SYBASE®

Introduction to PocketBuilder

PocketBuilder™

2.1

Contents

About This Book

Audience

This guide is for anyone who wants to build applications with PocketBuilder™ but is not familiar with PowerBuilder®.

How to use this book

This book provides an overview of PocketBuilder features and the PocketBuilder development environment and a tutorial that leads you through the step-by-step process of creating and deploying PocketBuilder applications. For information about the tutorial and the lessons it includes, see Chapter 3, "About the PocketBuilder Tutorial."

Related documents

PocketBuilder documentation   The PocketBuilder documentation set also includes the following manuals:

- *User's Guide* - Gives an overview of the PocketBuilder development environment and explains how to use the interface. Describes basic techniques for building the objects in a PocketBuilder application, including windows, menus, DataWindow® objects, and user-defined objects. An appendix summarizes the differences between PocketBuilder and PowerBuilder.

- *Resource Guide* - Presents advanced programming techniques and information about connecting to and synchronizing with a database.

The PocketBuilder reference set is made up of four manuals that are based on PowerBuilder documentation:

- *Connection Reference* - Describes the database parameters and preferences you use to connect to a database in PocketBuilder.

- *DataWindow Reference* - Lists the DataWindow functions and properties and includes the syntax for accessing properties and data in DataWindow objects.

- *Objects and Controls* - Describes the system-defined objects and their default properties, functions, and events.

- *PowerScript Reference* - Describes syntax and usage for the PowerScript® language including variables, expressions, statements, events, and functions.

Online Help Reference information for PowerScript properties, events, and functions is available in the online Help with annotations indicating which objects and methods are applicable to PocketBuilder.

SQL Anywhere® Studio documentation PocketBuilder is tightly integrated with SQL Anywhere (formerly Adaptive Server Anywhere), UltraLite®, and MobiLink, which are components of SQL Anywhere Studio. You can install these products from the PocketBuilder setup program. Documentation for SQL Anywhere Studio is included in a separate collection on the PocketBuilder Technical Library CD and in online Help. For an introduction to these products, see Chapter 1, "PocketBuilder Overview."

Windows CE documentation If you do not have a basic familiarity with Windows CE devices, you should consult a book that describes them. For information about developing applications for Microsoft Windows CE, see the Microsoft Web site at http://msdn2.microsoft.com/en-us/library/ms950422.aspx. You can also find helpful information at the Pocket PC Developer Network Web site at http://www.pocketpcdn.com.

Sample applications The PocketBuilder installation provides a Code Examples workspace with targets that illustrate many of the product's features. Commented text inside events of target objects helps explain the purpose of the sample code. The example workspace is installed in the Code Examples subdirectory under the main PocketBuilder directory.

More applications on the Web You can find more sample PocketBuilder applications and techniques in the PocketBuilder project on the Sybase CodeXchange Web site at http://pocketbuilder.codexchange.sybase.com/. There is a link to this page on the Windows Start menu at Program Files>Sybase>PocketBuilder 2.0>Code Samples.

If you have not logged in to MySybase, you must log in to the Sybase® Universal Login page to access CodeXchange. If you do not have a MySybase account, you can sign up. MySybase is a free service that provides a personalized portal into the Sybase Web site.

Other sources of information Use the Sybase Getting Started CD, the SyBooks CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.

- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

    Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

    Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

    To access the Sybase Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.

Sybase EBFs and software maintenance

❖ **Finding the latest information on EBFs and software maintenance**

1 Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.

2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.

3 Select a product.

4 Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

    Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the "Technical Support Contact" role to your MySybase profile.

5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Conventions

The formatting conventions used in this manual are:

Formatting example	To indicate
Retrieve and Update	When used in descriptive text, this font indicates: • Command and function names • Datatypes such as integer and char • Database column names such as emp_id and f_name • User-defined objects such as dw_emp or w_main
variable or *file name*	When used in descriptive text and syntax descriptions, oblique font indicates: • Variables, such as *myCounter* • Parts of input text that must be supplied, such as *pklname*.pkd • File and path names
File>Save	Menu names and menu items are displayed in plain text. The greater than symbol (>) shows you how to navigate menu selections. For example, File>Save indicates "select Save from the File menu."
`dw_1.Update()`	Monospace font indicates: • Information that you enter in a dialog box or on a command line • Sample script fragments • Sample output fragments

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

P A R T   1   # Welcome to PocketBuilder

This part is an overview of PocketBuilder, the PocketBuilder development environment, and the PocketBuilder tutorial.

PocketBuilder Overview

About this chapter

This chapter describes the major features of PocketBuilder and the database software that is integrated with it.

Contents

Topic	Page

About PocketBuilder

PocketBuilder is a smart-client application development tool that enables you to build applications for handheld devices in an object-centric, graphical, desktop environment, and then deploy those applications to supported Pocket PC and Smartphone devices or emulators.

Whether you already use PowerBuilder or are new to Sybase application development tools, PocketBuilder can help you create applications for the mobile environment quickly and efficiently. You can also use PocketBuilder to migrate existing PowerBuilder desktop applications to handheld devices.

The target platform for PocketBuilder applications is the Windows CE operating system, including its more recent Windows Mobile enhancements. The Windows CE API is a subset of the API for traditional Windows platforms. The most obvious difference between Windows CE and desktop Windows platforms is the screen size (real estate) available to deployed applications. There are also stylistic differences for applications deployed to Windows CE platforms.

Highlights

PocketBuilder:

- Provides a highly productive 4GL Integrated Development Environment (IDE) for mobile development

- Delivers the same Rapid Application Development (RAD) platform that PowerBuilder developers rely upon today

- Extends the patented Sybase DataWindow to mobile environments, enabling dynamic data access with display formatting and data manipulation capabilities, with mininal or no coding required

- Introduces tight integration with SQL Anywhere Studio, simplifying the creation of database-powered mobile enterprise applications and enabling synchronization with enterprise-level data

- Provides native objects and controls for integration with diverse Pocket PC programs and tools, including the Pocket Outlook Object Manager, Subscriber Identification Modules (SIMs), various bar code scanners, biometric scanners, digital cameras, and GPS devices.

- Supports SMS messaging, getting and setting screen orientation, and signing of CABs and applications.

Supported platforms and devices

The PocketBuilder IDE runs on Microsoft Windows 2000 (with Service Pack 2), Windows XP, Windows 2003, and Windows Vista desktops for application development. You can deploy PocketBuilder applications to Pocket PC and Smartphone devices and emulators.

For more information on supported devices and emulators, see the *Release Bulletin* for your version of PocketBuilder. The *Release Bulletin* also includes pointers to Web sites where you can download supported emulators.

PocketBuilder features

PocketBuilder provides a feature-rich, highly productive, easy-to-use environment for mobile application development. It extends the tried and true core features of Sybase's 4GL desktop development tool, PowerBuilder, with enhanced capabilities for handheld devices.

Object-oriented development

PocketBuilder is an object-oriented development environment, with full support for inheritance, polymorphism, and encapsulation. These powerful features enable you to build robust applications for mobile deployment rapidly and address critical business needs quickly.

PocketBuilder provides hundreds of built-in functions and many ready-to-use components. For example, it furnishes system functions to control the Soft Input Panel (SIP), custom Today items, and a separate object for the Toolbar, on Pocket PC devices. The GUI environment has painters for building objects and components graphically, enabling you to develop sophisticated applications quickly and easily. Whether you inherit windows, user objects, and menus or build all your own classes, PocketBuilder is the key to RAD for mobile environments.

DataWindow technology

Sybase's DataWindows technology offers data access, data manipulation, and sophisticated data presentation for mobile devices, all with minimal or no coding. With DataWindow technology, you can build complex SQL, define validation rules, filter, sort, and manipulate data with point-and-click ease, and display data using sophisticated formats, including free-form styles, graphs, grids, complex groupings, and tabular structures.

If you are a PowerBuilder developer, you can save time and leverage your investment in existing DataWindows that run in distributed n-tier and Web environments by deploying them to mobile platforms.

Workspaces and targets

Workspaces and targets provide the core interface for building PocketBuilder applications. A PocketBuilder workspace provides the graphical area in which you create applications and enables you to work on multiple applications simultaneously.

Targets are built in a workspace and are required to create an executable PocketBuilder application. Targets generally include a collection of windows and other objects that perform various functions. Since this is the same workspace and target paradigm used by PowerBuilder, PowerBuilder developers will already be familiar with the PocketBuilder environment.

Wizard-driven development

PocketBuilder provides wizards that make development almost effortless. You can create new targets, build new objects and projects, and create complex SQL, all with easy-to-use GUI wizards. You can also kick-start your development with the Pocket PC and Smartphone application creation wizards and get the fundamental pieces in place for your mobile application. Wizards also guide you through the rest of your development, including the building of DataWindows and the deployment of your PocketBuilder targets.

Small-footprint, full-featured, SQL database

PocketBuilder offers tight integration with Sybase SQL Anywhere (formerly Adaptive Server Anywhere) and UltraLite databases. SQL Anywhere provides rich enterprise functionality, including full transaction processing, referential integrity, stored procedures, triggers, row-level locking, automatic event scheduling, and automatic recovery.

UltraLite offers many of the same features in a relational database designed expressly for synchronization between small, mobile, and embedded devices and enterprise databases. For more information, see "SQL Anywhere database management" on page 7 and "UltraLite features" on page 9.

Advanced data synchronization

PocketBuilder includes a comprehensive set of technologies for scalable, bidirectional synchronization of data between enterprise database systems and mobile Windows CE devices. The synchronization wizards in PocketBuilder lead you step-by-step through the synchronization requirements.

The synchronization technologies are optimized for both occasionally connected and near real-time environments. These technologies support the secure communication of information for remote and mobile users over a wide variety of synchronous, asynchronous, wireless, dial-up, and Internet protocols. For more information, see "MobiLink synchronization" on page 10.

SQL Anywhere Studio integration

PocketBuilder is integrated with several components of Sybase SQL Anywhere Studio:

- The SQL Anywhere relational database management system (RDBMS)

- The UltraLite RDBMS

- MobiLink data synchronization technology

- The Sybase Central administration tool

These components are for development purposes only and are integrated into PocketBuilder through wizards, painters, and dialog boxes. For more information, see "MobiLink synchronization" on page 10. Full documentation for SQL Anywhere Studio, including SQL Anywhere, MobiLink, and UltraLite, is available on the iAnywhere Web site at http://www.ianywhere.com/developer/product_manuals/sqlanywhere/ and in the SQL Anywhere Studio online Help.

For specific information about using SQL Anywhere, MobiLink, and UltraLite with PocketBuilder, see the chapter on managing the database in the PocketBuilder *User's Guide* and the information on database connectivity in the *Resource Guide* and the *Connection Reference*.

SQL Anywhere database management

The SQL Anywhere RDBMS is the core of SQL Anywhere Studio. SQL Anywhere provides a series of tools for storing and managing data. You can use these tools to enter data into your database, to change your database structure, and to view or alter your data.

SQL Anywhere is intended for tasks that require a full-featured SQL database. It is designed to operate in varied environments. By taking advantage of available memory and CPU resources, SQL Anywhere provides very good performance in environments with ample resources. It also operates very well in environments with limited physical and database administration resources, including mobile computing and embedded database environments, and with workgroup servers.

SQL Anywhere
characteristics

SQL Anywhere excels in all the following roles:

- **A workgroup database server**   Workgroups ranging in size from a few people to several hundred people can use SQL Anywhere as a multiuser database server. It provides a high-performance database for workgroups, and is well suited for (but not limited to) environments where administration and hardware resources are limited.

  SQL Anywhere can employ multiple CPUs and use up to 64GB of memory. Multigigabyte SQL Anywhere databases are currently in production use.

- **An embedded database**   Many applications require a database "behind the scenes." These include Personal Information Managers, document management systems—just about any application that stores information. SQL Anywhere is designed to be the database for these applications.

  A key requirement of embedded databases is that they be able to run entirely without administration. SQL Anywhere has demonstrated this facility in many demanding commercial applications.

- **Mobile computing**   Handheld, laptop, and notebook computers are now common in many workplaces. SQL Anywhere is intended to be the SQL database for these computers. With MobiLink synchronization and SQL Remote replication, SQL Anywhere extends transaction-based computing throughout the enterprise.

SQL Anywhere features

SQL Anywhere provides all of the following features:

- **Full SQL RDBMS**   SQL Anywhere is a transaction-processing RDBMS with full recovery capabilities, online backup, referential integrity actions, stored procedures, triggers, row-level concurrency control, schedules and events, a rich SQL language, and all the features you expect in a full SQL RDBMS.

- **Economical hardware requirements**   SQL Anywhere requires fewer memory and disk resources than other database management systems.

- **Easy to use**   SQL Anywhere is self-tuning and easy to manage. You can use SQL Anywhere without the extensive database administration efforts usually associated with RDBMSs.

- **Standalone and network use**   SQL Anywhere can be used in a standalone manner, for example as an embedded database in a data-centric application, or as a network server in a multiuser client/server or three-tier environment. As an embedded database system, it can be started automatically by an application when required.

- **High performance**   Although SQL Anywhere is designed with simple administration and modest resource requirements in mind, it is a scalable, high-performance DBMS. SQL Anywhere can run on multiple CPUs, has an advanced query optimizer, and provides performance monitoring and tuning tools.

- **Industry-standard interfaces**   SQL Anywhere provides a native ODBC driver for high performance from ODBC applications, and an OLE DB driver for use from ActiveX Data Object (ADO) programming environments. It comes with Sybase jConnect for JDBC as well as an iAnywhere JDBC driver, and supports embedded SQL and Sybase Open Client interfaces.

- **A cross-platform solution**   SQL Anywhere can be run on many operating systems, including Windows, Windows PocketPC, Windows Mobile 2003, Novell NetWare, Sun Solaris, and Linux.

UltraLite features

UltraLite is a small-footprint RDBMS with synchronization features for small, mobile, and embedded devices that have very limited resources. UltraLite offers:

- **Robust data management**   Data held on small devices is as important as data in enterprise databases. UltraLite brings transaction processing, referential integrity, and other benefits of relational databases to small devices.

- **Powerful synchronization**   UltraLite uses MobiLink synchronization technology to synchronize with industry-standard database management systems. MobiLink provides flexible, programmable, and scalable synchronization that can manage thousands of UltraLite databases.

- **Straightforward development**   The integration of UltraLite components with the object-based programming interface of PocketBuilder improves development productivity. The graphical PocketBuilder tool enables you to design and modify UltraLite databases quickly.

MobiLink synchronization

MobiLink is a session-based synchronization system that allows two-way synchronization between a main database, called the consolidated database, and many remote databases.

The consolidated database, which can be any ODBC-compliant database, holds the master copy of all the data. Remote databases can be either SQL Anywhere or UltraLite databases. Synchronization begins when a MobiLink remote site opens a connection to a MobiLink synchronization server.

During synchronization, a MobiLink client at the remote site uploads database changes that were made to the remote database since the previous synchronization. On receiving this data, the MobiLink synchronization server updates the consolidated database, and then sends back all relevant changes to the remote site.

MobiLink characteristics

The MobiLink synchronization server is adaptable and flexible. MobiLink behavior can be adjusted using a comprehensive range of command-line options for the MobiLink synchronization server, *dbmlsrv9* or *mlsrv10*, and the SQL Anywhere synchronization client, *dbmlsync*. You can set a number of options on the typical MobiLink server or client command line to manage the following:

- **Data coordination**   MobiLink allows you to choose selected portions of the data for synchronization. MobiLink synchronization also allows you to resolve conflicts between changes made in different databases. The synchronization process is controlled by synchronization logic, which can be written as a SQL, Java, or .NET application. Each piece of logic is called a script.

- **Automation**   MobiLink has a number of automated capabilities. The MobiLink synchronization server can be instructed to generate scripts suitable for snapshot synchronization, or instructed to generate example synchronization scripts. It can also automatically add users for authentication. Server-initiated synchronization allows you to push data updates to remote databases.

- **Monitoring and reporting**   MobiLink provides two mechanisms for monitoring your synchronizations: the MobiLink Monitor, and statistical scripts. You can monitor scripts, schema contents, row-count values, script names, translated script contents, and row values.

- **Performance tuning**   There are a number of mechanisms for tuning MobiLink performance. For example, you can adjust the upload cache size, degree of contention, number of database connections, number of worker threads, logging verbosity, or BLOB cache size.

- **Data security**   Data integrity can be ensured and protected by the use of certificates and encryption.

Synchronization features

MobiLink synchronization provides the following features:

- **Choice of communication streams**   Synchronization can be carried out over TCP/IP, HTTP, or HTTPS. Additionally, Windows CE devices can synchronize using ActiveSync.

- **Two-way synchronization**   Changes to a database can be made at any location. You can also choose to perform upload-only or download-only synchronizations.

- **File-based synchronization**   Downloads can be distributed as files, enabling offline distribution of synchronization changes. This allows you to create a synchronization file once and distribute it widely.

- **Server-initiated synchronization**   You can initiate MobiLink synchronization from the consolidated database. This means you can push data updates to remote databases, as well as cause remote databases to upload data to the consolidated database.

- **Remote-initiated synchronization**   Synchronization between a remote database and a consolidated database can be initiated at the remote database.

- **Session-based synchronization**   All changes can be uploaded in a single transaction and downloaded in a single transaction. At the end of each successful synchronization, the consolidated and remote databases are consistent.

- **Transactional integrity**   Either a whole transaction is synchronized or none of it is synchronized. This ensures transactional integrity in each database.

- **Data consistency**   MobiLink operates using a loose consistency policy. All changes are synchronized with each site over time in a consistent manner, but different sites might have different copies of data at any instant.

- **Wide variety of hardware and software platforms**   A MobiLink consolidated database system can be one of a variety of widely used database management systems: Sybase SQL Anywhere, Sybase Adaptive Server Enterprise, Oracle, IBM DB2, or Microsoft SQL Server.

  Remote databases can be SQL Anywhere or UltraLite databases. The MobiLink synchronization server runs on Windows or UNIX platforms. SQL Anywhere runs on Windows, Windows CE, or UNIX systems. UltraLite runs on Palm, Windows CE, VxWorks, or Java-based devices.

- **Flexibility**   The MobiLink synchronization server uses SQL, Java, or .NET scripts to control the upload and download of data. The scripts are executed according to an event model during each synchronization. Event-based scripting provides great flexibility in the design of the synchronization process, including such features as conflict resolution, error reporting, and user authentication.

- **Scalability and performance**   MobiLink synchronization is scalable: a single server can handle thousands of simultaneous synchronizations, and multiple MobiLink servers can be run simultaneously using load balancing. The MobiLink synchronization server is multi-threaded and uses connection pooling with the consolidated database. MobiLink provides extensive monitoring and reporting facilities.

- **Ease of getting started**   You can construct simple MobiLink installations quickly and add more complex refinements incrementally for full-scale production work.

Getting started with PocketBuilder

Now that you have been introduced to PocketBuilder, you can learn more about developing PocketBuilder applications by reading Chapter 2, "The PocketBuilder Development Environment." After that, you can prepare to do the PocketBuilder tutorial by reading Chapter 3, "About the PocketBuilder Tutorial," and then you can start doing the tutorial.

The PocketBuilder Development Environment

About this chapter

This chapter introduces the PocketBuilder development environment, which you use in the tutorial in Part 2. It also describes the building blocks of a PocketBuilder application.

Contents

Topic	Page
About PocketBuilder applications	13
The PocketBuilder environment	16
PocketBuilder objects	20

For more information

For a more detailed description of the PocketBuilder development environment, see the PocketBuilder *User's Guide*.

About PocketBuilder applications

What's in a PocketBuilder application?

A PocketBuilder client application contains:

- **A user interface**   Menus, windows, and window controls that users interact with to direct an application.

- **Application processing logic**   Event and function scripts in which you code business rules, validation rules, and other application processing. PocketBuilder allows you to code application processing logic as part of the user interface or in separate modules called custom class user objects.

PocketBuilder applications are event driven

In a client application, users control what happens by the actions they take. For example, when a user clicks a button, chooses an item from a menu, or enters data into a text box, one or more events are triggered. You write scripts that specify the processing that should happen when events are triggered.

Windows, controls, and other application components you create with PocketBuilder each have a set of predefined events. For example, each button has a Clicked event associated with it and each text box has a Modified event. Most of the time, the predefined events are all you need. However, in some situations, you may want to define your own events.

PowerScript language

You write scripts using PowerScript, the PocketBuilder language. Scripts consist of PowerScript commands, functions, and statements that perform processing in response to an event.

For example, the script for a button's Clicked event might retrieve and display information from the database; the script for a text box's Modified event might evaluate the data and perform processing based on the data.

The execution of an event script can also cause other events to be triggered. For example, the script for a Clicked event in a button might open another window, triggering the Open event in that window.

PowerScript functions

PowerScript provides a rich assortment of built-in functions that can act on the various components of your application. For example, there is a function to open a window, a function to close a window, a function to enable a button, a function to update the database, and so on.

PocketBuilder has system functions to control the display of the Soft Input Panel (SIP) on a Pocket PC device or emulator, and window events that respond to changes in the SIP display. The SipUp and SipDown window events occur when the SIP is opened and closed, respectively.

You can also build your own functions to define processing unique to your application.

Object-oriented programming with PocketBuilder

Each menu or window you create with PocketBuilder is a self-contained module called an object. The basic building blocks of a PocketBuilder application are the objects you create. Each object contains the particular characteristics and behaviors (properties, events, and functions) that are appropriate to it. By taking advantage of object-oriented programming techniques such as encapsulation, inheritance, and polymorphism, you can get the most out of each object you create, making your work more reusable, extensible, and powerful.

Window object size

The default window object size is smaller in PocketBuilder than in PowerBuilder, since it is tailored to the size of a Pocket PC screen. Main windows in PocketBuilder applications are also automatically resized to fit the device where they are deployed, and are automatically reoriented if the window layout settings on the device are changed. Main and response windows are currently the only window types available for PocketBuilder applications.

Database connectivity

PocketBuilder provides easy access to information stored in Sybase SQL Anywhere and UltraLite databases. The ODBC database driver for SQL Anywhere is installed with PocketBuilder, as well as a native driver for UltraLite. If you need to access an enterprise database from a PocketBuilder application, you can use MobiLink synchronization technology or convert the enterprise database to a SQL Anywhere database.

Deployment options

For a PocketBuilder project, you must select one of the following deployment options for each project:

- Desktop

- Pocket PC Device (ARM)

- Smartphone Device (SPARM)

- Pocket PC emulator (ARM)

- Smartphone emulator (SPARM)

An application that you deploy to the desktop looks slightly different from the same application deployed to a handheld device or emulator. The desktop application has its own title bar with a maximize, minimize, and close button. Even if you select Close or SmartMinimize icons for a window, these do not display in the window when it is run or debugged on the desktop. Desktop deployment is for testing and demonstration purposes only.

Using the Windows CE Start Menu   By default, PocketBuilder applications are deployed to the *Program Files* directory of a Pocket PC device or emulator and to the *Storage\Program Files* directory on a Smartphone device or emulator, but you can change the deployment directory in the Project painter. On a Pocket PC device, users can run the PocketBuilder applications you deploy by tapping on an application executable file in the directory where it is deployed. (You can also select a Project painter option to launch the application immediately upon deployment.)

Users can take advantage of the built-in PocketBuilder application list utility as a selection vehicle for deployed applications. However, you can also select a Project painter option to deploy an application shortcut to the *Windows\Start Menu\Programs* directory on the Pocket PC or to the *Storage\Windows\Start Menu\Accessories* directory on the Smartphone. That way users are able to find the deployed applications quickly using the Start menu.

CAB file creation and distribution

In PocketBuilder you can generate a CAB file with all the objects from a project and the project executable file. You can use the CAB file to distribute the project to multiple devices.

Online Help and
documentation

PocketBuilder online Help can be accessed using Help buttons and menu items, or by selecting the F1 key from anywhere in PocketBuilder. There are jumps in several places from the online Help to books in HTML format. Manuals are also available on the Sybase Web site.

For more information

For additional information about PocketBuilder, see the *User's Guide*.

The PocketBuilder environment

Workspaces and
targets

In PocketBuilder, you work with one or more PowerScript targets in a workspace. A PowerScript target is a client/server or multitier executable application or a server component.

You can add as many targets to the workspace as you want, open and edit objects in multiple targets, and build and deploy multiple targets at once.

The first lesson in the tutorial shows you how to create a workspace and a PowerScript target.

The development
environment

When you start PocketBuilder, it opens in a window that contains a menu bar and the PowerBar at the top, and the System Tree and Clip windows on the left.
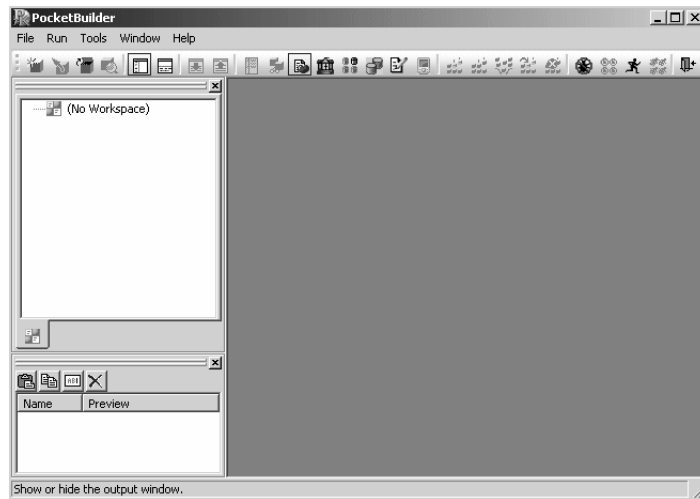
System Tree

The System Tree window can serve as the hub of your development activities. You use it to open, run, debug, and build your targets, and for drag-and-drop programming.

|	
Clip window	The Clip window lets you store code fragments that you use frequently.
Output window	The output of a variety of operations (migration, builds, deployment, project execution, object saves, and searches) displays in an Output window at the bottom of the main window. The Output window opens automatically when output information is generated, but you can open the Output window at any time by clicking the Output window toolbar button.
Painters	Once you have created a workspace and a PowerScript target, you build the components of the target using painters. Painters provide an assortment of tools for enhancing and fine-tuning the objects in a target.
	PocketBuilder provides a painter for each type of object you build. For example, you build a window in the Window painter. There you define the properties of the window and add controls, such as buttons and text boxes.
Wizards	Wizards simplify the creation of applications, objects, and components. PocketBuilder has many wizards including ones that support the conversion of PowerBuilder targets, as well as separate wizards to generate template applications for the Pocket PC and the Smartphone.
To-Do List	The To-Do List displays a list of development tasks you need to do for the current target. Entries on the To-Do list can be created automatically by most PocketBuilder wizards. You can also type in entries or import them from a text file and then link them to a task that you want to complete.
Browser	The Browser lets you see all the objects, methods, variables, and structures that are defined for or available to your PowerScript target. Objects in the Browser can be displayed in alphabetic or hierarchical order. The Browser displays methods with their complete prototypes (signatures), which include the datatypes of all arguments and return values.
PowerBar	The PowerBar displays when you begin a PocketBuilder session. The PowerBar is the main control point for building PocketBuilder applications. You can use the New, Inherit, or Open buttons on the PowerBar to open all of the PocketBuilder painters. From the PowerBar, you can also open the Browser, debug or run the current application, and build and deploy the workspace.
PainterBar	When you open a painter or editor, PocketBuilder displays a new window that has a workspace in which you design the object you are building. PocketBuilder also displays one or more PainterBars with buttons that provide easy access to the tools available in the painter or editor.

StyleBar

The StyleBar displays when you open any painter that can contain text controls, such as the Window painter. Using buttons on the StyleBar, you can modify text properties such as the font and point size.

PowerTips

When you leave the mouse pointer over a button for a second or two, PocketBuilder can display a brief description of the button (a PowerTip). The ability to display PowerTips is toggled on and off by selecting the Show PowerTips menu item in any toolbar pop-up menu.
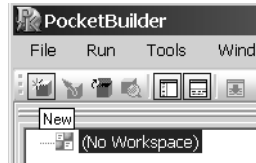
You can also include brief descriptive text for all toolbar buttons by selecting ShowText from any toolbar pop-up menu.

Customizing the environment

In addition to displaying text in toolbar buttons, you can move the toolbars around, add new toolbars, and customize existing ones. You can add buttons for opening painters and performing other activities.

You can also rearrange the System Tree, Clip, and Output views, set up custom layouts for each painter, choose whether PocketBuilder opens your last workspace at start-up with or without painters and editors open, customize shortcut keys, and change the colors and fonts used in scripts.

PowerBar buttons

The buttons in the PowerBar give you quick access to the most common PocketBuilder tasks:

Table 2-1: Buttons in the PowerBar

Button	Use to
New	Create new workspace, target, component, or other object, or open a tool.
Inherit	Inherit from menu, user object, or window.
Open	Open an existing application, DataWindow, function, menu, pipeline, project, query, structure, user object, window, HTML page, HTML frame, style sheet, or script file.
Preview	Preview a window or DataWindow object.

Button	Use to
SysTree	Show or hide the System Tree window.
Output	Show or hide the Output window.
Next	Move to the next line in the Output window.
Previous	Move to the previous line in the Output window.
To-Do List	Display a list of development tasks you need to do. These can be self entered or entered automatically by PocketBuilder wizards.
Browser	View object information (such as object properties or global variables) and copy, export, or print it.
Clip Window	Show or hide the Clip window.
Library	Create and maintain libraries of PocketBuilder objects.
DB Prof	Specify how to connect to a database.
Database	Maintain databases, control user access to databases, and manipulate data in databases.
Edit	Edit a file.
Sel Emul	Select and launch an emulator.
I. Build	Start an incremental build of the workspace.
F. Build	Start a full build of the workspace.
Deploy	Build and deploy the workspace.

Introduction to PocketBuilder

Button	Use to
Skip	When a series of operations is in progress, such as a full deploy of the workspace, skip to the next operation.
Stop	Stop a build or deploy operation or series of operations.
Debug	Debug the current target.
Sel_Debug	Select a target and debug it.
Run	Run the current target.
SelRun	Select a target and run it.
Exit	Exit from PocketBuilder.

PocketBuilder objects

The basic building blocks of a PowerScript target are objects:

Table 2-2: Basic building blocks of a PowerScript target

Object	Use
Application	Entry point into an application
Window	Primary interface between the user and a PocketBuilder application
DataWindow	Retrieves and manipulates data from a relational database or other data source
Menu	List of commands or options that a user can select in the currently active window
Global function	Performs general-purpose processing
Query	SQL statement used repeatedly as the data source for a DataWindow object
Structure	Collection of one or more related variables grouped under a single name

Object	Use
User object	Reusable processing module or set of controls, either visual or nonvisual
Project	Packages application for distribution to users

These objects are described in more detail in the sections that follow.

Application object

The Application object is the entry point into an application. It is a discrete object that is saved in a PocketBuilder library (PKL file), just like a window, menu, function, or DataWindow object.

The Application object defines application-level behavior, such as which fonts are used by default for text, and what processing should occur when the application begins and ends.

When a user runs the application, an Open event is triggered in the Application object. The script you write for the Open event initiates the activity in the application. When the user ends the application, the Close event in the Application object is triggered.

The script you write for the Close event typically does all the cleanup required, such as closing a database or writing to a preferences file. If there are serious errors during execution that are not caught using PocketBuilder's exception handling mechanism, the Application object's SystemError event is triggered.

**Figure 2-1: Application life cycle**
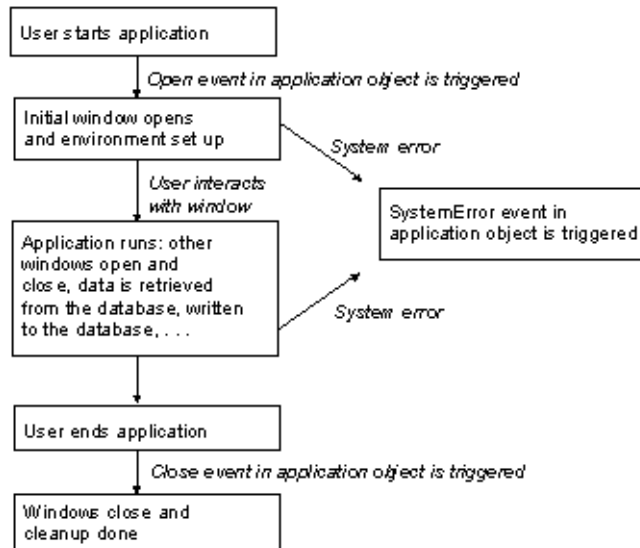
Windows

Windows are the primary interface between the user and a PocketBuilder application. Windows can display information, request information from a user, and respond to the user's mouse or keyboard actions.

A window consists of:

• Properties that define the window's appearance and behavior (for example, a window might have a title bar and a Minimize box)

• Events triggered by user actions

• Controls placed in the window

Windows can have various kinds of controls, as illustrated in the following picture:
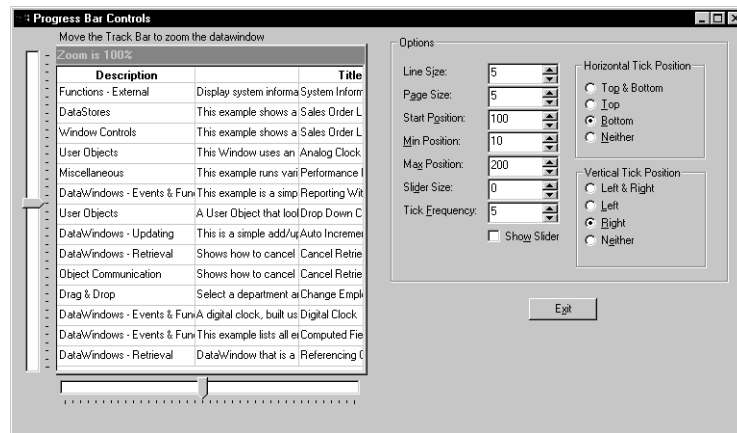
On the left of the window is a DataWindow control with horizontal and vertical trackbars. On the right is a group box that contains static text controls (containing descriptive labels), edit mask controls (as they appear when the SpinControl property is on), a check box, and two smaller group boxes with radio buttons. Under the main group box is a command button.

DataWindow objects

A DataWindow object is an object that you use to retrieve and manipulate data from a relational database or other data source (such as an Excel worksheet or dBASE file).

Presentation styles   DataWindow objects also handle the way data is presented to the user. You can display the data in the Freeform, Graph, Grid, Group, or Tabular presentation style.

There are many ways to enhance the presentation and manipulation of data in a DataWindow object. For example, you can include computed fields, pictures, and graphs that are tied directly to the data retrieved by the DataWindow.
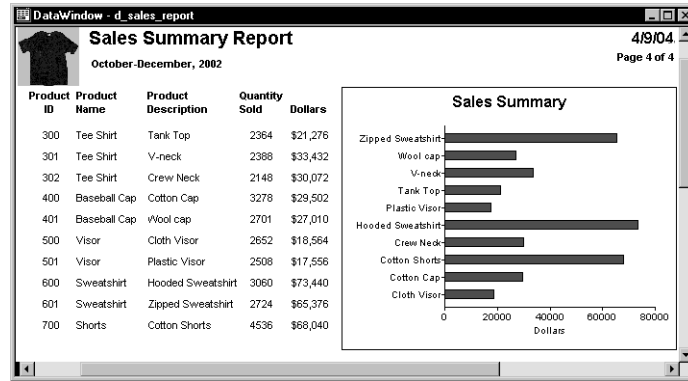
Display formats, edit styles, and validation   You can specify how to display the values for each column, and you can validate data entered by users in a DataWindow object. You do this by defining display formats, edit styles, and validation rules for columns.

For example:

• If a column can take only a small number of mutually exclusive values, you can have the data appear as radio buttons in a DataWindow so users know what their choices are.
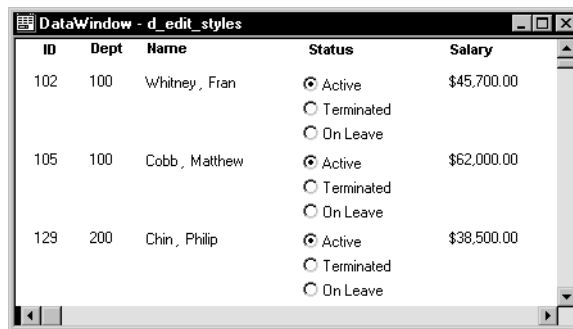
- If the data includes phone numbers, salaries, and dates, you can format the display to suit the data.

- If a column can take numbers only in a specific range, you can specify a simple validation rule for the data. This can spare you from writing code to make sure users enter valid data.

Menus

Menus are lists of items that a user can select from a menu bar for the active window. The items on a menu are usually related. They provide the user with commands (such as Open and Save As on the PocketBuilder File menu) or alternate ways of performing a task (for example, the items on the Edit menu in the Window painter correspond to buttons in the PainterBar).

You can select menu items with the mouse or with the keyboard, or use accelerator (mnemonic access) keys defined for the items. You can define your own keyboard shortcuts for any PocketBuilder menu item from a dialog box that you open with the Tools>Keyboard Shortcuts menu item.

A drop-down menu is a menu under an item in the menu bar. A cascading menu is a menu that appears to the side of an item in a drop-down menu.
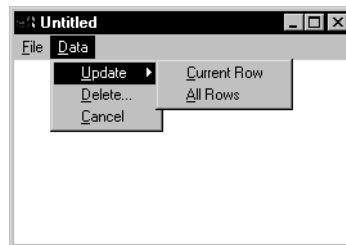
Each choice in a menu is defined as a Menu object in PocketBuilder. The preceding window shows two Menu objects on the menu bar (File and Data), three Menu objects on the drop-down Data menu (Update, Delete, and Cancel), and two Menu objects on the cascading menu beside Update (Current Row and All Rows).

Global functions

PocketBuilder lets you define two types of functions:

- Object-level functions are defined for a particular type of window, menu, or other object type and are encapsulated within the object for which they are defined. These are further divided into system functions (functions that are always available for objects of a certain object class) and user-defined functions.

- Global functions are *not* encapsulated within another object, but instead are stored as independent objects.

  Unlike object-level functions, global functions do not act on particular instances of an object. Instead, they perform general-purpose processing such as mathematical calculations or string handling.

Queries

A query is a SQL statement that is saved with a name so that it can be used repeatedly as the data source for a DataWindow object. Queries enhance developer productivity, because they can be coded once but reused as often as necessary.

Structures

A structure is a collection of one or more related variables of the same or different data types grouped under a single name. In some languages, such as Pascal and COBOL, structures are called records.

Structures allow you to refer to related entities as a unit rather than individually. For example, you can define the user's ID, address, access level, and a picture (bitmap) of the employee as a structure called user_struct, and then refer to this collection of variables as user_struct.

There are two kinds of structures:

- Object-level structures are associated with a particular type of object such as a window or menu. These structures can always be used in scripts for the object itself. You can also choose to make the structures accessible from other scripts.

- Global structures are not associated with any object or type of object in an application. You can declare an instance of the structure and reference it in any script in an application.

User objects

Applications often have features in common. For example, several applications might have a Close button that performs a certain set of operations and then closes the window, or they might have DataWindow controls that perform the same type of error checking. Several applications might all require a standard file viewer.

If you find yourself using the same application feature repeatedly, you should define a user object. You define the user object once and use it as many times as you need.

User objects can be visual or nonvisual. They can be further divided into standard or custom user objects. Standard user objects, whether visual or nonvisual, are system objects that are always available with PocketBuilder. You can also use controls for external visual objects that were created outside PocketBuilder. The main types of user objects are:

- **Visual user objects**   These are reusable controls or sets of controls that have a consistent behavior. For example, a visual user object could consist of several buttons that function as a unit. The buttons could have scripts associated with them that perform standard processing. Once an object is defined, you can use it as often as you need.

- **Nonvisual user objects**   These are reusable processing modules that have no visual component. Standard class user objects are nonvisual objects that inherit events and properties from built-in system objects. You typically use nonvisual objects to define business rules and other processing that acts as a unit.

   For example, you might want to calculate commissions or perform statistical analysis in several applications. To do this, you could define a custom class user object. To use a custom class user object, you create an instance of the object in a script and call its functions.

   Custom class user objects, which define functions and variables, are the foundation of PocketBuilder multitier applications. This is because you typically use nonvisual components for applications that are run on a server.

Libraries

You save objects, such as windows and menus, in PocketBuilder libraries (PKL files). When you run an application, PocketBuilder retrieves the objects from the library. Applications can use as many libraries as you want. When you create an application, you specify which libraries it uses.

Projects

You can create Project objects that build your executable applications. You can build for the desktop or a device, specify deployment options, CAB file packaging, and version information, and sign the application and CAB files using certificates.

CHAPTER 3   **About the PocketBuilder Tutorial**

About this chapter   This chapter describes what you will do in the PocketBuilder tutorial and
                            how to get set up for it.

Contents

Topic	Page
Learning to build a PocketBuilder application	27
How you will proceed	28
Setting up for the tutorial	29

Learning to build a PocketBuilder application

The PocketBuilder tutorial has five lessons that show you how to get
started with PocketBuilder. The tutorial provides step-by-step instructions
for these development tasks:

• Building a basic application with no database connection

• Customizing the PocketBuilder development environment

• Connecting to a database

• Building a DataWindow-based employee application with database
  connection to SQL Anywhere and deployment to a Pocket PC device

• Building a DataWindow-based sales application (using a skeleton
  application that is provided for you) with database connection to SQL
  Anywhere and deployment to a Pocket PC device

How you will proceed

Table 3-1 describes what you will do in each of the tutorial lessons.

Table 3-1: Tutorial lessons and what you will accomplish

Lesson	What you will do
1	Start PocketBuilder; begin familiarizing yourself with the development environment; use the Workspace wizard and the PocketPC Application Creation wizard to create an application object, a window, and a menu in a PocketBuilder workspace and target; run the application; build the application; and deploy the application to a PocketPC device or emulator
2	Explore the PocketBuilder environment and customize the workspace
3	Create a database profile and look at the demonstration database and the Database painter
4	Create a workspace and target; create and preview a new DataWindow object; attach the DataWindow object to a DataWindow control; code the custom event to connect to the data source; test the application on the desktop; deploy and run the application on a handheld device or emulator
5	Set up the SQL Anywhere remote and consolidated databases; modify a skeleton Sales application that is provided to you; create a DataWindow for sales order information; add menu items to the application menu; create a MobiLink connection; create the main application window; test the application on the desktop; deploy the application to a Pocket PC device; run the application; and troubleshoot the application

Tutorial solutions   The solutions for the tutorial lessons are in the Tutorial\Solutions directory.

How long it will take

You can do the tutorial in about three hours, or you can stop after any lesson and continue at another time.

If you are interrupted
You can save your work and exit PocketBuilder at any time. When you are ready to continue, open the tutorial workspace and continue where you left off.

What you will learn

This tutorial will not make you an expert in PocketBuilder. Only experience with building real-world applications can do that. It will give you hands-on experience, though, and provide a foundation for continued growth.

You will learn basic PocketBuilder techniques and concepts, including those listed in Table 3-2:

Table 3-2: Features demonstrated in the PocketBuilder tutorial

How to use the	To
Application painter	Define an Application object and application-level scripts
Window painter	Create a window and modify its properties
Database painter	Define a database profile and connect to the database
DataWindow painter	Define DataWindow selection and display options
Menu painter	Define menus and menu items
Layout view	Design how a window, menu, and DataWindows will look when you run the application
Script view	Define scripts for applications, windows, window controls, and menus
Project painter	Create an executable version of an application

In addition, you will learn how to customize the PocketBuilder development environment, deploy applications to a PocketPC device, and create a MobiLink connection and use MobiLink synchronization.

Setting up for the tutorial

Before you start the tutorial, you need to make sure that you can connect to a database and that you have the tutorial files.

SQL Anywhere version
The databases in this tutorial are SQL Anywhere 10 databases. If you are using SQL Anywhere Studio 8.x or 9.x with Adaptive Server Anywhere databases, see the tutorial lessons in the previous version of this book on the Sybooks Web site at http://sybooks.sybase.com/nav/detail.do?docset=614#dt3.

Connecting to a database

The tutorial uses the SADemo_10 demonstration database that installs with PocketBuilder 2.1. PocketBuilder also installs command scripts that you use to create remote and consolidated databases for the SalesDB tutorial lesson. The *MakeDB.cmd* script creates Adaptive Server Anywhere 9 databases and the *MakeDB10.cmd* script creates SQL Anywhere 10 databases. SQL Anywhere databases require the Sybase SQL Anywhere database and server engines. If you do not already have SQL Anywhere on your local machine or server, you must install it now.

When you install SQL Anywhere 10 to your device, make sure you select the International Components for Unicode (ICU) check box. For the SalesDB tutorial lesson, you also need to select the MobiLink Synchronization check box in the SQL Anywhere 10 installation program. If you did not select these check boxes when you installed SQL Anywhere, you must reinstall it to the device with these components selected. You can reinstall SQL Anywhere from the desktop Start menu when you are connected to the device with ActiveSync.

Some Pocket PC 2002 devices can freeze while running SQL Anywhere 10 applications. For lessons using SQL Anywhere 10 databases, you should use Pocket PC 2003, Windows Mobile 5, or Windows Mobile 6 devices.

If you are using Adaptive Server Anywhere 9
You can install Adaptive Server Anywhere 9 from the PocketBuilder 2.0 CD. However, you must purchase SQL Anywhere 10 separately. If you want to use the Adaptive Server Anywhere version that installs with PocketBuilder, please see the tutorial lessons in the previous version of this book on the Sybooks Web site at http://sybooks.sybase.com/nav/detail.do?docset=614#dt3.

Requirements

To test the applications you develop in the tutorial on the Pocket PC, you must have an ARM-based device or emulator.

Before you begin Lesson 4, "Creating an Employee List," make sure you have the PocketBuilder Virtual Machine (VM) as well as SQL Anywhere installed on your deployment device or emulator. For Lesson 5, "Creating a Sales Application," you also need MobiLink Client on the deployment device or emulator. For information, see the PocketBuilder or SQL Anywhere *Installation Guide*.

The Tutorial directory

The tutorial also uses the files listed in Table 3-3.

Table 3-3: Files required by the PocketBuilder tutorial

File	Contents
ASADemo_10.db and *ASADemo_10.log*	The database and log file that you use in Lesson 3, "Connecting to the Database" and Lesson 4, "Creating an Employee List." These files are installed in the *Code Examples\SA10DemoData* directory.
MakeDB10.cmd	A command file located in the *Tutorial\SalesDB\db* directory that creates the remote and consolidated databases for Lesson 5, "Creating a Sales Application"
tutorial.ico	An icon located in the *Tutorial\HelloWorld* directory

When you install PocketBuilder, these files are installed in the *Tutorial* directory, which is a subdirectory of the PocketBuilder installation directory.

The Tutorial\Solutions directory

The *Tutorial\Solutions* directory has solutions for the HelloWorld, EmployeeList, and SalesDB lessons. The EmployeeList and SalesDB solution PKLs in the main Tutorial\Solutions directory presume you are using Adaptive Server Anywhere 9 databases. A separate subdirectory contains solutions for these lessons when you use SQL Anywhere 10 databases.

The solutions contain all the objects and scripts that you create in the tutorial, as well as workspace and target files. You can use this solutions library as a reference if you need to.

PART 2   **PocketBuilder Tutorial**

This part is a tutorial that shows you how to get started with PocketBuilder. It provides step-by-step instructions for creating a:

- Basic application with no database connection

- Customized PocketBuilder development environment

- Connection to the demonstration database

- DataWindow-based employee application with database connection to SQL Anywhere and deployment to a PocketPC device

- DataWindow-based sales application with database connection SQL Anywhere, use of MobiLink synchronization, and deployment to a PocketPC device

L E S S O N   1

Creating a Basic Hello World Application

This lesson provides the information you need to start PocketBuilder and create a basic Hello World application. No database access is needed for this basic application.

In this lesson you:

- Create a new workspace
- Create a target
- Specify an icon for the application
- Add a button and text to the window
- Run the application on the desktop
- Build and deploy the application
- Run the application on the device

How long does it take?
About 25 minutes.

Create a new workspace

Where you are
> Create a new workspace
  Create a target
  Specify an icon for the application
  Add a button and text to the window
  Run the application on the desktop
  Build and deploy the application
  Run the application on the device

The workspace is where you build, edit, and debug PocketBuilder targets. You can build several targets within a single workspace. You can also run the targets from the desktop.

Now you start PocketBuilder and create a new workspace.

1   Double-click the PocketBuilder icon on the desktop (representing PK20.EXE) in the Sybase>PocketBuilder 2.0 path
or
Select Programs>Sybase>PocketBuilder 2.0>PocketBuilder from the Windows Start menu.

If the Welcome to PocketBuilder dialog box displays
Select the Don't Show This Dialog Again check box if you want to keep PocketBuilder from displaying the Welcome dialog box every time you start PocketBuilder. Select the Reload Last Workspace On Starting PocketBuilder check box if you want PocketBuilder to load the most recently used workspace when you start a PocketBuilder session. When you are finished with the Welcome dialog box, click Close This Dialog.

The PocketBuilder development environment displays.

If this is the first time you are opening PocketBuilder on your machine, you see only a top-level entry in the System Tree to indicate that no workspace is currently open. Otherwise, the System Tree might show a workspace with targets and objects in it.
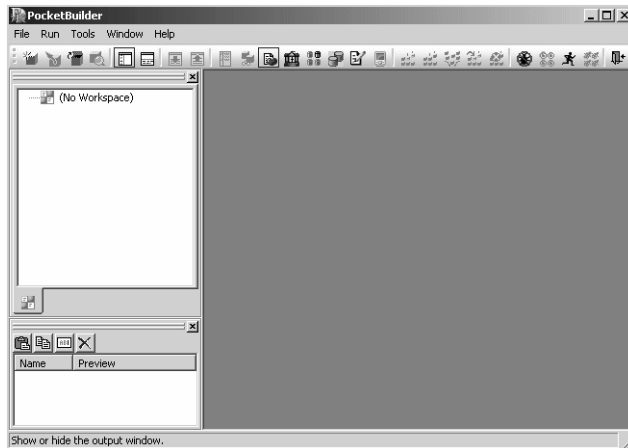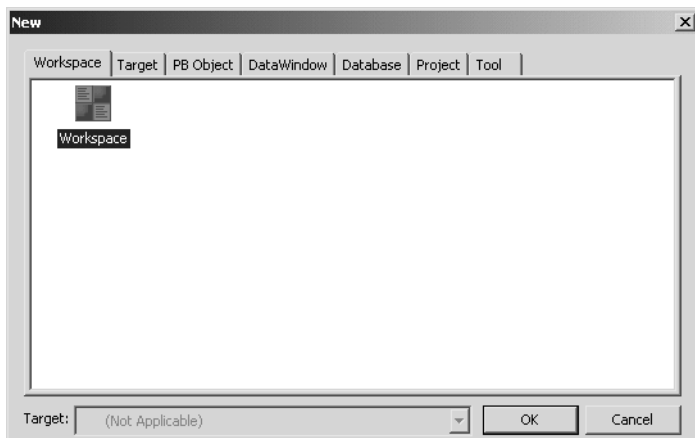

2   Select New from the File menu
or
Click the New button in the PowerBar.

The Workspace page of the New dialog box displays.

The only option in the Workspace page is Workspace, so it is already selected.

If the New dialog box displays a different page
PocketBuilder displays the page of the New dialog box that was used
before the dialog box was last closed. In this exercise, make sure that the
Workspace page displays by clicking the Workspace tab.

---

3 Click OK.

The New Workspace dialog box displays.

4 Navigate to the \PocketBuilder 2.0\Tutorial\HelloWorld directory.

The Tutorial directory is located under the PocketBuilder 2.0 directory.
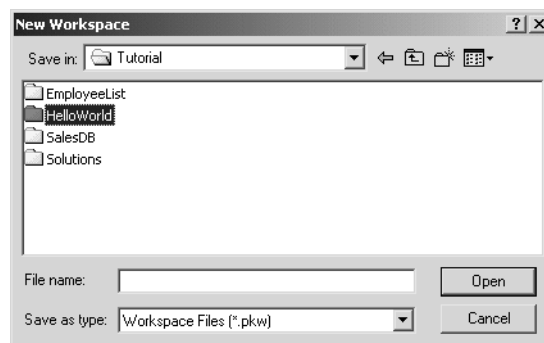
5 Type basic_tutorial **in the File name text box.**

**6 Click Save to save the new workspace as basic_tutorial in the
\PocketBuilder 2.0 \Tutorial\HelloWorld directory.**

The New Workspace dialog box closes and the workspace you created
appears as the first item in the System Tree.

Create a target

Where you are
Create a new workspace
> Create a target
Specify an icon for the application
Add a button and text to the window
Run the application on the desktop
Build and deploy the application
Run the application on the device

Now you create a new target using the PocketPC Application Creation Wizard. Based on the choices you make, the wizard creates precoded events, menus, windows, and user objects in addition to the application object.

About the Smartphone Application Creation Wizard
To deploy an application to a Smartphone instead of a Pocket PC, you would use the Smartphone Application Creation Wizard.

---

1   **Select New from the File menu and click the Target tab**
     or
     Right-click basic_tutorial in the System Tree, select New from the pop-up menu, and click the Target tab.

     The Target page of the New dialog box displays.
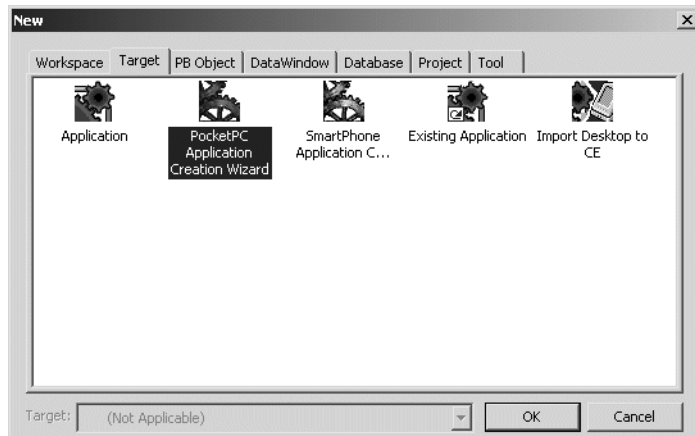
2   Select the PocketPC Application Creation Wizard icon and click OK.

The PocketPC Application Creation Wizard displays. In most wizards, the first page explains what the wizard is used for. As you step through the wizard, you can press F1 to get Help on most fields.

**3   Click Next until the Specify New Application and Library page displays.**

4   Type basic_tutorial **in the Application Name text box.**

When you click Next (or you click in the Library or Target text box), the wizard will automatically assign file names to a library and target that use this application name. It assigns the library a PKL extension and the target a PKT extension.
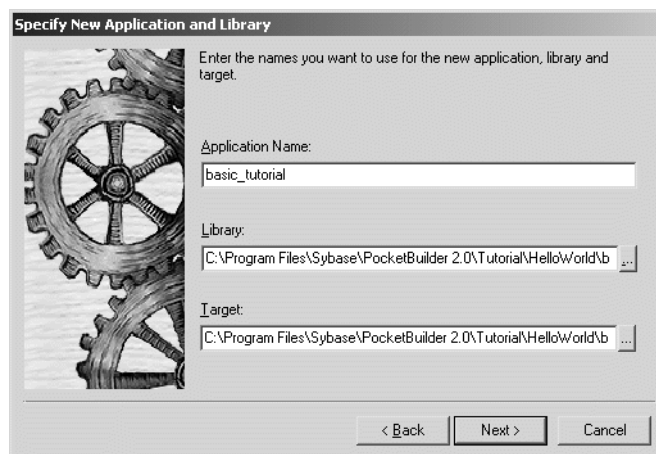
**5   Click Next until the Specify Connectivity page displays.
Select None for the Connection Options if it is not already selected.**

You accept the default library search path and window and menu names. This lesson does not require database connectivity. (You could add a connection object later if you wanted to.)
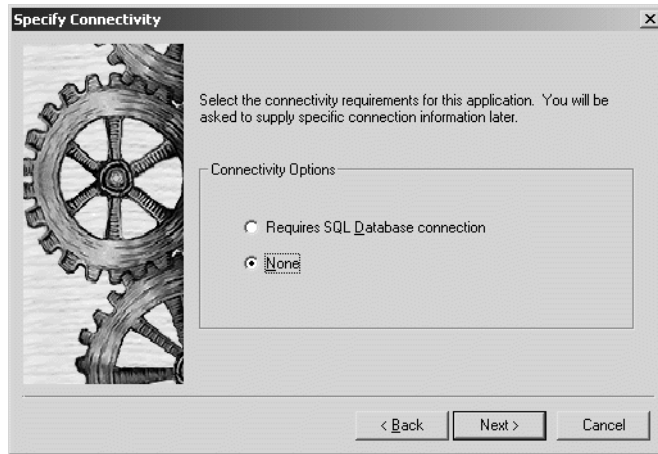
6   Click Next.

The Specify Project Object page displays.
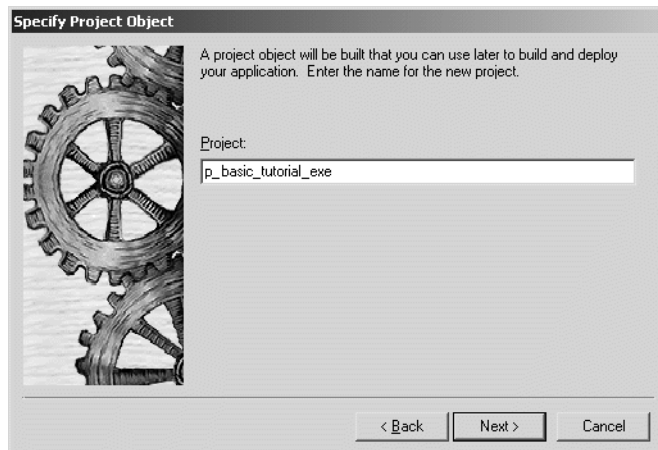
The wizard will create a project object that you can open in the Project painter. The Project painter allows you to streamline the generation of the files your target needs, to create an executable for desktop deployment, and rebuild your application easily when you make changes to the application.

7   Click Next until the Specify Deployment Configuration page displays.

You accept the default names for the project name and the executable file name and you do not choose to build dynamic libraries.

You will also now accept the default directories in the Specify Deployment Configuration wizard page. The Target directory (with a default of \Program Files) is the directory on the device or emulator where you want to deploy your application. The Build directory is the directory where you want to build your application. The default "." is the directory that contains the main PocketBuilder application library.

Target directory for a Smartphone
If you were creating a Smartphone application rather than a Pocket PC application, you would typically use \Storage\Program Files as the target directory.
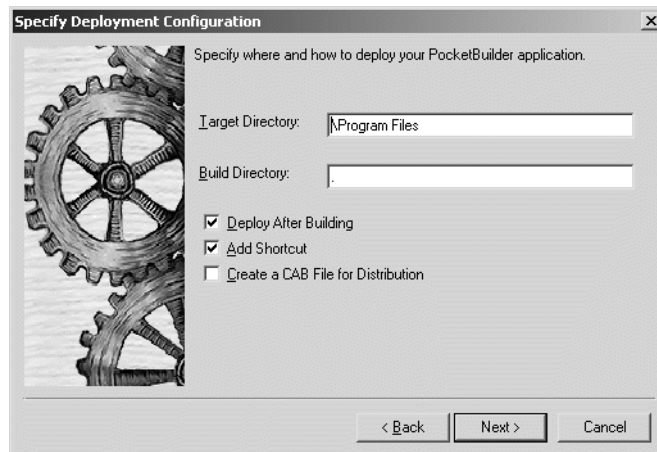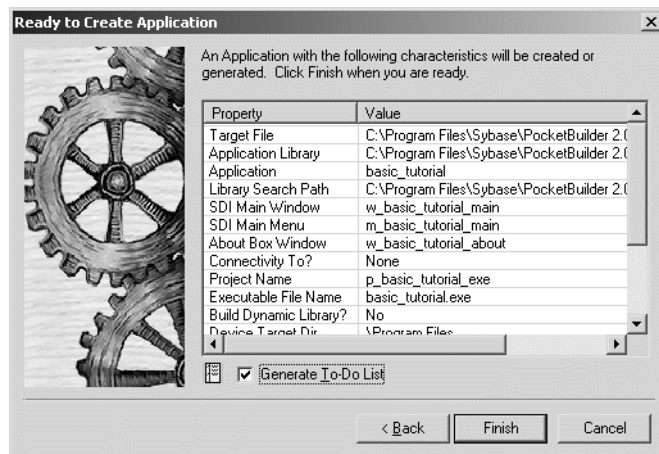
8   Click Next.

The Specify Version Information page displays.
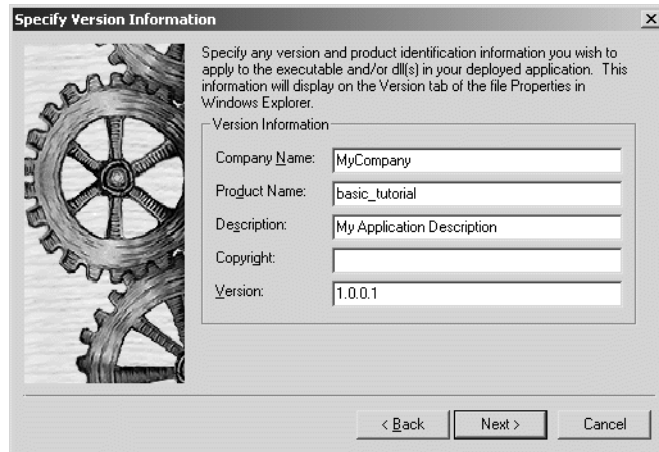
You accept the default version information.

9   Click Next to display the Ready To Create Application page.

This is the last wizard page. It lists your current selections so that you can review them and use the Back button to go back and change them if necessary.

**10 Notice that the Generate To-Do List check box is selected.
Click Finish.**

The PocketPC Application Creation Wizard creates the *basic_tutorial.pkt*
target and the *basic_tutorial.pkl* library, and sets the new basic_tutorial
application as the default application.

You can expand the System Tree to view all the objects that have been
created by the PocketPC Application Creation Wizard. The System Tree
does not display the file extension of the pbtutor target, but it does display
the directory where the target file is saved.

The *basic_tutorial.pkl* library displays under the basic_tutorial target in
the System Tree. It contains the target Application object, which has the
same name as the target object, basic_tutorial, but displays under the
library file.
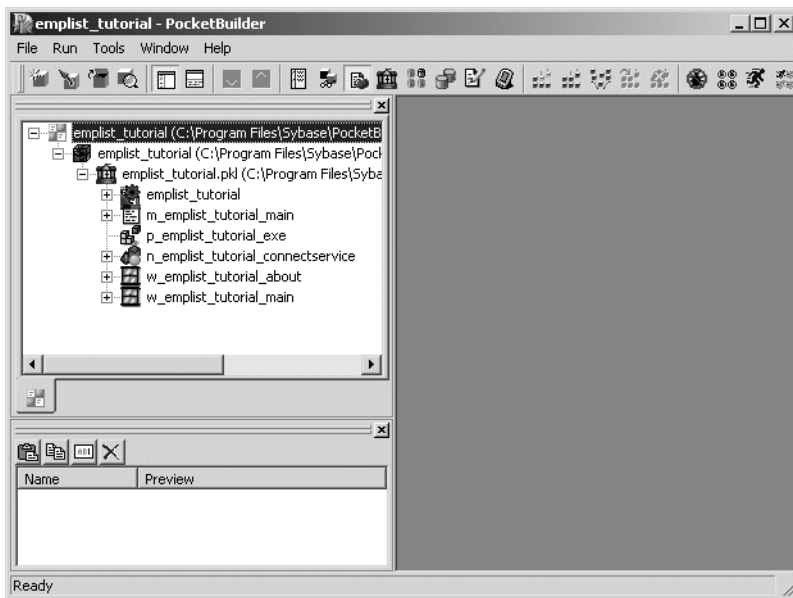
Other objects generated by the wizard also display under the library file.
One of these is the main window w_basic_tutorial_main.

11   Click the To-Do List button in the PowerBar.

The To-Do List was generated by the PocketPC Application Creation Wizard. The To-Do List is created automatically by most wizards to guide you through the continued development of objects of different types that you will need for building an application. Some To-Do List entries are hot-linked to get you quickly to the painter (and the specific object you need) or to a wizard.
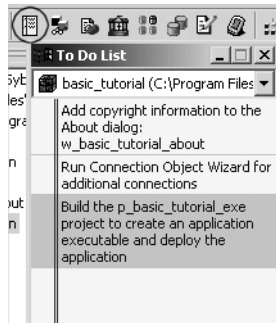
Next you specify an icon for the application and then you work in the main window.

12   Close the To-Do List.

You do not use it in this lesson. For information about the To-Do List, see the PocketBuilder *User's Guide*.

Specify an icon for the application

Where you are
Create a new workspace
Create a target
> Specify an icon for the application
Add a button and text to the window
Run the application on the desktop
Build and deploy the application
Run the application on the device

Now you specify an icon for the application from the Properties view in the Application painter. The icon appears in the PocketBuilder workspace when you minimize the application during execution. PocketBuilder also includes the icon automatically when you create an executable file.

The icon you specify is also used for a shortcut on a device, the image that displays in the Pocket PC File Explorer, and the image in the Pocket PC Recently Run List.

1 **Double-click the basic_tutorial Application object in the System Tree**
or
Right-click the basic_tutorial Application object in the System Tree and select Edit from the pop-up menu.

The basic_tutorial Application object is located under the basic_tutorial library, which is under the basic_tutorial target object that you created with the Template Application wizard. Different views of the Application object display in the Application painter.

2   Make sure the Properties view displays in the Application painter.

If the Properties view is not open, you can open it by selecting View>Properties from the menu bar. The menu item is grayed if the Properties view is already open.
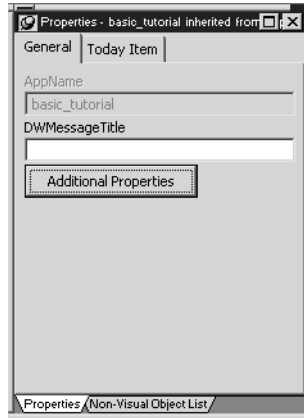
3   Click the Additional Properties button in the Properties view.

A tabbed Application property sheet displays.

4   Select the Icon tab.

**5   Click Browse.
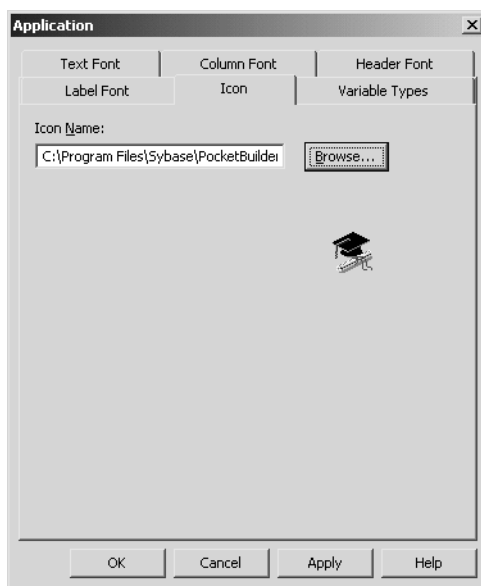Navigate to the PocketBuilder 2.0\Tutorial\HelloWorld directory.**

**6   Select the tutorial.ico file.
Click Open.**

If you do not see the ICO file extension
You do not see ICO file extensions if the Hide File Extensions for Known File Types check box is selected in the Options dialog box of your Windows Explorer.

The tutorial icon displays on the Icon page of the Application property sheet.

7 **Click OK.**
Click the Save button in PainterBar1 or select File>Save.
Click the Close button in PainterBar1 or select File>Close.

Add a button and text to the window

Where you are
Create a new workspace
Create a target
Specify an icon for the application
> Add a button and text to the window
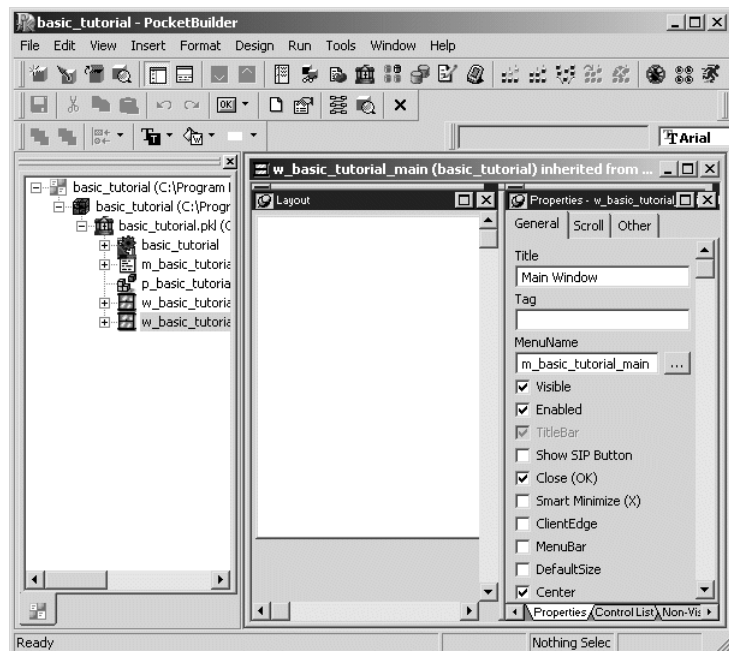Run the application on the desktop
Build and deploy the application
Run the application on the device

Now you add a button and text to the application's main window. When you run the application on the desktop, the main window displays in the position and size that you specify.

Window size on mobile devices
On mobile devices, the main window always displays full size and centered.

1   Double-click w_basic_tutorial_main in the System Tree.

The Window painter opens the application's main window.

2   Change the title of the window to Hello World on the General page in the Properties view.

3   Select the Center check box on the General page in the Properties view if it is not already selected.

Now when you run the application on the desktop, the main window will be centered. On the PocketPC or Smartphone, this does not affect the window position.

4   Select the Close (OK) check box on the General page in the Properties view if it is not already selected.

Now when you run the application on the desktop, you can close it by clicking the window's Close button in the upper right. On the PocketPC, the Close button becomes an OK button that you can use to Close the application.

On the Smartphone
The Close (OK) button property is ignored on the Smartphone.

5   Select Insert>Control>CommandButton from the PocketBuilder menu.
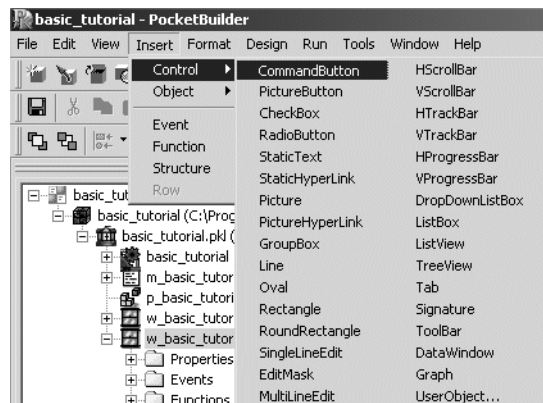
6   In the Layout view, click in the space near the upper left corner of the window.

A command button with the label *none* displays in the window.

**7   Click the command button.
In the Properties view, change the name property to cb_ok.
Change the text property to OK.**
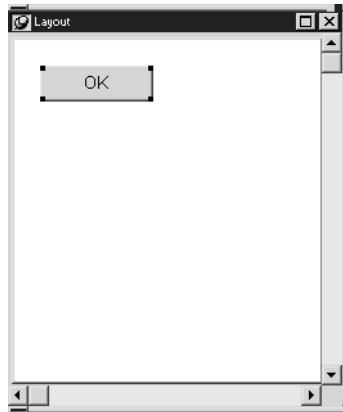
**8   Select Insert>Control>StaticText from the PocketBuilder menu.
Click in the space below the command button.**

A static text control with the label *none* displays in the window.

**9   Be sure the static text control is selected.
In the Properties view, change the name property to st_hello.
Delete the default text property and leave the property blank.**

10   Double click the command button in the Layout view.

The Script view displays with the cb_ok clicked event selected.

11 Click in the blank script area and type the following code for the clicked event:

```
st_hello.text = "Hello World"
```
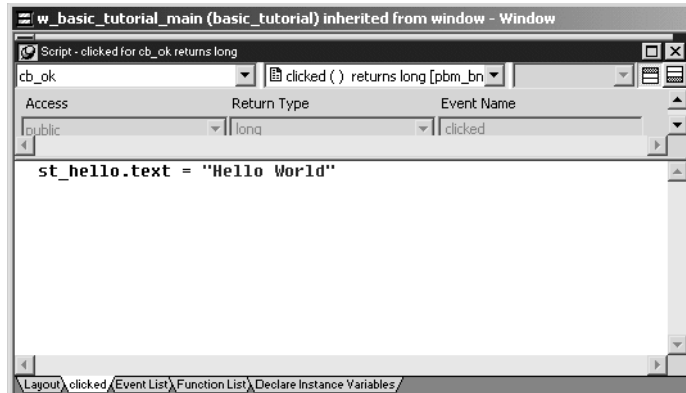

**12 Select File>Close from the PocketBuilder menu.
Click Yes when you are prompted to save your changes.**

Next you run the application.
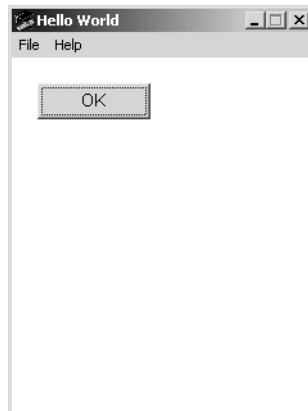
Run the application on the desktop

Where you are
Create a new workspace
Create a target
Specify an icon for the application
Add a button and text to the window
> Run the application on the desktop
Build and deploy the application
Run the application on the device

Now you run the application on the desktop to see how it works. By running the application, you can see the window and menus that were created for you when PocketBuilder generated the application based on your choices.
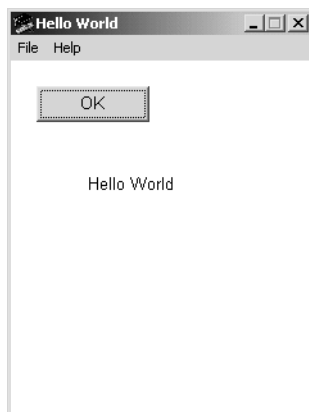
1   Click the Run button in the PowerBar.

The window displays.

2 Click the OK button.

Hello World displays in the window.

3 In Hello World, select File>Exit.

The application closes and you return to the PocketBuilder development environment.

When you exit and restart PocketBuilder, you might want to have PocketBuilder in the state it was in when you exited, with the workspace and painters you were working in open.
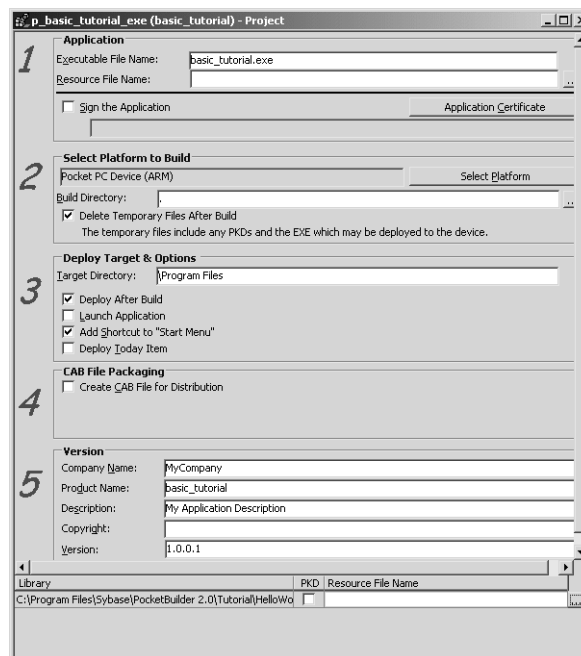
Build and deploy the application

Where you are
Create a new workspace
Create a target
Specify an icon for the application
Add a button and text to the window
Run the application on the desktop
> Build and deploy the application
  Run the application on the device

Now you can build the application and deploy it to a device or emulator by running the project object that was created by the PocketPC Application Creation Wizard in the "Create a target" task on page 39 of this lesson.

1   Double-click the p_basic_tutorial_exe project object in the System Tree.

The Project painter opens. The build platform is PocketPC Device (ARM), which was specified by the wizard.

2   Select the Add Shortcut to "Start Menu" check box if it is not already selected.

Unless the number of Start menu items has already reached the device limit, deploying with this option adds a Start menu item for the application on the device.

3   Select Run>Build and Deploy Workspace or click the Deploy button in the PainterBar.

The project is built and deployed and the application is copied to the device's \Program Files directory.

Next you run the application on the Pocket PC device.

Run the application on the device

Where you are
Create a new workspace
Create a target
Specify an icon for the application
Add a button and text to the window
Run the application on the desktop
Build and deploy the application
> Run the application on the device

Now you can run the application on the Pocket PC device.

1   On the Pocket PC device, tap the Start menu.

2   Tap Programs and then tap the basic_tutorial application icon.

Another way to run the application
You can also tap the Start menu, then tap the PocketBuilder 2.0 menu, and then tap the application you want to start—in this case basic_tutorial.exe.

The Hello World application starts.
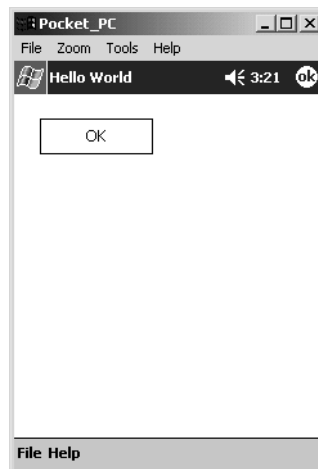
3 Tap the OK button.

The text Hello World displays in the Pocket PC window.
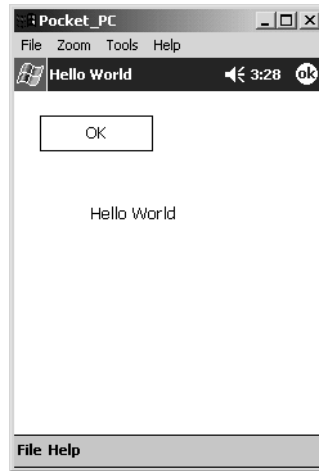
4 Tap the circular *ok* button (the one on the menu at the far right).

The Hello World application closes and you return to the Programs directory.

5 Tap the Close button to close the Programs directory.

The application closes and you return to the Pocket PC Start menu.

When you exit and restart PocketBuilder, you might want to have PocketBuilder in the state it was in when you exited, with the workspace and painters you were working in open.

6 In PocketBuilder, select Tools>System Options from the menu bar and then click the Workspaces tab.

7 **Make sure that the Reload Workspace on Startup and the Reload Painters When Opening Workspace check boxes are selected.**
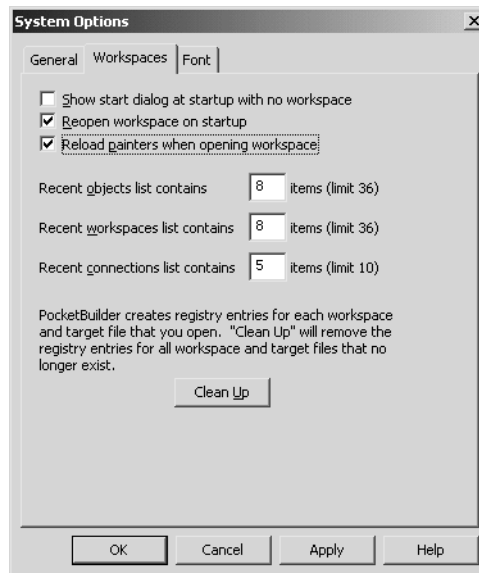
8 **Click OK.**

Now when PocketBuilder starts up, it opens the workspace and the painters that were open when you exited. If your were coding in PocketBuilder when you exited, the last script you were working on opens at the last line you edited.

LESSON 2   Customizing the PocketBuilder Environment

This lesson provides the information you need in order to become familiar with the PocketBuilder environment and to customize the workspace. This lesson is optional—you can skip to Lesson 3 if you want to.

In this lesson you:

* Manipulate the System Tree window

* Open an object

* Manipulate views

* Set up the toolbars

How long does it take?
About 25 minutes.

Manipulate the System Tree window

Where you are
> Manipulate the System Tree window
  Open an object
  Manipulate views
  Set up the toolbars

The Workspace page in the System Tree provides you with an overview of your work. By expanding the workspace and the objects it contains, you can see the content and structure of your target.

You can work directly with all the objects in the workspace. For example, you can edit, run, search, or regenerate a window using its pop-up menu in the System Tree. In this exercise you reposition, close, and open the System Tree. You can reposition the System Tree in relation to the main window using its drag bar. You can also change the way the System Tree, Clip, and Output windows are arranged.

1   Click the Output window button in the PowerBar to display the Output window.

**2   Select Tools>System Options from the menu bar.
Clear the Horizontal Dock Windows Dominate check box on the General page and click OK.**

If they did not already do so, the System Tree and Clip windows now occupy the full height of the main window on the left side.

**3   Click and hold the drag bar at the top of the System Tree.
Drag the System Tree to position it above, below, or to the right of the painter workspace.**

The painter workspace is the gray (blank) area, initially to the right of the System Tree, where painters display when you open an object.

When you start dragging the System Tree, a gray rectangular outline displays. It indicates the area that the System Tree would occupy if you released the mouse button.

4   When the gray rectangular outline is positioned where you want the System Tree to display, release the mouse button.

The System Tree displays in the new location.

5   **Close the System Tree by clicking the SysTree button in the PowerBar.**

The current workspace remains open, but the System Tree closes. Closing the System Tree leaves more space for the painter workspace views.

6   **Reopen the System Tree by clicking the SysTree button in the PowerBar again.**

7   **Select Tools>System Options from the menu bar. Select the Horizontal Dock Windows Dominate check box on the General page and click OK.**

You change back to the default selection for this design-time property.

8   **Close the Clip and Output windows by clicking their buttons on the PowerBar or by clicking the small *x* in the corner of each window.**

9   **Right-click the workspace at the top of your System Tree and select Close from the pop-up menu.**

The workspace closes. No workspaces display in the System Tree.

Open an object

Where you are
Manipulate the System Tree window
> Open an object
Manipulate views
Set up the toolbars

Now you open an object created by the Template Application wizard.

1 Select File>Recent Workspaces from the menu bar, then basic_tutorial from the cascading menu.

2 In the System Tree, expand the basic_tutorial workspace, the basic_tutorial target, and the basic_tutorial.pkl library.

3 In the basic_tutorial.pkl library, double-click the basic_tutorial Application object
or
Right-click the basic_tutorial Application object and select Edit from the pop-up menu.

The Application painter opens. It displays different views of the basic_tutorial Application object. Your view layout scheme might look different. To display the default layout, select View>Layouts>Default.
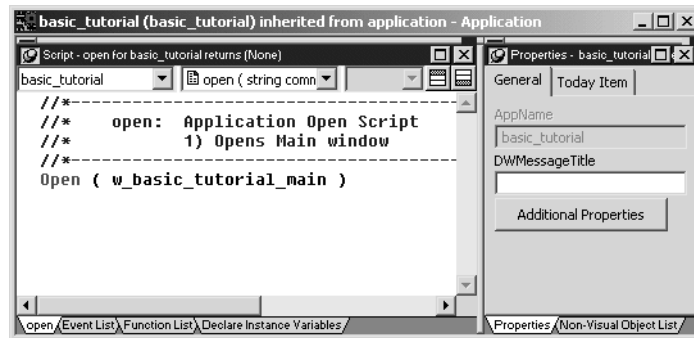
The default Application painter layout displays two stacks of tabbed panes. The left stack contains tabs for a Script view (open tab—it is set to the Open event on the Application object), an Event List view, a Function List view, and the Declare Instance Variables view. The right stack contains tabs for the Properties view and a Non-Visual Object List view.

4   Look at the code in the Open event in the Script view.

The PowerScript code that was generated by the wizard in the Application Object Open event calls a PowerScript function to open the main window in the application.

Manipulate views

Now you learn to control the location and appearance of PocketBuilder painter views. You can add views to a painter workspace by selecting them from the View menu in the workspace menu bar.

You can add multiple views of the same type and you can combine views into a stack of panes with selection tabs at the bottom. You can resize a view by grabbing and dragging the separator bars that surround it or that surround neighboring views in the painter workspace.

These exercises demonstrate how you can change the appearance of Application painter views, but you can manipulate views in all painters in the same way.

Now you:

- Add an extra Script view

- Display view title bars

- Float and dock views

- Manipulate tabbed views

- Save a view layout scheme

- Reset the default view layout scheme

Add an extra Script view

The default Application painter layout actually has two Script views. One of the Script views displays the script for an Application object event, and the other Script view, the Declare Instance Variables tab page, displays the declared variables for the object instance or the entire application. Both of these Script views are in the same stack of tabbed views (panes).

Now you add a third Script view that is not part of a stack of tabbed panes. You can add multiple Script views to your painter layout, but no two Script views can display the same script at the same time.

1   Select View>Script from the menu bar.

A new Script view displays. It is not attached to a stack of tabbed panes. It lists the Application object in the left drop-down list box. The other two drop-down lists are empty and the right drop-down list is grayed.
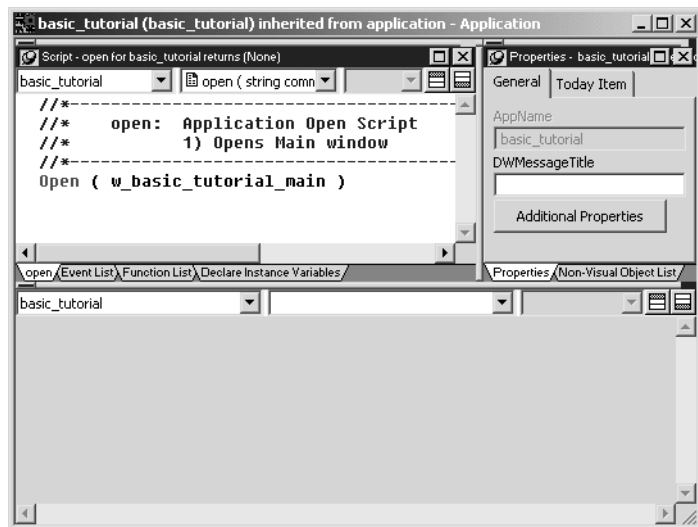
If an existing Script view shows the Open event, the new Script view is empty. Otherwise it displays the Open event.

2   Select the Close event from the second drop-down list box.

If another Script view is already open to the Close event, an error message displays in the PocketBuilder status bar.

Display view title bars

Now you display a view title bar by pinning it to the painter workspace background. If a title bar is unpinned, you see it only when your cursor pauses near the top edge of a view.

1   Move the cursor to the top edge of the extra Script view you just added.

The view title bar rolls down. It contains a pushpin button on the left and a maximize/minimize button and a close button on the right. The name of the view displays on the left side of the title bar, next to the pushpin button.

2   Click the pushpin in the title bar
or
Right-click the view title bar and click Pinned from the pop-up menu.

The pushpin button and the Pinned menu item are toggle switches. You can click the pushpin button or the pop-up menu item to pin and unpin the view title bars.

Float and dock views

Now you float and dock a view in the painter workspace. Floating a view enables you to move it around outside the painter frame.

1   Right-click the title bar of an unstacked view you want to float
or
Right-click the tab of a view in a stack of tabbed panes.

For example, you can right-click the application object's Function List tab.

If you want to right-click the title bar of an unstacked view and the title bar is not pinned, move the cursor over the title bar area and wait until it displays before you right-click it.

2   Click Float in the pop-up menu.

When a view is floated, the Float menu item is not enabled. When a view is docked, the Dock menu item is not enabled.

3   Drag the view around the screen.

Notice that the floating property allows you to move the view outside the painter workspace.

4   Right-click the title bar of the floating view.
Click Dock in the pop-up menu.

The view returns to its original location.

Manipulate tabbed views

Now you separate a view from a stack of tabbed panes and place it above the stack. You then return it to the stack and change its position in the stack.

1   Press and hold the mouse button on the Function List tab.
Drag the tab onto the vertical separator bar between the two default stacks in the Application painter.
Release the mouse button.

When you release the mouse button, the Function List view is no longer part of a stack. If you drag the tab too far and release it over the right stack with the Properties view and Non-Visual Object List, the Function List becomes part of that stack.

Alternate way to float a view from a stack
If you hold the Ctrl or Shift key down as you drag a tabbed pane from a stack, the pane becomes a floating view.

2   Press and hold the mouse button on the Function List title bar.
Drag it over the stack from which you separated it.
Release the mouse button when the gray rectangular outline of the Function List view overlaps the stack.

The Function List view returns to its original stack, but it is added as the last pane in the stack.

3   Press and hold the mouse button on the Function List tab.
Drag it sideways over the other tabs in the same stack.
Release the mouse button when the small gray rectangular outline overlaps another tab in the stack of tabbed panes.

The Function List view moves to the position in the stack where you release the mouse button.

Save a view layout scheme

You can save view layout schemes for a PocketBuilder painter and use them every time you open the painter.

1   Arrange the views in the painter as you like.

2   Select View>Layouts>Manage from the menu bar.

3   Click the New Layout button in the Layout dialog box.

4   Type a name for your layout in the text field, click the background of the dialog box, and then click the *x* button in the upper right corner of the dialog box to close it.

Your layout scheme is saved. Now, when you select View>Layouts, you see your layout listed on the cascading menu.

Saving the toolbars and System Tree layouts
PocketBuilder saves the customizations you make to the toolbars and System Tree separately from the view layout. It retains those settings and reapplies them to every workspace you access and every view layout you select.

Reset the default view layout scheme

Each PocketBuilder painter has a default view layout scheme. You can always reset the layout scheme to this default layout.

1   Select View>Layouts from the menu bar.

2   Choose Default from the cascading menu.

The default view layout scheme displays in the painter workspace.

Set up the toolbars

Where you are
Manipulate the System Tree window
Open an object
Manipulate views
> Set up the toolbars

A painter workspace always includes the PowerBar and other PainterBar toolbars that you can use as you work. The buttons in the toolbars change depending on the type of target or object you are working with. You can also customize the toolbars to include additional functionality.

Now you change the appearance of the toolbars to:

• Show labels on toolbar buttons

• Float the toolbars

• Reposition the toolbars

Show labels on toolbar buttons

You can learn a toolbar button's function by placing the cursor over it to view its PowerTip. A PowerTip is pop-up text that indicates a button's function.

You can also display a label on each toolbar button.

1   Move the pointer to any button on the PowerBar, but do not click.

The button's PowerTip displays.

2   Select Tools>Toolbars from the menu bar.

The Toolbars dialog box displays.

3   Select the Show Text check box, then click the Close button.

PocketBuilder displays a label on each of the buttons in the PowerBar and the PainterBars.

Float the toolbars

You can float the toolbars so that you can move them around the painter workspace as you work.

1   **Right-click anywhere in the PowerBar.**

The pop-up menu for the toolbars displays. From the pop-up menu you can set the toolbar's location to the left, top, right, or bottom of the workspace. You can also set it to floating.

About pop-up menus
Throughout PocketBuilder, pop-up menus provide a fast way to do things. The menu items available in the pop-up depend on the painter you are using and where you are in the workspace when you click the right mouse button.

---

2   **Select Floating from the pop-up menu.**

The PowerBar changes to a floating toolbar. You can adjust its shape.

3   Move the pointer to an edge or border area in the PowerBar.
Press and drag the PowerBar toward the left side of the workspace.
Release the mouse button when the PowerBar becomes a vertical bar.
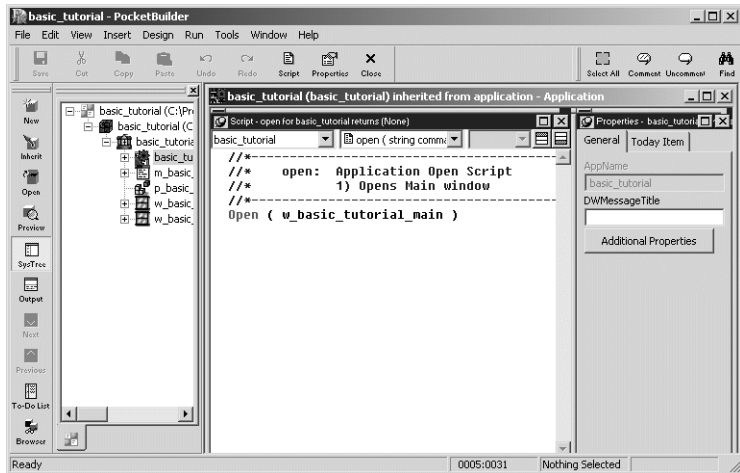
The PowerBar is docked at the left side of the frame.

Reposition the toolbars

You can customize the position of the toolbars to suit your work style.

1   Select Tools>Toolbars from the menu bar.

The Toolbars dialog box displays. The selected radio button in the Move group box indicates the position of the currently selected toolbar.
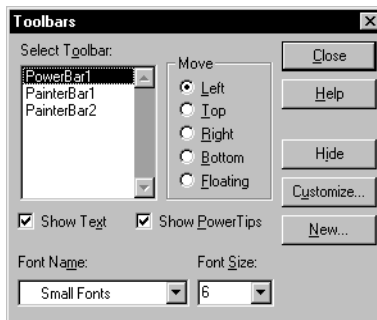
2   Click Top.

This repositions the PowerBar at the top of the workspace.

Radio buttons are grayed if a selected toolbar is hidden
If a selected toolbar is hidden (not visible) in the painter, you cannot select where it appears in the workspace. In this case, the radio buttons are grayed and you must first click the Show button before you can select a radio button. The Show button replaces the Hide button when a toolbar is hidden.

---

3   Click PainterBar1 in the Select Toolbar list box and select Right. Click Close in the Toolbars dialog box.

4   Right-click PainterBar2 and select Left from the pop-up menu.

You have swapped the locations of the two painter bars.

5   Arrange the toolbars to suit your preferences.

You can also drag the toolbars to the top, bottom, left, or right of the painter workspace. When a toolbar is in a fixed location, it has a drag bar at the left or top of its buttons. You can click the drag bar and drag the mouse to move the toolbar around the painter workspace.

PocketBuilder applies toolbar configuration properties to all painters and saves them for the next PocketBuilder session.

6   Close the Application painter.

L E S S O N 3   **Connecting to the Database**

This lesson shows you how to connect to a SQL Anywhere demonstration database and how to use the Database painter to look at the table definitions for this database. You will use this database connection in Lesson 4, "Creating an Employee List."

SQL Anywhere version
The databases in this tutorial are SQL Anywhere 10 databases. If you are using SQL Anywhere Studio 8.x or 9.x with Adaptive Server Anywhere databases, see the "Connecting to the Database" lesson in the previous version of this book that remains available on the Sybooks Web site at http://sybooks.sybase.com/nav/detail.do?docset=614#dt3.

Some Pocket PC 2002 devices can freeze while running SQL Anywhere 10 applications. For lessons using SQL Anywhere 10 databases, you should use Pocket PC 2003, Windows Mobile 5, or Windows Mobile 6 devices.

In this lesson you:

•   Define the tutorial data source

•   Create a database profile for the tutorial database

•   Look at table definitions in the tutorial database

How long does it take?
About 20 minutes.

About database data sources and the Database painter

In many organizations, database specialists maintain the database. If this is true in your organization, you might not need to create and maintain tables within the database. However, to take full advantage of PocketBuilder, you should know how to work with databases.

Defining a data source   Using the ODBC administrator or other database connection utilities, you can define a database as a data source for your application. You can access the ODBC Administrator from the Database Profiles dialog box. The definitions of ODBC data sources are stored in the *odbc.ini* registry key.

Using database profiles to connect   Once you define a data source, you can create a database profile for it in the PocketBuilder Database painter. A database profile is a named set of parameters that specifies a connection to a particular data source or database. Database profiles provide an easy way for you to manage database connections that you use frequently. When you are developing an application, you can change database profiles to connect to a different data source.

When database connections occur   PocketBuilder can establish a connection to the database in either the design-time or runtime environment. PocketBuilder connects to a database when you open certain painters, when you compile or save a PocketBuilder script that contains embedded SQL statements, or when you run a PocketBuilder application that accesses the database.

To maintain database definitions with PocketBuilder, you do most of your work using the Database painter. The Database painter allows you to:

- Create, alter, and drop tables

- Create, alter, and drop primary and foreign keys

- Create and drop indexes

- Define and modify extended attributes for columns

- Drop views

In this exercise you:

- Define the tutorial data source

- Create a database profile for the tutorial database

- Look at table definitions in the tutorial database

Define the tutorial data source

Where you are
> Define the tutorial data source
  Create a database profile for the tutorial database
  Look at table definitions in the tutorial database

In this exercise you create an ODBC data source for the *ASADemo_10.db* database that installs with PocketBuilder in the *Code Examples\SA10DemoData* subdirectory. This is a version of the *asademo.db* Adaptive Server Anywhere 9 demonstration database that has been updated for SQL Anywhere 10.

About the SQL Anywhere demonstration database
The SQL Anywhere 10 demonstration database, *demo.db*, is significantly altered from the *asademo.db* database. When you install SQL Anywhere 10, an ODBC data source is automatically created for the *demo.db* database, but this data source is not used in the current tutorial.

---

1   Click the Database Profile button in the PowerBar
    or
    Select Tools>Database Profile from the menu bar.

   PocketBuilder displays the Database Profiles dialog box, which includes a tree view of the installed database interfaces and defined database profiles for each interface. You can click the + signs or double-click the icons next to items in the tree view to expand or contract tree view nodes.
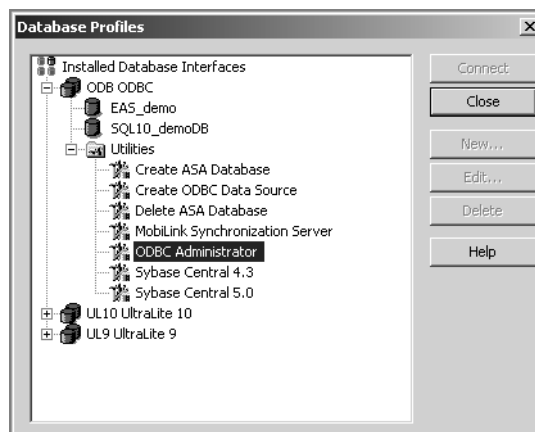
**2   Expand the ODB ODBC node, then expand the Utilities node.
Double-click on the ODBC Administrator item.**

The ODBC Administrator opens.

**3   Select the System DSN tab is selected and click the Add button.
Select SQL Anywhere 10 from the Create New Data Source dialog box
and click Finish.**

The ODBC Configuration for SQL Anywhere 10 dialog box displays.
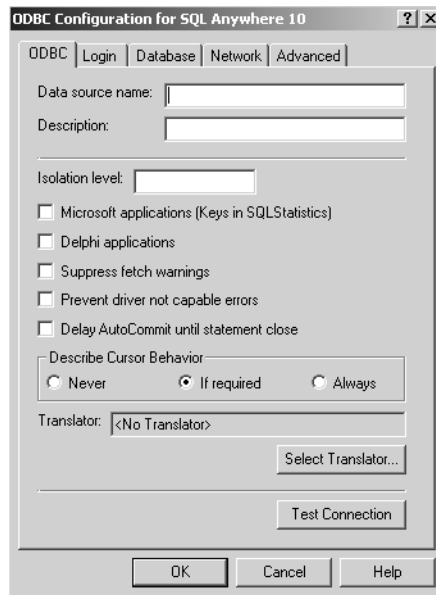
4   In the Data Source Name text box on the ODBC tab, type
DemoDB_SA10**.
On the Login tab, type** DBA **in the User ID text box and** sql **in the
Password text box.**

The "Supply user ID and password" radio button must be selected before
you can enter a user ID or password.

**5   On the Database tab, click the Browse button.
Browse to the** ASADemo_10.db **file in the PocketBuilder Code
Examples\SA10DemoData directory.**

6 **Type** ASADemo **for the Database Name,** ASADemo **for the Server Name,** dbeng10 -c 8M **for the Start Line.**

7 **Select the ODBC tab and click the Test Connection button.**

You should receive a success message.

8 **Click OK twice to close the message box, save the DemoDB_SA10 ODBC profile, and return to the Database Profile painter.**

Create a database profile for the tutorial database

Where you are
Define the tutorial data source
> Create a database profile for the tutorial database
Look at table definitions in the tutorial database

Now that you have defined *ASADemo_10.db* as an ODBC data source, you can create a profile for this data source in PocketBuilder.

If you closed the Database Profiles dialog box after the last exercise, open it again by clicking the Database Profiles button in the PocketBuilder toolbar.

**1   Right-click ODB ODBC in the Database Profile painter.
Click the New button.**

The Database Profile Setup dialog box displays.
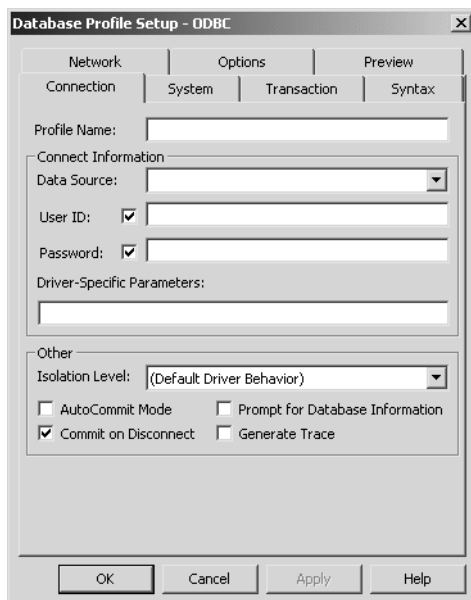
2   Type or select the required profile information.

Profile setup	Value
Profile Name	SA Demo
Data Source	DemoDB_SA10
User ID	DBA
Password	sql

3   Select the Preview tab.

The PowerScript connection syntax for the new profile is shown on the Preview tab. If you change the profile connection options, the syntax changes accordingly.
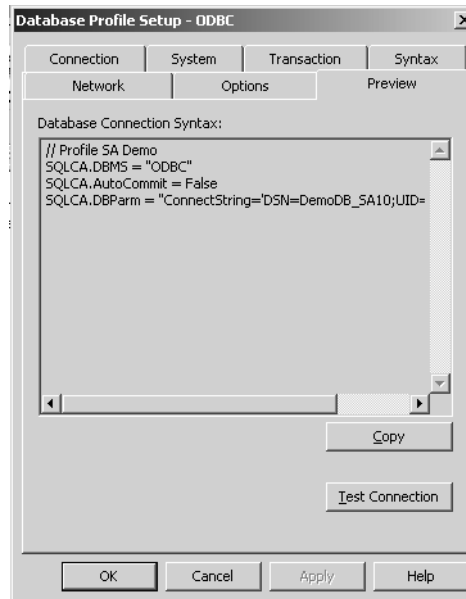
4   Click the Test Connection button.

A message box tells you that the connection is successful.

5   Click OK to close the message box.
Click OK to close the Database Profile Setup dialog box.

You return to the Database Profiles painter.

6   Select SA Demo and click the Connect button.
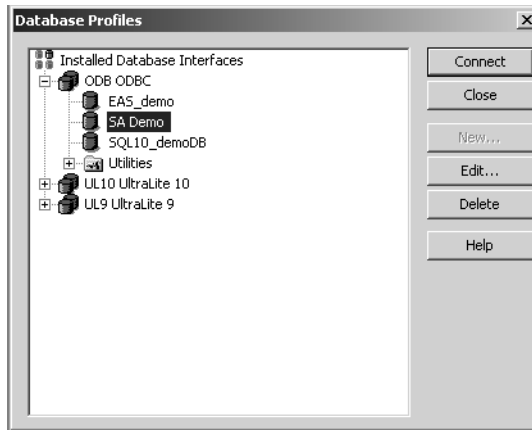
PocketBuilder connects to the database and the Database Profile painter closes.

What happens when you connect When you connect to a database in the development environment, PocketBuilder writes the connection parameters to the Windows registry. Each time you connect to a different database, PocketBuilder overwrites the existing parameters in the registry with those for the new database connection. When you open a PocketBuilder painter that accesses a database, you automatically connect to the last database used. PocketBuilder determines which database this is by reading the registry.

Look at table definitions in the tutorial database

***

Where you are
Define the tutorial data source
Create a database profile for the tutorial database
> Look at table definitions in the tutorial database

***

Now that you are connected to the tutorial database, you can look at the definitions for the tables in that database. This exercise helps you become familiar with the Database painter and the tables you will use in the next tutorial lesson.

1 Click the Database button in the PowerBar.

PocketBuilder connects to the database and the Database painter opens. The Database painter title bar identifies the active database profile. There is also a green check mark in the icon for the active database profile in the Objects view of the painter.

The Objects view of the Database painter displays all existing database profiles in a tree view under the Installed Database Interfaces heading. The SA Demo database profile is visible under the ODB ODBC node in the tree view.

***

If the Objects view is not open
The Objects view is part of the default view layout scheme. To reset to this scheme, select View>Layouts>Default. You can also open an Objects view by selecting View>Objects from the menu bar.

***

2 Expand the SA Demo database node in the Objects view.

Notice the folders under the SA Demo database node.
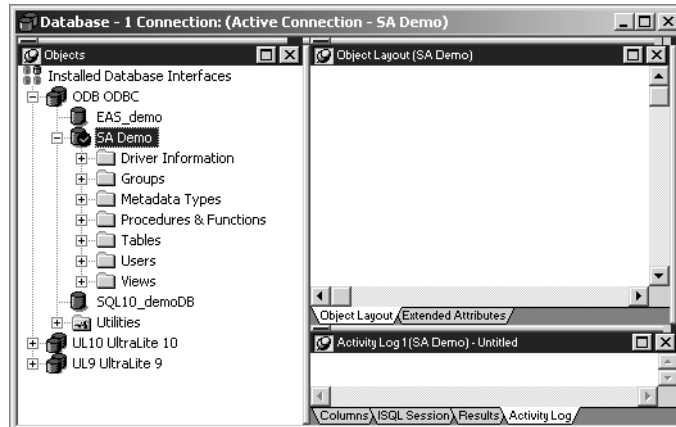
3 Expand the Tables folder.

You see the list of tables in the database.

4 Right-click the customer table and select Add To Layout from the pop-up menu
or
Drag the customer table from the Objects view to the Object Layout view.

Dragging an object from one view to another
When you start dragging an object from the Objects view to another view, the pointer changes to a barred circle. If you continue moving the cursor to a view that can accept the object, the barred circle changes back to a pointer with an additional arrow symbol in a small box. When you see this symbol, you can release the object.

---

5   Repeat step 4 for the product table.

Widening the Object Layout view
You can widen the Object Layout view by dragging its separator bars toward the painter frame. If the Object Layout view is part of a stack, you might find it easier to separate it from the stack before you change its size.

The Object Layout view shows the two tables you selected.
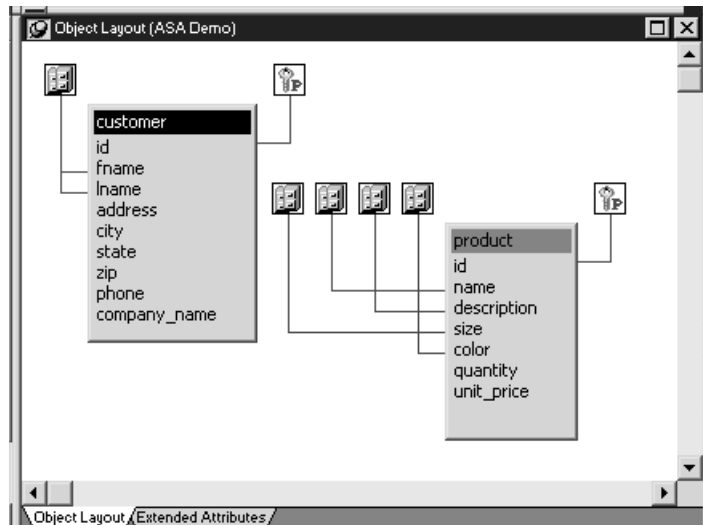
Viewing table data types, comments, keys, and indexes
In the Object Layout view, you can see a description for each column, as well as icons for keys and indexes. If you do not see this, right-click a blank area inside the view and select Show Referential Integrity and Show Index Keys from the pop-up menu. If you select Show Datatypes, you also see the datatype for each column in the selected tables.

---

6   Right-click the title bar of the customer table in the Object Layout view and select Alter Table from the pop-up menu
or
Right-click the customer table in the Objects tree view and select Alter Table from the pop-up menu.

The Columns view displays the column definitions for the table.

**7   Right-click a column in the customer table in the Object Layout view.
      Select Properties from the pop-up menu.**

The Properties view for a column has five tabs, one for general database
properties and the others for column extended attributes.
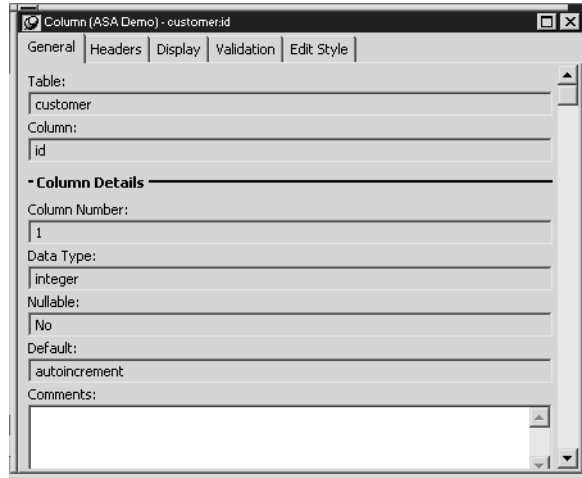
About extended attributes
PocketBuilder stores extended attribute information in system tables of the
database. Extended attributes include headers and labels for columns,
initial values for columns, validation rules, and display formats.

You can define new extended attributes or change the definitions of
existing extended attributes from the pop-up menus of items in the
Extended Attributes view of the Database painter.

8   Close the Database painter.

LESSON 4 — Creating an Employee List

The DataWindow object is one of the most powerful features of PocketBuilder. A DataWindow provides data access and programming capabilities. A DataWindow can connect to a database, retrieve rows, display the rows in various presentation styles, and update the database. In this lesson you create an Employee list using a DataWindow.

You must complete Lesson 3 to define and connect to the data source that you use in this lesson.

SQL Anywhere version
The databases in this tutorial are SQL Anywhere 10 databases. If you are using SQL Anywhere Studio 8.x or 9.x with Adaptive Server Anywhere (ASA) databases, see the "Creating an Employee List" lesson in the previous version of this book that remains available on the Sybooks Web site at http://sybooks.sybase.com/nav/detail.do?docset=614#dt3.

When you install SQL Anywhere 10 to your device, make sure you select the International Components for Unicode (ICU) check box. If you did not install the ICU components, reinstall SQL Anywhere to the device with this check box selected before you begin this lesson.

In this lesson you:

*   Create a workspace and target

*   Create and preview a new DataWindow object

*   Attach the DataWindow object to a DataWindow control

*   Modify window properties and add a retrieve call

*   Run the application on the desktop

*   Copy the DSN file and database to the Pocket PC

*   Build, deploy, and run the application on the device

How long does it take?
About 45 minutes.

Create a workspace and target

Where you are

> Create a workspace and target
> Create and preview a new DataWindow object
> Attach the DataWindow object to a DataWindow control
> Modify window properties and add a retrieve call
> Run the application on the desktop
> Copy the DSN file and database to the Pocket PC
> Build, deploy, and run the application on the device

1   Click the New button in the PowerBar or select File>New from the menu bar.

The New dialog box displays.

2   In the Workspace tab, select Workspace and click OK.
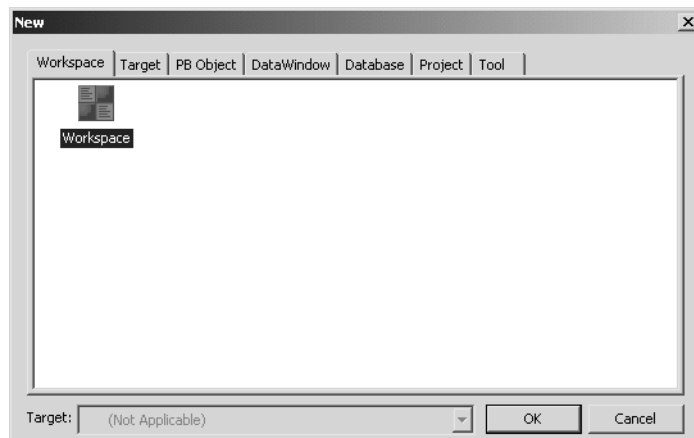
3   Navigate to the \PocketBuilder 2.0\Tutorial\EmployeeList directory.

4   Type emplist_tutorial in the File name text box.

5   Save the new workspace as emplist_tutorial in the \PocketBuilder 2.0 \Tutorial\EmployeeList directory.

6   Select File>New from the menu bar and click the Target tab
or
Right-click emplist_tutorial in the System Tree, select New from the pop-up menu, and click the Target tab.

The Target page of the New dialog box displays.

7   Select the PocketPC Application Creation Wizard icon and click OK.
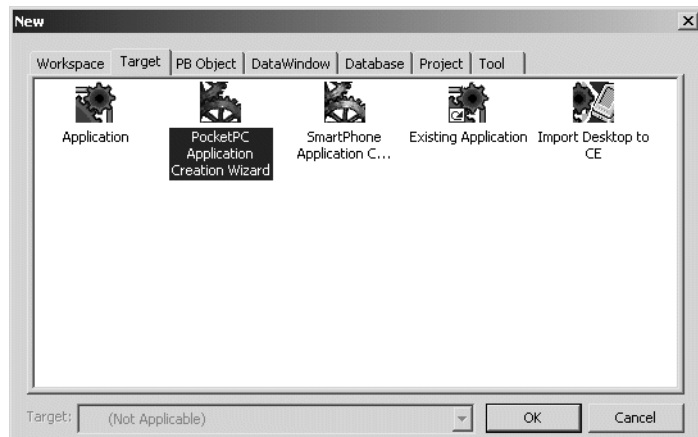
The PocketPC Template Application wizard displays. In most wizards, the first page explains what the wizard is used for. As you step through the wizard, you can press F1 to get Help on most fields.

8   Click Next until the Specify New Application and Library page displays.

9   Type emplist_tutorial **in the Application Name text box.**

When you click Next or place the cursor in the Library or Target text box, the wizard automatically assigns file names to the library and target that use this application name. It assigns the library a PKL extension and the target a PKT extension.
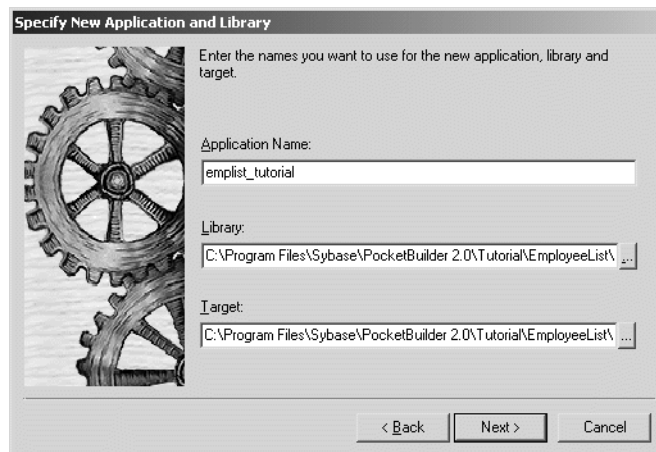
10   Click Next until the Specify Connectivity page displays.

You accept the default library search path and window and menu names. This lesson requires database connectivity.

11   Select the Requires SQL Database connection check box.
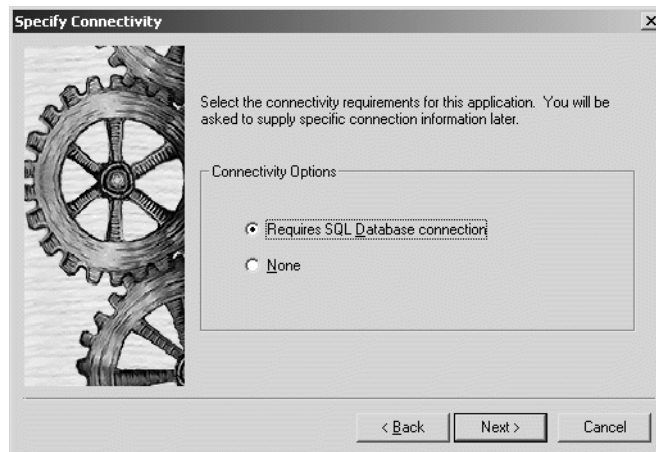
12   Click Next.
Select the SA Demo database profile from the Profile Name list if it is
not already selected.
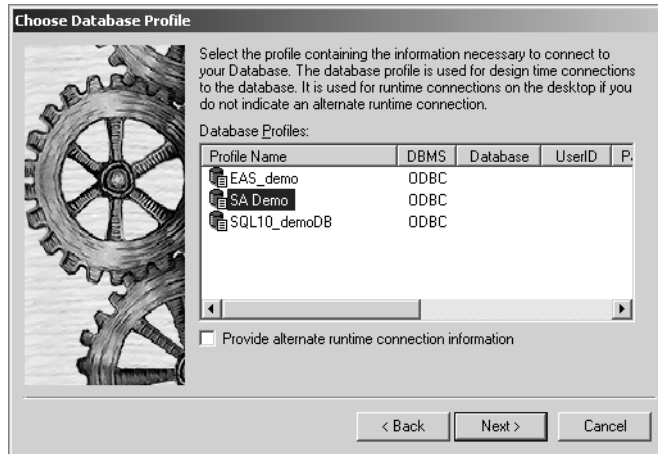
13   Click Next.
Select the Script radio button to specify that the application get its
connection information from script.
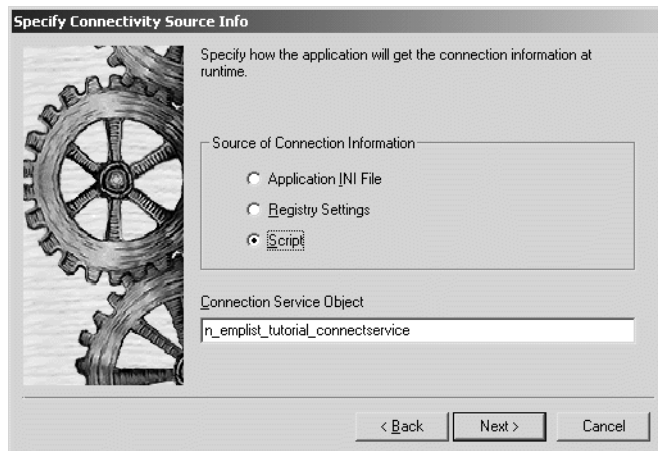
14   Click Next.

The Specify Project Object page displays.

The wizard will create a project object that opens in the Project painter. The Project painter allows you to streamline the generation of files that your target needs, create an executable for deployment, and rebuild your application easily when you make changes to the application.

15   Click Next until the Specify Deployment Configuration page displays.

You accept the default names for the project name and the executable file name, and you do not choose to build dynamic libraries.

The Specify Deployment Configuration page displays fields for a target directory and a build directory. The Target directory (with a default of *\Program Files*) is the directory on the device or emulator where you want to deploy your application. The Build directory is the directory where you want to build your application. The default value is a single dot (".") representing the directory that contains the main application library.
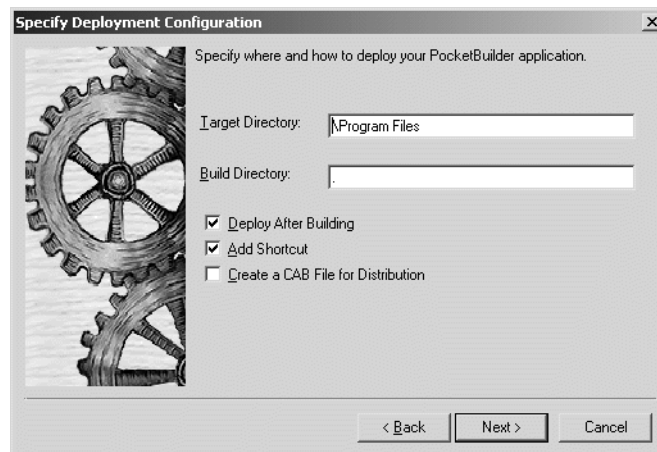
16   Click Next to accept the default names for the target and build directories.

The Specify Version Information page of the wizard displays.

17 Click Next to accept the default version information.

You accept the default deployment configuration and version information. The last wizard page lists your current selections so that you can review them or use the Back button to go back and change them if necessary.

18 Click Finish.

The PocketPC Application Creation Wizard creates the *emplist_tutorial.pkt* target and the *emplist_tutorial.pkl* library, and sets the new emplist_tutorial application as the default application.

You can expand the System Tree to view all the objects that have been created by the PocketPC Application Creation Wizard. The System Tree does not display the file extension of the emplist_tutorial target, but it does display the directory where the target file is saved.

The *emplist_tutorial.pkl* library displays under the basic_tutorial target in the System Tree. It contains the target Application object, which has the same name as the target object but displays under the library file.

Other objects generated by the wizard also display under the library file. One of these is the main window w_emplist_tutorial_main.

Next you create a DataWindow object for your application.

Create and preview a new DataWindow object

Where you are
  Create a workspace and target
> Create and preview a new DataWindow object
  Attach the DataWindow object to a DataWindow control
  Modify window properties and add a retrieve call
  Run the application on the desktop
  Copy the DSN file and database to the Pocket PC
  Build, deploy, and run the application on the device

Now you create a new DataWindow object and display it in the DataWindow painter. Like other painters, the DataWindow painter has an assortment of views that you can open simultaneously.

About the Design view in the DataWindow painter
The Design view in the DataWindow painter is similar to the Layout view in other painters. You can open only one Design view at a time.

The Design view is divided into four areas called bands: header, detail, summary, and footer. You can modify the contents of these bands. For example, you can change their sizes, add objects (controls, text, lines, boxes, or ovals), and change colors and fonts.

In the Preview view of the DataWindow painter, you can see how the object looks in an application at runtime, complete with table data.

1 Select File>New.
In the DataWindow tab, select Grid from the list of presentation styles and click OK.
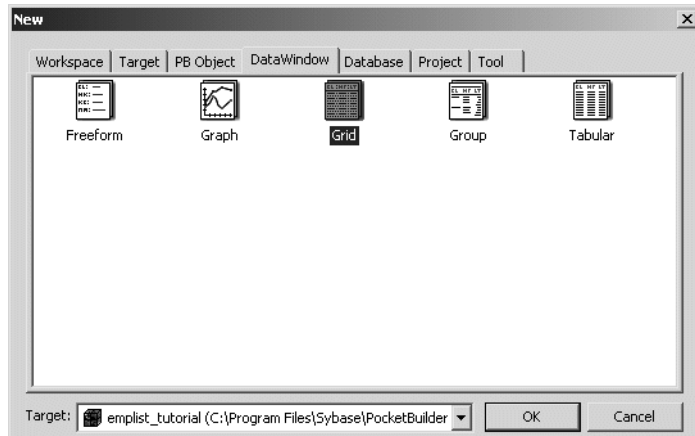
The Choose Data Source for Grid DataWindow page of the DataWindow wizard displays.

2 Choose SQL Select as the data source.
Be sure the Retrieve on Preview check box is selected.
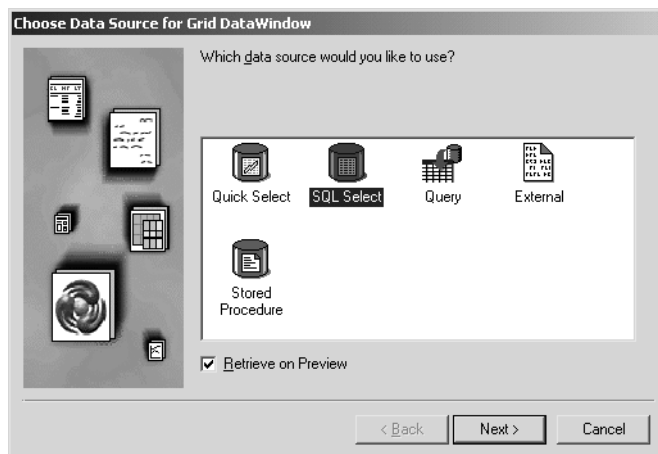
3   Click Next.

The Select painter displays the tables in the database.
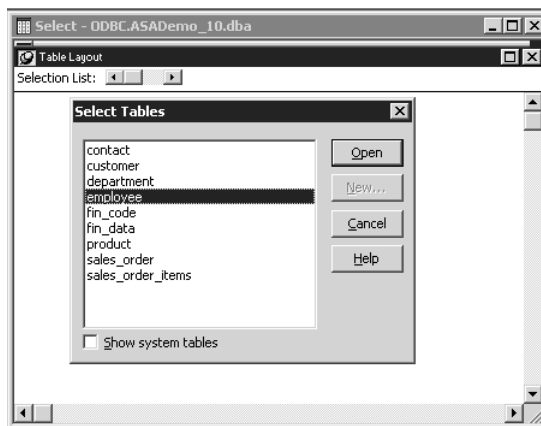
4   Select the employee table and click Open.

The Select painter displays the employee table. In the Select painter, you select table columns for the DataWindow.

5   Select the emp_fname column first, then the emp_lname column.

If you selected emp_lname first, click the column again to clear the selection, then select the columns in the order described for this tutorial.

6   Click the Return button in the painter menu or select File>Return to DataWindow Painter from the menu bar.

The Select Color and Border Settings page displays.

The DataWindow wizard asks you to select the colors and borders for the new DataWindow object. By default, there are no borders for text or for columns.

7   Click Next.

You accept the border and color defaults. The DataWindow wizard
summarizes your selections.
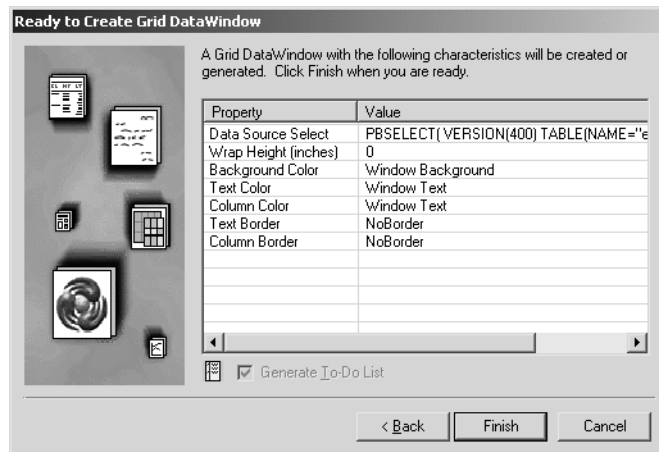
8   Click Finish.

PocketBuilder creates the new DataWindow object and opens the
DataWindow painter. In the Design view, PocketBuilder displays a Header
band with default headings and a Detail band with the columns you
selected.

**9   In the Design view, select the column borders (the light gray vertical
    lines) one at a time and drag them to adjust the column widths of the
    DataWindow so they are small enough to fit on a Pocket PC screen.**

When you are done, the columns should look something like this:
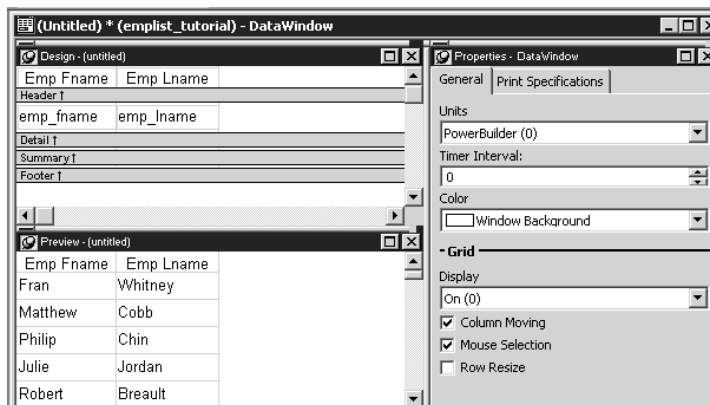
The Preview view below the Design view displays the DataWindow as it
appears during execution. PocketBuilder displays data for all customers.

Changing the layout of the DataWindow painter
If the Preview view is not displayed, select View>Preview from the menu
bar. If Preview is grayed, it is already displayed and you cannot select it.
You can open only one Preview view at a time. If your DataWindow
painter layout does not look at all like this, you can change your layout to
the default layout by selecting View>Layouts>Dafault from the menu bar.

**10 In the Design view, click the Emp Fname header.
    In the Properties view, change the Text property to First Name.
    Change the Alignment property to Left (0).
    Click the Font tab and select Bold.**

**11 In the Design view, click the Emp Lname header.
    In the Properties view, change the Text property to Last Name.
    Change the Alignment property to Left (0).
    Click the Font tab and select Bold.**

12 Select File>Save from the menu bar.

The Save DataWindow dialog box displays with the insertion point in the
DataWindows box.

13 Type `d_sqlgr_employee` **in the DataWindows text box.**

This names the DataWindow object. The prefix d_ is standard for
DataWindow objects.

14 (Optional) Type the following comments in the Comments box:

```
This DataWindow object retrieves employee first
names and last names.
```
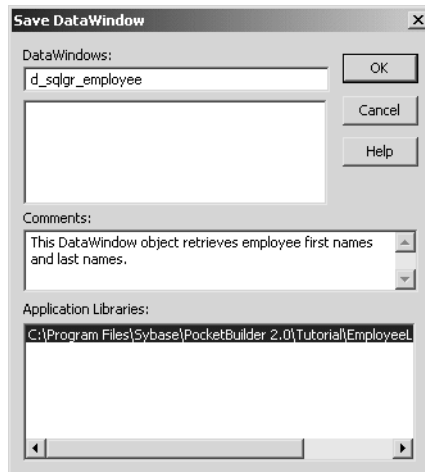

15 Click OK.

PocketBuilder saves the DataWindow object and closes the Save
DataWindow dialog box.

Attach the DataWindow object to a DataWindow control

Where you are
Create a workspace and target
Create and preview a new DataWindow object
> Attach the DataWindow object to a DataWindow control
Modify window properties and add a retrieve call
Run the application on the desktop
Copy the DSN file and database to the Pocket PC
Build, deploy, and run the application on the device

Now you attach the DataWindow object to a DataWindow control in the w_emplist_tutorial_main window.

1   **Expand the emplist_tutorial.pkl branch in the System Tree.**
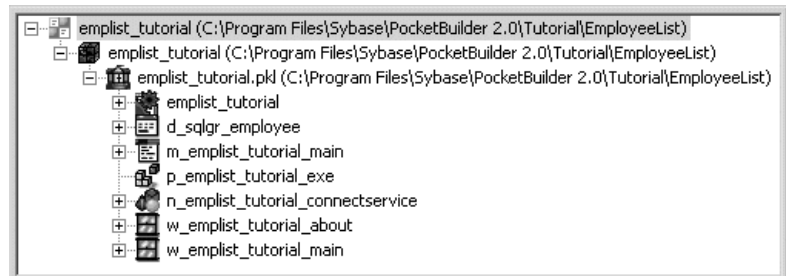
2   **Right-click w_emplist_tutorial_main and select Edit from the pop-up menu**
     or
     Double-click w_emplist_tutorial_main in the System Tree.

The Window painter displays the w_emplist_tutorial_main window.

**3   Make sure the Layout view displays.
    Select Insert>Control>DataWindow from the menu bar and click
    inside the main window in the Layout view.**

An empty DataWindow control displays in the window.
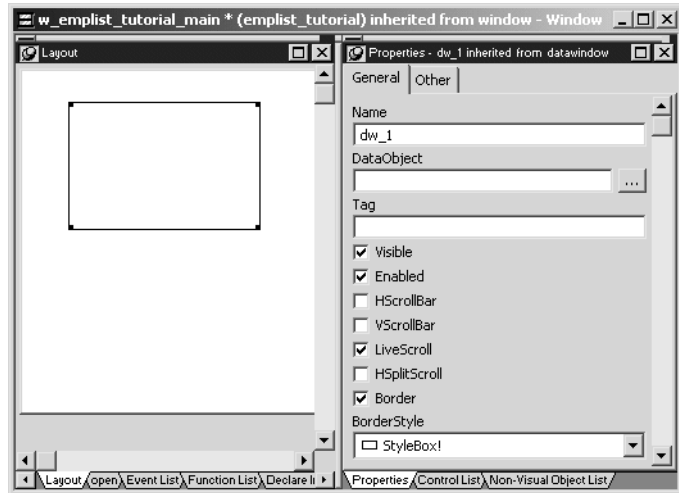
**4   In the Properties view for the DataWindow, select HScrollBar and
    VScrollBar.**

The scroll bars enable users to scroll through the list of employees when
they run the application.

Now you need to associate the d_sqlgr_employee DataWindow object that
you created in the last exercise with the DataWindow control that you just
added to the window.

5   In the Properties view, click the browse button next to the DataObject text box and select the d_sqlgr_employee DataWindow.
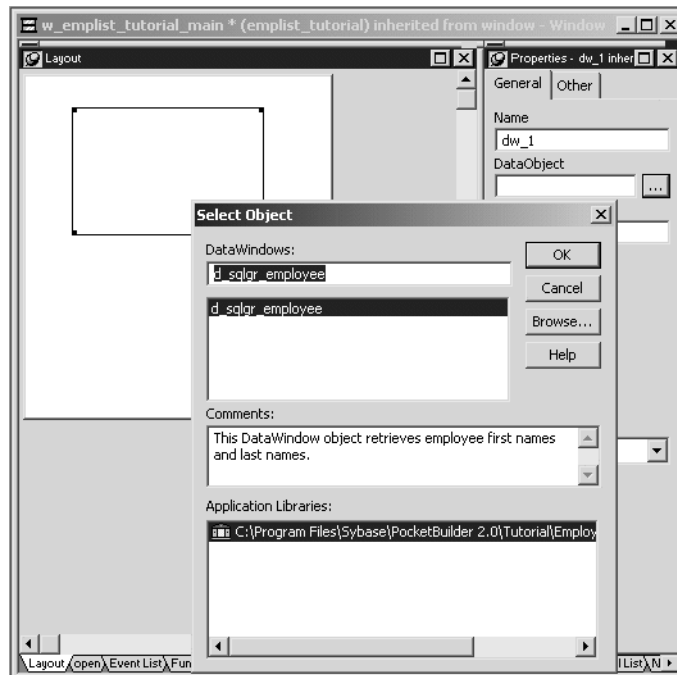
6   Click OK.

PocketBuilder associates the d_sqlgr_employee DataWindow object with the DataWindow control.

The Layout view now shows the d_sqlgr_employee DataWindow headings inside the DataWindow control, but you do not see any data yet. The DataWindow does not execute its SELECT statement until you run the application.

7   Select the DataWindow control and then and drag its borders so the control takes up most of the space in the main window.

When you are done, the window should look something like this:
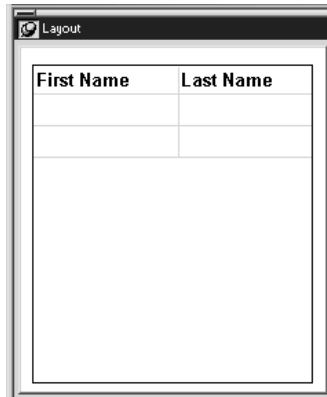
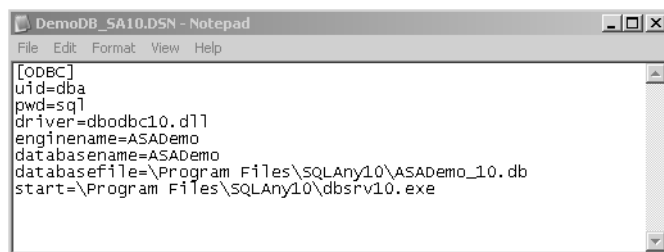# Modify window properties and add a retrieve call

Where you are
Create a workspace and target
Create and preview a new DataWindow object
Attach the DataWindow object to a DataWindow control
> Modify window properties and add a retrieve call
Run the application on the desktop
Copy the DSN file and database to the Pocket PC
Build, deploy, and run the application on the device

Now you modify the w_emplist_tutorial_main window properties to make the window easier to use.

1   Click in the main window, but outside of the DataWindow control. In the Properties view, change the title from Main Window to Employees.

2   Select the Close (OK) check box, if it is not already selected. Select the Show SIP Button check box.

The Visible and Enabled window properties are selected by default. You select the Close (OK) property to give users a way to close the window. You select the Show SIP Button property so the Soft Input Panel (SIP) on the Pocket PC will be available to the user.
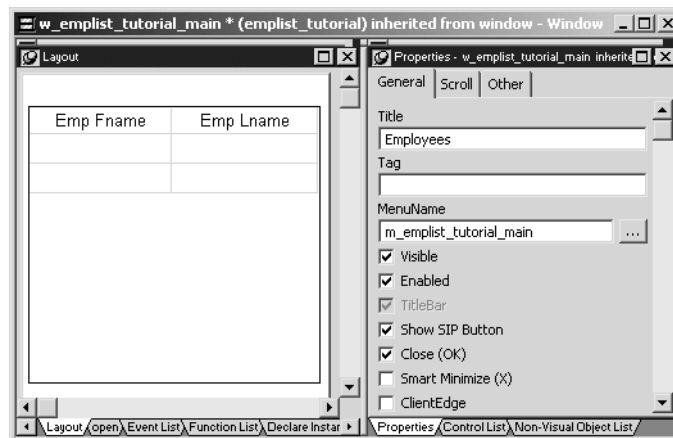
**3   In the main window's Properties view, click the Scroll tab page.
Select the HScrollBar check box and the VScrollBar chec kbox.**

You add scrollbars to the window to give users a way to scroll to see data that is not visible in the window.
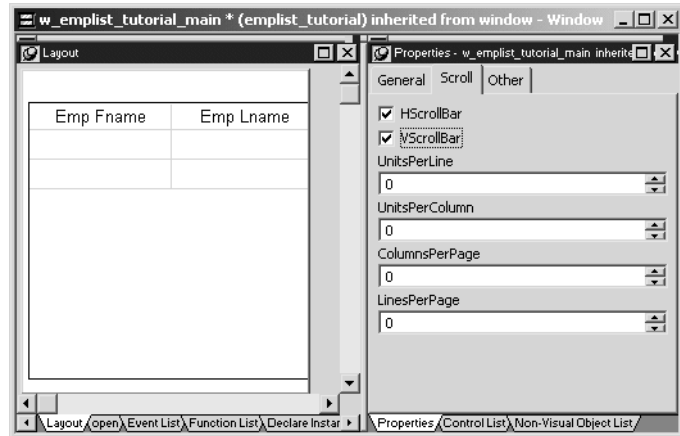
Next you code the open event for the window.

4   Select the "open" tab from the main stack of views.

The label for the Script view tab shows the name of a default event. Typically the default event is an event that is already scripted. In this case, code has been added by the wizard to the window open and ue_postopen events when you generated the application, but "open" is the first event script that is run when the window is opened.

Note that the only uncommented line in the open event calls the ue_postopen event.

**5   Select ue_postopen from the second drop-down list in the Script view.
Uncomment the last two lines in the ue_postopen script.**

You can uncomment the lines by removing the slash marks at the beginning of the lines.

The code for the SetTransObject for DataWindow dw_1 associates the default SQLCA transaction object that is connected to the database with the DataWindow control.

The Retrieve function call retrieves data from the database and populates the DataWindow object associated with the DataWindow control:

```
dw_1.SetTransObject(SQLCA)
dw_1.Retrieve()
```

You do not need to make any further changes to the ue_postopen script, since the DataWindow control that you added to the window is labeled dw_1 by default.
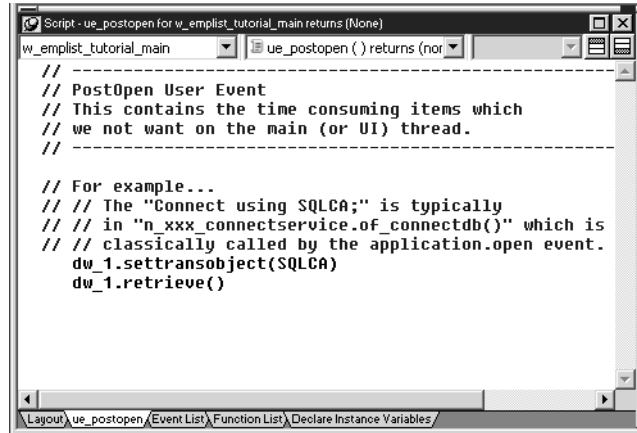
6 **Click the Save button in the PainterBar or select File>Save from the menu bar.**

Run the application on the desktop

Where you are
Create a workspace and target
Create and preview a new DataWindow object
Attach the DataWindow object to a DataWindow control
Modify window properties and add a retrieve call
> Run the application on the desktop
Copy the DSN file and database to the Pocket PC
Build, deploy, and run the application on the device

Now you run the application again to test the capabilities of the DataWindow.

1   Click the Run button (the running person icon) in the PowerBar or select Run>Run from the menu bar.

If you did not already save your work, PocketBuilder prompts you to save your changes. If you see this prompt, click Yes.

The application begins running, connects to the database, and displays data retrieved from the database in the DataWindow of the main application window.

**2   Close the application.
Close the Window painter.**

# Copy the DSN file and database to the Pocket PC

Where you are
Create a workspace and target
Create and preview a new DataWindow object
Attach the DataWindow object to a DataWindow control
Modify window properties and add a retrieve call
Run the application on the desktop
> Copy the DSN file and database to the Pocket PC
Build, deploy, and run the application on the device

A data source name (DSN) file is a data structure that contains the information about a specific database that an ODBC driver needs in order to connect to it. Included in the DSN, which resides either in the registry or as a separate text file, is information such as the name, directory, and driver of the database, and, optionally, the ID and password of the user.

For the this tutorial, you rely on a registry DSN for the design-time database connection, but a file DSN on the handheld device. The ODBC profile name for the registry DSN has the same name as the file DSN that you copy to the Pocket PC, so you do not need to change the of_getconnectioninfo function script of the n_emplist_tutorial_connectservice user object that you generated with the wizard.

**1   Open the DemoDB_SA10.DSN in a text editor.
Examine the contents of this DSN file.**

The DSN file is installed in the *Code Examples\SA10 Demo Data* subdirectory by the PocketBuilder setup program. It references the default path to SQL Anywhere 10 on a Pocket PC device. If you installed SQL Anywhere to a nondefault location, you must change at least the start line in this DSN file. For example, if you installed SQL Anywhere on the storage card, you might need to add *\Storage* before the default path to the database file and the database start line.

ODBC driver setting
By default, PocketBuilder assumes the driver for a remote database is
dbodbc9.dll. If you are using SQL Anywhere 10, you must set the driver
to *dbodbc10.dll*, either in the DSN file or in the database connection string.
The DSN file in the *Code Examples\SA10 Demo Data* subdirectory sets
the ODBC driver for SQL Anywhere 10.

Next, you copy the DSN to the root directory of the Pocket PC device and
the *ASADemo_10.db* database and its log file, *ASADemo_10.log*, to the
\Program Files\SQLAny10 directory of the device.

**2   If your Pocket PC is not on, turn it on, and connect to it from the
desktop using ActiveSync.**

ActiveSync displays on the desktop and connects to your Pocket PC.
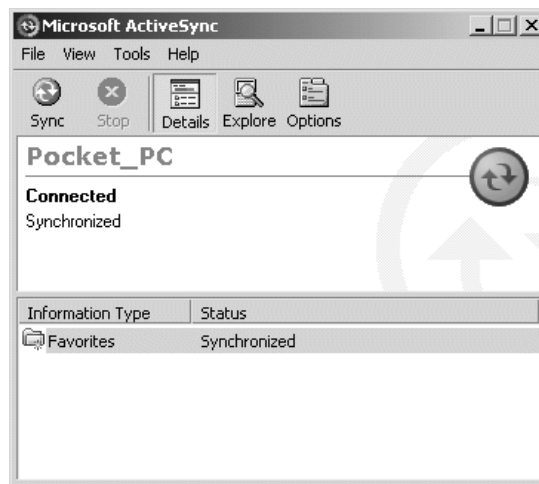
3   In ActiveSync, click Tools>Explore Device.

A view of files in the Pocket PC displays.
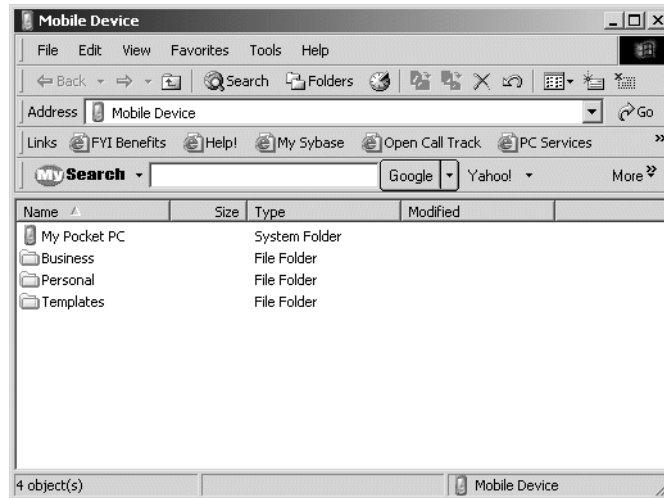
4   Double-click My Pocket PC (or My Windows Mobile-Based Device).

This brings you to the root directory (\ ).

5 Using the Windows Explorer on the desktop, copy the DemoDB_SA10.DSN file to the Pocket PC root directory.

To copy this file, you can drag it from the PocketBuilder *Code Examples\SA10 Demo Data* directory in Windows Explorer and drop it in the Pocket PC root directory displaying in the Explorer window that you opened from ActiveSync, or you can use the Edit>Copy and Edit>Paste menu items in the same or separate Explorer windows.
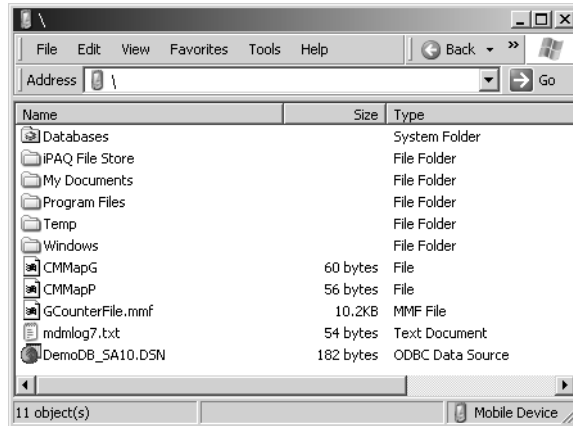
6 In the Explorer window that you opened from ActiveSync, change directories to the main SQL Anywhere directory on the Pocket PC.

The default directory for SQL Anywhere on a Pocket PC device is */Program Files/SQLAny10*.

7 Using Windows Explorer, copy `ASADemo_10.db` **from the PocketBuilder Code Examples\SA10 Demo Data directory on the desktop to the \Program Files\SQLAny10 directory of the device.**
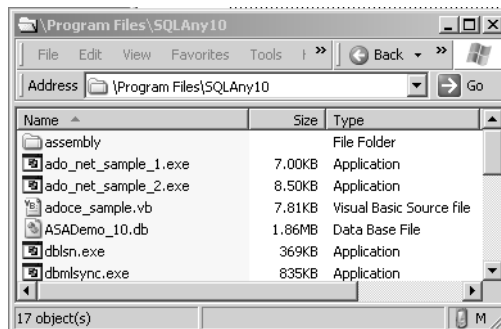
You might need to close PocketBuilder, or to disconnect from the database in PocketBuilder. before you can copy the database file from the desktop to the mobile device.

8 Repeat the previous step for the log file, `ASADemo_10.log`**.**

Now you are ready to build and deploy the application.

Build, deploy, and run the application on the device

Where you are
Create a workspace and target
Create and preview a new DataWindow object
Attach the DataWindow object to a DataWindow control
Modify window properties and add a retrieve call
Run the application on the desktop
Copy the DSN file and database to the Pocket PC
> Build, deploy, and run the application on the device

Now you can build the application, deploy it to a device or emulator, and run it on the device or emulator.

1   Double-click the p_emplist_tutorial_exe project object in the System Tree.

The Project painter opens.

The build platform is PocketPC Device (ARM) specified by the wizard.
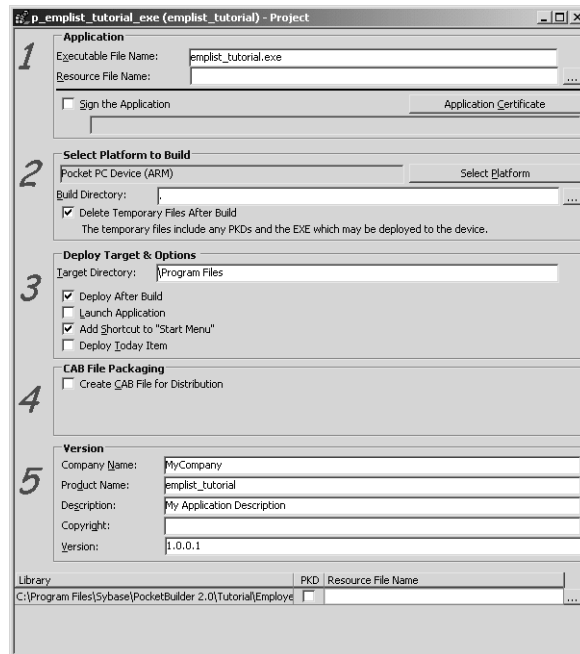
2   Select the Add Shortcut to "Start Menu" check box if it is not already selected.

When you deploy the application, this setting adds the application name to Start Menu list.

Devices can restrict the number of menu items in their Start menus, so this does not guarantee that the application name will be visible in the Start menu. If there is already a maximum number of menu items in the list when you deploy the application, you can open Settings>Menu from the Start menu and clear the check box for one of the menu items you do not need to display, then select the check box for the deployed application or redeploy it with the same Add Shortcut setting.

3   Select Run>Build and Deploy Workspace or click the Deploy button on the PainterBar.

The project is built and deployed and the application is copied to the device's \Program Files directory.

Next you run the application on the Pocket PC device.

4   On the Pocket PC device, tap the Start menu.

5   Tap emplist_tutorial, or if it does not display in the Start menu, tap Programs and then tap the emplist_tutorial application icon.

Another way to run the application
You can also tap the Start menu, then tap the PocketBuilder 2.1 menu, and then tap the application you want to start, in this case emplist_tutorial, from the PocketBuilder application list.

The application starts on the Pocket PC device, first displaying the SQL Anywhere log screen, and then the main application window.
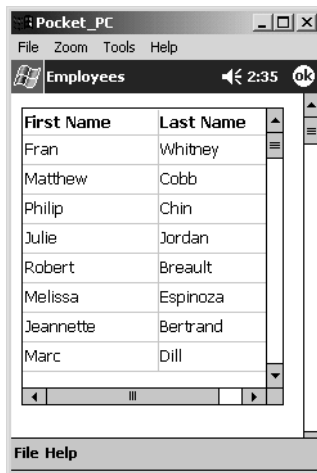
6   Scroll through the employee list using the DataWindow scroll bars.

7   Tap the circular *ok* button to exit the application.

The Employees application closes and you return to the Programs menu.

L E S S O N   5   Creating a Sales Application

In this lesson, you complete a skeleton sales application that is provided for you. The application uses MobiLink synchronization to synchronize the data in Pocket PC and server databases.

When you install SQL Anywhere 10 to your device, make sure the MobiLink Synchronization and International Components for Unicode (ICU) check boxes are selected. If you did not install these components, reinstall SQL Anywhere to the device before you begin this lesson.

In this lesson, you:

- Set up the SalesDB databases

- Begin modifying the SalesDB application

- Create a DataWindow for sales order information

- Add menu items to the application menu

- Create a MobiLink connection

- Create the main application window

- Test the application on the desktop

- Deploy the application to a device

- Run the application

- Troubleshoot the application

How long does it take?
About 60 minutes.

Set up the SalesDB databases

Where you are
> Set up the SalesDB databases
  Begin modifying the SalesDB application
  Create a DataWindow for sales order information
  Add menu items to the application menu
  Create a MobiLink connection
  Create the main application window
  Test the application on the desktop
  Deploy the application to a device
  Run the application
  Troubleshoot the application

The SalesDB application uses one consolidated database on the PC to store all data. A remote database is placed on the Pocket PC device and data is transferred to the consolidated database using MobiLink synchronization. In this lesson, you set up the remote and consolidated databases.

1   Locate the PocketBuilder 2.0\Tutorial\SalesDB\db directory.

2   Double-click MakeDB10.cmd.
When the command file finishes running, press any key to continue.

This creates and populates the remote and consolidated databases. The corresponding Data Source Name (DSN) entries are also be created.

SQL Anywhere 10
The *MakeDB10.cmd* script creates SQL Anywhere 10 database only. If you are using ASA 9 or ASA 8, you must run the *MakeDB.cmd* script instead, and you should use the "Creating a Sales Application" lesson in the previous version of this book. (That book is available on the Sybooks Web site at http://sybooks.sybase.com/nav/detail.do?docset=614#dt3.)

If you just installed SQL Anywhere and the *MakeDB10.cmd* script does not run, you might need to reboot your machine. The script requires the proper SQL Anywhere environment variables to be set.

Now you verify that the databases are created correctly.

3   From the Start Menu, select Programs>SQL Anywhere 10>Sybase Central.
Close the tips window and the Welcome to Sybase Central window if either or both of these windows display.

4   Display the Folders view in the left pane of Sybase Central, then right-click SQL Anywhere 10 in this view, and click Connect in the pop-up menu.

You display the Folders view by selecting View>Folders from the Sybase Central menu. After you click Connect, the Connect dialog box displays.

Using the Tasks view
You can also diplay this dialog box from the Tasks view of Sybase Central by clicking the "View and edit schema or perform maintenance on a database" item.

---

5   Select the ODBC Data Source name radio button.
Click Browse and select *SalesDB*, and then click OK.
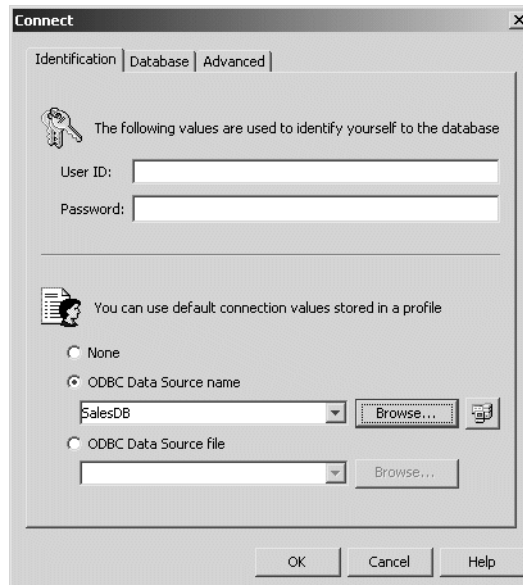
6 **Verify that the SalesDB tables are created and sample data is provided by expanding the Tables node in the Folders view, selecting a table, and, in the right pane of Sybase Central, clicking the Data tab.**
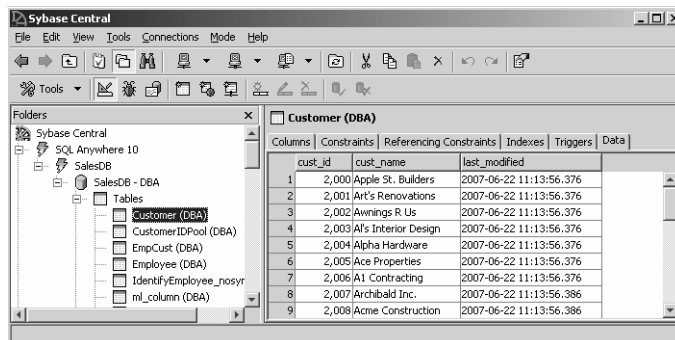
7 **Repeat steps 4-6 for the** *SalesDB_remote* **remote database. Notice that there is no data populated for this database.**
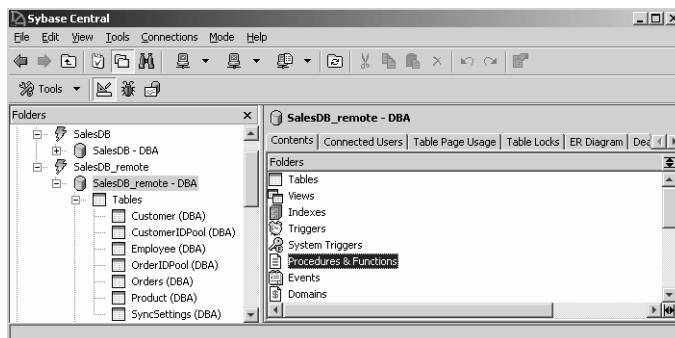
Data will be transferred to this database during the first synchronization.

The consolidated database does not yet have the required MobiLink synchronization system tables. You add them next with the MobiLink Create Synchronization Model wizard.

8   In Sybase Central, select the Tools>MobiLink 10>Setup MobiLink Synchronization menu item.

The Create Synchronization Model wizard displays.
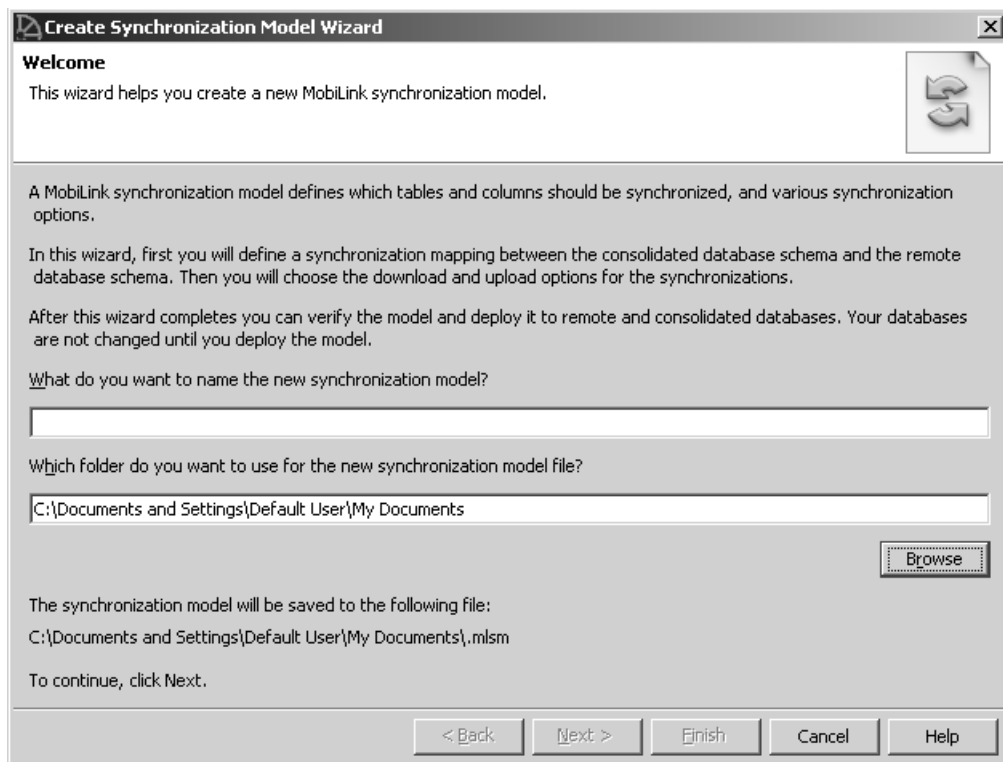
9   Type Sales Tutorial for the name of the synchronization model and click Next.

The Consolidated Database Schema page of the wizard displays.

10   Click the "Choose a consolidated database" button.

The Connect to Consolidated Database dialog box displays.

11 Select the ODBC Data Source radio button, select or browse to the SalesDB data source name in the corresponding drop-down list, and click OK.

The SalesDB database already has MobiLink system tables in it. If it did not, a message box would display, prompting you to install MobiLink system tables to the database under the owner 'DBA'. Clicking Yes in this message box would add MobiLink system tables to the database.

An alternative way to add MobiLink system tables is to run the *syncsa.sql* script from the Interactive SQL utility while connected to the consolidated database. The *syncsa.sql* script is installed by the SQL Anywhere 10 setup program in the SQL Anywhere 10/MobiLink/Setup directory.

Note that the database user who runs the setup script is the only user who will has permission to change the MobiLink system tables that are required for configuring MobiLink applications.

12 Click Next.

The Remote Database Schema wizard page displays.

13 Select the "Yes, use an existing remote database" radio button and click the "Choose a remote database" button.

The Connect to Remote Database dialog box displays.

14 Select the ODBC Data Source radio button, select or browse to the SalesDB_remote data source name in the corresponding drop-down list, click OK, then click Next on the wizard page.

The Existing Remote Database page of the wizard displays.

**15   Click Select All to select all the tables for synchronization, then clear
the check box for the Sync Settings table, and click Next.**
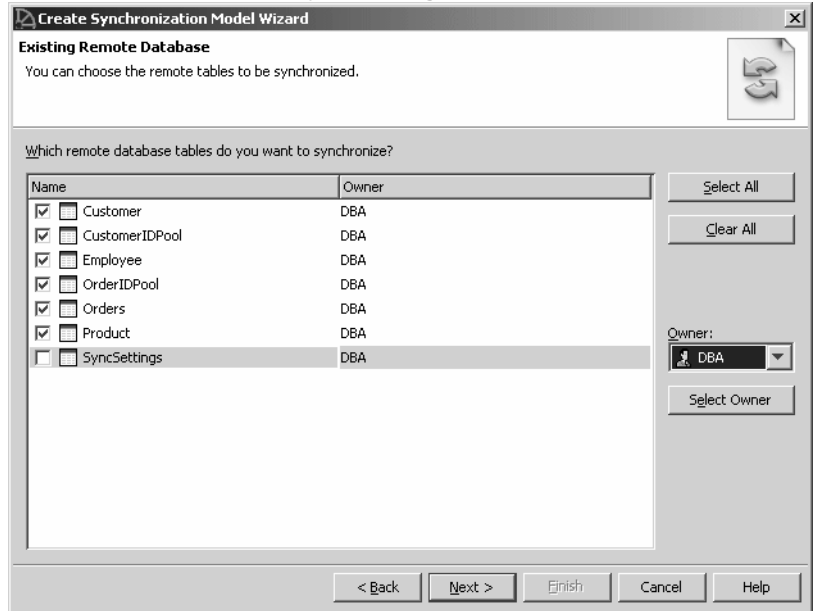
After you click Next, the Table Synchronization Mapping page displays.
It allows you to change the table mappings, as well as the direction of
synchronization. In this tutorial, the tables in the remote and consolidated
databases have the same names and the synchronization is bidirectional for
all tables.

The Download Type page of the wizard displays. A timestamp-based
download is the default download type.

**16   Click Next through the remaining wizard pages to examine the
remaining defaults, then click Finish.**

The Sales Tutorial synchronization model displays under the MobiLink
node in the Folders view of Sybase Central. The MobiLink plug-in is set
in the Model mode. If you switch to the Admin mode, the synchronization
model is no longer is listed in the Folders view, but it is listed again if you
switch back to the Model mode.

17 Click the Events tab of the Sales Tutorial synchronization model in the right pane of Sybase Central.

Default scripts display for five synchronization events and each of the tables that you mapped for synchronization. The scripted events for each table are: download_curser, download_delete_cursor, upload_delete, upload_insert, and upload_update.

The download_cursor event downloads data from the consolidated tables to the remote table if the timestamps indicate the data has changed since the last synchronization. The upload_insert event inserts new data from the remote database into the consolidated database and the upload_update event updates the consolidated database with any changes to existing data that were made in the remote database.

18 Close Sybase Central.
    Click Yes if a dialog box prompts you to save changes to the Sales Tutorial synchronization model.

Begin modifying the SalesDB application

Where you are
Set up the SalesDB databases
> Begin modifying the SalesDB application
Create a DataWindow for sales order information
Add menu items to the application menu
Create a MobiLink connection
Create the main application window
Test the application on the desktop
Deploy the application to a device
Run the application
Troubleshoot the application

1   If PocketBuilder is not running, from the Start Menu, select Programs>Sybase>PocketBuilder 2.0> PocketBuilder to start PocketBuilder.

Before you start developing the SalesDB application in PocketBuilder, you must specify how to connect to the database by creating a database profile. Since SalesDB is an application for the Pocket PC, you create a database profile only for the remote database.

**2   Select Tools> Database Profile.
Select ODB ODBC in the list box and then click the New button.**

The Database Profile Settings dialog box displays.

3   Type SalesDB_remote for the Profile Name.
Select SalesDB_remote for the Data Source.
Uncheck User ID and Password (since they are provided by the DSN
file on the Pocket PC device).
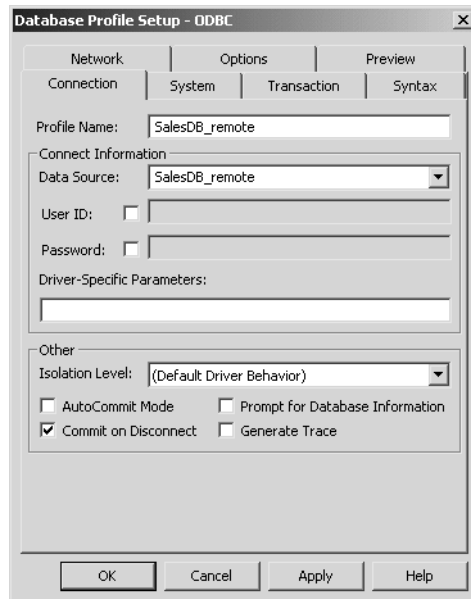
4   Select the Preview tab to examine the database connection syntax.

The Preview tab also has a Test Connection button you can click to test the connection.

5   Click OK to return to the Database Profiles painter, expand the ODB
ODBC,select SalesDB_remote, and click Connect.

The SQL Anywhere remote database starts and a connection is established. You can verify this by selecting Tools>Database Profile from the menu bar. SalesDB_remote should have a green check mark next to it.

Now you set up the Sales database workspace.

6   **Select File>New from the PocketBuilder menu bar.**
On the Workspace tab, select Workspace and click OK.
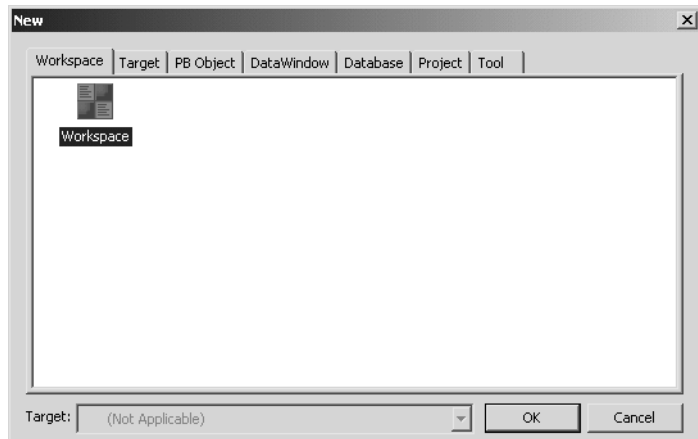
7   **Save the new workspace as** salesdb_tutorial **in the**
\PocketBuilder 2.0 \Tutorial\SalesDB directory.
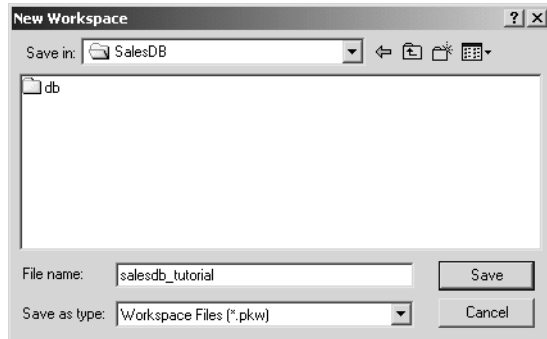
**8   Select File>New and then in the Target tab, select Existing Application
       and click OK.**
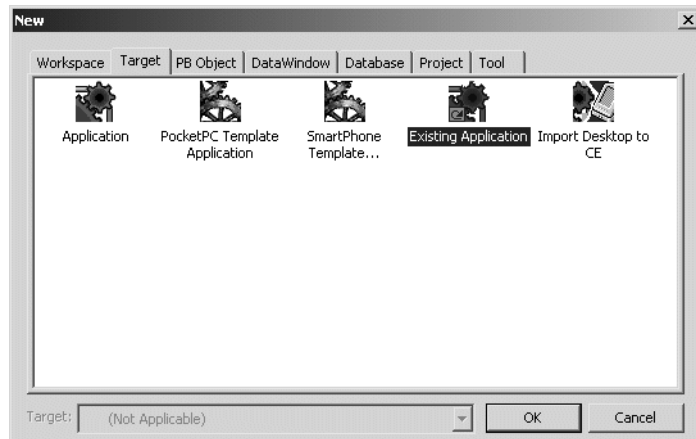
**9   Locate and expand salesdb_tutorial.pkl in the \Tutorial\SalesDB
       directory.**

10   Select the salesdb_tutorial application object and then click Next.
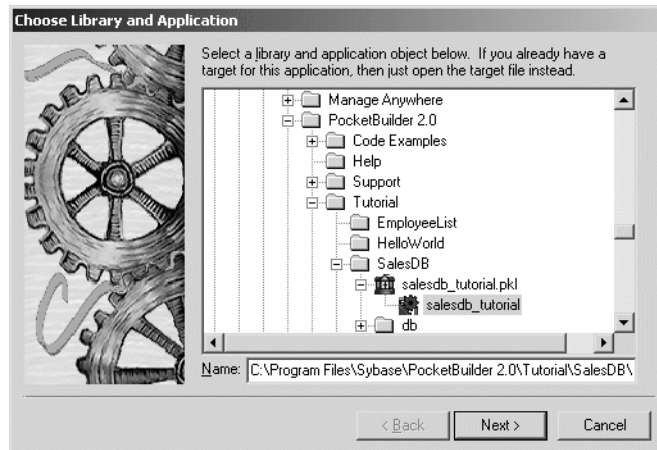
11   Click Next again and then click Finish to accept the default locations. If prompted, click Yes to migrate the existing application to the latest version.

The existing SalesDB application is now ready to be modified. Next you create a function called f_conn to handle the connection code.

12   Select File -> New from the menu bar.
In the PB Object page, select Function and then click OK.
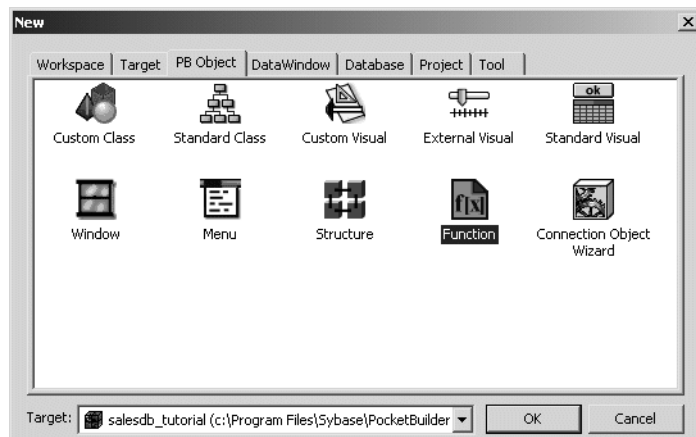
The Function painter opens. You use the Function painter to define and code the new f_conn function.

13   Select (None) at the bottom of the drop-down list for the return type.
Type `f_conn` **for the function name.**
Do not include an argument name or a Throws statement.

14   In the script view (the code editor below the function properties), type the following code on separate lines:

```
sqlca.dbms='odb'
sqlca.dbparm="ConnectString='DSN=SalesDB_remote'"

// establish DB connection
connect using sqlca;
```

This code establishes a connection to the SalesDB_remote database. The same parameter can be used to initiate a connection on the desktop or in the Pocket PC environment.

Here is what the definition of your new f_conn function should look like:
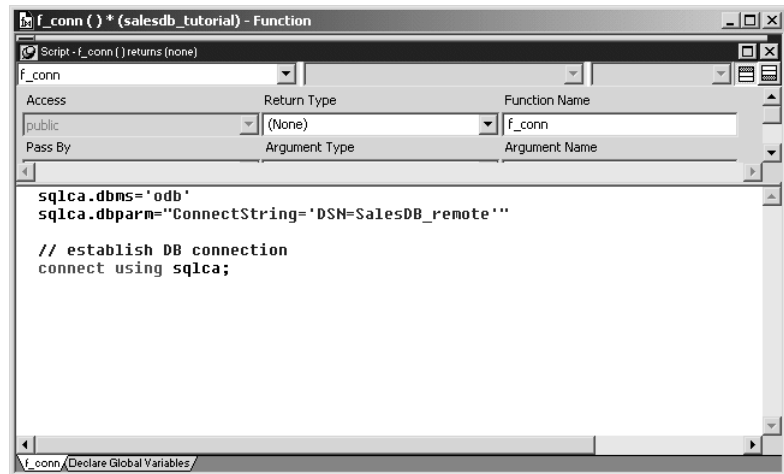
15 **Select File>Save, then OK, to save the f_conn function.**

16 **Close the Function painter.**

17 **Expand salesdb_tutorial.pkl in the System Tree in the file.**

In the System Tree, you can see the f_conn function that you just created.

The f_conn function is used when the application first starts. It is called from the ue_postopen event inside the salesdb_tutorial application. The database connection is established in this event and any connection error is reported to the user. The database connection is closed when the application terminates and is handled by the f_disconn function that is provided for you.

Next you create a DataWindow to access sales data in your application.

Create a DataWindow for sales order information

Where you are
Set up the SalesDB databases
Begin modifying the SalesDB application
> Create a DataWindow for sales order information
Add menu items to the application menu
Create a MobiLink connection
Create the main application window
Test the application on the desktop
Deploy the application to a device
Run the application
Troubleshoot the application

A DataWindow is a powerful PocketBuilder object that allows you to access data and manipulate the data visually in a variety of ways. The DataWindow you build will display sales order information in a scrollable window.

Now you build the d_orders DataWindow object.

1   Select File>New from the menu bar.
   In the DataWindow page, select Freeform and then click OK.
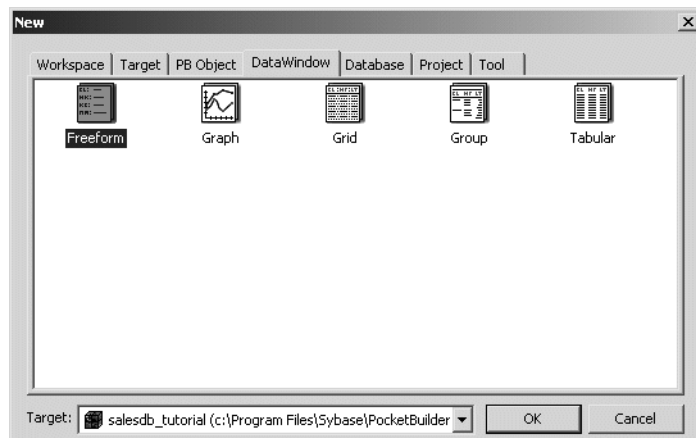
2   Select SQL Select as the data source and check Retrieve on Preview.
   Click Next.

3   In the Select Tables dialog box, click customer, orders, and product.
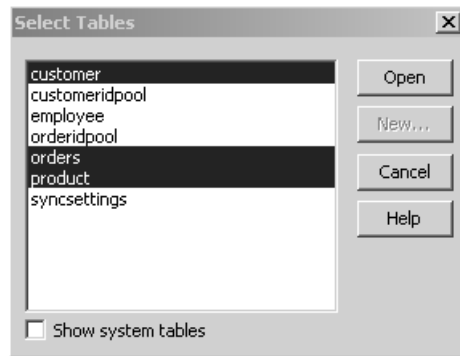
4   Click Open.
In the Table Layout window, click the following items in this order:

```
order_id from orders
cust_name from customer
prod_name from product
quant from orders
price from product
disc, status, and notes from orders
```

The order in which the items are clicked determines the order in which the columns appear in the DataWindow. You can rearrange the order later if needed.

The Syntax tab at the bottom displays the query that PocketBuilder will use to retrieve the data. Next, you modify the query to sort the result by order_id.

**5   Click the Sort tab (at the bottom of the painter workspace) if it is not
    already selected.
    Drag "orders"."order.id" from the left pane to the right pane.
    Be sure that the Ascending check box is selected.**

If you do not see the tabs at the bottom
Select View>Layouts>(Default) from the menu bar.

---
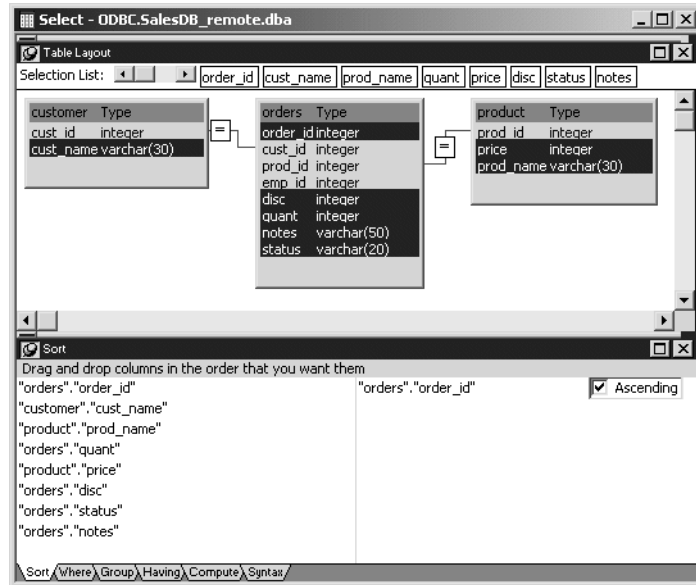
6   Click the Syntax tab.

The Syntax tab displays the selected columns as well as the sorting
criteria. Notice how all of the SQL is generated without your having to
type anything yourself.

This completes the data selection process. Next you manipulate the layout
of the data displayed in the DataWindow object.

**7   Select File>Return to DataWindow Painter.
    Click Next to accept the default Color and Border settings.
    Click Finish to generate the DataWindow.**

**8   Select File>Save and name the DataWindow d_orders.
    Optionally add a comment to describe the DataWindow.
    Click OK.**

9 In the Detail band of the Design view, use Ctrl/click to select the *Order Id:* **label and** *orders_order_id* **column.**
Press the Delete key.

These are the first items under the Header band. The order_id is for internal use and is not displayed to users, so you can delete it.

10 Select the *Cust Name:* **label and in the Properties view, change the Text property (at the bottom of the view) to** *Customer:***.**
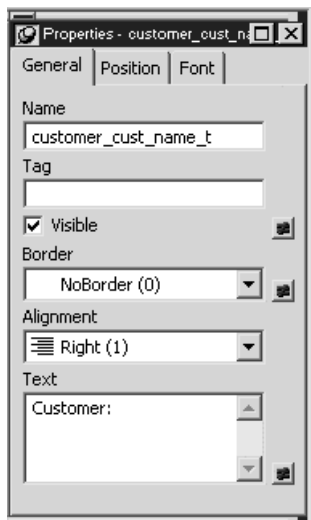
11 Using the same method as the previous step, change *Prod Name:* **to** *Product:***,** *Quant:* **to** *Quantity:,* **and** *Disc:* **to** *Discount:***.**

The spacing of controls in a window is particularly important in Pocket PC development. Next you make all the controls fit in a single screen.

12   Select Edit>Select>Select All from the menu bar, and then drag all the labels and column names to the top of the band.
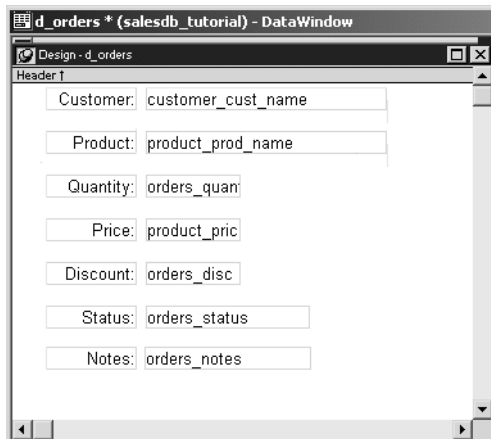
13   Align and space the labels and column names so they are closer together and look like this:
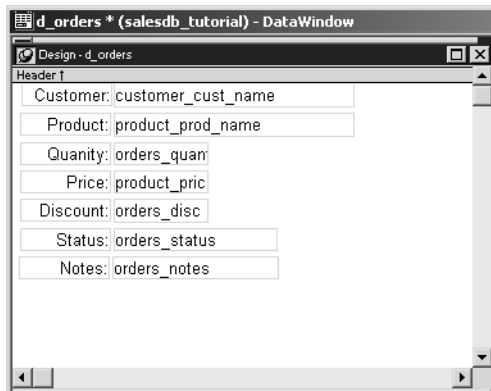
You can align controls in a DataWindow object by selecting them with Ctrl/click and then selecting one of the Format>Align cascading menu choices. You can equalize the space between controls by selecting them with Ctrl/click and then selecting one of the Format>Space cascading menu choices.

**14   Select File>Save to save the DataWindow.
Close the DataWindow painter.**

Add menu items to the application menu

Where you are
Set up the SalesDB databases
Begin modifying the SalesDB application
Create a DataWindow for sales order information
> Add menu items to the application menu
Create a MobiLink connection
Create the main application window
Test the application on the desktop
Deploy the application to a device
Run the application
Troubleshoot the application

Now you add new menu items to the m_salesdb menu.

1   In the System Tree, double-click m_salesdb.

The Menu painter opens.

2   In the Menu painter tree view, right click Order and then select Insert Menu Item.
In the blank box that displays, type File and press Enter.

3   Right click File and select Insert Submenu Item from the popup menu.
For the submenu item text, type *Synchronize* and press Enter.

4   Repeat the previous step but type *Sync Options...* for the submenu item text.

You will script the Synchronize and Sync Options menu items after you create synchronization objects using the MobiLink Synchronization for ASA wizard.

5   Repeat the step a third time but type *Exit* for the submenu item text.

The File menu name properties should be m_file, m_synchronize, m_syncoptions, and m_exit. If any menu name is incorrect, clear the Lock Name check box in the Properties view and correct the menu name. The Menu painter should now look like this:
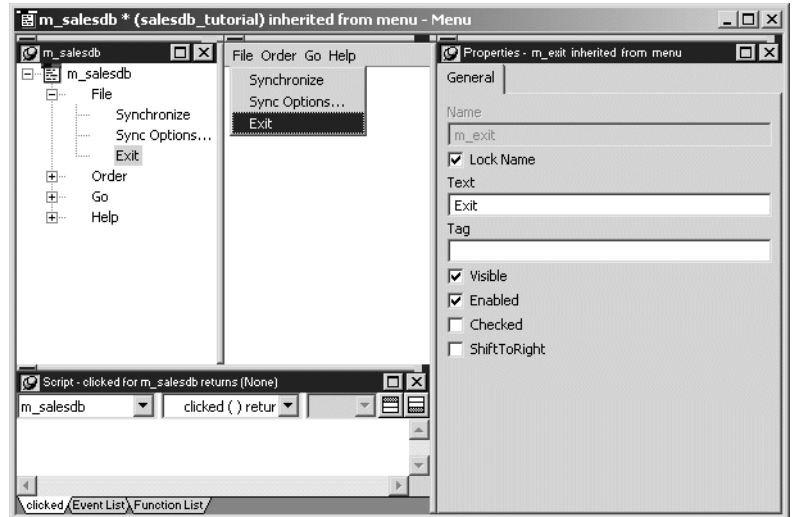
6   Double click the Exit menu item in the main menu tree view.
Be sure the drop-down menu at the top of the Script view displays the clicked( ) event for the m_file.m_exit object.
Type the following code in the Script view.

```
// Terminate application
f_disconn()
Halt Close
```

7   Select File>Save to save the changes.

8   Close the Menu painter.

Create a MobiLink connection

Where you are
Set up the SalesDB databases
Begin modifying the SalesDB application
Create a DataWindow for sales order information
Add menu items to the application menu
> Create a MobiLink connection
Create the main application window
Test the application on the desktop
Deploy the application to a device
Run the application
Troubleshoot the application

Now you generate a MobiLink connection for the remote application using the MobiLink Synchronization wizard.

1 **Select File>New.**
In the Database tab, select MobiLink Synchronization for ASA.
Click OK.
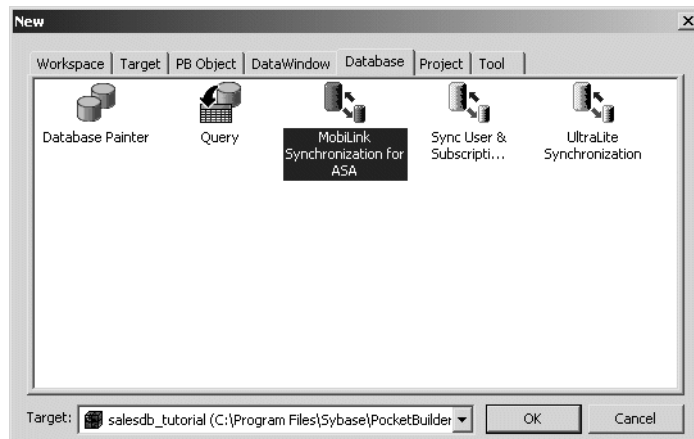
2 **Read the first overview screen.**
Click Next.
Read the second screen to learn about what the wizard provides.
Click Next again.

3   Click Next to accept salesdb_tutorial.pkl as the default library for storing the generated MobiLink objects.

**4   Select SalesDB_remote from the drop-down menu.
Click Test Connection to ensure the DSN is in working order.**
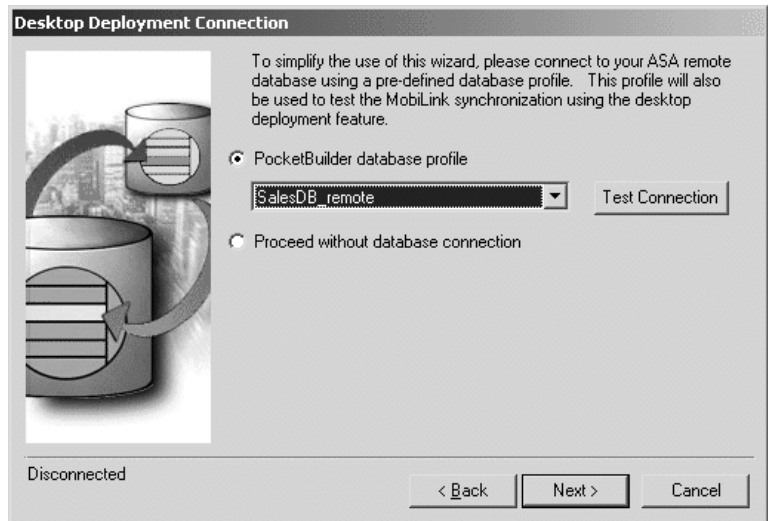
A *Connection successful* message should display in the lower-left corner of the wizard page. If it does not, verify that the DSN is correctly configured in PocketBuilder and the ODBC Administrator. The ODBC Administrator is located in the Objects view of the Database painter in the ODB ODBC utilities.

5 Click Next.
   Choose Browse and then select SalesDB_remote.DSN in the
   \Tutorial\SalesDB directory.
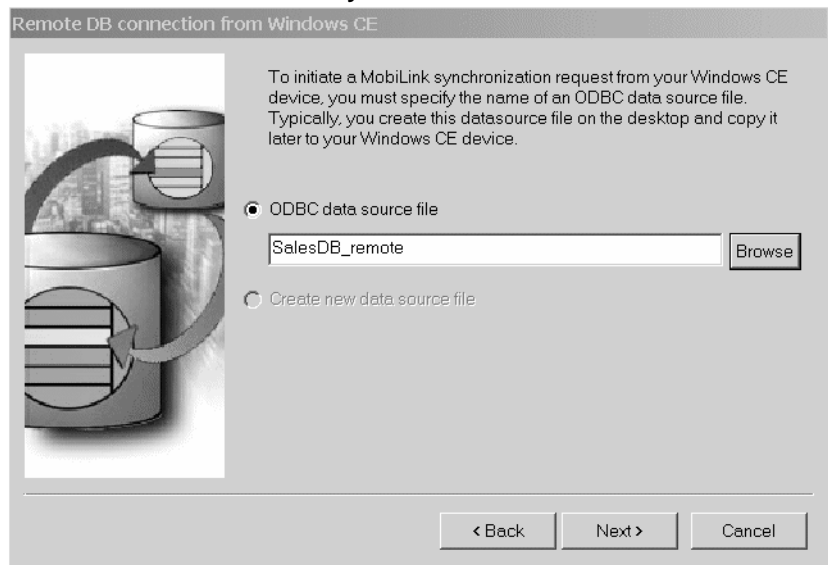
6 Click Next to display the wizard page showing publications in the
   remote database.
   Select the salesapi publication.
   Click Next.
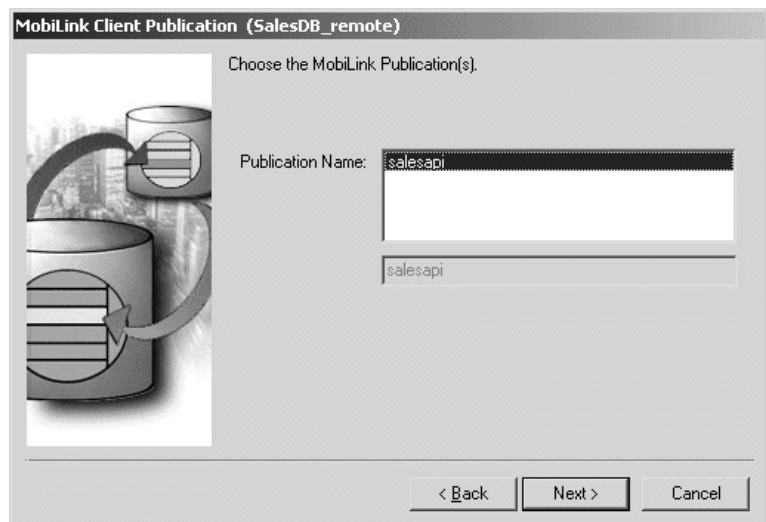
7   Click Next to accept the default names for the generated MobiLink objects.
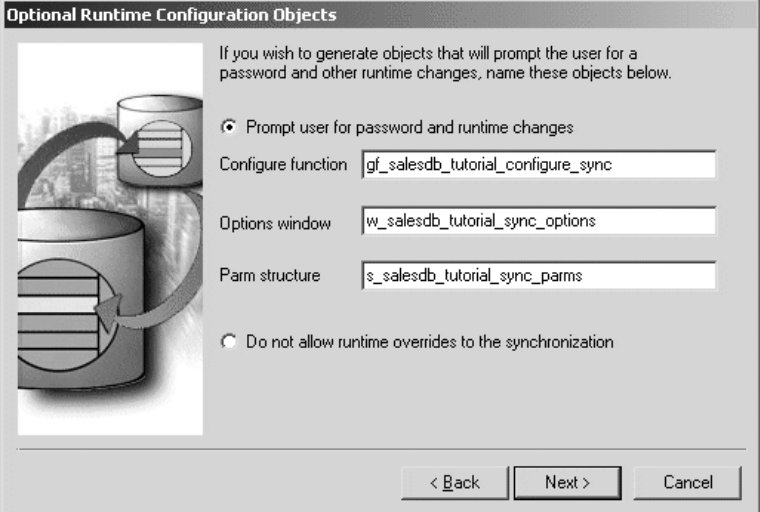
The default names are nvo_salesdb_tutorial_sync for the nonvisual object that will control the MobiLink synchronization client, and gf_salesdb_tutorial_sync for a global function that instantiates the nonvisual object and makes the call to launch a synchronization request.

After you click Next, the MobiLink Client Display Options page of the wizard displays.

8   In the Mobilink Client Display Options page, click Next to accept the default option.

The default display option causes the wizard to generate a synchronization status window named w_salesdb_tutorial_sync.

After you click Next, the Optional Runtime Configuration Objects wizard page displays. On this page, you specify names of wizard-generated objects that allow users to change synchronization settings at runtime. You can also intruct the wizard not to generate these objects. However, for this tutorial, you leave the default selections.

If this were a production environment
In this lesson, you give the user significant control of the synchronization parameters. In a production environment however, giving the user a high degree of control over the synchronization parameters is usually not good practice.

9   **In the Optional Runtime Configuration Objects page, click Next.**

 You accept the default selection and instruct the wizard to generate synchronization objects that allow the end user to make runtime changes. The Override Registry Settings wizard page displays.

The MobiLink wizard stores synchronization information for each MobiLink user in the Windows registry on the device and on the desktop. The Override Registry Settings wizard page lets you specify how PocketBuilder adds or modifies registry settings when redeploying and configuring end-user applications.

10   **In the Override Registry Settings wizard page, keep all the default values and then click Next.**

The MobiLink Client Logging Options page displays.

11   **Check the** *Show all except -vc and -vp (-v+)* **check box.**
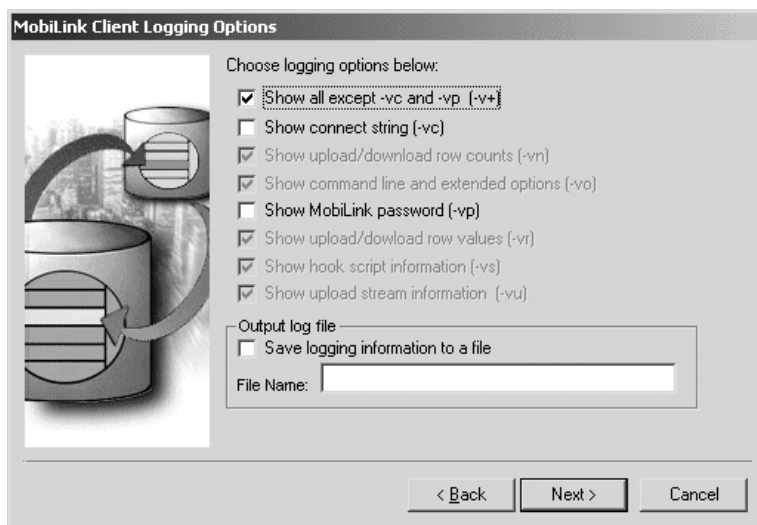
This wizard page determines the amount and type of information shown in the MobiLink status window during synchronization.

Improving performance for deployed applications
The *Show all except -vc and -vp (-v+)* option can result in the display of many status messages that affect the synchronization speed. To improve performance in a deployed application, show fewer messages.

---

12 Click Next.

The MobiLink Client Additional Options page displays. It allows you to include additional MobiLink command line options or extended options and assign the MobiLink host and port. You must also select the MobiLink server version you want to use. The default version is 9, but for 10.x SQL Anywhere databases, you must select 10 for the MobiLink server version.

13 In the Extended Options text box, type the value *lt=share***.**
    Select 10 for the MobiLink server version.

Table-locking significantly reduces data transfer times, so for your application to function optimally, you set the LockTable (lt) options to *share*.

14 Click Next.

The last wizard page contains a summary of your choices, including the default selections. If you need to, you can go back and change any options that are incorrect.

15 Click Finish to generate the MobiLink synchronization components in your project.

The MobiLink Synchronization wizard creates the following synchronization objects:

Synchronization object	Description
nvo_salesdb_tutorial_sync	Non-visual user object that controls the MobiLink synchronization client.
gf_salesdb_tutorial_sync	Global function that creates the user object and initiates synchronization requests.
s_salesdb_tutorial_sync_parms	Structure that stores the MobiLink command line parameters. (All the variables accessible from the wizard are MobiLink command line options.)
gf_salesdb_tutorial_configure_sync	Global function that handles a user request to change the synchronization options and stores the values in the Windows/Windows CE registry.

The MobiLink Synchronization wizard also creates two windows:

Window	Use
w_salesdb_tutorial_sync	Displays the synchronization status information
w_salesdb_tutorial_sync_options	Sets synchronization options at runtime

Now you enable the w_salesdb_tutorial_sync_options Sync Options window in the Menu painter.

16 Double-click m_salesdb in the System Tree to open the Menu painter. In the Menu painter tree view, expand the File menu and double-click the *Sync Options…* **menu item.**

17 Check to be sure that the drop-down lists at the top of the Script view display the clicked( ) event for the m_file.m_syncoptions object. If not, select that object and event from the drop-down lists.

18 Type the following code in script view:

```
// Open the Sync Options window
gf_salesdb_tutorial_configure_sync()

// Fetch data
f_refresh_orders(-1)
```
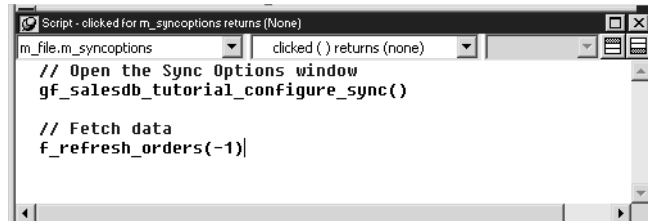

Next, to enable synchronization when the user clicks Synchronize in the File menu, you add a call to the gf_salesdb_tutorial_sync global function.

19 In the Menu painter tree view, double-click Synchronize.
Make sure the drop-down menu at the top of the script view displays the clicked() event for the m_file.m_synchronize object.

20 Type the following code in script view:

```
      // Start synchronization
IF gf_salesdb_tutorial_sync(string(::g_emp_id),"")<>0 THEN
MessageBox("Error"," MobiLink Synchronization Error.")
      END IF
      // Fetch data
      f_refresh_orders(-1)
```
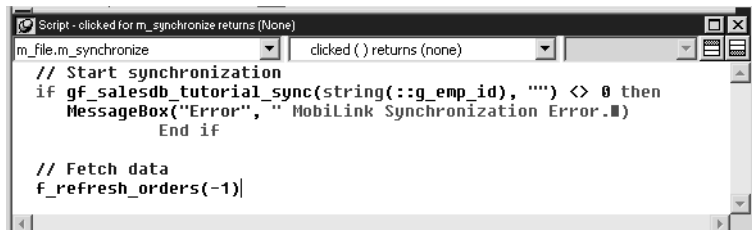

21 Select File>Save to save the changes.
Close the Menu painter.

Create the main application window

Where you are
Set up the SalesDB databases
Begin modifying the SalesDB application
Create a DataWindow for sales order information
Add menu items to the application menu
Create a MobiLink connection
> Create the main application window
Test the application on the desktop
Deploy the application to a device
Run the application
Troubleshoot the application

Now you create a w_orders window.

1   Select File -> New.
In the PB Object tab, select Window and click OK.
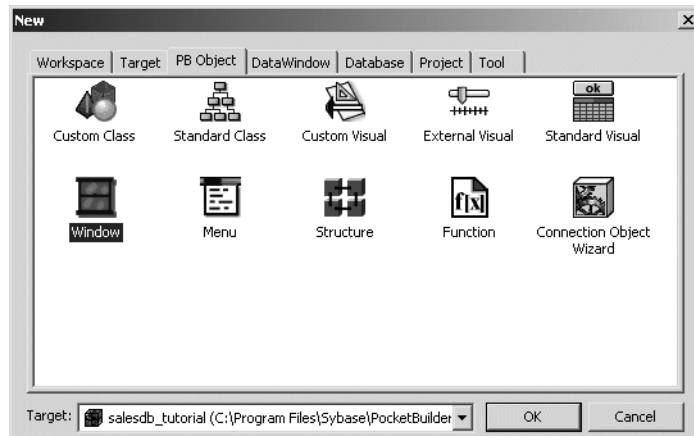
2   On the General tab of the window Properties view, make sure the following values are set:

Window property	Value
Title	SalesDB tutorial
MenuName	m_salesdb
Visible	selected
Enabled	selected

Window property	Value
Show SIP Button	selected
Close (OK)	selected

Selecting the Show SIP Button property makes the Soft Input Panel in the Pocket PC available to the user. Selecting the Close (OK) property adds an OK icon to the title bar of the window in the Pocket PC; when the user clicks OK, the application closes.

3   On the Scroll tab of the window Properties view, select the HScrollBar and VScrollBar check boxes.

4   Select File>Save to save the window.
Type `w_orders` **as the name of the window.**
Click OK.

5   Select Insert>Control>DataWindow.
Click the window in the Layout view to insert a DataWindow control in the w_orders window.
Click and drag the borders of the DataWindow object to size it, leaving space at the bottom of the window for navigation buttons.
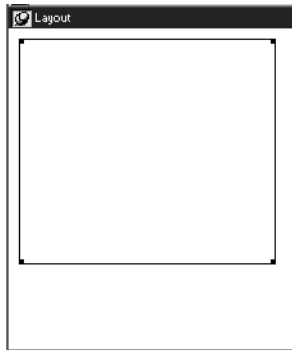
6   Change the DataWindow control properties on the General tab of the Properties view to these values:

DataWindow control property	Value
Name	dw_orders
DataObject	d_orders

DataWindow control property	Value
HScrollBar	checked
VScrollBar	checked

You should now see the column labels inside the DataWindow control.

7   Select Insert>Control>CommandButton.
  Click in the window below the DataWindow control to add the
  command button.
  Name the button cb_prev with command button text < *Previous***.**

8   Repeat the preceding step with the following settings:

Command button name	Value
cb_next	Next >
cb_approve	Approve...
cb_deny	Deny...

9   Rearrange the position of the DataWindow control and buttons so the
  resulting window looks like this:
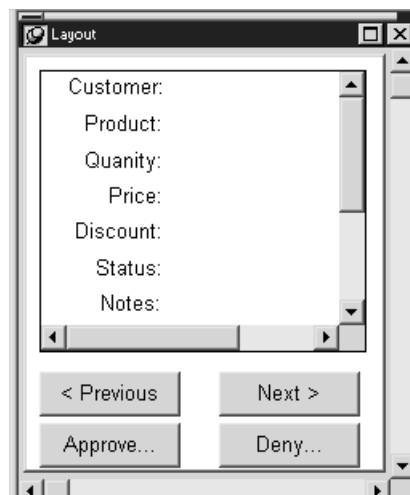
Next you associate events with the buttons.

10   Double click the cb_prev button.
Make sure the clicked( ) event displays in the Script view.
Type the following code in the Script view.

```
f_scroll(-1)
```

This function scrolls data in the DataWindow back by one row.

11   In the Script view, select the each of the other command buttons from
the first drop-down list and type the code shown below for the clicked
event of that command button:

Command button name	Code
cb_next	f_scroll(1)
cb_approve	f_approve_deny(APPROVE)
cb_deny	f_approve_deny(DENY)

12   Select File>Save to save the changes.
Close the Window painter.

Now you must tell the application to open the window when the
application starts.
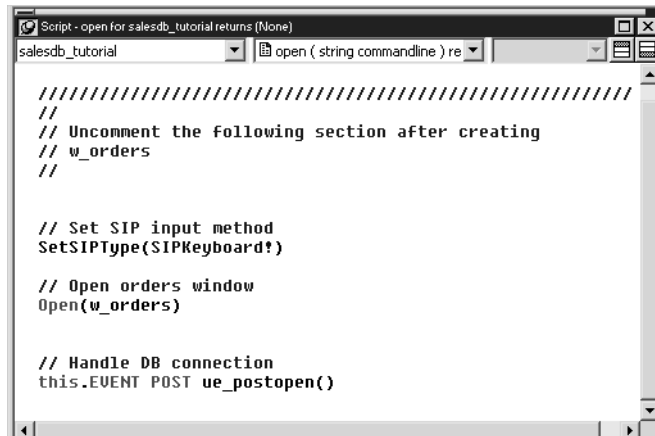
13   In the System Tree, double-click the salesdb_tutorial Application
object.
In the second drop-down list in the Script view, select the *open* **event.**
Uncomment the section following the text: Uncomment the
following section after creating w_orders. **Leave the**
comments that start with a double slash (//) unchanged.

Uncommenting code
You can use comments in code to document your scripts or prevent
statements within scripts from executing. In this step and the next few
steps, you uncomment scripts that have been commented to prevent them
from executing. The slash-and-asterisk method of commenting code looks
like this: /* *Code* */, where *Code* can refer to any number of lines of code.
You uncomment the code by deleting the /* and the */ characters that
surround the code.

The double slash method of commenting code can comment out only a
single line of code at a time. The tutorial uses this method to include
descriptive text. If you remove the comments from the descriptive text, the
PocketBuilder compiler will indicate a compilation error.

Here is what you should see in the Script view now:

This code had to be commented out before you created the w_orders window. Since you created the window at the beginning of this lesson, you can now uncomment the code without causing the PocketBuilder compiler to issue an error message.

14 **In the second drop-down list, select the** *ue_postopen* **event.**
Uncomment the section following the text: Uncomment the
following section after creating w_orders**.**
Leave the comments that start with a double slash (//) unchanged.

15 **Select File>Save to save the changes.**
Close the Application painter.

16 **In the System Tree, double-click the m_salesdb menu.**
Expand Order, and double-click the Delete submenu item.
Uncomment the section following the text: Uncomment the
following section after creating w_orders**.**
Leave the comments that start with a double slash (//) unchanged.

17 **Select File>Save to save the changes.**
Close the Menu painter.

**18   In the System Tree, double-click the f_scroll function.
Uncomment the section following the text:** Uncomment the
following section after creating w_orders**.
Leave the comments that start with a double slash (//) unchanged.**

**19   Select File>Save to save the changes.
Close the Function painter.**

**20   Repeat steps 18 and 19 for f_scroll_last, f_approve_deny,
f_refresh_orders and f_set_dir_btn_enabled.**

The SalesDB application is now ready to be tested.

Test the application on the desktop

Where you are
Set up the SalesDB databases
Begin modifying the SalesDB application
Create a DataWindow for sales order information
Add menu items to the application menu
Create a MobiLink connection
Create the main application window
> Test the application on the desktop
Deploy the application to a device
Run the application
Troubleshoot the application

Before you run your application in PocketBuilder to preview and test it, you need to start the MobiLink server.

**1   Close any painters that are open in PocketBuilder.
Select Tools>Database Profile to open the Database Profiles tool.**

**2   Expand the ODB ODBC node and then expand the Utilities node.
Double-click the MobiLink Synchronization Server utility.**
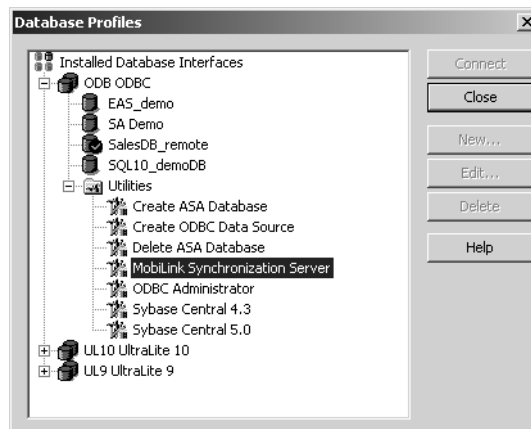
**3   For the MobiLink version, select SQL Anywhere 10.
For the ODBC connection string, type** *DSN=SalesDB***.**
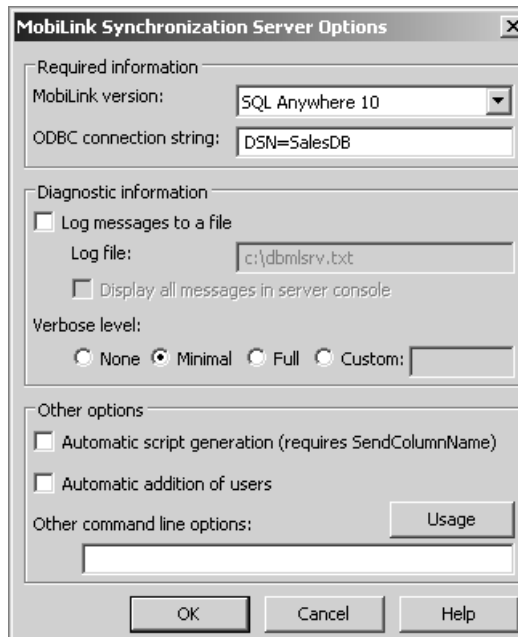
4   Click OK.

The MobiLink Synchronization Server runs and is ready to accept requests.
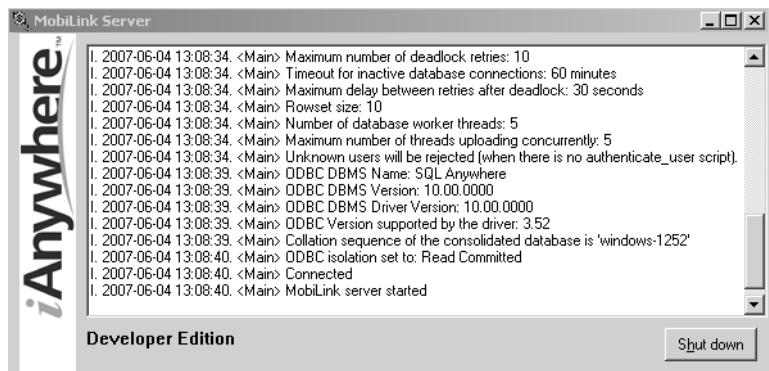
5   Close the Database Profiles painter.

6   Select Run>Run from the menu bar.

The SalesDB application connects to the local copy of the
SalesDB_remote remote database and the Sync Options dialog box opens.

In the Sync Options dialog box, you can type syncronization information
including the MobiLink user name and password.

**7   For testing purposes, type 50 as the user name in the MLUser text
box, leave the MLPassword text box blank, and maintain the default
for all other options.**

If you use 50 as the user name now, when you run the application on your
device or emulator you should use another user name such as 51 or 52.

Default protocol and host
For ease of testing, the default protocol is TCP/IP and the default host is
the machine name or the machine IP number. The default port is 2439 for
TCP/IP, 80 for HTTP, and 443 for HTTPS.

8   Click OK to synchronize.

First the synchronization completes.
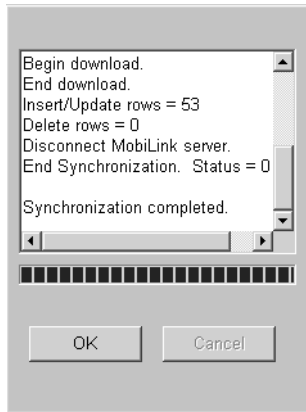
9   Click OK when the synchronization completes.

The application starts on the desktop.
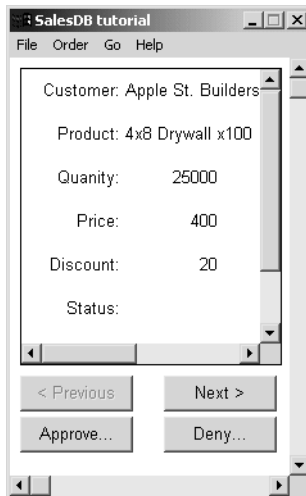
Since you can run PocketBuilder applications from the design-time environment, it is possible to develop and test applications without a device or emulator.

Now you test your application.

10 Explore the layout of the application.
Click the Order>New menu item and add a new order.

The application handles orders that have been modified in the remote
database.

11 Click the Next, Approve, and Deny buttons.

Here the Approve button is clicked and the Good Work note is selected.
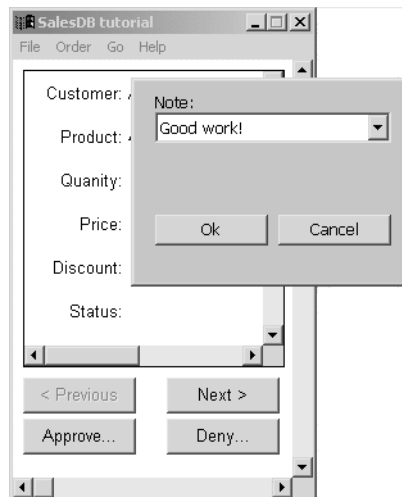
12 Click the OK button to close the Note.
Select File>Exit to exit the application and return to PocketBuilder.

Deploy the application to a device

Where you are
Set up the SalesDB databases
Begin modifying the SalesDB application
Create a DataWindow for sales order information
Add menu items to the application menu
Create a MobiLink connection
Create the main application window
Test the application on the desktop
> Deploy the application to a device
Run the application
Troubleshoot the application

Before an application can be deployed, a project is needed to specify how the application is deployed.

Create a project

1   Select File>New.
In the Project tab, select Application and click OK.
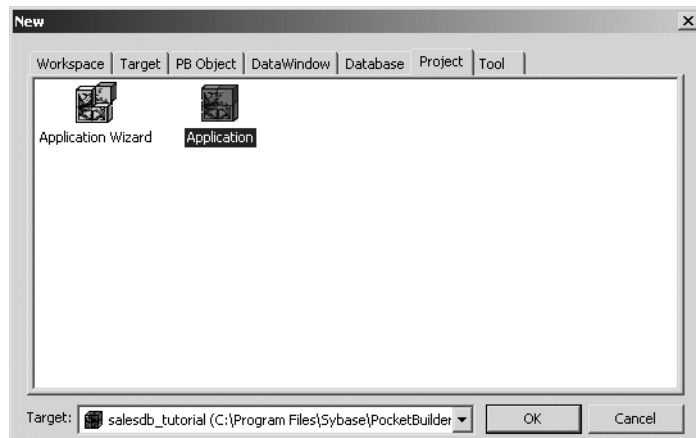
The Project painter displays. The default executable file name is salesdb_tutorial.exe.
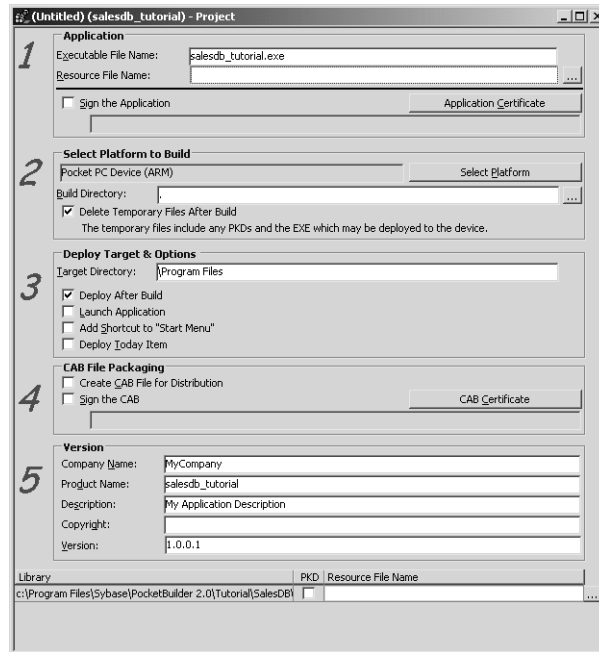
Pocket PC Device (ARM) is the platform for deployment to an ARM device. Use the default \Program Files location for deployment to the device. For your own applications, you might want to deploy to an application-specific directory that you create.

Select Platform
The Select Platform button allows you to choose a different deployment platform, such as a Smartphone or emulator. You can also launch the Emulator Manager from the Select a Build Platform dialog box that you open by clicking the Select Platform button.

2   **Select the Add Shortcut to "Start Menu" check box to add a Start menu item for the deployed application on the device.**

3 **Select File>Save to save the project.**
Type p_salesdb_tutorial for the project name and click OK.

Before you can deploy SalesDB to a device or emulator, you must deploy the the local copy of the remote database and the DSN file to the Pocket PC or emulator. You do that next.

Copy the remote database and log files to the Pocket PC

Now you copy the remote database and log file to the Pocket PC. You copy the log file because synchronization subscriptions were added to the remote database during the initialization phase. Synchronization subscriptions can also be added during the first execution of the application. When adding the subscriptions at runtime, you need to deploy only the database file to the device, not the log file.

1 **On your desktop machine, use a text editor to open SalesDB_remote.dsn in the \Tutorial\SalesDB directory.**

2 **Make sure that the paths for the database file and the start line (set to the default location) are correct for your Pocket PC device.**

Since you are using SQL Anywhere 10 on the device, you must add the following line:

```
driver = dbodbc10.dll
```

You will also need to change the database file and start directories:

```
databasefile=\Program Files\SQLAny10\SalesDB_remote.db
start=\Program Files\SQLAny10\dbsrv10.exe
```

3   Open ActiveSync while a connection to the device is established.
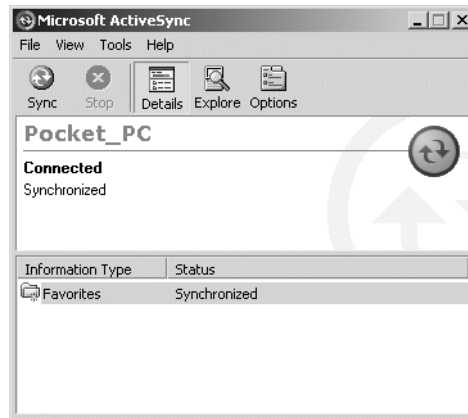
4   Click Explore in the ActiveSync toolbar.
Double-click My Pocket PC.

My Pocket PC is the root directory on the Pocket PC device.

5   Open another Windows Explorer window.
Copy SalesDB_remote.DSN from your PocketBuilder 2.0
Tutorial\SalesDB directory to the root directory on your Pocket PC
device (My Pocket PC or My Device).

You can drag and drop files from the desktop to the Pocket PC in the same
way that you drag and drop files from one directory to another on the
Windows desktop.

6   In the Explorer window for your device, go to the \Program
Files\SQLAny10 directory.

This is the default directory for SQL Anywhere 10 on a Pocket PC. If you
installed SQL Anywhere in a different directory, be sure to use that
directory instead.

7   Locate the \Tutorial\SalesDB\db\fresh directory on your desktop.
Copy SalesDB_remote.db and SalesDB_remote.log to the
SQL Anywhere 10 directory on your device.

Build and deploy the application

Now you can build the application and deploy it to a device or emulator by running the project object you created in "Create a project" on page 157 of this lesson. However, because the registry location for SQL Anywhere 10 is not the same as the registry location on the device, you must first modify the registry path that you generated with the MobiLink Synchronization for ASA wizard.

1   In the PocketBuilder System Tree, double-click the nvo_salesdb_tutorial_sync user object that you generated with the MobiLink Synchronization for ASA wizard.

The User Object painter opens.

2   Click the Function List tab in the main stack view of the User Object painter.

3   Double-click "determine_asa_regpath () returns integer" in the functions list.

The script for determining the SQL Anywhere registry path displays. Script constants display the registry path for the 8, 9, and 10 versions of the database server. In the MobiLink Synchronization for ASA wizard, you selected 10 as the server version. You can verify this by selecting the Declare Instance Variables tab and examining the value of the *is_mlserverversion* variable.

4   Comment out the constant string that sets the registry path for SQL Anywhere 10.

The path assigned by the wizard ("HKEY_LOCAL_MACHINE\Software\Sybase\SQL Anywhere\10.0") is valid on the desktop, but not on the device.

To comment out this string, you can add two forward slashes (//) in front of the line that starts: `constant string ASA10_REGPATH =`.

5   Add the following assignment after the line you just commented out:

```
constant string ASA10_REGPATH = HKEY_CURRENT_USER\Software\Sybase\SQL Anywhere\10.0
```

This line assigns the *ASA10_REGPATH* variable to "HKEY_CURRENT_USER\Software\Sybase\SQL Anywhere\10.0"

This is the registry path used on the device for SQL Anywhere.

6   **Save your changes and close the User Object painter.**

7   **Double-click the p_salesdb_tutorial project object in the System Tree.**

The Project painter opens.

8   **Select Run>Build and Deploy Workspace or click the Deploy button on the PainterBar.**

The project is built and deployed and the application is copied to the device's \Program Files directory.

Next you run the application on the Pocket PC device.

Run the application

Now you run the SalesDB application.

About troubleshooting the application
If you have problems running the application, read the troubleshooting
information in

**1   Tap PocketBuilder 2.1 in the Start Menu on the Pocket PC, or tap the
      Start>Programs item and select the File Explorer.**

   If you tap PocketBuilder 2.1, the PocketBuilder Application List utility
   displays.

**2   Navigate to the directory where the SalesDB sample application is
      installed (\Program Files by default).**

3   Tap SalesDB_tutorial.exe to start the SalesDB application.

   When you run SalesDB the first time, the Sync Options window pops up
   and you can type your MobiLink user name and password.

**4   Using the Soft Input Panel, type 51 in the MLUser text box and leave
      the MLPassword text box blank.**

Single-user devices only
The databases you generated using the *MakeDB10.cmd* batch file are intended for single-user devices only. SQL Anywhere does have the capability to handle multiple user devices, but this is beyond the scope of this lesson.

---

5   Click OK.

This automatically launches a synchronization request. Data that is relevant to the employee is downloaded to the Pocket PC after the first synchronization. Any changes made in the SalesDB application will be updated in the consolidated database during the next synchronization.
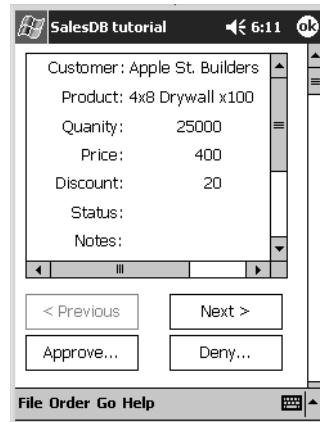
6   Browse the sales data using the Next and Previous buttons. Examine the menu items at the bottom: File, Order, Go, and Help.

7   Click the Order menu item to add a new sales order or remove a sales order.

Troubleshoot the application

Where you are
Set up the SalesDB databases
Begin modifying the SalesDB application
Create a DataWindow for sales order information
Add menu items to the application menu
Create a MobiLink connection
Create the main application window
Test the application on the desktop
Deploy the application to a device
Run the application
> Troubleshoot the application

If you are having any problems with your application, follow these steps to correct the problems.

1   Make note of the problems you are having with your application.

2   Read the questions and answers in the table that follows.

The table provides you with some information that can help you troubleshoot your application:

Question	Answer
When I start the SalesDB application, the *Connect to Adaptive Server...* dialog box displays. The application cannot establish a connection to the database.	This is most likely due to an incorrect DSN file. Check the *SalesDB_remote.DSN* located at the root directory of the device or emulator and make sure that the database file and start properties point to the correct locations. For SQL Anywhere 10.0, make sure the DSN contains a separate line for the *dbodbc10.dll* driver assignment.
When I initialize synchronization from the device or emulator, the MobiLink window shows the error *Error: Protocol version mismatch*.	Check the version of the SQL Anywhere engine on the device or emulator and the version of the MobiLink server. Versions for both components need to match.

Question	Answer
When I initialize synchronization from the device or emulator, the MobiLink window shows the error *Communication error occurred while receiving data from the MobiLink server*.	Make sure the MobiLink server is running and check the MobiLink server log for more details.
When I initialize synchronization from the device, MobiLink stalls while displaying an error message about connecting to the MobiLink server.	In the SalesDB application on the device, select File>Sync Options and click the ML Server tab. Then type the host and port of the MobiLink server you want to connect to and try to synchronize again.

This completes the lesson. This lesson demonstrates the foundation of building a PocketBuilder application using SQL Anywhere databases and MobiLink synchronization technology.

For more information, see the chapter on SQL Anywhere database technology in the PocketBuilder *User's Guide* and the chapter on MobiLink technology in the PocketBuilder *Resource Guide*. Also see the SQL Anywhere Studio documentation that is related to these technologies.

What to do next

Congratulations. You have completed all the lessons in the tutorial. Now you know the basics of application development with PocketBuilder.

The Preface to this book includes a guide to the PocketBuilder documentation. To further your understanding, you should continue with these books:

> *User's Guide*
> *Resource Guide*

All the PocketBuilder books are available in the Online Books and on the Sybase Web site at http://www.sybase.com/support/manuals/.

Index

user objects
   about   25
   class   26
   types of   26
   visual   26

V

views
   docking   68
   floating   68
   manipulating   66
   pinning   68
   saving layout schemes   70
   stacks   69
views, types of
   Design (DataWindow painter)   94
   HTML Preview (DataWindow painter)   98
   Object Layout (Database painter)   85
   Objects (Database painter)   83
   Preview (DataWindow painter)   98
visual user objects   26

W

window properties, modifying   104, 146
windows
   about   22
   creating   146
   size   49
Windows CE Start menu   15
wizards
   for DataWindow creation   95
   using for development   6
workspaces
   creating   36, 88
   in PocketBuilder environment   16
   what they provide   5