added in the "Data Sources" tab panel.

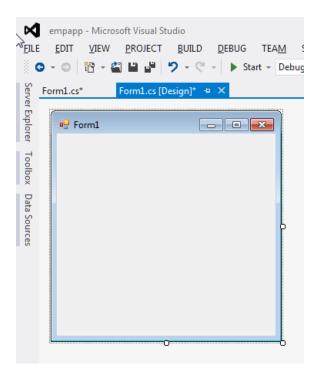
You can make the data source window visible using the top level menu View → Other Windows→ Data sources

FILE EDIT	VIEV	V PROJECT BUILD	DEBUG TEAM SC	QL 1	OOLS VISUALSV	N TEST	ANALYZE	WINDOW
G - O	$\diamond$	Code	F7	1	🞜 🚚 🖕 🖷 🕴	ə 🍫 📘	위험점	d ↓ 1
	<b>G</b>	Designer	Shift+F7					
Form1.cs*		Open		- H	+ @ Fo	orm10		
	9) 51	Open With				MILLO .		
orer u	si д	Solution Explorer	Ctrl+W, S					
. u	s 👯	Team Explorer	Ctrl+∖, Ctrl+M					
	s:	Server Explorer	Ctrl+W, L					
	E.	SQL Server Object Explorer	Ctrl+ Ctrl+S					
Data Sources		Bookmark Window	Ctrl+W, B					
î P	$\mathbb{Z}$	Call Hierarchy	Ctrl+W, K					
	45	Class View	Ctrl+W, C					
	$\bigcirc$	Code Definition Window	Ctrl+W, D					
	27	Object Browser	Ctrl+W, J					
-	പ	Error List	Ctrl+W, E					
Ē	<b>K</b> ¢	Output	Ctrl+W, O	ve	ntArgs e)			
	C	Start Page						
	പ്	Task List	Ctrl+W, T					
}	ŵ	Toolbox	Ctrl+W, X					
[]		Find Results		F				
		Other Windows		Þ	Command Wine	dow	Ctrl+W	A
		Toolbars		× i	Data Sources	•	Shift+A	lt+D
	15.71 16.31	Full Screen	Shift+Alt+Enter		Web Browser	3	Ctrl+W	w
	-	All Windows	Shift+Alt+M	€	Page Inspector			
	G	Navigate Backward	Ctrl+-	Z	Load Test Runs			
	0	Navigate Forward	Ctrl+Shift+-	C	Layer Explorer			



# 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. You will have to drag the left side of the form to make it wider so it may accommodate the fields from employees.

🖳 Form								
E 14 - 4	0 of {0}	🕨 🕨 🕂 🗡	<b>1</b>					
				0				
	ADDRESS_DATA_	ADDRESS_DATA_	BIRTHDAY	CITY	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MIDDLE
*								
9								

# **Step3:** Build and Run the windows application

	ADDRESS_DATA_	ADDRESS_DATA_	BIRTHDAY	CITY	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MID	C -
•	146 Pamell Place		28/03/1947	Chocorua	00164	Alvin 🔒	Toliver	А	
θ	120 Tenby Dr		15/05/1954	Chocorua	00165	Тепу	Smith	D	
θ	19 Union Square		20/03/1954	Boscawen	00166	Rick 😶	Dietrich		-
θ	143 Pine St		5/03/1937	Marlow	00167	Janet	Kilpatrick		-
θ	87 West Rd		23/10/1932	Meadows	00168	Noman 🔒	Nash		
θ	51 Maple St		13/08/1938	Etna	00169	Susan 🤑	Gray	0	-
θ	140 Searles Rd		3/06/1957	Jefferson	00170	Brian	Wood		
θ	67 Underhill St		29/01/1932	Sandown	00171	Aruwa	D'Amico		-
	1		11				1	4	

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:

- Create a New application
- 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 **empapp2**.

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

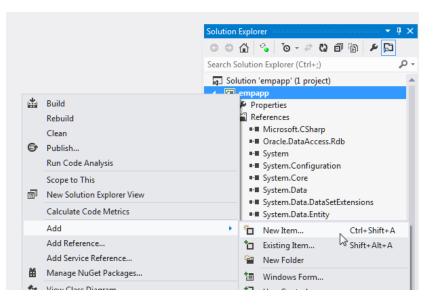
# 7.2 Add connection to Rdb database in Server Explorer

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: Using the Microsoft Solution Explorer, right click on the empapp2, and select Add → New Item



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

Installed		Sort by:	Default *	E		Search Installed Templates (Ctrl+E)
✓ Visual C# Items Code		~*^	Bitmap File	Visual C# Items	^	Type: Visual C# Items A DataSet for using data in your
Data General		Ľ	Class Diagram	Visual C# Items		application
Web Windows Fo	rms	8	Code Analysis Rule Set	Visual C# Items		
WPF Reporting		Z,	Code File	Visual C# Items		
Workflow Graphics		<b>۳</b>	CoffeeScript File	Visual C# Items		
Online		k	Cursor File	Visual C# Items		
		įΠ	Custom Control	Visual C# Items		
		J	DataSet	Visual C# Items		
		C#	Debugger Visualizer	Visual C# Items		
		œ	Domain Service Class	Visual C# Items		
		⇔.	Dynamic Data Field	Visual C# Items		
		Ð	EF 5.x DbContext Generator	Visual C# Items		
		<b>``</b>	HTML Page	Visual C# Items		
			t	10		
Name:	DataSet1.xsd					
						Add Cancel

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

-	empapp - Microsoft Visual Studio
FILE	EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS VISUALSVN TEST
	ver Explorer
2	× 樟油酸 腳
Iorer Toolbox	<ul> <li>Data Connections</li> <li>MY_SQS_SERVER:GENERIC DISK1:[my_test_/</li> <li>Tables</li> <li>CANDIDATES</li> </ul>
ox Data Sources	<ul> <li>▷ E COLLEGES</li> <li>▷ E DEGREES</li> <li>▷ E DEPARTMENTS</li> </ul>
ources	▶ ■ EMPLOYEES         ▶ ■ JOB_HISTORY         ▶ ■ JOBS         ▶ ■ JOBS         ▶ ■ SALARY_HISTORY         ▶ ■ SALARY_HISTORY         ▶ ■ WORK_STATUS         ▶ ■ Domains         ▶ ■ Synonyms

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

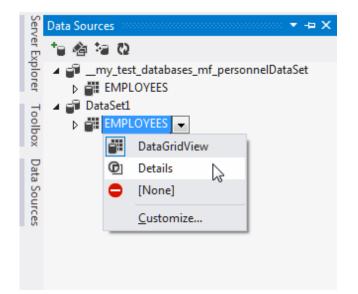
91	EMPLOYEES 🔝
	ADDRESS_DATA_1
	ADDRESS_DATA_2
	BIRTHDAY
	CITY
	EMPLOYEE_ID
ę.	FIRST_NAME
	LAST_NAME
	MIDDLE_INITIAL
f	POSTAL_CODE
f	SEX
	STATE
ę	STATUS_CODE
5	EMPLOYEESTableAdapter
śQL	Fill,GetData ()

Step3: Ensure the password is added to the connection string. Select EMPLOYEESTableAdapter, and right click to view its properties. Click in the ConnectionString field and append your password in the below format Password=yourpassword

Properties	<b>→</b> ₽ ×
EMPLOYEESTableAdapte	r TableAdapter 👻
□ □ □ □	
Code Generation	
BaseClass	System.ComponentModel.
ConnectionModifier	Internal
Modifier	Public
Name	EMPLOYEESTableAdapter
🗆 Data	
Connection	_my_test_databases_mf_p
Name	_my_test_databases_mf_p
Modifier	Internal
ConnectionString	J=True;password=mypass
Provider	Oracle.DataAccess.RdbCli
DeleteCommand	(None)
GenerateDBDirectMeth	oc True 🔻
ConnectionString	
Information used to conn	ect to a Data Source, such as alog=x;Integrated Security=SS

### 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.

Form1.cs [De	sign]* 🗢 🗙 DataSet1.xsd*			
Server Explorer Toolbox Data Sources		ADDRESS DATA 1: ADDRESS DATA 2: BIRTHDAY: CITY: EMPLOYEE ID: FIRST NAME: LAST 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.

🖳 Form1	7-7-83	
	▶ ▶ ₽ ×	<u>n</u>
	ADDRESS DATA 1:	1/C Densell Disco
		146 Fameli Flace
	ADDRESS DATA 2:	
	BIRTHDAY:	Friday ,28 March 1947
	CITY:	Chocorua
	EMPLOYEE ID:	00164
	FIRST NAME:	Alvin
	LAST NAME:	Toliver
	MIDDLE INITIAL:	A
	POSTAL CODE:	03817
	SEX:	M
	STATE:	NH
	STATUS CODE:	1
<u>.</u>		

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
- Methods Generation

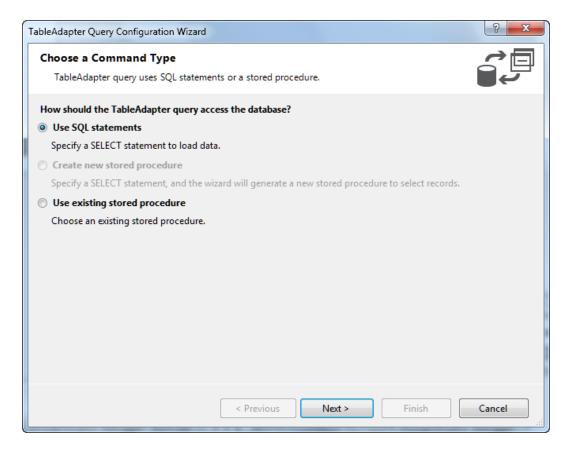
# 8.1 Using the Query Configuration Wizard



Server Explorer Toolbox Data Sources	DataSet1.xsd* 4 ADDRESS_DATA_1 ADDRESS_DATA_2 BIRTHDAY CITY EMPLOYEE_ID FIRST_NAME LAST_NAME LAST_NAME MIDDLE_INITIAL POSTAL_CODE SEX STATE STATE FILGEData ()	₩ □ ×	Add         Configure         Cut         Copy         Paste         Delete         Rename         Autosize         Preview Data         View Code         Properties	Ctrl+X Ctrl+C Ctrl+V Del F7 Alt+Enter	· · · · · · · · · · · · · · · · · · ·	Query <u>R</u> elation <u>K</u> ey <u>C</u> olumn	Ctrl+L	
--------------------------------------	--	-------------	---	--	---------------------------------------	--	--------	--

**Step1:** In the DataSet XSD window right click dataset and click **add→Query** 

The Table Adapter Query Configuration Wizard will be shown.



### 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	? x
Choose a Query Type =	<u>}</u>
Choose the type of query to be generated	
What type of SQL query would you like to use?	
<u>SELECT</u> 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).	
© <u>U</u> PDATE	
Changes existing data in a table.	
DELETE Removes rows from a table.	
○ INSERT	
Adds a new row to a table.	
< <u>P</u> revious <u>N</u> ext > <u>F</u> inish C	ancel

### **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.	
Type your SQL statement or use the Query Builder to construct it. What data should be loaded into the <u>W</u> hat data should the table load?	ie table?
SELECT CITY, FIRST_NAME FROM EMPLOYEES WHERE EMPLOYEE_ID=:EMPLOYEE_ID	~
(	Query Builder
< <u>P</u> revious <u>N</u> ext > <u>F</u> inish	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  $\ensuremath{\mathsf{UPDATE}}$  and  $\ensuremath{\mathsf{DELETE}}$  .

# 8.2 Using the Query Builder

Instead of entering the SQL text directly 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:** Startjng with Query Configuration Wizard used in the previous walkthrough, Click **Query Builder** to bring up the builder window .

TableAdapter Query Configuration Wizard	? X
Specify a SQL SELECT statement The SELECT statement will be used by the query.	
Type your SQL statement or use the Query Builder to construct it. What data should be loaded into the tak What data should the table load?	ble?
SELECT CITY, FIRST_NAME FROM EMPLOYEES WHERE EMPLOYEE_ID=:EMPLOYEE_ID	
Qu	ery Builder
< Previous Next > Finish	Cancel

**Step2:** This will bring up a window allowing you to choose columns for your query, in this case it will be preloaded with the query we typed in before.

Query Build	der							2	X
	EMPLOYEES ADDRESS_DAT ADDRESS_DAT BIRTHDAY CITY	;) [A_1 [A_2]	-						•
•								4	
	Column	Alias	Table		Sort Type	Sort Order	Filter	Or	Â
	CITY		EMPLOYEES	<b>V</b>					
	FIRST_NAME		EMPLOYEES	<b>V</b>					Ŧ
•			III					Þ	
SELECT FROM WHERE	CITY, FIRST_N/ EMPLOYEES (EMPLOYEE_ID		(EE_ID)						
•									•
	0 of 0	► H %	0						
Execu	ite Query						ОК	Cancel	

**Step3:** Within the **EMPLOYEES** table check the columns name and Parameters to the query (Filter) as shown below.

uery Buik	der							9	X
	EMPLOYEES	A_1 A_2	-						•
•								Þ	
	Column	Alias	Table	Outp	Sort Type	Sort Order	Filter	Or	
۱.	СПТҮ		EMPLOYEES	1					
	FIRST_NAME		EMPLOYEES	<b>V</b>					÷
•	7	1		:				F	
SELECT FROM WHERE	CITY, FIRST_N/ EMPLOYEES (EMPLOYEE_ID				/ATA_2				
•									F.
	0 of 0	► H %	0						
Execu	ite Query						ОК	Cancel	

**Step4:** Click **Execute Query** to execute the generated query. You will be prompted for the employee\_id. Enter a valid id and press **OK**.

Que	ry Parameters	S X
T	o run this query, enter values	for its <u>p</u> arameters.
	Name	Value
	EMPLOYEE_ID	NULL
		OK Cancel
		h.

**Step5:** The results of the operation will be displayed in the results window.

	EMPLOYEES  * (All Column ADDRESS_DA ADDRESS_DA BIRTHDAY CITY	ATA_1							
•								+	-
	Column	Alias T	able	Outp	Sort Type	Sort Order	Filter	Or	
•	CITY	E	MPLOYEES		21				
	FIRST_NAME	Ε	MPLOYEES	<b>V</b>					
	EMPLOYEE_ID	E	MPLOYEES				= :EMPLOYE		
•								•	
FROM		; ID = :EMPLOYEE FIRST_NAME		ESS_DAT	ADDRESS_DAT	r			
•	Etna	Susan		le St					
*	NULL	NULL	NULL		NULL				
14 ···	(   1 of 1		0						

# $\label{eq:step6:Click ``OK'' to accept the query .}$

Table	eAdapter	r Query Configuration Wizard	? ×
s		a SQL SELECT statement	
	The SEL	LECT statement will be used by the query.	
		SQL statement or use the Query Builder to construct it. What data should be loaded into the table	e?
	-	a should the table load?	
	ROM	CITY, FIRST_NAME, ADDRESS_DATA_1, ADDRESS_DATA_2 EMPLOYEES	~
	VHERE	(EMPLOYEE_ID = :EMPLOYEE_ID)	
			_
			<u> </u>
		Quer	y Builder
		<pre>&lt; Previous Next &gt; Finish C</pre>	ancel
		C Licklogs Revr > Tunisu	ancer

**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:

**Step1:** Press NEXT in the Query Configuration Wizard to bring up the choose Methods To Generate panel.

TableAdapter Quer	y Configuration Wizard	? X
Choose Meth	ods to Generate	<u>}</u>
The TableAda	apter methods load and save data between your application and the database.	
Which methods	do you want to add to the TableAdapter?	
🔽 F <u>i</u> ll a DataTab	ole	
	d that takes a DataTable or DataSet as a parameter and executes the SQL statement or SEL ed on the previous page.	ECT stored
Method name:	FillBy	
📝 <u>R</u> eturn a Dat	aTable	
	d that returns a new DataTable filled with the results of the SQL statement or SELECT store ed on the previous page.	ed
Method name:	GetDataBy	
	< <u>P</u> revious <u>N</u> ext > <u>F</u> inish	Cancel

# Step2: Select the Fill A DataTable and Return a DataTable . Step3: Change the Return a DataTable method name to GetDataBy\_EMPID Step4: Press FINISH.

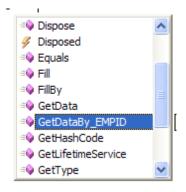
The final dataset in the designer show look like the following:

	EMPLOYEES 🔊
	ADDRESS_DATA_1
	ADDRESS_DATA_2
	BIRTHDAY
	CITY
	EMPLOYEE_ID
ę	FIRST_NAME
	LAST_NAME
	MIDDLE_INITIAL
f	POSTAL_CODE
ę	SEX
	STATE
ę	STATUS_CODE
5	EMPLOYEESTableAdapter
śQL	Fill,GetData ()
SQL	FillBy,GetDataBy_EMPLOYEE_ID (:EMPLOYE

**Step5:** To execute the above functionality in your code, instantiate the table adapter as follows:

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

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



Step6: The methods provide by the TableAdapter may be used within your code

```
// Get Employee details by employee_id
dataGridView1.DataSource=adapt.GetDataBy_EMPID(00164);
```

Show Navigation

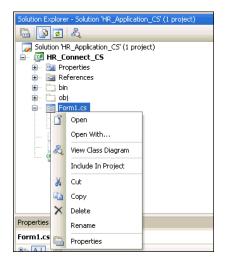
# **Chapter 9** Copying a Form

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	plorer - HR_Connect_CS
	\$
	on 'HR_Application_CS' (1 proj
	D Connect CS Build
🖃 🎬	
	Rebuild
- -	Clean
	Publish
	Add 🔸
	Add Reference
	Add Web Reference
ಪ್	View Class Diagram
	Set as StartUp Project
	Debug +
Proper 👗	Cut
	Paste
×	Remove
Poli	Rename
Pro	Unload Project
Pro 🛅	Properties

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

Solution Explorer - Solution 'HR_Application_CS' (						
Solution 'HR_Application_CS' (1 project)						
📄 📴 HR_Connec	🖃 📴 HR_Connect_CS					
	🖬 🔤 Properties					
🗄 🔤 Referenc	es					
🗄 – 🛅 bin						
🗄 📄 obj						
Copy of f						
E- 🔳 Forr	Open					
	Open With					
- D HR_ &	View Class Diagram					
HR_	Include In Project					
*	Cut					
	Сору					
×	Delete					
Properties	Rename					
Copy of Form1.	Properties					

**Step4:** Right-click on Form1.cs, and select **Include In Project**.

Solution Explore	r - Solution 'HR_Applicatio					
🖹 🗿 👩 🕹						
Solution 'HR_Application_CS' (1 proj						
IR_Connect_CS						
	operties ferences					
🗄 🔂 bin						
😐 🗂 ob	j					
	rm1.cs					
Form1.Designer.cs						
Form1.resx     Form2.cs						
Open						
	Open With					
ಪ್	View Class Diagram					
	Include In Project					
¥	🔏 Cut					
Properties Copy						
Form2.cs × Delete						
₿₽	Rename					
Custom	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.

The examples in this chapter are based on Visual Studio 2012. If you are using another version of Visual Studio, the user interface may have changed. Please refer to your Visual studio documentation to see how the same objectives may be achieved.

This chapter contains:

- <u>Creating new Windows Form Application</u>
- <u>Reference the Oracle Rdb EntityFramework Provider</u>
- Adding an Entity Data Model
- Create a new Entity
- <u>Creating Complex Types</u>
- <u>Generating Database Schema from Model</u>

# **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.

# **10.2 Reference the Oracle Rdb EntityFramework Provider**

Before you can use EntityFramework 6 with Oracle Rdb you should install the latest version of the Oracle Rdb Entity Framework provider using the Microsoft NuGet package installer.

Microsoft NuGet is the software development package manager for Microsoft .NET. The NuGet client tools produce and consume these packages, including non-Microsoft software, such as ORDP.NET.

The <u>NuGet Gallery</u> is the central repository for hosting and consuming packages.

Starting with release 7.3.4.0.0 ORDT, you can use the Oracle Rdb Entity NuGet package to configure your Microsoft Projects to use ORDT in conjunction with Entity Framework release 6 (EF6).

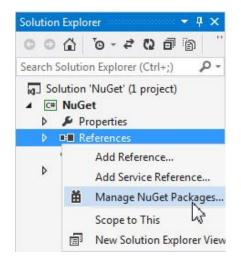
## Note: Currently the Oracle Rdb Entity NuGet package only configures your projects for use with EF6, you still must manually download and install ORDT. This may change in future releases of ORDT.

The following steps show how to use NuGet to help configure your project to use ORDT with EF6.

Step1: In Visual Studio IDE, select View > Solution Explorer from the main menu.

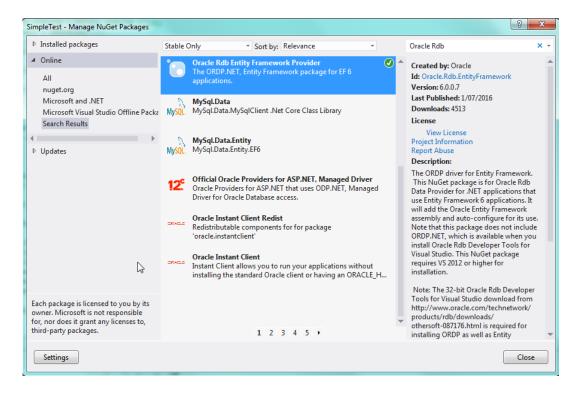
VIEV	V PROJECT	BUILD	DEBUG	TEAM	TOOL
$\diamond$	Code		F	7	
3	Open				
	Open With				
	Solution Explor	rer N	C	trl+Alt+L	
***	Team Explorer	43	C	trl+ Ctrl-	+M
	Server Explorer		C	trl+Alt+S	
R	Architecture Ex	plorer	С	trl+ Ctrl-	+R
Ð	SQL Server Obj	ect Explor	er C	trl+ Ctrl-	+S

# **Step2:** Right-click **References** in the Solution Explorer and choose **Manage NuGet Packages**.



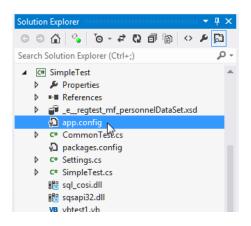
**Step3:** In the **Manage NuGet Packages** window, ensure that you have selected **nuget.org** in the left panel, under the **Online** tab. On the upper right side of the window, there is a search bar. Search for **Oracle Rdb**.

Select Oracle Rdb Entity Framework Provider (with the Id Oracle.Rdb.EntityFramework) in the search results and click Install.



The package installation will not download any software, but it will setup your environment so that you may use the NuGet package to configure your projects. Once the package has been installed, a green tick mark will be placed against the product name.

**Step4:** To confirm that ORDP.NET and Entity Framework have been automatically configured, from the Solution Explorer window, open **App.config**.



You will see that ORDP.NET and Entity Framework configuration information has been added to the project.

app.com	fig* 😐 🗙 NuGet.config	NuGet Gallery	Package.nuspec	Variables.wxi	Ident.cs	Ŧ
	<pre></pre>					÷
é	<startup></startup>					
	<pre><supportedruntime pre="" ve<=""></supportedruntime></pre>	rsion="v4.0" sku='	".NETFramework,Vers	ion=v4.5" />		
_						
Ė	<entityframework></entityframework>					
Ė	<providers></providers>					
	<provider invarian<="" td=""><td>tName="System.Data</td><td>a.SqlClient" type="</td><td>System.Data.Entit</td><td>ty.SqlServer.SqlPro</td><td>oviderSe</td></provider>	tName="System.Data	a.SqlClient" type="	System.Data.Entit	ty.SqlServer.SqlPro	oviderSe
	<pre></pre>	tName="Oracle.Data	Access.RdbClient"	type="Oracle.Data	Access.RdbClient.R	RdbProvi
	<pre></pre>					
	<pre><defaultconnectionfa< pre=""></defaultconnectionfa<></pre>	ctory type="Oracle	e.DataAccess.RdbCli	ent.Entity.RdbCor	nnectionFactory, Or	acle.Da
_						
Ė	<connectionstrings></connectionstrings>					

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

# **10.3 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	¢#	Class	Visual C# Items	^	Type: Visual C# Items A project item for creating an ADO.NET Entit
General Web	S C#	Interface	Visual C# Items		Data Model.
Windows Forms WPF		Windows Form	Visual C# Items		
Reporting Workflow		User Control	Visual C# Items		
nline Templates		Component Class	Visual C# Items		
	<b>~</b>	User Control (WPF)	Visual C# Items		
	7.4x	About Box	Visual C# Items		
	92	ADO.NET Entity Data Model	Visual C# Items		
	<b>A</b> 2	ADO.NET EntityObject Generator	Visual C# Items		
		ADO.NET Self-Tracking Entity Generator	Visual C# Items		
		Application Configuration File	Visual C# Items		
		Application Manifest File	Visual C# Items		
	C#	Assembly Information File	Visual C# Items		
		Bitmap File	Visual C# Items	~	
ame: Model1.e		Lise Liseram	Veral C # those		

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 Data Model Wizard	? 🗙
Choose Model Contents	
What should the model contain?	
Generate from Empty model	
Generates the model from a database. Classes are generated from the model when the project is	
compiled. This wizard also lets you specify the database connection and database objects to inclu in the model.	
< Previous Next > Finish Cance	*

**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?
O 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.ssd][res://*/Model2.ssd];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
<pre></pre>

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 also we have chosen to select all tables, just as a part of example. But, you have an option to select any specific table.

**Step6: Check** the box against **Table**s node.

Entity Data Model Wizard	? 🗙
Choose Your Database Objects	
Which database objects do you want to include in your model?	
Cons (NULL)     CustomersDb (NULL)     Grave Views     Stored Procedures	
Pluralize or singularize generated object names	
Include foreign key columns in the model	
Model Namespace:	
Model	
< Previous Next > Finish Car	ncel

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

Visual Studio will generate the model then display it.

	itomersDb 혽	😤 Con 🛞	
E Prope		Properties     Profestid	
	stomerId	ConsType	
Pp	stname 1	* Constitumber	
😑 Navig	ation Properties	<ul> <li>Navigation Properties</li> </ul>	
Co	ns	CustomersDb	

# 10.4 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 Project→ Add New Data Source.

You will be presented with the Data Source Configuration Wizard.

Data Source Configuration Wizard	? 🗙
Choose a Data Source Type	
Where will the application get data from?	
Database Service Object SharePoint	
Lets you choose objects that can later be used to generate data-bound controls.	
< 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 chosen **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?	: is missing from a t.
Image: Second system assemblies	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.

Data Sources	▼ -¤ X
🛅 🌆 🚏 📭	
CustomersDb ♥ B ↓ CustomersD ♥ B ↓ CustomersD B ↓ Firstname M Firstname M Sumame D thess B ♥ Cons B ♥ CustomersDbs	

# 10.4.1 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**.

10 ba 17 0s						
□     □     □     Conserverable     ►       □     □     □     Conserverable     ►       ■     ■     Customerable       ■     ■     ■     Extranse       ■     ■     Surname       ■     ■     Enbides       ■     ●     Conse       ■     ●     Conserve       ■     ♦     Conserve       ■     ♦     CustomersDbs	▶ N   Ф X	*	ConHistId	ConsType	ConsYear	
		<	W		>	
		<u>×</u>				

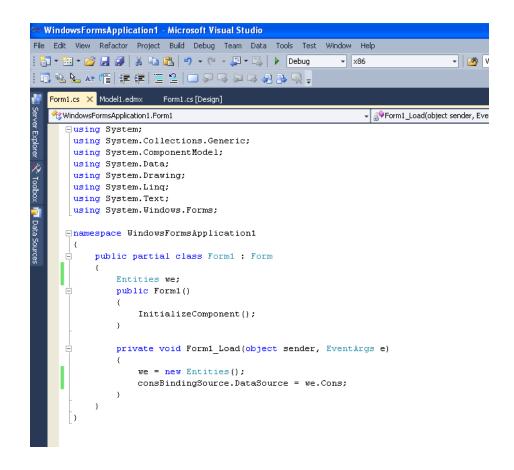
**Step2: Save** and **rebuild** the solution before continuing.

# 10.4.2 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				
i 🛛 🖣 🔤 of 3 📄 🛃 🛱 🔀 🔙				
		ConHistId	ConsType	ConsYear
	•	21	p1	1230
		22	p1	1230
		23	p1	1230
	*			
	<	Ш		>

# 10.5 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**.

stalled 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
Data General Web	3 <b>(1</b>	Interface	Visual C# Items	Data Model.
Windows Forms WPF		Windows Form	Visual C# Items	
Reporting Workflow		User Control	Visual C# Items	
nline Templates	-	Component Class	Visual C# Items	IJ
	<b>~</b>	User Control (WPF)	Visual C# Items	
	<b>*</b>	About Box	Visual C# Items	
	93	ADO.NET Entity Data Model	Visual C# Items	
	<u></u>	ADO.NET EntityObject Generator	Visual C# Items	
		ADO.NET Self-Tracking Entity Generator	Visual C# Items	
		Application Configuration File	Visual C# Items	
	2	Application Manifest File	Visual C# Items	
	ď	Assembly Information File	Visual C# Ibems	
		Bitmap File	Visual C# Items	
ame: Model1.ee	dmx	Class Disaram	Stead of the Beam	1

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.

Entity Data Model Wizard	<u>?</u> ×
Choose Model Contents	
What should the model contain?	
Generate from Empty model database	
Creates an empty model as a starting point for visually designing a conceptual model from the toolbox. Classes are generated from the model when the project is compiled. You can specify a database connection later to map the conceptual model to the storage model.	
< Previous Next > Finish Car	ncel

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



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

🔫 Emp_	Table	8
😑 Propertie	s	
₩Emp_ Last_ First_ Title	Name	
😑 Navigatio	on Proper	ties

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

Model3.ed	mx	Model1.edmx Program.cs	Start	Page		
				_		
		Add	•		Entity	
		Diagram	•		Association	
		Zoom	•		Inheritance	
		Grid	•		Function Import	
		Scalar Property Format	•			,
		Select All				
	æ	Mapping Details				
	A	Model Browser				
		Update Model from Database				
		Validate				
	Ŀ,	Properties		s you	visualize and design you	r Entity Data Model.
	_					
		Create new entities	in the	model	by dragging items from t	he <u>Toolbox</u> ,

The Add Entity form will be displayed.

Add Entity				?×
Properties				
Entity name:				
Entity1				
Base type:				
(None)				~
Entity Set:				
Entity1Set				
Key Property				
🔽 Create key proj	perty			
Property name:				
ld				
Property type:				
Int32				~
		<u>0</u> K	Cance	el

**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	~
	<u> </u> <u> </u> <u> </u> <u> </u>

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

1odel3.edmx* Model1.edmx	Program.cs Start Page	
C Emp_Table Scalar Properties Emp_Id Navigation Prope	Add       Rename       Cut       Copy       Delete       Collapse       Table Mapping       Stored Procedure Mapping       Show in Model Browser       Update Model from Database       Validate       Poperties	Scalar Property Association Inheritance Function Import

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

	F	Properties		🚽 🕂 🗙
		Model3.Emp_Table.Last	Name Property	-
😪 Emp_Table 🛛 🔕				
		Concurrency Mode	None	
Scalar Properties		Default Value		
	Ð	E Documentation		
🎦 Emp_Id		Entity Key	False	
st_name		Getter	Public	
		Name	Last_Name	
Navigation Properties		Nullable	True	
		Setter 😽	Public	
		Туре	String	*

# **10.6 Creating Complex Types**

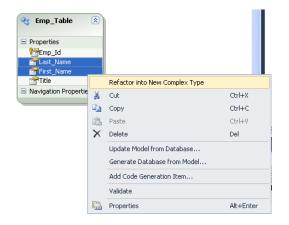
Complex types are non-scalar properties of entity types which enable scalar properties to be organized within entities. This makes easier to work with objects, allowing the grouping of related properties in Entities.

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

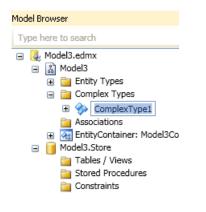
To create a Complex type:

**Step1:** Hold the **CRTL** key and **Select** the properties in the data model that you want to group together

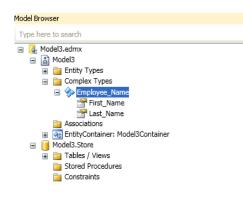
Step2: Right click in the context menu, select Refactor into New Complex Type



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.

Model3.Emp_Table.Emp_Name Property					
₿∎ <b>2</b> ↓					
🖃 Code	Generation				
Getter		Public			
Setter		Public			
🗄 Gener	al				
Concur	rrency Mode	None			
🗄 Docum	entation				
Name		Emp_Name			
Туре		Employee_Name			

# 10.6.1 Using the Complex Type created

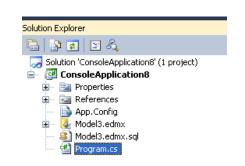
The Complex type you created may be used as follows:

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

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

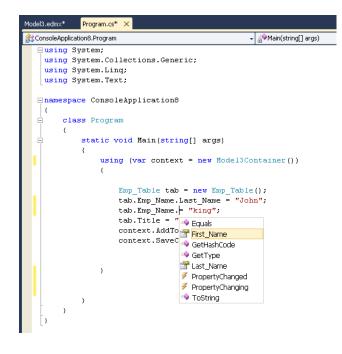
**Step2:** Design a new Entity with Complex Types. See <u>Create a new Entity</u>, <u>Creating Complex Types</u>

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

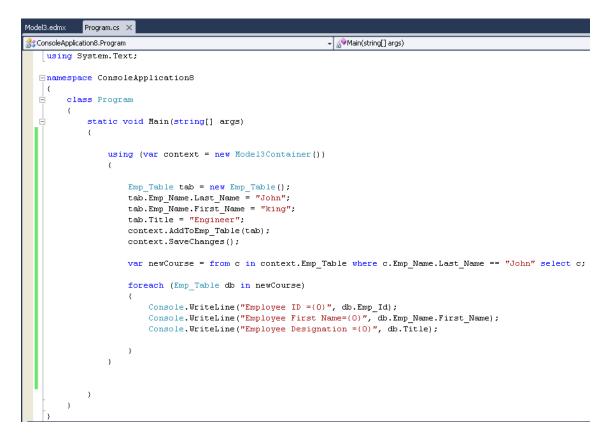


The Console Application's solution explorer looks as follows:

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



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	

# **10.7 Generating Database Schema from Model**

Beginning with Visual Studio 2010 and Entity Framework V3.5, you can generate a database schema from an existing Entity Data 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:

🖏 Solution Explorer 🛛 🖓 Team Explorer	💐 Class View				
Properties	<del>▼</del> ╀ ×				
Model3 ConceptualEntityModel	-				
8∎ 2↓   =	•				
Code Generation Strategy	Default 🔨				
Lazy Loading Enabled	True				
Connection					
Connection String	metadata=res://*/Model3.csdl res://*/Mc				
Metadata Artifact Processing	Embed in Output Assembly				
Database Script Generation					
Database Generation Workflow	TablePerTypeStrategy.xaml (VS)				
Database Schema Name	dbo				
DDL Generation Template	SSDLToOracleRdb.tt (VS) 🛛 💌				
🗆 Schema	SSDLToSQL10.tt (VS)				
Entity Container Access	SSDLToOracleRdb.tt (VS)				
Entity Container Name	modersconcamer				
Namespace	Model3				
Pluralize New Objects	True 👱				
DDL Generation Template The T4 text template used to transform SSDL to DDL in the Generate Database wizard.					

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

😤 Emp_Table 🛛 🛞				
Properties				
₩Emp_Id ☆Title ☆Emp_Name				
Navigation Properties				
		Add		•
		Diagram		•
		Zoom		•
		Grid		•
		Scalar Property Format		•
		Select All		
	æ	Mapping Details		
	2	Model Browser		
		Update Model from Database		
		Generate Database from Model		
		Add Code Generation Item		
		Validate		_
		Properties	Alt+Enter	

**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	
< Previous Next > Finish Cance	3

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

Show Navigation

# **10.8 Using Fluent API**

Fluent API provides a mechanism to change the default behavior of the database SQL generated by Entity Framework when your POCO objects are transformed into Oracle Rdb table definitions.

The following example shows how to use fluent API with ORDT.

**Note:** This example assumes you are already familiar with using Entity Framework **DbContext** to create code first data objects.

To begin with we will create our new model :

## Visual C#:

```
public class ComicsEntities : DbContext
  #region Properties
  public DbSet<Series> Series { get; set; }
  public DbSet<Issue> Issues { get; set; }
  #endregion
  #region Methods
  protected override void OnModelCreating(
    System.Data.Entity.ModelConfiguration.ModelBuilder modelBuilder)
  {
   base.OnModelCreating(modelBuilder);
    // the fluent code will be added here
  }
  #endregion
}
public partial class Issue
  #region Properties
  public int IssueID { get; set; }
  public string Title { get; set; }
  public DateTime Time { get; set; }
 public int SeriesID { get; set; }
  public virtual Series Series { get; set; }
  #endregion
}
public class Series
  #region Properties
```

```
public int SeriesID { get; set; }
public string Name { get; set; }
public DateTime StartDate { get; set; }
public string Publisher { get; set; }
public virtual ICollection<Issue> Issues { get; set; }
#endregion
}
```

Once the model has been created we can use fluent API to configure the model further. Generally this is done by overriding the standard **OnModelCreating** method of **DbContext**:

### Visual C#:

```
protected override void OnModelCreating (ModelBuilder modelBuilder)
{
 base.OnModelCreating(modelBuilder);
  modelBuilder.Entity<Series>().
   Property(s => s.Name).
   IsRequired().
   HasMaxLength(50);
 modelBuilder.Entity<Series>().
    Property(s => s.SeriesID).
    HasDatabaseGenerationOption (DatabaseGenerationOption.None);
 modelBuilder.Entity<Series>().
    HasMany(s => s.Issues).
    WithRequired(i => i.Series).
   HasForeignKey(i => i.SeriesID);
  modelBuilder.Entity<Series>().
    Ignore(s => s.PublisherID);
 modelBuilder.Entity<Issue>().
   Property(i => i.Title).
   IsRequired().
   HasMaxLength(50).
   HasColumnName("Name");
}
```

In the code above, we can use **Fluent API** to start to configure our Comics model.

We have set the maximum size of the Series Name column to 50 characters.

By default the ID fields are database-generated, that is, the field will be declared as an Identity column in the table. In the above example this behaviour has been suppressed for the **SeriesID** column.

As we want a one to many relationship between **Series** and its **Issues** we have declared the **Series** entity to have the **HasMany** property.

As we are not interested in storing the **Publisher** information in the database, we have suppressed column generation using **Ignore**. We have also chosen to change the column name that will be used in the database for the **Issue title**.

Now we can tie these components together in a small program that will generate the appropriate database objects for us, based on our model and the fluent API modifications:

# Visual C#:

```
class Program
 static void Main(string[] args)
   using (ComicsEntities context = new ComicsEntities())
     var series = new Series
      {
       SeriesID = 1,
       Publisher = "NoWhereComics",
       Name = "FabAndFamous",
       StartDate = DateTime.Now
      };
     var issue = new Issue
      {
       Time = DateTime.Now,
       Title = "Entity Framework Funnies",
       Series = series
     }:
     context.Series.Add(series);
     context.SaveChanges();
   }
 }
}
```

Running this example will generate a new database by the name **ComicsEntities** with the configurations provided in the **OnModelCreating** method. By default, the name of the database will be taken from the context name. To change the generated database name you need to change the constructor **DbContext** and pass in a string parameter specifying the required name:

## Visual C#:

```
public CommicsEntities() : base("MyComicDatabase")
```

{ }

**Note:** Please see the section <u>Creating and Dropping Oracle Rdb Databases</u> if you intend to create or drop databases.

# 10.8.1 Database Creation and Removal

Fluent API provides methods to control the creation and deletion of database instances. In the prior example the database will be created for you automatically, however you may wish to override this by specifying a **Database Initializer**.

The database initializer is called when the given DbContext type is used to access a database for the first time. The default strategy for Code First contexts is an instance of **System.Data.Entity.CreateDatabaseIfNotExists<TContext>**, thus in the prior example a database would be created if it did not already exist.

The following example shows how to change the database initializer:

## Visual C#:

In this example the database will be recreated each time the application is run.

Please see the Microsoft Entity Framework documentation for more information on database contexts and initialization.

**Note:** See the following section <u>Creating and Dropping Oracle Rdb Databases</u> if you intend to create or drop databases.

# **10.8.2 Creating and Dropping Oracle Rdb Databases**

Currently, the creation and deletion of databases is only supported when using Thin connectivity within ORDP, that is, only connections using JDBC servers will accept these database operations.

By default JDBC servers do not allow CREATE and DROP database operations. The following requirements must be satisfied before these database operations may be carried out:

- The application is using Thin Connectivity.
- The JDBC server is running Oracle JDBC for Rdb Release 7.3.4.0.0 or later.
- The JDBC server is configured to allow your username access to server root and the CREATE or DROP database operations.
- The username provided has adequate OpenVMS privileges and quotas to carry out the operation.
- The username provided has appropriate Oracle Rdb privilege and authorization to carry out the operation.

Please see your **Oracle JDBC for Rdb User Guide** for more information on support of CREATE and DROP database.

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