

Getting to Know the Tools

One main limitation of the SQL CE 2.0 database released in 2002 was the need to manage the database either using a query analyzer tool on mobile devices or by running a program to manipulate data on a device. To allow rapid application development, Microsoft has added tight integration between SQL Server Compact Edition 3.x, SQL Server 2005, and Visual Studio 2005.

Using SQL Server Management Studio, developers and administrators can manage the SQL Server Compact Edition database located on a device or on a desktop. You can design tables, columns, constraints, and indexes using SQL Server Management Studio.

Similarly you can use Visual Studio features to create, design, and manipulate the SQL Server Compact Edition database. Visual Studio also helps you to deploy a database on a device.

In this chapter you will be familiarized with the available tools. You will also learn how to use SQL Server Management Studio and Query Analyzer to manage and manipulate the SQL Server database. You will become familiar with the Visual Studio interface and will learn how to include SQL Server Compact Edition in your project. You will also learn how to set up a connection with the SQL Server Compact Edition database and finally how to deploy the application on platforms.

CHAPTER 3

IN THIS CHAPTER

- Using SQL Server Management Studio
- Using the Query Analyzer
- Using Visual Studio
- Summary

Using SQL Server Management Studio

SQL Server Management Studio provides an integrated environment to manage the SQL Server family of databases. Using SQL Server Management Studio, you can manage SQL Server 2005 database, SQL Server Compact Edition 3.x database, Integration Services, Analysis Services, and Reporting Services.

SQL Server Management Studio is installed on your computer along with the SQL Server 2005 installation. By default the `SqlWb` executable is installed at `C:\Program Files\Microsoft SQL Server\90\Tools\Binn\VSShell\Common7\IDE`.

NOTE

SQL Server Management Studio is also known as Management Studio, SSMS, or Work Bench. Many developers also use the term Work Bench for SQL Server Management Studio.

SQL Server Management Studio has replaced Enterprise Manager and Query Analyzer. Using SQL Server Management Studio, you can also manage SQL Server 7 and the SQL Server 2000 database.

The biggest advantage of SQL Server Management Studio is that you do not need to learn a new interface for managing the SQL Server Compact Edition database. If you are already using SSMS for managing SQL Server, you can use the same interface for managing the SQL Server Compact Edition 3.0 database.

Starting SQL Server Management Studio

You can start the SQL Server Management Studio by clicking on Start | All Programs | Microsoft SQL Server 2005 | SQL Server Management Studio. Using SSMS you can create a project and a solution. The project contains a connection to a database and the corresponding SQL scripts. A container solution contains related projects. A project provides a mechanism to organize files, not the database objects. To create a new project and solution use the File | New menu option and specify the project and the solution name.

The SSMS user interface provides two types of windows—component windows and document windows. Component windows display windows listed under the View options such as Object Explorer, Registered Server, Solution Explorer, and so on. Document windows are used for queries, scripts, and files. The SQL Server Management Studio windows are shown in Figure 3.1. You can use various options to customize SSMS windows by using the Tools | option.

This section will describe these windows. The Bookmark feature is a new feature that you can use to create a shortcut to a line in a project file. To create a bookmark, choose Toggle Bookmark from the Edit menu.

Getting Connected to SQL Server Compact Edition Database

When you start SQL Server Management Studio, it starts with the Connection dialog box. In this dialog box you will specify that you wish to connect to SQL Server Compact

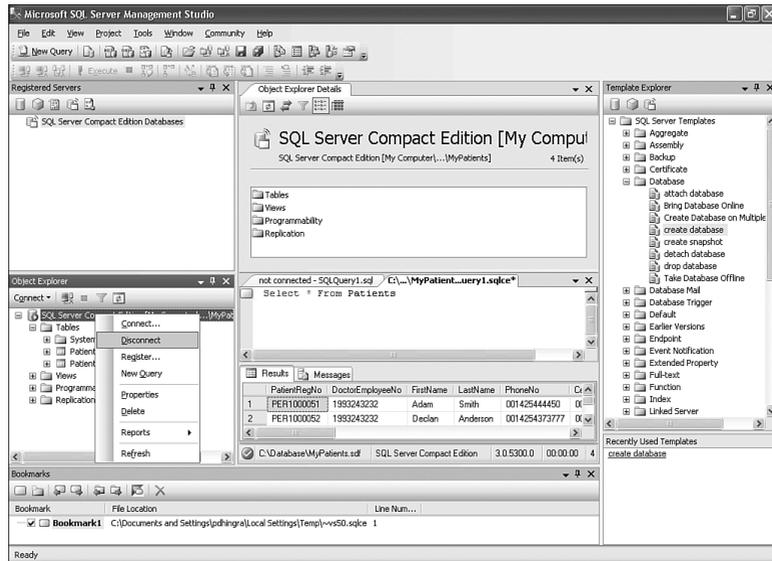


FIGURE 3.1 SQL Server Management Studio Windows

Edition database. You need to specify the path of the SQL Server Compact Edition database file and initiate the connection. To connect to the SQL Server Compact Edition database follow the steps given below:

1. Start the SQL Server Management Studio. The Connect to Server dialog box will appear as shown in Figure 3.2.
2. Select SQL Server Compact Edition Server Type.



FIGURE 3.2 Connect to SQL Server CE

3. Specify the name of the SQL Server Compact Edition database file in order to open an existing database. You can choose Browse for More options to browse the database file.
4. Click on the Connect button to connect to the database.

This example demonstrates a simple scenario—opening an SQL Server Compact Edition database. Using the dialog box shown in Figure 3.2, you can open a password-protected encrypted database and create a new SQL Server Compact Edition database.

Using Object Explorer

Object Explorer is an SQL Server Management Studio component. Using this component you can connect to SQL Server, Integration Services, Analysis Services, Reporting Services, and an SQL Server Compact Edition database.

Object Explorer is displayed as part of the SQL Server Management Studio user interface. If Object Explorer is not visible, click on the View | Object Explorer menu option.

As shown in Figure 3.3, the Object Explorer window displays a tree structure view for all the objects. For each server type, Object Explorer displays a different set of nodes. For an SQL Server Compact Edition database, it displays Tables, Views, Programmability, and Replication nodes.

Figure 3.1 shows a connection to both SQL Server database and the SQL Server Compact Edition database. All available connections are shown in a tree view. You can expand the tree nodes to see underlying objects. Object Explorer can display up to 65,536 objects.

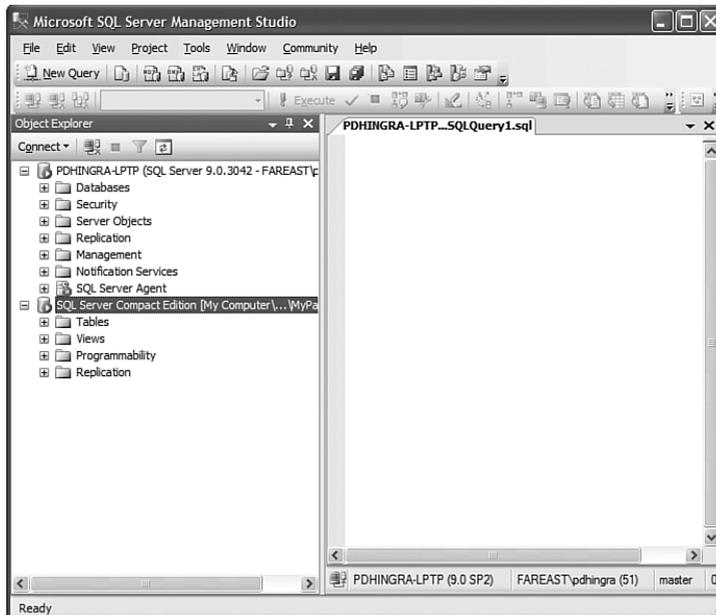


FIGURE 3.3 Object Explorer

You can right click on any node revealing a list of operations that can be performed on the object. For SQL Server Objects, the Filter option is available to allow filtering of objects. For SQL Server Compact Edition, the Filter option is not available.

Disconnecting from the SQL Server Compact Edition Database

To disconnect from the SQL Server Compact Edition database, simply select the database. Right click and then click on the Disconnect option.

After disconnecting from the SQL Server Compact Edition database, Object Explorer is refreshed. You can use the Tools | Options menu option to customize a window layout of SSMS as shown in Figure 3.4.

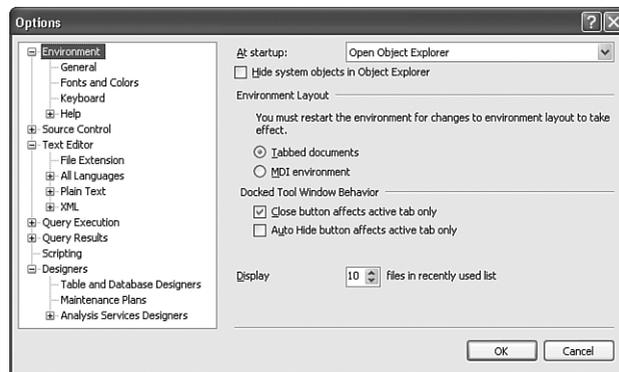


FIGURE 3.4 Object Explorer

Connecting to SQL Server Compact Edition Database from Object Explorer

To reconnect to the SQL Server Compact Edition database, click on the Connect button in Object Explorer. You will be shown options to connect to SQL Server Database Engine, Analysis Services, Integration Services, Reporting Services, and the SQL Server Compact Edition database. Select the SQL Server Compact Edition database option.

You will be shown the Connect to Server dialog box. Specify the path of the SQL Server Compact Edition database as you did earlier, thereby connecting to the SQL Server Compact Edition database.

Registering a Database Server

Using the Register Server option you can store the connection information. While connecting to an SQL Server Compact Edition database that is already registered, you do not need to resupply all the connection parameters. The Register Server Option is useful for connecting to databases that you connect to frequently. Use the Register Server Option to store connection information for SQL Server Compact Edition, SQL Server, Analysis Services, Integration Services, and Reporting Services.

At the time of registration, you store the connection information in SQL Server Management Studio. To store this information, use the Register a Database Server option.

1. To register a server, select the SQL Server Compact Edition database in SQL Server Management Studio Object Explorer.
2. Right click on database and click the Register option as shown in Figure 3.5.

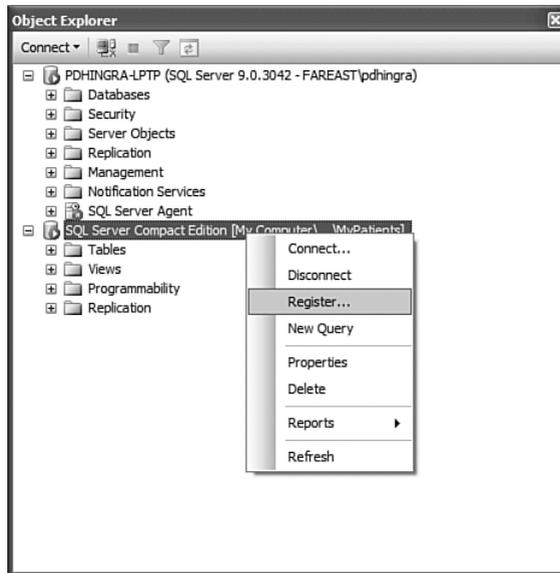


FIGURE 3.5 Register a Database Server

3. In the Register Server dialog box, specify where you want to place the database in the Server Group box. By default the SQL Server Compact Edition database is selected in Figure 3.6.
4. To create a new group, click on the New Group button.
5. In the New Server Group dialog box, enter a Group name and Group description as shown in Figure 3.7.
6. Click Save.
7. The new Group name is entered under SQL Server Compact Edition database.
8. Click Save.

Connecting to a Registered Database

You can view all registered servers and connect to one of the registered databases. You can access all the Registered Servers by clicking on the View | Registered Servers option.

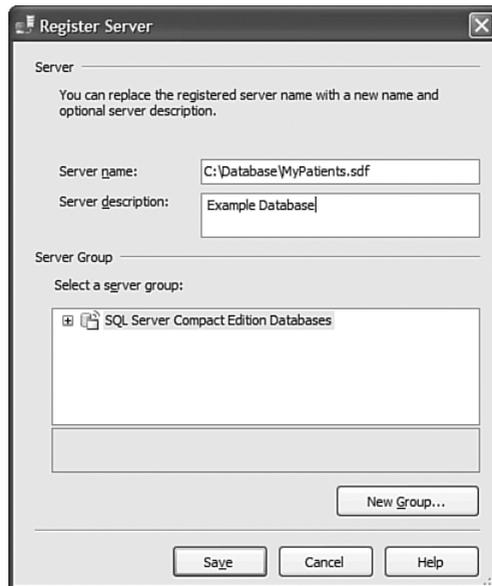


FIGURE 3.6 Specify a Server Name

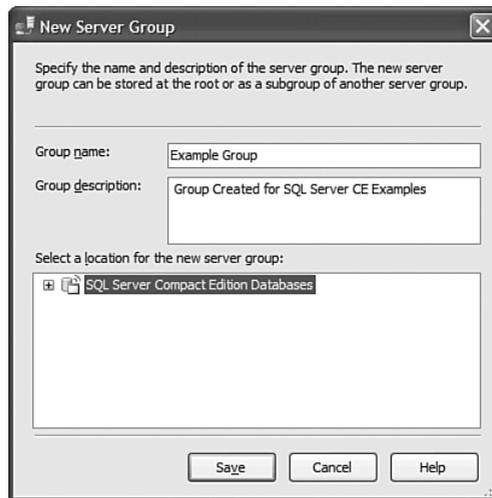


FIGURE 3.7 Specify a New Server Group

Clicking on Registered Server will start the Registered Server Explorer tab. Register Server Explorer has five tabs in the top menu. The five tabs correspond to five types of database server that you can connect using SQL Server Management Studio. Earlier in this chapter you learned that when using SQL Server Management Studio you can manage the SQL Server 2005 database engine and Integration Services. As shown in Figure 3.8, select the tab for SQL Server Compact Edition to access the Example Registered Server Group that you created earlier.

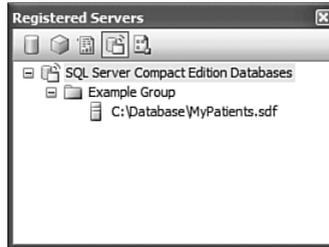


FIGURE 3.8 List of Registered Servers

Select your Database Group and Database and click to connect.

Using Tables Node

You will be using the Tables node to define the tables and columns structures. You can right click on the Tables node and use options to create new tables. You can expand the Tables node by right clicking on any existing table and view/modify the properties of the table.

You can also expand the table to see the columns. You can see the column properties by right clicking on all columns.

Using Views Node

SQL Server Compact Edition provides INFORMATION_SCHEMA views to help you obtain metadata information about the SQL Server Compact Edition database. The INFORMATION_SCHEMA view gets included in each SQL Server Compact Edition database. You can click on views folders to see INFORMATION_SCHEMA views available. When you click on Views, your result will look similar to one shown in Figure 3.9.

Tables

The Tables view contains one row for each table that is accessible to the current user.

Columns

The Columns view contains one row for each column that is accessible to the current user.

Indexes

The Indexes view contains one row for each index that is accessible to the current user.

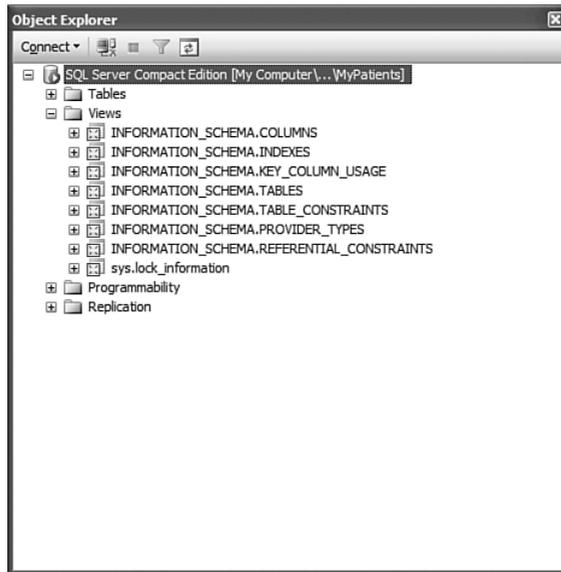


FIGURE 3.9 Information Schema Views

Key_Column_Usage

The Key_Column_Usage view contains one row for each column that is defined as key in the current database.

Table_Constraint

The Table_Constraints view contains one row for each table constraint in a database.

Referential_Constraints

The Referential_Constraints view contains one row for each foreign key constraint in a database.

Provider_Types

The Provider_Types view contains the data type supported in SQL Server Compact Edition.

In the upcoming chapters you will learn how to utilize these views for problem solving.

Using the Programmability Node

The programmability node is more useful in SQL Server 2005. In SQL Server 2005 this node contains subnodes for Stored Procedures, User Defined Data Types, etc. In the SQL Server Compact Edition database, Programmability nodes display the data type available in SQL Server CE.

Using the Replication Node

Merge Replication is a mechanism to synchronize data between SQL Server and SQL Server Compact Edition. In Merge Replication, SQL Server acts as Publisher of data and

SQL Server Compact Edition as Subscriber of data. In the Object Explorer of the SQL Server Compact Edition database you have a Replication node. The Replication node will have a Subscriptions subnode. This node will show the subscriptions available for SQL Server Compact Edition database. If you have set up the subscription, this node will display the subscription information as shown in Figure 3.10. You can check out the subscription properties by right clicking on the subscription.

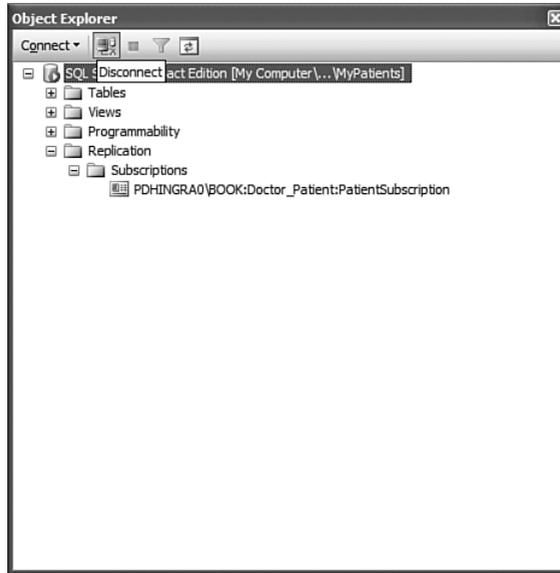


FIGURE 3.10 Replication Node

If the SQL Server Compact Edition database has not been configured to act as Subscriber, the Subscription node under Replication node will be empty.

You will learn more about Merge Replication and Subscription in Chapter 12.

Using the Query Editor

SQL Server Management Studio has a query editor. To write queries for SQL Server CE, click on either the New Query button or the File | New Query menu option. Clicking on New Query will start the new query editor or ask you to specify a connection. Query Editor can be connected or disconnected to an SQL Server Compact Edition database. Query Editor color highlights query syntax and provides tracking indicators.

You can also open existing queries using the File | Open | File... option.

From Object Explorer you can select the SQL Server Compact Edition database, right click, and select New Query to open a query editor.

The Query Editor has an upper pane in which you can write queries. The lower pane displays the results of query execution. The results displayed are read-only. In the results pane there are two tabs: Results and Messages. The Result tab displays the data after successful execution. The Message tab displays an error message if the query has errors or it displays a message indicating the number of rows impacted as a result of a query. You can click on the Message tab to see the message. The track changed indicator in Query Editor shows which lines of an SQL script are changed.

NOTE

SQL Server Compact Edition supports a subset of Transact-SQL commands. SQL Server 2005 supports complete Transact-SQL commands. Chapter 5, *Defining the Database Structure*, and Chapter 6, *Manipulating the Database*, provide the syntax and examples of Transact-SQL that are supported in SQL Server Compact Edition.

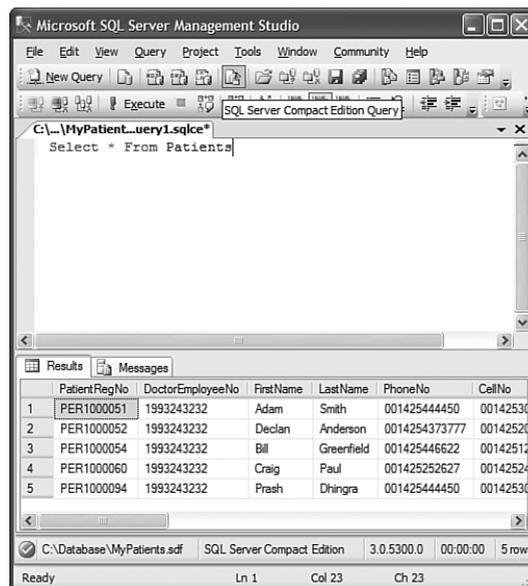


FIGURE 3.11 Query Editor

The results are displayed in the Results section as shown in Figure 3.11.

NOTE

SQL Server Management Studio provides a query builder feature to design queries. For an SQL Server database, use the Design Query in the Editor menu option. You will not see this menu option when you write queries for an SQL Server Compact Edition database. The Query Builder feature is not available for an SQL Server Compact Edition database.

Using a Graphical Execution Plan

When you write an enterprise application, it is very likely that you will want to isolate and tune performance bottlenecks. SQL Server Management Studio fulfills this key objective and helps you to understand where the most time is used in query execution. SQL Server Management Studio provides a solution to seeing the execution plan for queries. The query execution plan gives you insight and enables you to tune queries.

Using the Estimated Execution Plan

The Estimated Execution Plan option parses the query and then generates the execution plan. In this option, SQL Server Compact Edition does not execute the query but generates the execution plan. You can see the graphical representation of the execution plan with Query Display Estimated Execution plan.

Using the Actual Execution Plan

The Actual Execution plan generates the query and then shows the plan used by SQL Server Compact Edition. You can see the graphical representation of execution plan with Query | Include Actual Execution plan.

In Chapter 15, *SQL Server Compact Edition Performance Tuning*, you will learn how to exploit this feature of SQL Server Management Studio in order to debug and tune SQL Server Compact Edition database queries.

Managing the Database

Using SSMS you can create a new SQL Server Compact Edition database. Not only can you modify the properties of the database, you can also drop the database. With Management Studio you can even do additional operations such as the following:

- Verify the SQL Server Compact Edition database.
- Repair the SQL Server Compact Edition database.
- Shrink the SQL Server Compact Edition database.
- Compact the SQL Server Compact Edition database.

In this section you will also use SSMS interface to complete these operations.

Using Replication Wizards

For enterprise applications you will be synchronizing the data between SQL Server and the SQL Server Compact Edition database. The SQL Server Compact Edition database provides a built-in mechanism Merge Replication and Remote Data Access to synchronize the data with SQL Server. To implement Merge Replication you need to configure a Web Server, set up the Distributor, set up the Publication, and set up the Subscription.

Before using a Merge Replication you need to provide configuration details—central database, remote database, Snapshot agent folder, security mechanism, etc. You can set up this configuration either programmatically or by using Wizards. SQL Server Management

Studio provides tools and wizards to set up publication, subscription, and a snapshot folder. It even generates a template code that can be used for Data Synchronization. The set of wizards remains the same whether you are setting up Merge Replication between two SQL Server instances or between SQL Server and SQL Server CE database.

Using the Publication Wizard

For database synchronization between SQL Server and SQL Server CE database, SQL Server first needs to publish the data. The database instance publishing the data is called Publisher. By using the SQL Server 2005 Publication Wizard you can specify the Publisher, Publication type, snapshot folder, and articles to publish.

Using the Subscription Wizard

The SQL Server CE database synchronizing data with backend SQL Server is called Subscriber. The subscription wizard allows you to create and manage subscriptions for SQL Server and SQL Server CE databases. You can use the properties dialog box for modifying properties of existing subscriptions.

Using the Configure Web Synchronization Wizard

You need to set up a Web Server for both Merge Replication and Remote Data Access. SQL Server Management Studio provides a Configure Web synchronization wizard. This wizard helps you to specify the Web Server, virtual directory, and authentication mechanism.

To use Web Synchronization Wizard, Publication Wizard, and Synchronization Wizard, you first need to learn the Merge Replication and Remote Data access features. You will learn how to use these wizards in Chapter 12, *Synchronizing Data with Merge Replication*.

NOTE

You can run the Configure Web Synchronization Wizard from SSMS or by running the executable `Connwiz30.exe`.

Using the Query Analyzer

You have probably already used the SQL Query Analyzer tool to execute a query against an SQL Server 2000 database. Microsoft developed a device version of Query Analyzer, similar to SQL Server Query Analyzer, to manage the SQL CE 2.0 database on a device and SQL Server Query Analyzer functionality is extended to devices too. With SQL Server Compact Edition 3.0 you can manage the database using SQL Server Management Studio on a desktop. With SQL Server Compact Edition 3.0 you also get the Query Analyzer tool to manage the database on a device and on an emulator.

Query Analyzer allows you to not only execute queries but it also allows you to view database object information and manipulate database objects.

Installing the Query Analyzer

On a desktop, you have SQL Server Management Studio tool to manage an SQL Server Compact Edition database. On a device, you will use the SQL Server Compact Edition Query Analyzer tool. The Query Analyzer gets installed when you install SQL Server Compact Edition on a device or on an emulator. If you are a developer and deploying an application using Visual Studio in debug mode, both SQL Server Compact Edition and Query Analyzer are installed.

You can also manually install SQL Server Compact Edition and Query Analyzer onto a device. To install the Query Analyzer on a device you should install the .NET Compact framework and the SQL Server Compact Edition database. The procedures for deploying .NET Compact Framework and SQL Server Compact Edition database is explained in Chapter 2, *Platform Support and Installation*.

Getting Connected to the SQL Server Compact Edition Database

Query Analyzer is a graphical tool that manages an SQL Server Compact Edition database on a device and on an emulator. With SQL Server Compact Edition 3.x you can use Query Analyzer version 3.x. As discussed earlier, SQL CE 2.0 also provides support for Query Analyzer. Query Analyzer for SQL CE 2.0 and SQL Server Compact Edition 3.x are different executables. Both versions can exist on the same device. Query Analyzer for SQL CE 2.0 will not connect to an SQL Server Compact Edition 3.x database.

Start the Query Analyzer on a device or emulator by clicking on Start | Query Analyzer as shown in Figure 3.12.



FIGURE 3.12 Start Query Analyzer

NOTE

You need to use Query Analyzer for SQL Server 3.0 to open and manage an SQL Server Compact Edition 3.x database. When you click on an .sdf file on a device or emulator, Query Analyzer for SQL Server Compact Edition opens the database. In case you have installed Query Analyzer for SQL CE 2.0 and SQL Server Compact Edition 3.x, the Query Analyzer installed last will open the .sdf file.

Figure 3.13 shows the layout of Query Analyzer. It has four tabs—Objects, SQL, Grid, and Notes.



FIGURE 3.13 Connecting the Database

1. To connect to the SQL Server Compact Edition database using Query Analyzer, click on the Connect button inside the rectangle as shown in Figure 3.13.
2. Specify the path of the database and click Connect as shown in Figure 3.14.

The database is open. It has Objects, SQL, Grid, and Notes tabs as shown in Figure 3.15. Under the Objects tab you can browse the database objects. Under the SQL tab you can write SQL queries. You can view the results of SQL queries under the Grid tab. The Notes tab shows remarks such as errors in executing query, time taken to execute queries, and rows impacted.

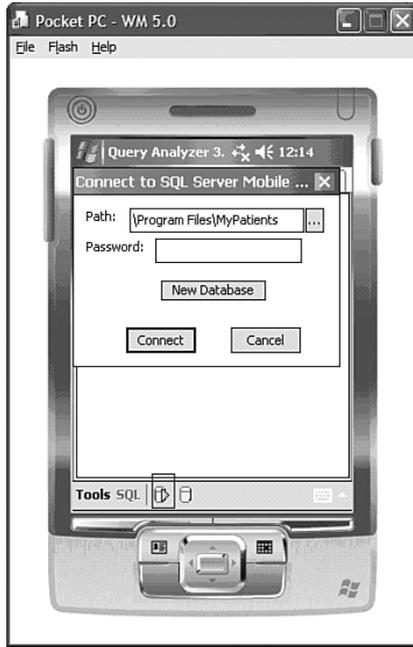


FIGURE 3.14 Specify the SQL Server CE Database File



FIGURE 3.15 Tools Menu

Using the Tools Menu

At the bottom of the Query Analyzer dialog box there is a Tools option. The Tools option is available under all four tabs. As shown in Figure 3.15, the Tools menu has subitems—Refresh, Logging, Fonts, About, and Exit.

Refresh

The Objects tab displays the tree view of the SQL Server Compact Edition database object. You can click on the Refresh menu to refresh the tree view.

NOTE

In SQL Server Management Studio, you refresh the tree view by right clicking on tree and clicking Refresh.

Logging

Logging is an interesting option. You can click on Logging to enable and disable this feature. After enabling logging, any DDL or DML operation that you do using the Object Explorer Graphical menu option is logged in the SQL tab. The SQL statement equivalent to your action is logged in the SQL tab. For example, if you use the New Table menu option, the Create Table command logs into the SQL Tab.

The statements logged or written in SQL Tab are not saved automatically. Use the Save menu option if you need to save the statements for future usage.

1. Click on the Tools | Logging option as shown in Figure 3.16.



FIGURE 3.16 Logging Option

2. Select a table. Click on the **Execute Query** button.
3. Clicking this button while a table is selected shows the content of the table as shown in Figure 3.17.
4. You can check the statement that Query Analyzer has executed. Click on the **SQL** tab.
5. The **SQL** tab shows the “**SELECT**” statement as shown in Figure 3.18. If you drop a table, column, or index, you will see an equivalent statement in the **SQL** tab.

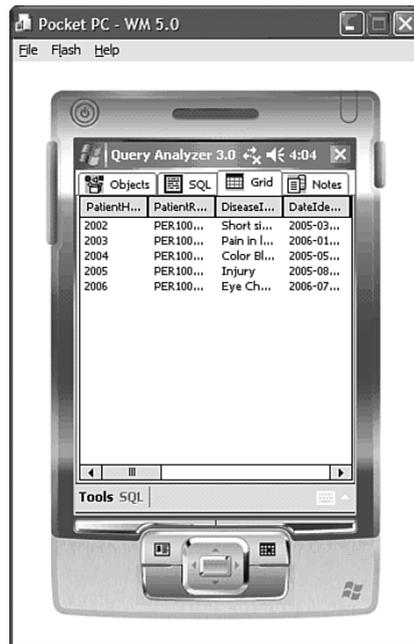


FIGURE 3.17 Table Content

Using the **Logging** option, you can record the SQL statements corresponding to actions that you do using the graphical tool. If you need to insert one hundred rows into a database and you are not keen on writing SQL Statements, you can generate the first few SQL statements by using the **Logging** option and then copy and paste to convert them into a hundred rows.

Fonts

You use the **Font** option to change the setting in Query Analyzer. Click on the **Font** menu option to change the font in the **Font** dialog box as shown in Figure 3.19.

About

The **About** menu option displays the current versions of Query Analyzer and the SQL Server Compact Edition database.

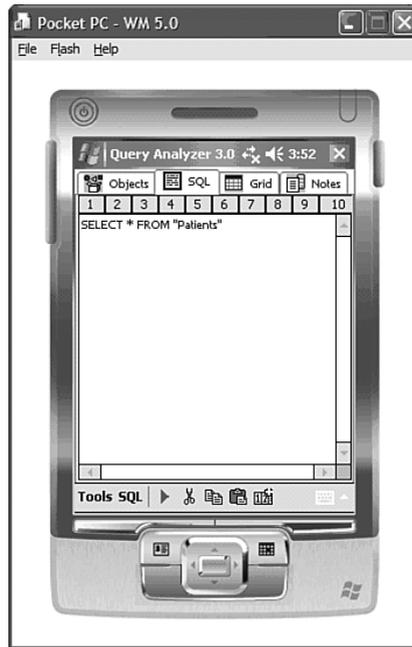


FIGURE 3.18 SQL Statement



FIGURE 3.19 Font Dialog Box

Exit

The purpose of the Exit menu is to close the Query Analyzer. Clicking on X in the right-hand corner only minimizes the application and leaves it running in the background. In the Visual Studio Tools section you will learn to use the `MinimizeBox` property to change the default behavior.

Using Tabs

The tabs on the Query Analyzer allow you to access and manipulate database definition and data contents. You can use the Object tab for a database object and the SQL tab for writing SQL queries. The Grid tab shows the results of queries. The Notes tab contains information regarding query execution time and number of rows impacted.

Objects

The Object tab shows the tree view of database objects such as the one shown in Figure 3.20. This relates to Object Explorer in SQL Server Management Studio. At the top of the tree is the Database node. The Database node shows the available connection. The open database connection will have a green icon and a closed database will have a red icon. Under each database there are tables, columns, and indexes.

Clicking on the Tables node displays a set of icons as shown in Figure 3.21.



FIGURE 3.20 Database Objects



FIGURE 3.21 Icons with Tables Object

The second icon is for displaying and hiding system tables. You can toggle the option by clicking on it. Figure 3.22 shows a Hide System Table. Clicking on the Hide System Table icon hides the table.



FIGURE 3.22 Hide System Table

Figure 3.23 shows a Show System Table. Clicking on the Show System Table button shows system tables.



FIGURE 3.23 Show System Table

The icon with a plus sign shown in Figure 3.24 is used to create a new table. The icon is shown when you select Tables node.



FIGURE 3.24 Create New Table

You can click on a specific table in Object tree. When you select a table in Object tree, icons appear at the bottom as shown in Figure 3.25.



FIGURE 3.25 Icons with Table

Clicking on the button shown in Figure 3.26 displays the content of the table you have selected.



FIGURE 3.26 Execute Query

Clicking on the button shown in Figure 3.27 opens a dialog box for adding a new column object.



FIGURE 3.27 New Column

Clicking on the button shown in Figure 3.28 opens a dialog box for adding a new index object.



FIGURE 3.28 New Index

SQL

The SQL tab shown in Figure 3.29 is an editor to write and execute SQL queries. This relates to Query Editor in SQL Server Management Studio.

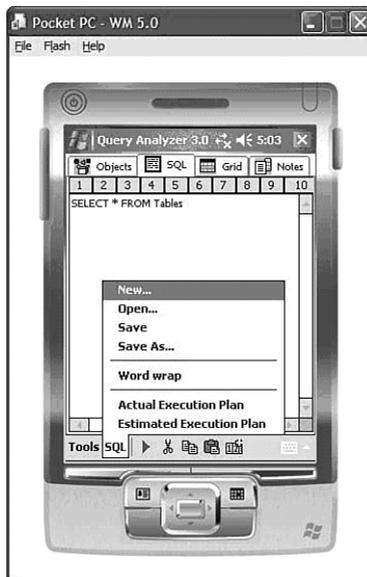


FIGURE 3.29 SQL Tab and Menu Options

When you move to the SQL Tab, the Query Analyzer shows the SQL menu option on the lower left-hand side. The SQL option has suboptions to open a new query, open an existing query, save a query, and to show the execution plans. You will be using these options in upcoming chapters.

The Word Wrap option is specific to Query Analyzer. The Word Wrap option is useful for the small screen and wraps an SQL statement to the visible part. A similar option is not available in SQL Server Management Studio.

Next to the SQL menu option there is the Execute icon. (This icon is shown in Figure 3.27.) Use this function to execute queries.

On the right side of the Execute icon you have standard Cut, Copy, and Paste icons as shown in Figure 3.30. These are standard features and you can utilize them while writing queries.

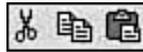


FIGURE 3.30 Cut, Copy, and Paste Buttons

The icon on the far right as shown in Figure 3.31 is the Preset button. This saves a frequently used SQL statement.



FIGURE 3.31 Preset Button

You will be using the Preset button in conjunction with ten SQL statement buttons. These ten buttons are shown in Figure 3.32.



FIGURE 3.32 Ten SQL Statement Buttons

You may have noticed these ten buttons at the top of SQL Query Editor.

The Preset button allows you to save up to ten frequently used SQL statements. By clicking on one of these buttons you can invoke an SQL statement.

Saving SQL Statements

To save SQL statements, click on the PreSet button. It will start the Button Presets dialog box. Type the SQL statements and click OK. Figure 3.33 demonstrates a scenario in which two SQL statements are written.

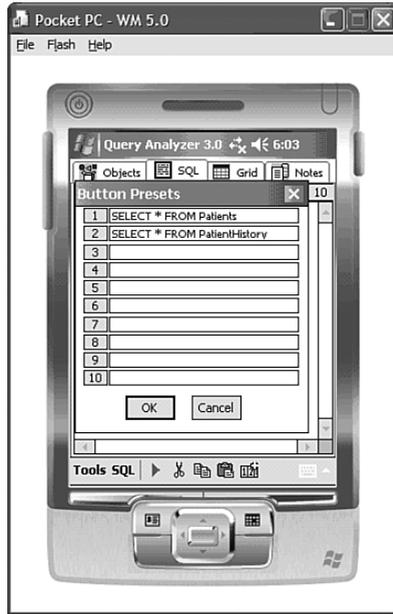


FIGURE 3.33 Button Preset Dialog Box

Type an SQL statement. Highlight the statement as shown in Figure 3.34 and then click on the Preset button.

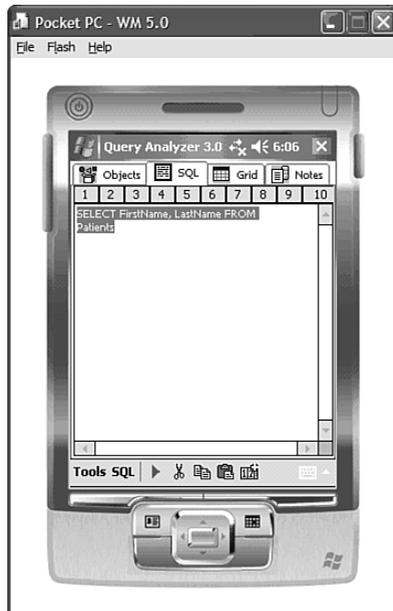


FIGURE 3.34 Highlighted SQL Statement

The Preset dialog will open. Click on the button number where you want to store an SQL statement. The example shown in Figure 3.35 stores a statement at number three.

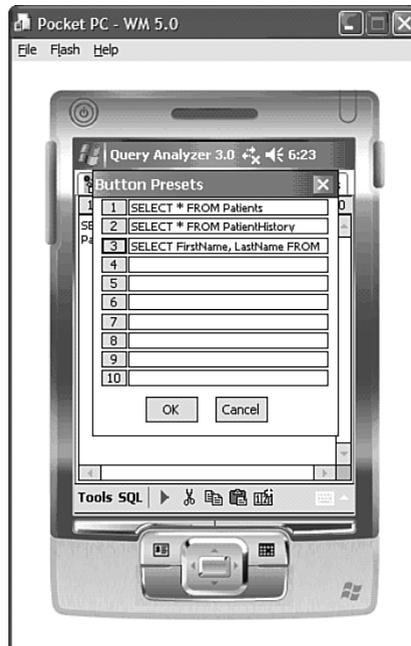


FIGURE 3.35 Adding SQL in Button Preset

Using a Saved SQL Statement

Once you have saved SQL statements, you can retrieve them by simply clicking on the button on top of SQL Query Editor. The SQL statement corresponding to the button number will be copied to SQL Query Editor.

Grid

The Grid node displays the results of query execution. When you double click on the Execute Query button, the Query Analyzer automatically activates the Grid tab and displays the results. If the query has an error, the Notes tab will activate. Figure 3.36 demonstrates the Grid tab.

Notes

When you execute a query, the Notes section contains the amount of time it took to execute the query similar as shown in Figure 3.37. In the event the query has an error, the Notes tab will display the error message.

Now you know how to utilize Query Analyzer functionality. In the next three chapters you will learn more about the SQL Server Compact Edition database and you will use Query Analyzer interface for other operations such as creating, compacting, and deleting the database.

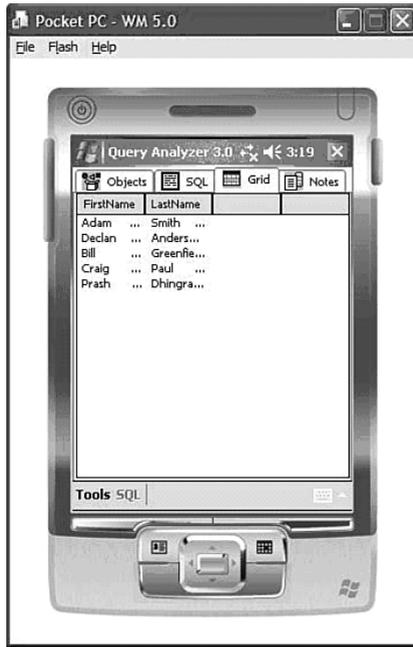


FIGURE 3.36 Grid Tab



FIGURE 3.37 Notes Tab

Using Visual Studio

SQL Server Compact Edition database is fully integrated with Visual Studio 2005. The integration enables rapid development of SQL Server Compact Edition applications. In this section you will create a Visual Studio project. Once the project is created, you will then explore the user interface of Visual Studio.

NOTE

Using Visual Studio 2005, you can create managed applications in C#.NET and VB.NET. In Visual Studio, you can create native applications using Microsoft Visual C++. Microsoft Visual C++ is also known as an embedded Visual C++ 4.0. Microsoft Embedded Visual Basic is not supported by SQL Server Compact Edition.

Creating a Smart Device Project

One primary goal of Visual Studio integration with SQL Server Compact Edition is to provide a single development environment where a developer can create a database, design forms, and write code to access and manipulate the database. Using Visual Studio 2005, you will do the following:

- Connect to an SQL Server Compact Edition database on a desktop or on a device.
- Manage an existing database or create a new database.
- Design SQL Queries with Query Designer.
- Automatically deploy SQL Server Compact Edition and .NET Compact Framework binaries onto a device.

The following example demonstrates the Visual Studio 2005 features that help the development of a Mobile devices application using SQL Server Compact Edition.

1. Start Visual Studio. Click on File | New Project
2. The New Project dialog starts. The dialog allows you to choose project types—Visual Basic, Visual C#, Visual J#, Visual C++, etc. In the example shown in Figure 3.38, Visual C# node is selected.
3. You can build Windows, Smart Device, Database, or Starter Kits projects. In this example, select Smart Device. The Smart Device project can be built for Pocket PC 2003, Smartphone 2003, Windows CE 5.0, Windows Mobile 5.0 Pocket PC, Windows Mobile 5.0 Smartphone, Windows Mobile 6.0 Professional Edition, and Windows Mobile 6.0 Standard Edition. Select Windows Mobile Platform.

NOTE

You need to have Windows Mobile 5.0 SDK and Windows Mobile 6.0 SDK installed in order to have the Windows Mobile 5.0 and Windows Mobile 6.0 project options available.

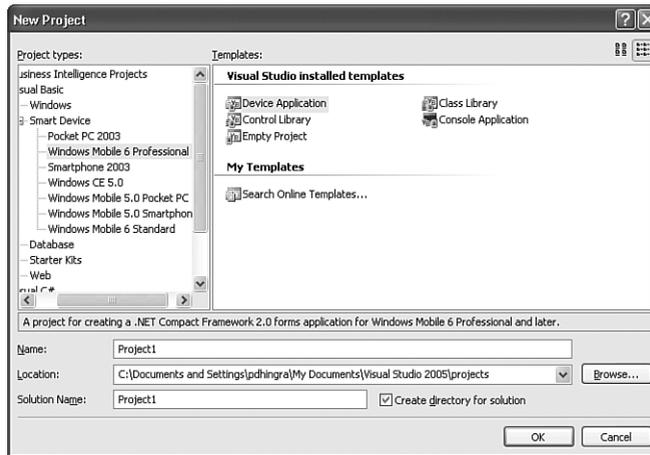


FIGURE 3.38 New Project

4. Select Device Application from the templates.
5. Specify the project name and click OK to create the project.

You will get a blank form. Using the Tool box you can add various controls to the form. In Chapter 7, *Programming with ADO.NET*, you will create examples and build a user interface for an SQL Server Compact Edition based application.

Setting Up References

Every project has a reference file that contains the run-time requirement of application for which the project is built. You should add a reference to `SqlServerCe.dll` in your C# or VB.NET program. To add a reference, right click on References in Solution Explorer and choose Add Reference. The Add Reference dialog box shown in Figure 3.39 allows you to browse and add components. Under the .NET tab, select `System.Data.SqlServerCe.dll` and click OK.

Once you have set up the reference, you need to add `System.Data.SqlServerCe` namespace in your program as shown in Listing 3.1.

LISTING 3.1 Include namespace C#.NET

```
using System.Data.SqlServerCe;
```

LISTING 3.1 Include namespace VB.NET

```
Imports System.Data.SqlServerCe
```

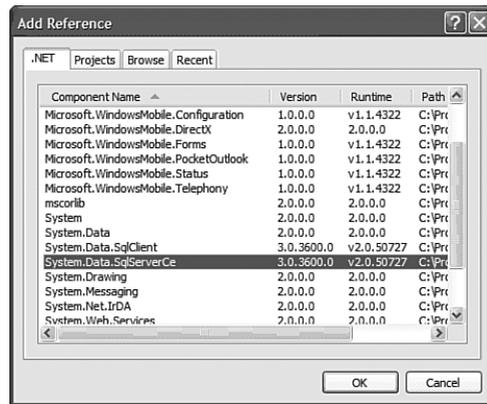


FIGURE 3.39 Add Reference Dialog Box

Using Data Sources

Visual Studio 2005 provides great features for connecting to databases. In this section you will learn Visual Studio features that assist in developing applications using the SQL Server Compact Edition database. Using the Data Sources dialog box, you can create new data sources and view existing SQL Server Compact Edition data sources. Within the Data Source window you can add a new data source, edit an existing dataset, and configure a dataset with Wizard.

NOTE

The Data Source menu option becomes available after you open a project in Visual Studio 2005.

The Data Source Window

You can view data sources by clicking on Data | Show Data Sources options. The Data Source window is a central place to see the data available for a project. You can drag objects from a Data Source window onto forms to create data-bound controls.

Configuration Wizard

Data Source Configuration Wizard helps manage the connection to Data Source. Using the Data Source Configuration Wizard, you can create and edit data sources. You can create a dataset consisting of one or more SQL Server Compact Edition tables. To connect to a data source click on Data | Add New Data Source. In Visual Studio you can create Data source from databases, Web Services, or Objects. You will get a dialog box similar to the one shown in Figure 3.40.

Click on Database and then click on the Next Button. The next dialog box as shown in Figure 3.41 allows you to choose the connection and connection string.

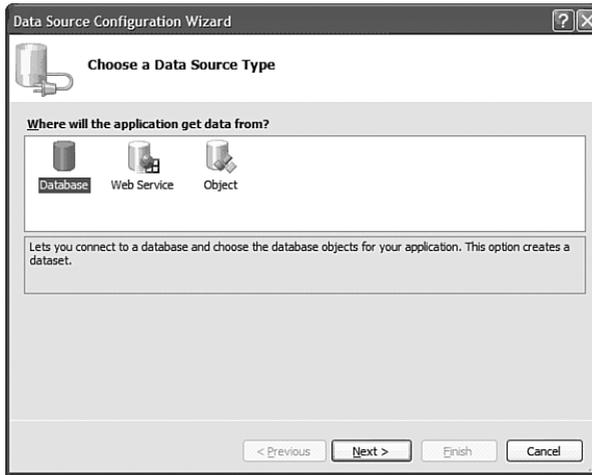


FIGURE 3.40 Choose a Data Source Type



FIGURE 3.41 Choose Data Connection

Click on New Connection. The Add Connection dialog box shown in Figure 3.42 allows you to connect to a selected data source. You can also click on the Change button to choose a different data source. For SQL Server Compact Edition you will choose .NET Framework Data Provider.

You can click on browse and select a different database file or create a new SQL Server Compact Edition database file by clicking on the Create button.

The password field allows you to specify the password for a password protected SQL Server Compact Edition database.

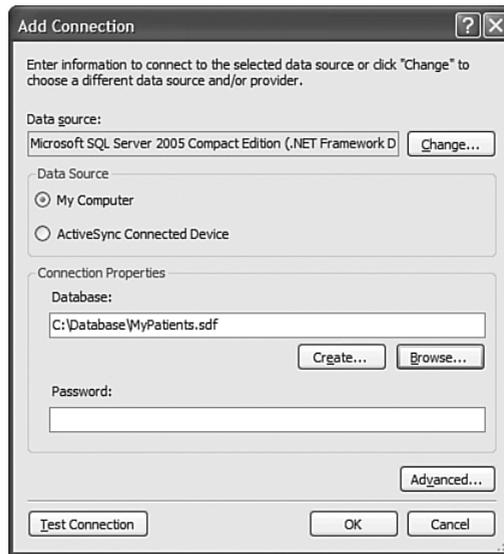


FIGURE 3.42 Add Connection Dialog Box

You can click on the Test Connection button to verify that connection to the database works.

Click the OK button.

The next dialog box shown in Figure 3.43 allows you to choose from the available objects that are displayed in a tree structure. After selecting the objects, you should provide a name for the dataset.

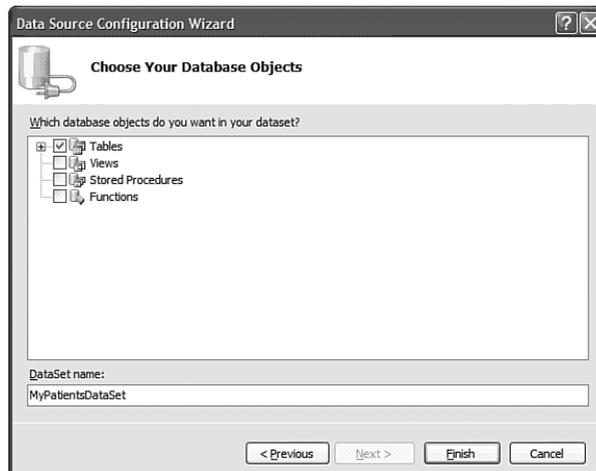


FIGURE 3.43 Choose Database Objects

On clicking Finish, a dataset schema will be added to Solution Explorer.

NOTE

For the database objects selected, Visual Studio created a strongly typed dataset and adapter. The dataset contains selected tables and columns. A strongly typed dataset has columns and tables defined at design time.

The typed dataset allows data binding at design time. The strongly typed dataset will return all the rows in a table. When designing, select the dataset and add an SQL statement to retrieve a subset of rows.

Adding SQL Server CE to a Visual Studio Project

You can also add an SQL Server CE database the same way you add a C# or VB class. In Solution Explorer right click on Add | Existing Item as shown in Figure 3.44. You can also use the Project | Add Existing Item menu option.

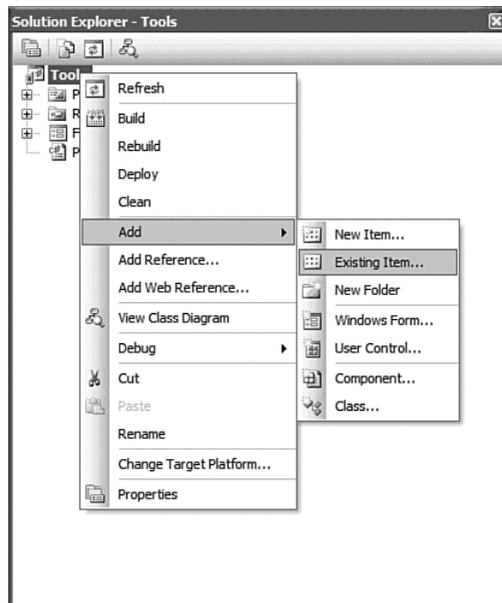


FIGURE 3.44 Adding SQL Server CE Database to a Project

In the Add Existing Item dialog box, select the Files of Type Microsoft SQL Server 2005 Compact Edition database file.

Browse to a specific file that you want to add to the project as shown in Figure 3.45. You will notice that dataset schema file is added to your project as shown in Figure 3.46.

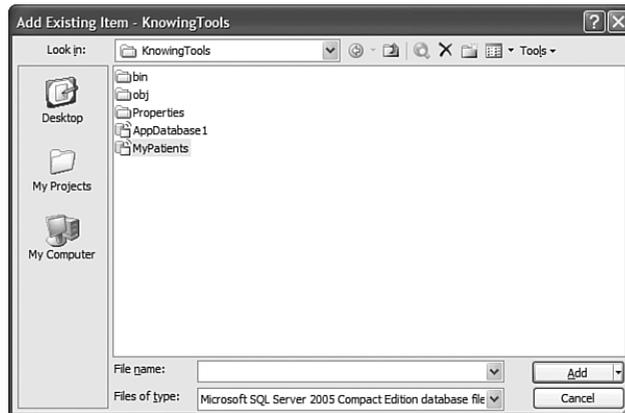


FIGURE 3.45 Add Database to a Project

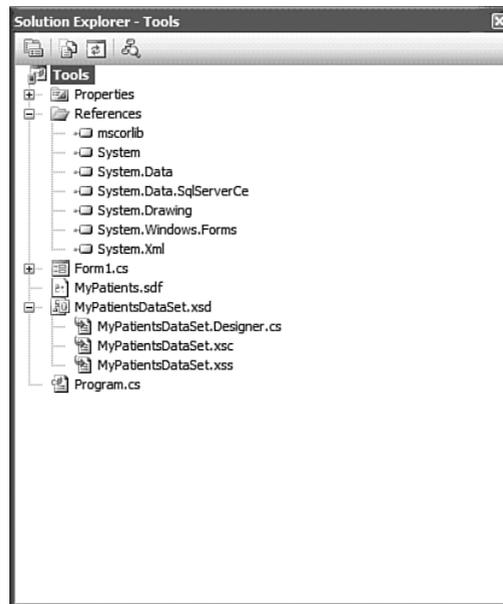


FIGURE 3.46 Data Set Schema in Solution Explorer

NOTE

In the Data Source window you can click on Edit Data Source with the Designer option to modify a dataset schema.

By default Visual Studio set the Copy to Output Directory property to Copy if Newer and Build Action property value to Content.

You can choose one of three values for Copy to Output Directory.

- The Copy if Newer value copies the database file from the project folder to bin (output folder) if the database file in the project folder is newer.
- The Copy Always value always copies the database file from the project folder to a bin folder.
- The Do Not Copy value does not copy the database file. You should copy the database yourself.

You can choose one of following values for the Build Action property:

- Compile value to compile the element.
- Content to include the element as content.
- Embedded Resource to treat the element as embedded resource.
- Do nothing with the element.

The SQL Server CE database should be included as content in a project. Visual Studio 2005 does this automatically for you.

Deploying Your Smart Device Application Using Visual Studio

Once you have built the application using Visual Studio, you will then want to execute it. For a desktop-based application you can run the executable from a command prompt or using the menu option Debug | Start without debugging.

For a device-based application you need to transfer the executable from a developer machine to a device. Visual Studio allows you to deploy the application directly on a device or on an emulator. You can use the following steps to deploy a smart device application onto a device or an emulator. You can also choose the device or emulator where you wish to deploy the application.

Emulators are virtual devices. Using Emulators available with Visual Studio 2005 you can develop and test the application on an emulator and you do not need a physical device.

1. Click Debug | Start without debugging.
2. The Deploy dialog box shown in Figure 3.47 is displayed. Choose whether to deploy an application on a device or on available emulators.
3. Select the device or emulator.
4. You can uncheck the button "Show me this dialog each time I deploy the application."
5. Click on the Deploy button.

If you opt not to get the Deploy dialog box, you will still choose whether to deploy on a device or on an emulator. From the Tool bar you can choose the option where to deploy your application. All the options in the Deploy dialog box are available in the Deployment Device combo box shown in Figure 3.48.

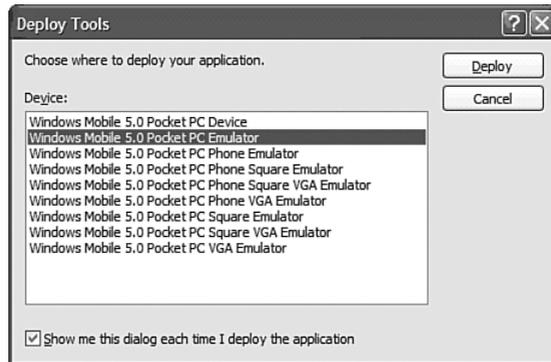


FIGURE 3.47 Deploy Dialog Box

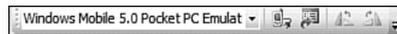


FIGURE 3.48 Deployment Device Combo Box

Deploying Multiple Times

When you develop a program, it is likely that you will need to improve it over many iterations. You will deploy the application on a device or on an emulator, test it, make changes in the program, and then deploy it again. By default when you click X on a .NET Compact Framework application, it does not close. It simply becomes minimized and keeps on running in the background. As one instance of the application is already running and you are trying to deploy another instance, you will receive an error. To remove the error you need to close the already running instance of the application. You can go to running programs and close the application.

In most of the examples mentioned in this book, the Minimize Box property is set to false. When you set this property to false, an OK button appears on the right corner of the main form of application instead of an X mark. Clicking on the OK button closes the application instead of minimizing it. You can click the OK button and close the application. Once the application is closed, you can modify the application and deploy the application again. You do not need to go to the running program section to stop the previous instance of the application.

Building a CAB File

Earlier versions of Visual Studio did not have support for building an installer project for .NET Compact Framework applications. Using Visual Studio 2005, you can build a Deployment project for Smart devices. Using Smart Device Cab Project you can drag and drop files, create folders, amend registry settings, and easily build the cabinet file for installation.

Cab files or Cabinet files are used to packages dll and executables. To build a CAB file for your application you should add a Smart Device CAB project to your solution. In the

following steps you will learn how to create a Smart Device Cab project to install smart device applications.

1. Click on File | Add | New Project.
2. In the Add New Project dialog box, choose the Project type as Other Project Type and under Other Project type choose Setup and Deployment project.
3. In the template section, select the Smart Device CAB Project.
4. Specify the name of the project. In the example shown in Figure 3.49, the name SmartDeviceCab1 is used.

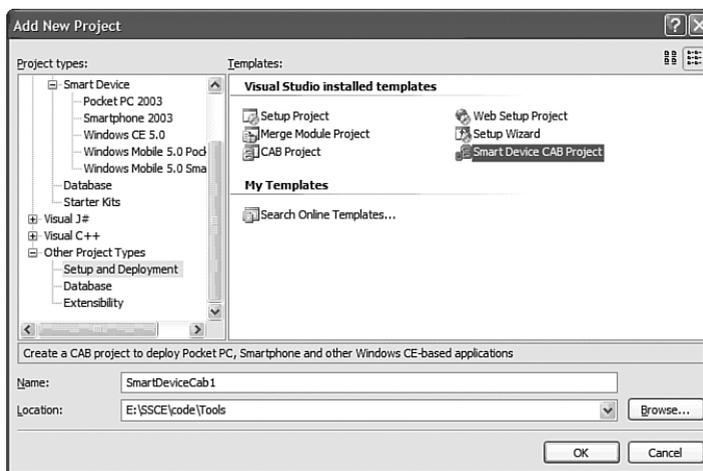


FIGURE 3.49 Setup and Deployment Project

5. Click OK.
6. Select the SmartDeviceCAB1 project.
7. Click on the View | Properties Window.
8. Change the ProductName as shown in Figure 3.50. In this example the name MyCAB is used.
9. To add the executable to the CAB file, open a file system editor. If it is not already open, right click on SmartDeviceCAB1 project, and click on View | File System option.
10. In the File System Editor, click on Application Folder.
11. Click on Action | Add | Project Output menu option as shown in Figure 3.51.

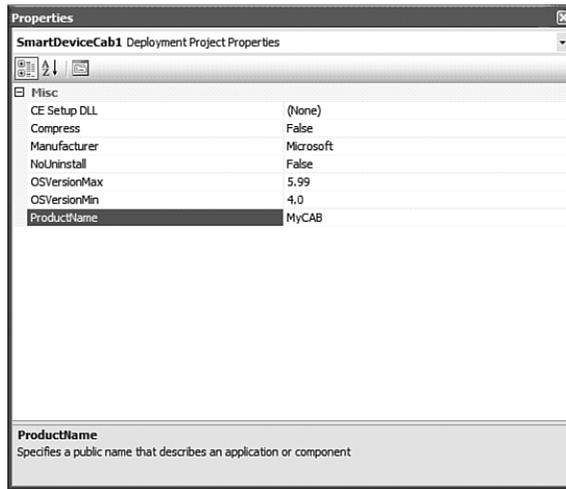


FIGURE 3.50 Properties of a CAB Project

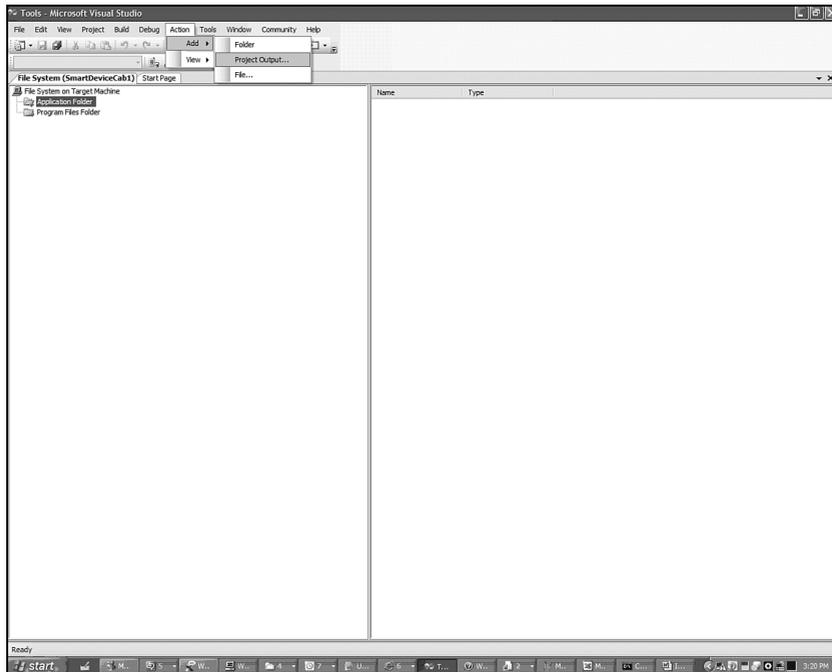


FIGURE 3.51 Project Output

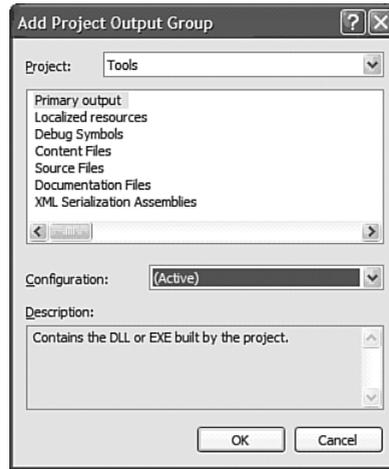


FIGURE 3.52 Add Primary Out

12. In the Add Project Output Group, click on Primary Output as shown in Figure 3.52 and click OK.
13. Click on Build | BuildsmartDeviceCab1. SmartDeviceCab1 is name of my deployment project. You may get a different menu option based on your project name.
14. Once you build the CAB file, you can find it under SmartDeviceCab1\Release folder.

You can copy the CAB file to a device or emulator. Just by clicking the file, the CAB file gets deployed on the device.

NOTE

For Smartphone you need to digitally sign the CAB file.

Changing Target Platform

In Visual Studio it is possible for you to change the target platform For example, you can change the Windows Mobile 5.0 Pocket PC platform to Windows Mobile 5.0 Smartphone or Windows Mobile 6.0 platforms by simply following these steps:

1. Click on the Project | Choose Target Platform menu option.
2. You will get a Change Target Platform dialog box as shown in Figure 3.53.
3. Choose the New Platform in the Change to pop-up box.
4. Click OK after selecting the new platform.
5. A message box similar to one shown in Figure 3.54 is displayed telling you that the project will be closed and then reopened. Click Yes to continue.

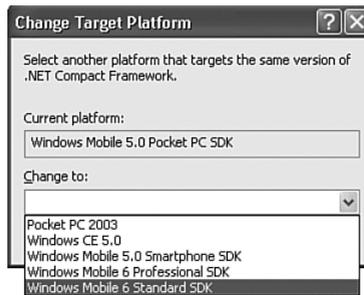


FIGURE 3.53 Change Target Platform

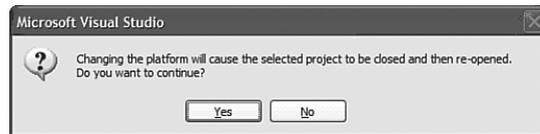


FIGURE 3.54 Information Message

Visual Studio 2005 and SQL Server Compact Edition integration have a number of features to develop, build, and deploy .NET Compact Framework applications. The Data Source window displays the data object for all types including the SQL Server Compact Edition data object. Data Source Configuration Wizard enables you to add the SQL Server CE database into a project.

Visual Studio 2005 has also added support for building an installer project for .NET Compact Framework based applications.

Creating a Desktop Project

One of the main differences between SQL Server Compact Edition 3.1 and SQL Server Mobile Edition 3.0 is desktop support. Now you can use SQL Server Compact Edition to build a desktop-based application. Desktop support also enables you to develop applications in which desktop, laptop, and mobile devices can use the same database.

To use SQL Server Compact Edition in a desktop, you will create a Windows based project and add a reference to SqlServerCe dll installed for desktop. Follow these steps to create a Windows project using SQL Server Compact Edition

1. Start Visual Studio. Click on File | New Project as shown in Figure 3.55.
2. The New Project dialog box appears. The dialog box allows you to choose project types—Visual Basic, Visual C#, Visual J#, Visual C++, etc. In this example, the Visual C# node is selected as shown in Figure 3.55.
4. Select Windows Application from the templates.

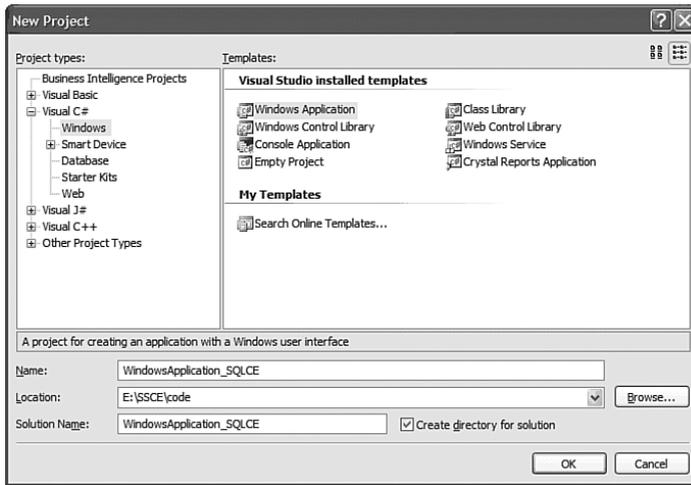


FIGURE 3.55 Desktop Project

5. Specify the name of the project and click OK to create the project. You will get a blank form. Using the Tool box you can add various controls to the form.
6. To use SQL Server Compact Edition you should add a reference to SqlServerCe.dll. In Solution Explorer, click on References. Right click and choose Add References.
7. In the Add Reference dialog box, click on the Browse button.
8. Browse to locate where you have installed SQL Server Compact Edition. Earlier you learned that SQL Server Compact Edition for desktop gets installed at <Disk>:\Program Files\Microsoft SQL Server Compact Edition\v3.1.
9. Select System.Data.SqlServerCe.dll and click the OK button as shown in Figure 3.56.

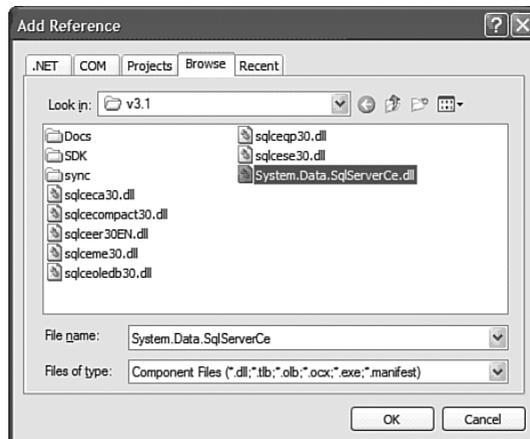


FIGURE 3.56 Add Reference to SqlServerCe Namespace

Once you have set up the reference, you need to add `System.Data.SqlServerCe` namespace in your program. At the top of the code include the `SqlServerCe` namespace as shown in Listing 3.2

LISTING 3.2 Include namespace C#.NET

```
using System.Data.SqlServerCe;
```

LISTING 3.2 Include namespace VB.NET

```
Imports System.Data.SqlServerCe
```

Now you can start writing code for a desktop application that uses SQL Server CE as a local store.

Using DataDirectory

You have learned how to specify the full path of an SQL Server Compact Edition database file. Using the `DataDirectory` feature you can specify the path of database in `DataDirectory`. You can set the `DataDirectory` property using Application Domain as shown in Listing 3.3.

With the `DataDirectory` feature you do not need to hardcode the full path of the database in connection string.

LISTING 3.3 DataDirectory C#.NET

```
AppDomain.CurrentDomain.SetData ("DataDirectory",@"C:\Database");
```

LISTING 3.3 DataDirectory VB.NET

```
AppDomain.CurrentDomain.SetData("DataDirectory", "C:\Database")
```

Instead of specifying the full path in a connection string, you will specify `DataDirectory` enclosed in | (pipe) symbol and the SQL Server Compact Edition filename as shown in Listing 3.4.

LISTING 3.4 DataDirectory C#.NET

```
myConn = new SqlCeConnection  
    ("Data Source=|DataDirectory|\Email.sdf;");  
myConn.Open();
```

LISTING 3.4 DataDirectory VB.NET

```
myConn = New SqlCeConnection _  
    ("Data Source=|DataDirectory|\Email.sdf;")  
myConn.Open()
```

Connecting to an Emulator

The device emulator is a desktop-based application that emulates the behavior of Windows Mobile or Windows CE based hardware. An emulator emulates an ARM processor. You can configure various properties of the emulator such as ROM size, orientation, and so on. Similar to a device, a partnership can now be created between ActiveSync and an emulator. The ActiveSync partnership is needed in applications that synchronize data between a desktop and a device.

There are three ways to connect to a device or an emulator:

- Develop a device project: Build and Deploy solution. A dialog box will appear where you can choose to develop or deploy the solution as shown in Figure 3.57.
- Click on the Tools | Connect to Device option. A dialog box similar to the one shown in Figure 3.57 will appear where you can choose the platform, the device/emulator, and select Connect.
- Click on Tools | Device Emulator Manager. A Device Emulator Manager dialog box appears showing a list of emulators. Select an emulator, right click, and select Connect.

NOTE

You can also use the `dvcemumanager.exe` file to start the Device Emulator Manager. This executable is located in the `Program Files\Microsoft Device Emulator\1.0` folder.

The connected emulator is shown with a green arrow.

Once the emulator is connected, you can deploy an application and execute the application on an emulator. A connected, but not cradled, emulator is shown in Figure 3.57

Cradling the Emulator

When the emulator starts, it acts as a device that is not cradled. A connected emulator can be used to execute an application locally on an emulator. If you need to synchronize data between a device and a server, the device needs to be cradled. Similar to a device, an emulator needs to be cradled. To cradle an emulator, you will choose the Cradle option in the Device Emulator Manager.

1. Select the emulator in Device Emulator Manager, right click, and select Cradle.
2. After the emulator is cradled, ActiveSync detects it, and the ActiveSync icon in the system tray will become green. You can also double click to start ActiveSync if it does not start automatically. Once the ActiveSync connection is established, it may display a dialog box to synchronize files. Click Cancel to establish a guest partnership between the emulator and the desktop as shown in Figure 3.58.

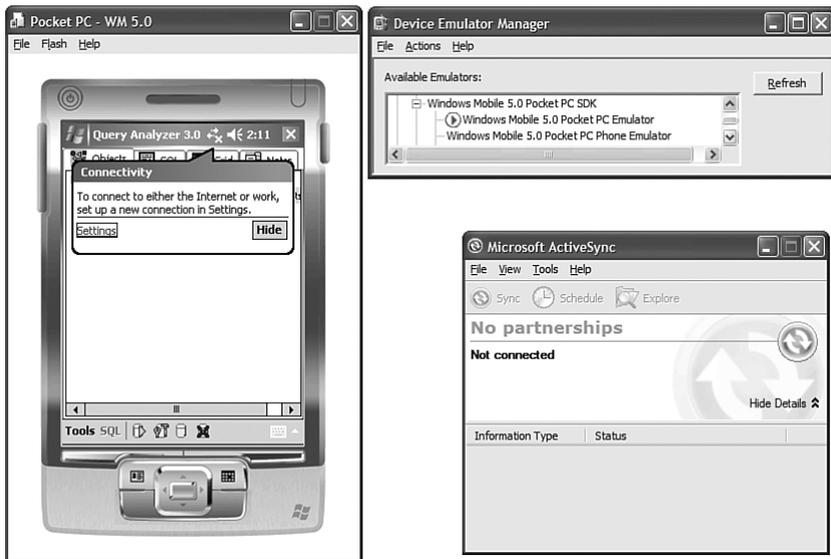


FIGURE 3.57 Not Cradled Emulator

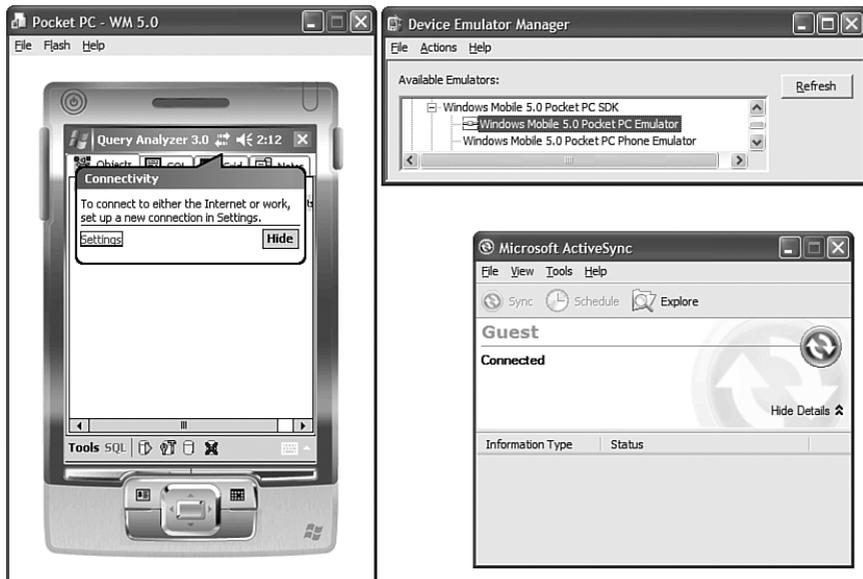


FIGURE 3.58 Cradled Emulator

Configuring ActiveSync for Emulator

The DMA option must be selected in the ActiveSync connection settings as follows:

1. Click on the File | Connection Settings menu option.
2. In the Connection Settings dialog box, check the box next to the “Allow Connection to one of following” option. Select DMA from the drop down box to enable DMA connections as shown in Figure 3.59.

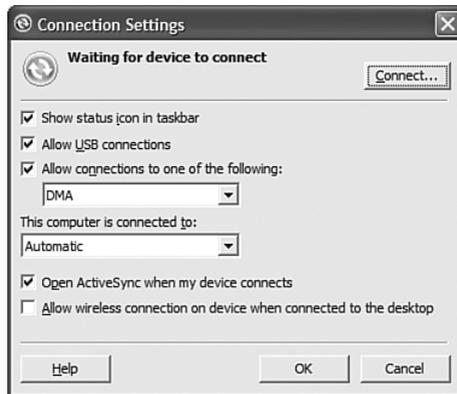


FIGURE 3.59 DMA Settings

Visual Studio 2005 supports the Direct Memory Access (DMA) transport channel to communicate with a device emulator. DMA is faster than TCP/IP and does not require network connectivity. DMA is also faster and more robust than transmitting over a network stack as it provides a direct communication between two Windows processes—Visual Studio and emulator.

NOTE

You do not need to use ActiveSync and you do not need to cradle the emulator if the application is running locally on an emulator. The emulator must be cradled through an ActiveSync connection if the application communicates to a backend server to exchange data. For example, Replication, RDA, or SqlClient usage requires an emulator to be cradled and connected through ActiveSync.

Using Windows Mobile 6.0

Microsoft released Windows Mobile 6.0 SDK in the beginning of 2007. Windows Mobile 6.0 is built on top of Windows Mobile 5.0. The taxonomy is changed between Windows Mobile 5.0 and Windows Mobile 6.0

Windows Mobile 5.0 differentiates devices such as Windows Mobile Pocket PC and Windows Mobile Smartphone. Pocket PCs and Pocket PCs with the phone edition have touch screen capability and accept screen input whereas Smartphone does not.

Windows Mobile 6.0 SDKs does not target specific devices. Windows Mobile 6.0 has the following two SDK:

- Windows Mobile Professional Edition targets the devices that have touch screen capability.
- Windows Mobile Standard Edition targets devices that do not have touch screen capability.

Once you install the Windows Mobile 6.0 SDKs, you will be able to develop corresponding project types in Visual Studio. Windows Mobile SDKs are installed in the same fashion as Windows Mobile 5.0 SDK.

New Features in Windows Mobile 6.0 SDK

Windows Mobile 6.0 devices include SQL Server Compact Edition and .NET Compact Framework in the ROM. Having these products in ROM have many benefits including:

- More remaining space available in RAM as .NET CF and SQL Server Compact Edition do not occupy RAM space.
- The reduction of the overall size of the installation package as it only contains the application component.
- A reduction in deployment effort and support level required for end users.

Windows Mobile 6.0 SDK provides Cellular Emulator. Using Cellular Emulator, you can test your application in various cellular communication situations from an emulator.

Using an emulator, you can also set the profile of your phone to headset, speaker phone, or car kit. The phone profile options and cellular emulator options in Emulator allow you to test your application in various real life scenarios.

Windows Mobile 6.0 provides a code signing utility, CABSignTool that signs together the Cab files and all of its constituent executables.

Windows Mobile 6.0 provides a Hopper unity to provide stress testing of mobile applications.

Windows Mobile 6 SDK provides Security Configuration Manager to configure the emulator with various security policies. Using these security policies, you can test the application with various security settings.

Windows Mobile 6.0 provides features and utilities to enhance the developer's experience. The inclusion of .NET CF and SQL Server compact Edition in ROM simplifies deployment. Device Emulator 2.0 included in Windows Mobile 6.0 allows you to test applications in real life scenarios without requiring a physical device.

You should consult the Windows Mobile documentation to get more details about Windows Mobile features.

Summary

This chapter provided you with an overview of SQL Server Management Studio, Query Analyzer, and Visual Studio tools. You will use SQL Server Management Studio and Query Analyzer to manage databases on a desktop and a device. SQL Server Management Studio provides an integrated environment to manage SQL Server, Integration Service, Analysis Services, and an SQL Server Compact Edition database. You also learned how to use Visual Studio to connect to a database. SQL Server Management Studio and Visual Studio add a lot of flexibility to develop database applications. In the remaining chapters of this book you will be using these features to develop more powerful examples.