SYBASE[®]

Introduction to PocketBuilder

PocketBuilder™

2.0

DOCUMENT ID: DC50059-01-0200-01

LAST REVISED: November 2004

Copyright © 2003-2004 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, the Sybase logo, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DataWindow .NET, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Impact, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SOL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, iAnywhere, iAnywhere Application Alerts, iAnywhere Mobile Delivery, iAnywhere Mobile Document Viewer, iAnywhere Mobile Inspection, iAnywhere Mobile Marketing Channel, iAnywhere Mobile Pharma, iAnywhere Mobile Sales, iAnywhere Pylon, iAnywhere Pylon Application Server, iAnywhere Pylon Conduit, iAnywhere Pylon PIM Server, iAnywhere Pylon Pro, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, M-Business Channel, M-Business Network, M-Business Server, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, My iAnywhere, My iAnywhere Media Channel, My iAnywhere Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSOL Access Module, OmniSOL Toolkit, Open Biz, Open Client, Open Client/Connect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Orchestration Studio, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, PocketBuilder, Pocket PowerBuilder, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, PowerSott, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report-Execute, Report Workbench, Resource Manager, RW-DisplayLib, RW-Library, S-Designor, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TotalFix, TradeForce, Transact-SQL, Translation Toolkit, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SOL Server, XA-Library, XA-Server and XP Server are trademarks of Svbase, Inc. 05/04

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book	vii

PART 1	WELCOME TO POCKETBUILDER
--------	--------------------------

CHAPTER 1	PocketBuilder Overview	3
	About PocketBuilder	3
	PocketBuilder features	4
	Object-oriented development	4
	DataWindow technology	5
	Workspaces and targets	5
	Wizard-driven development	6
	Small-footprint, full-featured, SQL database	6
	Advanced data synchronization	6
	SQL Anywhere Studio integration	7
	Adaptive Server Anywhere database management	7
	ASA features	8
	UltraLite features	9
	MobiLink synchronization	10
	Getting started with PocketBuilder	12
CHAPTER 2	The PocketBuilder Development Environment	13
	About PocketBuilder applications	13
	The PocketBuilder environment	16
	PocketBuilder objects	21
CHAPTER 3	About the PocketBuilder Tutorial	29
	Learning to build a PocketBuilder application	29
	How you will proceed	30
	How long it will take	30
	What you will learn	31
	Setting up for the tutorial	31

PART 2 POCKETBUILDER TUTORIAL

LESSON 1	Creating a Basic Hello World Application	35
	Create a new workspace	36
	Create a target	39
	Specify an icon for the application	47
	Add a button and text to the window	50
	Run the application on the desktop	54
	Build and deploy the application	57
	Run the application on the device	59
LESSON 2	Customizing the PocketBuilder Environment	63
	Manipulate the System Tree window	64
	Open an object	66
	Manipulate views	68
	Add an extra Script view	69
	Display view title bars	70
	Float and dock views	70
	Manipulate tabbed views	71
	Save a view layout scheme	72
	Reset the default view layout scheme	72
	Set up the toolbars	73
	Show labels on toolbar buttons	73
	Float the toolbars	74
	Reposition the toolbars	75
LESSON 3	Connecting to the Database	77
	Create a database profile for the ASA Demo DB database	79
	Connect to the ASA Demo DB database	83
	Look at table definitions in the ASA Demo DB database	84

LESSON 4	Creating an Employee List	89
	Create a workspace and target	90
	Create and preview a new DataWindow object	98
	Save the DataWindow object	. 104
	Attach the DataWindow object to a DataWindow control	. 106
	Modify the window properties	. 110
	Code the open event	. 112
	Run the application on the desktop	. 113
	Create a DSN for ASA 9	. 114
	Copy the DSN and the database to the Pocket PC	. 117
	Build and deploy the application	. 120
	Run the application on the device	. 122
LESSON 5	Creating a Sales Application	. 125
	Set up the ASA SalesDB databases	. 126
	Begin modifying the SalesDB application	. 131
	Create a DataWindow for sales order information	. 138
	Add menu items to the application menu	. 143
	Create a MobiLink connection	. 145
	Create the main application window	. 154
	Test the application on the desktop	. 160
	Deploy the application to a device	. 165
	Create a project	. 165
	Copy the remote database and log files to the Pocket PC	. 167
	If you are using an emulator	. 168
	Build and deploy the application	. 170
	Run the application	. 171
	Troubleshoot the application	. 173
Index		. 175

About This Book

Audience	This guide is for anyone who wants to build applications with PocketBuilder TM 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.
	• <i>Resource Guide</i> - 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:
	• <i>Connection Reference</i> - Describes the database parameters and preferences you use to connect to a database in PocketBuilder.
	• <i>DataWindow Reference</i> - Lists the DataWindow functions and properties and includes the syntax for accessing properties and data in DataWindow objects.
	• <i>Objects and Controls</i> - Describes the system-defined objects and their default properties, functions, and events.
	• <i>PowerScript Reference</i> - 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 Adaptive Server® Anywhere (ASA), 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 Windows CE documentation at http://msdn.microsoft.com/library/default.asp?url=/library/en- us/wceintro/html/cestart.asp. 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://pocketpb.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
informationUse the Sybase Getting Started CD, the SyBooks™ CD, and the Technical
Library 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 is downloadable 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 technical information about your product in an easy-to-use format.
- The Technical Library Product Manuals Web site is an HTML version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to the Technical Documents Web site (replacement for the Tech Info Library), the Solved Cases page, and Sybase newsgroups.

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

Sybase EBFs and software updates

* 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 <i>myCounter</i>
	• Parts of input text that must be supplied, such as <i>pklname</i> .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.

Welcome to PocketBuilder

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

CHAPTER 1 PocketBuilder Overview

About this chapter

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

Contents

Торіс	Page
About PocketBuilder	3
PocketBuilder features	4
SQL Anywhere Studio integration	7
Getting started with PocketBuilder	12

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 will help you create applications for the mobile environment or migrate them to it faster than ever before.

The target platform for PocketBuilder applications is Windows CE. The Windows CE API is a subset of the API for traditional Windows platforms. The most obvious difference between Windows CE and Windows 2000 or Windows XP 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 the first 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 devicesThe PocketBuilder IDE runs on Microsoft Windows 2000 (with Service
Pack 2) and Windows XP desktops for application development. You can
deploy PocketBuilder applications to Microsoft Windows Mobile 2002 and
2003 (First and Second Edition) devices and emulators. You can download
emulators from the Microsoft Web site. PocketBuilder supports these devices:

- Pocket PCs, both regular and Phone Edition
- Smartphones

PocketBuilder features

PocketBuilder provides a feature-rich, highly productive, easy-to-use environment for mobile application development. Its core features and capabilities are tried and true, since they are the fundamental components of PowerBuilder, Sybase's 4GL development tool.

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 Adaptive Server Anywhere (ASA) and UltraLite databases. ASA 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 "Adaptive Server 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 ASA relational database management system (RDBMS)
- The UltraLite RDBMS
- MobiLink and SQL Remote data synchronization technologies
- 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 ASA, MobiLink, and UltraLite, is available on the PocketBuilder Technical Library CD and in the SQL Anywhere Studio online Help.

For specific information about using ASA, 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*.

Adaptive Server Anywhere database management

The ASA RDBMS is the core of SQL Anywhere Studio. ASA 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.

ASA 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, ASA 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.

ASA characteristics ASA excels in all the following roles:

• A workgroup database server Workgroups ranging in size from a few people to several hundred people can use ASA 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.

ASA can employ multiple CPUs and use up to 64GB of memory. Multigigabyte ASA 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. ASA 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. ASA has demonstrated this facility in many demanding commercial applications.

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

ASA features

ASA provides all of the following features:

- **Full SQL RDBMS** ASA 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** ASA requires fewer memory and disk resources than other database management systems.
- **Easy to use** ASA is self-tuning and easy to manage. You can use ASA without the extensive database administration efforts usually associated with RDBMSs.

- Standalone and network use ASA 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 ASA is designed with simple administration and modest resource requirements in mind, it is a scalable, high-performance DBMS. ASA can run on multiple CPUs, has an advanced query optimizer, and provides performance monitoring and tuning tools.
- Industry-standard interfaces ASA provides a native ODBC 3.5 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 ASA 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 ASA 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*, and the ASA 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 ASA, Sybase Adaptive Server Enterprise, Oracle, IBM DB2, or Microsoft SQL Server.

Remote databases can be ASA or UltraLite databases. The MobiLink synchronization server runs on Windows or UNIX platforms. ASA 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.

CHAPTER 2

The PocketBuilder Development Environment

About this chapter	This chapter introduces the PocketBuilder develop which you use in the tutorial in Part 2. It also descri of a PocketBuilder application.	oment environment, bes the building blocks
Contents	Торіс	Page
	About PocketBuilder applications	13
	The PocketBuilder environment	16
	PocketBuilder objects	21
For more information	For a more detailed description of the PocketBuild environment, see the PocketBuilder <i>User's Guide</i> .	ler development

About PocketBuilder applications

What's in a PocketBuilder	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 Adaptive Server Anywhere (ASA) and UltraLite. The ODBC database driver for ASA 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 an ASA 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 (ARM)
- Pocket PC emulator
- Smartphone emulator

You can download Pocket PC and Smartphone SDKs from the Microsoft Web site. These SDKs include emulators that you can use as target platforms for your PocketBuilder applications.

An application that you deploy to the desktop looks slightly different from the same application deployed to a PDA 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 *StorageProgram 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 will be 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.



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.		
	TArialB I U ≣ ≣ ≣		
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.		
	RocketBuilder File Bun Lools Window Helt New New New New		

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:

Button	Use to
New	Create new workspace, target, component, or other object, or open a tool.
Inherit	Inherit from menu, user object, or window.
ک ت 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.
SysTree	Show or hide the System Tree window.
Output	Show or hide the Output window.
	Move to the next line in the Output window.
Frevious 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	Show or hide the Clip window.

Table 2-1: Buttons in the PowerBar

Button	Use to
É 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.
sàà I. Build	Start an incremental build of the workspace.
SốC F. Build	Start a full build of the workspace.
특별 Deploy	Deploy the workspace.
985 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.
Sebug	Debug the current target.
BB BB Sel_Debug	Select a target and debug it.
🚿 Run	Run the current target.
अहं अहं अहं अहं Sel <u>R</u> un	Select a target and run it.
∏+ Exit	Exit from PocketBuilder.

PocketBuilder objects

The basic building blocks of a PowerScript target are objects:

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

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

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:

		Line Circu	- Horizontal Tick Position-
Description Functions External DataStores Window Controls User Objects Miscellaneous DataWindows - Events & Fu User Objects DataWindows - Updating DataWindows - Retrieval	Title Display system informa System Inform Display system informa This example shows a Sales Order L This example shows a Sales Order L This Window uses an Analog Clock This example runs vari Performance I and This example is a simp Reporting With A User Object that lool Drop Down C This is a simple add/uj/Auto Incerner Shows how to cancel Cancel Retrie	Line Spec 5 4 Page Size: 5 4 Start Pgshor: 100 4 Men Postion: 10 4 Meg Position: 200 4 Silger Size: 0 4 Tick Erequency: 5 4 Showy Silder	Vertical text center C Top & Bottom C Jop C Bettom Vertical Tick Position C Left & Right C Left Bight Ngither
Dbject Communication Drag & Drop DataWindows - Events & Fu DataWindows - Retrieval	Shows how to cancel Cancel Retrie Select a department a Change Empl mA digital clock, built us Digital Clock m This example lists all et Computed Fie DataWindow that is a Referencing (Eg	

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.

	October-I	Summary December, 2002	Repor	τ		4/9/U Page 4 of
Product ID	Product Name	Product Description	Quantity Sold	Dollars	Sales Summary	
300	Tee Shirt	Tank Top	2364	\$21,276	Zipped Sweatshirt-	
301	Tee Shirt	V-neck	2388	\$33,432	Wool cap-	
302	Tee Shirt	Crew Neck	2148	\$30,072	V-neck-	
400	Baseball Cap	Cotton Cap	3278	\$29,502	Tank Top-	
401	Baseball Cap	Wool cap	2701	\$27,010	Hooded Sweatshift	
500	Visor	Cloth Visor	2652	\$18,564	Crew Neck-	
501	Visor	Plastic Visor	2508	\$17,556	Cotton Shorts-	
600	Sweatshirt	Hooded Sweatshirl	3060	\$73,440	Cotton Cap-	
601	Sweatshirt	Zipped Sweatshirt	2724	\$65,376	Cloth Visor	
700	Shorts	Cotton Shorts	4536	\$68,040	0 20000 40000 Dollars	60000 8000

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.

🖽 Data	Window	- d_edit_styles		_	
0	Dept	Name	Status	Salary	_
102	100	Whitney, Fran	 Active Terminated On Leave 	\$45,700.00	
105	100	Cobb , Matthew	 Active Terminated On Leave 	\$62,000.00	
129	200	Chin , Philip	 Active Terminated On Leave 	\$38,500.00	•

• If the data includes phone numbers, salaries, and dates, you can format the display to suit the data.

🖽 Data	Window - d	_formats		-	
0	Name	Phone	Salary	Start Date	
102	Whitney	(617) 555-3985	\$45,700.00	02/26/04	
105	Cobb	(617) 555-3840	\$62,000.00	07/02/04	
129	Chin	(404) 555-2341	\$38,500.00	08/04/04	
148	Jordan	(617) 555-7835	\$51,432.00	10/04/04	
160	Breault	(617) 555-3099	\$57,490.00	10/16/04	
•					► ►

• 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 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.

ះដ ម	ntitled		_ 🗆 ×
<u>F</u> ile	<u>D</u> ata		
	<u>U</u> pdate ►	Current Row	
	<u>D</u> elete	<u>A</u> ll Rows	
	<u>C</u> ancel		
		1	

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).

Menus

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 <i>not</i> 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 class (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.
- **Class user objects** These are reusable processing modules that have no visual component. You typically use class 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 class user object. To use a 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	Торіс	Page		
	Learning to build a PocketBuilder application	29		
	How you will proceed	30		
	Setting up for the tutorial	31		

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 Adaptive Server Anywhere and deployment to a PocketPC device
- Building a DataWindow-based sales application (using a skeleton application that is provided for you) with database connection to Adaptive Server Anywhere and deployment to a PocketPC device

How you will proceed

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

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 emulator
2	Explore the PocketBuilder environment and customize the workspace
3	Create a database profile and look at the ASA 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 Open event; run the application; build the application; and deploy
	the application to a Smartphone device
5	Set up the ASA database; 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
	1

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

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:

How to use the	10
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

Table 3-2: Features demonstrated in the PocketBuilder tutorialHow to use the \Box

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.

Connecting to a database the tutorial uses the ASA Demo DB database that installs with PocketBuilder and two ASA SalesDB databases that you set up. These are Adaptive Server Anywhere databases and require the Sybase Adaptive Server Anywhere engine. If you do not already have Adaptive Server Anywhere on your local machine or server, you must install it now. You can install it from the PocketBuilder CD. If you installed PocketBuilder in a nondefault location, you must make sure that the *odbc.ini* registry entry defining the ASA Demo DB as a data source points to the correct location of the Adaptive Server Anywhere engine.

Requirements	To test the applications you develop in the tutorial on the Pocket PC, you have either an ARM-based or an XScale-based device or a Pocket PC em Before you begin Lesson 5, "Creating a Sales Application," make sure have the PocketBuilder Virtual Machine (VM) as well as the ASA databa MobiLink Client installed on your deployment device or emulator. For information, see the PocketBuilder <i>Installation Guide</i> .			
The Tutorial directory	The tutorial also uses	s the files listed in Table 3-3:		
	Table 3-3: Files red	quired by the PocketBuilder tutorial		
	File	Contents		
	MakeDB.cmd A command file located in the Tutorial\SalesDB\db that creates and populates the remote and consolidar databases and creates the corresponding data source			
	tutorial.ico	An icon located in the Tutorial\HelloWorld directory		
	When you install Poo directory, which is a	cketBuilder, these files are installed in the <i>Tutorial</i> subdirectory of the PocketBuilder installation directory.		
	When you have finished the tutorial, you can delete the files.			
The Tutorial\Solutions directory	The <i>Tutorial</i> \Solutions directory has solutions for the HelloWorld, EmployeeList, and SalesDB lessons. 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.			

PocketBuilder Tutorial

PART 2

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 ASA demonstration database
- DataWindow-based employee application with database connection to Adaptive Server Anywhere and deployment to a PocketPC device
- DataWindow-based sales application with database connection Adaptive Server Anywhere, use of MobiLink synchronization, and deployment to a PocketPC device

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.

Ne	≌₩								x
	Workspace	Target	PB Object	DataWindov	w Database	Project 1	Fool	1	
	Worksp	асе							
] Taunahi 🗌	· · · · ·	11.5				_	or	
	rargec:	(Not Appli	cable)			T		UK	

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.

New Workspace	<u>? ×</u>
Save in: 🔂 Tutorial	- ⇐ Ҽ 📅 💷 -
EmployeeList HelloWorld SalesDB Solutions	
File name:	Open
Save as type: Workspace Files (*.pkw)	▼ Cancel

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.

Specify New Application	and Librar y
senen	Enter the names you want to use for the new application, library and target.
	basic_tutorial Library: C:\Program Files\Sybase\PocketBuilder 2.0\Tutorial\HelloWorld\b
1	Iarget: C:\Program Files\Sybase\Pocket8uilder 2.0\Tutorial\HelloWorld\b
	< Back Next > Cancel

5 Click Next until the Specify Connectivity page displays.

You accept the default library search path and window and menu names. Since this lesson does not require database connectivity, make sure None is selected.

Specify Connectivity		×
	Select the connectivity requirements for this application. You will be asked to supply specific connection information later.	
	< <u>B</u> ack Next > Cancel	

6 Click Next.

The Specify Project Object page displays.

Specify Project Object	
renarce	A project object will be built that you can use later to build and deploy your application. Enter the name for the new project.
500	Project:
a Contraction	,
10	
	< <u>B</u> ack Next> Cancel

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.

Specify Deployment Cor	nfiguration	×
anan	Specify where and how to deploy your PocketBuilder application.	
200	∐arget Directory: NProgram Files	
E COC	Build Directory:	
SCIN	Deploy After Building	
SA - C	Add Shortcut	
and	Create a CAB File for Distribution	
2		
	< <u>B</u> ack Next> (Cancel

8 Click Next.

The Specify Version Information page displays.

Specify Version Informat	ion		×
	Specify any version apply to the executa information will displa Windows Explorer. Version Information Company <u>N</u> ame: Product Name: Description: Copyright: <u>V</u> ersion:	and product identification information you wish to ble and/or dll(s) in your deployed application. This say on the Version tab of the file Properties in MyCompany basic_tutorial My Application Description 1.0.0.1	
		< Back Next > Cancel	

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.

Properties - basic_tutorial inherited from
General Today Item
AppName
basic_tutorial
DWMessageTitle
Additional Properties
Properties (Non-Visual Object List /

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.

Application				x
Text F	ont	Column Font	He	ader Font
Label	Font	Icon	Variat	le Types
Icon <u>N</u> ame	e:			
C:\Progr	am Files\Syb	ase\PocketBuilder	Browse	
				-
			1	
			5000	
	ок	Cancel	Apply	Help

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 Check 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.

C Layout	
ок	
<u> </u>	•

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.

- 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:

Script - clicked for c	b_ok returns long			
cb_ok	🗾 🖪 clicked () ret	💌 🗈 clicked () returns long [pbm_bn 💌 📃 🚍		
Access	Return Type	Event Name	_	
∎ublic ∢	▼ long	▼ clicked	•	
st_hello.t	ext = "Hello World"		<u>_</u>	
			v	
Layout Lolicked Even	t List \Function List \Declare Instance Varia	ables/	•	

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.

Hello World File Help	_ 🗆 🗙
ОК	

2 Click the OK button.

Hello World displays in the window.

SHello World File Help	<u> </u>
OK	
Hello World	

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.

4 Select Tools>System Options from the menu bar and then click the Workspaces tab.

5 Make sure the Reopen Workspace On Startup and the Reload Painters When Opening Workspace check boxes are selected.

System Options X			
General Workspaces Font			
Show start dialog at startup with no workspace Reopen workspace on startup Reload painters when opening workspace			
Recent objects list contains 8 items (limit 36)			
Recent workspaces list contains 8 items (limit 36)			
Recent connections list contains 5 items (limit 10)			
PocketBuilder creates registry entries for each workspace and target file that you open. "Clean Up" will remove the registry entries for all workspace and target files that no longer exist.			
Clean Up			
OK Cancel Apply Help			

6 Click OK.

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

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.

sí_p_t	oasic_tutorial_exe	(basic_	tutorial) - Project				_ 🗆 🗵
	Application						<u> </u>
1	Executable File Name	э:	basic_tutorial.exe	_			_
-	Resource File Name:						
	Sign the ApplicationApplication					e	
	Select Platform	to Build	1				
12	Pocket PC Device (A	ARM)				Select Platform	
—	Build Directory:	ŀ.		_			
	🔽 Delete Tempora	ry Files	After Build				_
	The temporary	files inc	lude any PKDs and the	EXE w	hich may be deploy	red to the device.	
	Deploy Target &	Optior	15				
1	Target Directory:	Progr	am Files	_			_
5	J Danlay Aftar Build						
-	Launch Application						
	Add Shortcut to "Start Menu" Deploy Ioday Item						
	CAB File Packagi	ng					
	Create CAB File	for Dist	ribution				
4							
·							
	Version						
	Company Name:	MyC	ompany				_
5	Product Name:	basic	tutorial				
	Description:	My A	- oplication Description				
	Convright:			_			
	Versien	1.0.	1				
	version:	p.0.	5.1				
Library	,			PKD	Resource File Nar	ne	Ϋ́
C:\Prog	gram Files\Sybase\Pock	ketBuild	er 2.0\Tutorial\HelloWo	Г			
1							

- 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 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.

Pocket_PC
File Zoom Tools Help
∰ Hello World 🛛 📢 3:28 🐽
OK
Hello World
File Help

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.
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.



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.



- 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

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

Set up the toolbars

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.

👯 basic_tutorial (basic_tutorial) inherited from application - App	olication
Script - open for basic_tutorial returns (None)	😰 Properties - basic_tutorial 🗖 🔀 d
basic_tutorial 💌 🗈 open (string comn 💌 📃 📃	General Today Item
<pre>//* open: Application Open Script //* 1) Opens Main window</pre>	AppName basic_tutorial
Open (w basic tutorial main)	
×	Additional Properties
Event List Function List Declare Instance Variables	Properties Non-Visual Object List
basic_tutorial	
	<u> </u>
	_
•	▶ <i> </i> /,

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]

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.

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.

PowerBar1 🛛												
New	M Inherit	C urr Open	Review	SysTree	Uutput	Next	Previous	To-Do List	F Browser	Clip	Dibrary	DB Prof
P Database	€⁄ Edit	Sel Emul	sắắ I. Build	sắắ F. Build	C.F Deploy	Skip	Stop	🛞 Debug	۵۵ ۵۵ Sel Dbg	🕉 Run	35년 영양 Sel Run	∏+ Exit

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.

LESSON 3 Connecting to the Database

This lesson shows you how to connect to the ASA 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."

In this lesson you:

- Create a database profile for the ASA Demo DB database
- Connect to the ASA Demo DB database
- Look at table definitions in the ASA Demo DB database

How long does it take? About 20 minutes.

About the ASA Demo DB 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. 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:

- Create a database profile for the ASA Demo DB database
- Connect to the ASA Demo DB database
- Look at table definitions in the ASA Demo DB database

Create a database profile for the ASA Demo DB database

Where you are

> Create a database profile for the ASA Demo DB database Connect to the ASA Demo DB database Look at table definitions in the ASA Demo DB database

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.



- Database Profiles
- 2 Select the ODB ODBC node by clicking on it.

3 Click the New button.

The Database Profile Setup dialog box displays.

Database Profile S	etup - ODBC			×
Network	Opt	ions	Preview	
Connection	System	Transactio	n Syntax	< İ
Profile Name: Connect Informat	ion			-
Data Source:				-
User ID: 🔽				
Password: 🔽				
Driver-Specific Pa	rameters:			
- Other				
Isolation Level:	(Default Driver	Behavior)		-
🗖 AutoCommit M	lode 🗔 i	Prompt for Dat	abase Informatio	on 📗
Commit on Dis	connect 🥅 🤅	Generate Trac	e	
L				
ОК	Cancel	Apply	Help	

Profile setup	Value
Profile Name	ASA Demo
Data Source	ASA 9.0 Sample
User ID	dba
Password	sql

4 Type or select the required profile information.

Here is what the profile setup should look like when you are done:

Database Profile S	etup - ODBC			<u>:</u>	×	
Network	Opt	ions		Preview	1	
Connection	System	System Transaction Syntax				
Profile Name:	ASA Demo					
Connect Informat	ion					
Data Source:	ASA 9.0 Samp	e		•		
User ID: 🔽	dba					
Password: 🔽	***					
Driver-Specific Pa	rameters:					
Other						
Isolation Level:	(Default Driver	Behavior)		•		
	node 🗖 i	Prompt for D	atahasi	e Information		
Commit on Dis	connect 🔲 (Generate Tr	ace			
ОК	Cancel	App	ply	Help		

5 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.

Database Profile Setup - ODBC	×
Connection System Transaction	Syntax
Network Options	Preview
Database Connection Syntax:	
// Profile ASA Demo SQLCA.DBMS = "ODBC" SQLCA.AutoCommit = False SQLCA.DBParm = "ConnectString="DSN=ASA 9.0 S	
	 ⊆opy
Iest	: Connection
OK Cancel Apply	Help

6 Click the Test Connection button.

A message box tells you that the connection is successful.

If the message box tells you the connection is not successful Close the message box and verify that the information on the Connection

close the message box and verify that the information on the Connection page of the Database Profile Setup dialog box is correct. Then check the configuration of the data source in the ODBC Administrator. You can run the ODBC Administrator by expanding the Utilities folder under the ODB ODBC node of the Database Profile painter and double-clicking the ODBC Administrator item.

7 Click OK to close the message box. Click OK to close the Database Profile Setup dialog box. Click Close to close the Database Profiles dialog box.

Connect to the ASA Demo DB database

Where you are

- Create a database profile for the ASA Demo DB database
- > Connect to the ASA Demo DB database Look at table definitions in the ASA Demo DB database

You use the ASA Demo DB database (asademo.db) in this lesson. The ASA Demo DB database is an Adaptive Server Anywhere database that is accessed through ODBC. In this exercise you connect to the ASA Demo DB database.



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 by clicking on the plus sign, and select ASA Demo.



3 Click the Connect button.

PocketBuilder connects to the database and the dialog box closes.

Look at table definitions in the ASA Demo DB database

Where you are

Create a database profile for the ASA Demo DB database Connect to the ASA Demo DB database

> Look at table definitions in the ASA Demo DB database

Now you look at the definitions for the Customer and Product tables in the ASA Demo DB database. This helps you become familiar with the Database painter and the tables you will use in this lesson.

What happens when you connect To look at the table definitions, you have to connect to the database. 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 the database, you automatically connect to the last database used. PocketBuilder determines which database this is by reading the registry.



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 connection.

The Objects view of the Database painter displays all existing database profiles in a tree view under the Installed Database Interfaces heading. The ASA Demo DB database 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 ASA Demo database node in the Objects view.

Notice the folders under the ASA Demo database node.



3 Expand the Tables folder.

You see the list of tables in the database.

Table names might have a prefix

The table names in the Select Tables dialog box might have a prefix such as *dba* or *dbo*. This depends on the login ID you are using. You can ignore the prefix.

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 data type 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.

In the Database painter, the Properties view is also called the Object Details view.

The title bar and tab headings for the Object Details view change dynamically depending on the current object selection. The title bar gives the object type, the database connection, and the object identifier.

The Object Details view for a column has five tabs, one for general database properties and the others for column extended attributes.

Column (ASA Demo) - customer.id	
General Headers Display Validation Edit Style	
Table:	-
customer	
Column:	
id	
- Column Details	_
Column Number:	
1	_
Data Type:	
integer	_
Nullable:	
No	
Default:	
autoincrement	
Comments:	
	<u>^</u>
1	- I - I

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.

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. Now you create an Employee list using a DataWindow.

Using the ASA Windows CE Royalty-Free Runtime

You can do this lesson using the ASA Windows CE Royalty-Free Runtime that is included in the PocketBuilder 2.0 installation. You do not need the SQL Anywhere 9 (developer version) that is also included in the PocketBuilder installation. The Royalty-Free Runtime Edition is a special version of the ASA database server. It is intended for use as a low-cost deployment option when the full functionality of ASA is not required.

In this lesson you:

- Create a workspace and target
- Create and preview a new DataWindow object
- Save the DataWindow object
- Attach the DataWindow object to a DataWindow control .
- Modify the window properties
- Code the open event
- Run the application on the desktop .
- Create a DSN for ASA 9
- Copy the DSN and the database to the Pocket PC
- Build and deploy the application
- 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
 Save the DataWindow object
 Attach the DataWindow object to a DataWindow control
 Modify the window properties
 Code the open event
 Run the application on the desktop
 Create a DSN for ASA 9
 Copy the DSN and the database to the Pocket PC
 Build and deploy the application
 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 Bight click emplise tutorial in the System Tree select New fr

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.

Specify New Application	and Library
man	Enter the names you want to use for the new application, library and target.
6613	Application Name:
	emplist_tutorial
SCI X	Library:
Thread	C:\Program Files\Sybase\PocketBuilder 2.0\Tutorial\EmployeeList\
SA	Iarget:
2	C:\Program Files\Sybase\PocketBuilder 2.0\Tutorial\EmployeeList\
	< Back Next > Cancel

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 ASA Demo database profile from the Profile Name list if it is not already selected.

Choose Database Profile			×
sector	Select the profile containing the your Database. The database pr to the database. It is used for run do not indicate an alternate runti Database <u>P</u> rofiles:	information necessary to connect to ofile is used for design time connection time connections on the desktop if yo me connection.	a L
JAL	Profile Name	DBMS Database UserID F	5,
	C ASA Demo	ODBC	-
10	Provide alternate runtime cor	nnection information]
	<	Back Next > Cancel	

13 Click Next.

Select the Script radio button to specify that the application get its connection information from script.

Specify Connectivity Sou	rce Info
manan	Specify how the application will get the connection information at runtime.
	Source of Connection Information
2 1	C Registry Settings
- Comos	
2/2	Connection Service Object
	n_emplist_tutorial_connectservice
	< <u>B</u> ack Next > Cancel

14 Click Next.

The Specify Project Object page displays.

Specify Project Object	
Anan	A project object will be built that you can use later to build and deploy your application. Enter the name for the new project.
	Project [p_basic_tutorial_exe
Strand Strand	
Yillinnessel	< <u>B</u> ack Next > Cancel

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, create an executable for desktop 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.

16 Accept the default names 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.

Specify Deployment Cor	figuration			×
ana	Specify where and ho	w to deploy your P	ocketBuilder applicati	on.
5 D	Iarget Directory:	∖Program Files		
	Build Directory: ✓ Deploy After Buil ✓ Add Shortcut	ding		
and a	☐ <u>C</u> reate a CAB Fili	e for Distribution		
		< <u>B</u> ack	Next >	Cancel

- 17 Click Next.
- 18 Accept the default version information.

19 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.

Ready to Create Application		
An Application with the following characteristics will be created or generated. Click Finish when you are ready.		iollowing characteristics will be created or when you are ready.
A CONTRACTOR	Property	Value 🔺
	Target File Application Library Application Library Library Search Path SDI Main Window SDI Main Window SDI Main Menu About Box Window Connectivity To? Database Profile Connection Source Profile Connection Source	C:\Program Files\Sybase\PocketBuilder 2.(C:\Program Files\Sybase\PocketBuilder 2.(dw.tutorial C:\Program Files\Sybase\PocketBuilder 2.(w_dw_tutorial_main m_dw_tutorial_main w_dw_tutorial_about SQL Database ASA Demo n_dw_tutorial_connectservice Script p_dw_httorial_eve
Generate Io-Do List		
		< <u>B</u> ack Finish Cancel

20 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 Save the DataWindow object Attach the DataWindow object to a DataWindow control Modify the window properties Code the open event Run the application on the desktop Create a DSN for ASA 9 Copy the DSN and the database to the Pocket PC Build and deploy the application 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. Click OK.

Ne	w								x
	Workspace	Target	PB Object [DataWindow	Database	Project 1	rool		
	ЕL: — НК: — КЕ: —]	K)]		
	Freefo	rm	Graph		Grid	Group	,	Tabular	
							-		
٦	arget:	emplist_tu	torial (C:\Prog	gram Files∖Syl	oase\PocketB	uilder 💌	ОК	Ca	ncel

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 ASA demo database.

III Select - asademo9.0DBC.asademo.dba	
S Table Layout	
Selection List:	
	_
Select Tables	_
contact Open customer department employee New fin_code Cancel product sales_order sales_order_items Help	
Show system tables	

4 Select the employee table. 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 and the emp_lname columns.

6 Click the Return button (circled below) or select File>Return to DataWindow Painter from the menu bar.





Select Color and Border Settings							
	Select color and bor	der options for your datawindow:					
	Background Color:	Window Background					
	Text						
	<u>C</u> olor:	Window Text					
	Bo <u>r</u> der:	NoBorder 💌					
		·					
	Columns						
	Color:	Window Text					
	Bor <u>d</u> er:	NoBorder 💌					
		☐ <u>S</u> ave as default					
		< Back Next > Cancel					

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.

A circl DataWindow with the following characteristics will be created energies and the second		
Column Border	NoBorder ▶ o List < <u>Back</u> FinishCancel	

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 Select the DataWindow 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 for deployment to a Pocket PC.

When you are done, the columns should look something like this:

🖽 (Untitled) * (e	mplist_tutorial) - Dat	aWindow					
Emp Fname	Emp Lname					General	Print Specifications
Header †							' '
emp_fname emp_lname						Units DewerBuilder (0)	
Detail †	1					Powerbu	liider (U)
						Timer Int	erval:
Summary†						0	
						Color	
					<u>-</u>	- Wi	ndow Background
└				<u> </u>		,	
Emp Fname	Emp Lname					- Grid —	
Fran	Whitney					Display	
Matthew	Cobb				-1	On (U)	an Mouing
Philip	Chin					Mous	e Selection
Julie	Jordan					Row I	Resize
Robert	Breault						
Melissa	Espinoza						
	Rows	1 to 9 of 75					11

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 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. Click the Font tab and select Bold.
- 11 Click the Emp Lname header. In the Properties view, change the Text property to Last Name. Change the Alignment property to Left. Click the Font tab and select Bold.

Save the DataWindow object

Where you are

Create a workspace and target Create and preview a new DataWindow object > Save the DataWindow object Attach the DataWindow object to a DataWindow control Modify the window properties Code the open event Run the application on the desktop Create a DSN for ASA 9 Copy the DSN and the database to the Pocket PC Build and deploy the application Run the application on the device

Now you name the DataWindow object and save it in the *emplist_tutorial.pkl* library.

Saving to another library

You can save objects to different application libraries, but to avoid complications, you save all your new objects in one library. You can also copy or move objects from one library to another using the Library painter.

1 Select File>Save from the menu bar.

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

2 Type d_sqlgr_employee in the DataWindows box.

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

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

5ave DataWindow	×					
DataWindows:	ОК					
d_sqlgr_employee						
	Cancel					
	Help					
Comments:						
This DataWindow object retrieves employee first names and last names.						
Application Libraries:						
C:\Program Files\Sybase\PocketBuilder 2.0\Tutoria	al\EmployeeL					
•	•					

This DataWindow object retrieves employee first names and last names.

4 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 Save the DataWindow object > Attach the DataWindow object to a DataWindow control Modify the window properties Code the open event Run the application on the desktop Create a DSN for ASA 9 Copy the DSN and the database to the Pocket PC Build and deploy the application 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 Select Insert>Control>DataWindow from the menu bar. Click the main window in the Layout view.

🖀 w_emplist_tutorial_	main * (emplist_tutorial) inherited from window - Window 💶 🗖 🗡
🥥 Layout	🔲 🗙 😡 Properties - dw_1 inherited from datawindow 🛛 🗙
	General Other
	dw_1
	DataObject
	Tag
•	
	Visible ↓ Epshlad
	□ H5crolBar
	↓ LiveScroll
	F HSplitScroll
	🔽 Border
	BorderStyle
4	→ StyleBox!
 Layout (open) Event List) 	Function List Declare II + Properties Control List Non-Visual Object List

A DataWindow control displays.

4 In the Properties view, select HScrollBar and VScrollBar to enable you to scroll through the list of employees when you run the application.

Now you need to associate the d_sqlgr_employee DataWindow object with the DataWindow control in the window.

- 🔳 w_emplist_tutorial_main * (emplist_tutorial) inh - 🗆 🗵 C Layout 🔲 🗙 😰 Properties - dw_1 inher 🔳 🗙 General Other Name dw_1 DataObject Select Object x DataWindows: OK d_sqlgr_employee Cancel d_sqlqr_employee Browse Help Comments: This DataWindow object retrieves employee first names * • and last names. -Application Libraries: 💼 C:\Program Files\Sybase\PocketBuilder 2.0\Tutorial\Employ F Layout open Event List Fun IList<u>)</u> N →
- 5 In the Properties view, click the button next to the DataObject text box. 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:

Last Name			
Luot Humo			

Modify the window properties

Where you are

Create a workspace and target Create and preview a new DataWindow object Save the DataWindow object Attach the DataWindow object to a DataWindow control > Modify the window properties Code the open event Run the application on the desktop Create a DSN for ASA 9 Copy the DSN and the database to the Pocket PC Build and deploy the application 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 the main window, not the DataWindow control. In the main window's Properties view, change the title to Employees. Select the Close (OK) checkbox. Select the Show SIP Button checkbox.

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) in the Pocket PC will be available to the user.



2 In the main window's Properties view, click the Scroll tab page. Select the HScrollBar checkbox and the VScrollBar checkbox.

You add scrollbars to the window to give users a way to scroll to see data that is not visible in the window.

🖀 w_emplist_tutoria	l_main * (emplist_	tutorial) inherited from window - Window 📃 🗆 🗙
🖉 Layout			💯 Properties - w_emplist_tutorial_main_inherite
		1 🔺	General Scroll Other
Emp Fname	Emp Lname		✓ HScrollBar ✓ VScrollBar UnitsPerLine 0 0 ▲ UnitsPerColumn ● 0 ▲ ColumnsPerPage ● 0 ▲ LinesPerPage ● 0 ▲
Layout open Event Li	st∑Function List∑Declar	e Instar ►	

Code the open event

Where you are

Create a workspace and target Create and preview a new DataWindow object Save the DataWindow object Attach the DataWindow object to a DataWindow control Modify the window properties > Code the open event Run the application on the desktop Create a DSN for ASA 9 Copy the DSN and the database to the Pocket PC Build and deploy the application Run the application on the device

Next you code the open event for the w_emplist_tutorial_main window.

- 1 Double-click the w_emplist_tutorial_main window (not the DataWindow control) in the Layout view.
- 2 Type the following code in the Script view:

dw 1.SetTransObject(SQLCA)

```
dw_1.Retrieve()
```



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

The code for the Retrieve function for dw_1 makes the DataWindow object associated with dw_1 retrieve the data from the database.

3 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 Save the DataWindow object Attach the DataWindow object to a DataWindow control Modify the window properties Code the open event > Run the application on the desktop Create a DSN for ASA 9 Copy the DSN and the database to the Pocket PC Build and deploy the application 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.

2 Click Yes.

The application begins running, connects to the database, and displays.



3 Close the application. Close the Window painter.

Create a DSN for ASA 9

Where you are

Create a workspace and target Create and preview a new DataWindow object Save the DataWindow object Attach the DataWindow object to a DataWindow control Modify the window properties Code the open event Run the application on the desktop > Create a DSN for ASA 9 Copy the DSN and the database to the Pocket PC Build and deploy the application 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, depending on the type of DSN, the ID and password of the user.

You might be able to omit this section if you are using SQL Anywhere 9 If you are using the Royalty-Free Runtime, you do need to create the DSN as described in this part of the lesson. If you are using the SQL Anywhere 9 (developer version) however, a DSN for ASA 9 will already be on the device if ASA is installed with the sample data option selected. If so, you can omit this section and simply open the DSN file and verify that the paths used are valid. With the Royalty-Free Runtime, the start line in the DSN points to rteng9.exe. With ASA 9, the start line points to dbsrv9.exe. 1 If you are using the SQL Anywhere 9 (developer version), you do not need to create a DSN, so you can skip to "Build and deploy the application" on page 120.

Note that a DSN for ASA 9 that looks like the following will already be on the device when ASA is installed with the sample data option selected.

🔊 ASA 9.0 Sample.dsn - Notepad 📃 🗖	×
File Edit Format Help	
[ODBC] uid=dba pwd=sql enginename=asademo databasename=asademo databasefile=\Program Files\Sybase\ASA9\asademo.db start=\Program Files\Sybase\ASA9\dbsrv9.exe	4

2 If you are using the Royalty-Free Runtime, open Notepad or WordPad and create a DSN file with the following content:

```
[ODBC]
uid=dba
pwd=sql
enginename=ASADemo
databasename=ASADemo
databasefile=\Program Files\Sybase\ASA\asademo.db
start=\Program Files\Sybase\ASA\rteng9.exe -q
```

What -q means

Notice that in this DSN, the start line ends with a -q that means *quiet*. If you use the -q, when you run the application on the Pocket PC, ASA starts without displaying a log screen and your application starts immediately.

3 Save the file as ASA 9.0 Sample.dsn in a temporary directory.

🖉 ASA 9.0 Sample.dsn - Notepad	×
File Edit Format Help	
[ODBC] uid=dba pwd=sql enginename=ASADemo databasename=ASADemo databasefile=\Program Files\Sybase\ASA\asademo.db start=\Program Files\Sybase\ASA\rteng9.exe -q	

Copy the DSN and the database to the Pocket PC

Where you are

Create a workspace and target Create and preview a new DataWindow object Save the DataWindow object Attach the DataWindow object to a DataWindow control Modify the window properties Code the open event Run the application on the desktop Create a DSN for ASA 9 > Copy the DSN and the database to the Pocket PC Build and deploy the application

Now you must copy the DSN to the root directory of the Pocket PC device and the asademo.db database to the \Program Files\Sybase\ASA directory of the device.

1 If your Pocket PC is not on, turn it on.

Run the application on the device

2 Open ActiveSync.

ActiveSync displays on the desktop and connects to your Pocket PC.



3 In ActiveSync, click Explore.

A view of files in the Pocket PC displays.

📱 Mobile Device				
File Edit View Fa	vorites Tools Help			
] 🗢 Back 🔹 🔿 🔹 🔁	🛛 🖓 Search 🛛 🔁 Folders	3 8 4 ×	n 🗉	• *= *=
Address 🔋 Mobile Devic	e			▼ 🖗 Go
Links 💣 FYI Benefits 🛛	🖹 Help! 🖉 My Sybase	@Open Call Track	PC Ser	rvices »
Search -		Google 🔻 Yał	noo! 🝷	More ¥
Name 🛆	Size Type	Modified		
🕘 My Pocket PC	System Folder			
Business	File Folder			
Personal	File Folder			
i Templates	File Folder			
]				
4 object(s)		🖉 Mo	obile Device	1.

4 Double-click MyPocketPC.

This brings you to the root directory $(\)$.

5 Using the Windows Explorer on the desktop, copy the ASA 9.0 Sample.dsn from the temporary directory where you saved it and paste it in the root directory in ActiveSync.

I \			<u>_ ×</u>
File Edit View Favo	orites Tool	s Help	(
	Q Search	Folders 🎯 🎦	12×10 画•指袖
Address [\			▼ 🔗 Go
🛛 Links 💣 FYI Benefits 🖉	Help! 🧉	My Sybase 🛛 🙆 Open C	Call Track @PC Services *
Name 🛆	Size	Туре	Modified
Databases		System Folder	
iPAQ File Store		File Folder	
My Documents		File Folder	
Drogram Files		File Folder	
🛅 Temp		File Folder	
🗀 Windows		File Folder	
ASA 9.0 Sample.dsn	184 bytes	Microsoft OLE DB Pr	9/17/2004 2:28:36
📾 CMMapG	60 bytes	File	5/20/2002 4:00:08
CMMapP	56 bytes	File	5/20/2002 4:00:08
GCounterFile.mmf	10.2KB	MMF File	5/20/2002 4:02:15
mdmlog7.txt	54 bytes	Text Document	9/27/2004 11:59:1
1 object(s) selected			Mobile Device

6 Now using Windows Explorer on the desktop again, copy the asademo.db from the \Program Files\Sybase\SQL Anywhere 9 directory on your desktop to the \Program Files\Sybase\ASA directory of the device.

📾 \Program Files\Syl	base\ASA				_ 🗆 ×
File Edit View I	Favorites	Tools Help			1
] 🕁 Back 🔹 ⇒ → 🖻] Q Sea	arch 🕒 Folders	3 4 4	Xn	∎• *= *
Address 🔄 \Program	Files\Sybase	ASA			▼ 🖓 Go
Links EFYI Benefits	Help!	My Sybase	Open Call Tr	ack 🕘 PC Se	ervices »
Search -			Google 🔻	Yahoo! 🔹	More
Name 🛆	Size	Туре	Modified		
ШLib		File Folder			
🛤 asademo.db	1.81MB	DB File	9/17/200	04 2:32:33	
🔊 rteng9.exe	2.87MB	Application	10/16/20	003 6:16:2	
3 object(s)				Mobile Devic	e //,

Now you are ready to build and deploy the application.

Build and deploy the application

Where you are

Create a workspace and target Create and preview a new DataWindow object Save the DataWindow object Attach the DataWindow object to a DataWindow control Modify the window properties Code the open event Run the application on the desktop Create a DSN for ASA 9 Copy the DSN and the database to the Pocket PC > 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 workspace and target" task beginning on page 90 of this lesson.

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

The Project painter opens.

Executable File N Resource File Nar	ame: emplist_tutorial.exe			_
Sign the App	lication		Application Certificat	e
Pocket PC Devic	m to Build e (ARM)		Select Platform	1
Build Directory: Delete Temp The tempor	, orary Files After Build ary files include any PKDs and the	e EXE which may be depli	oyed to the device.	
Deploy Targe	& Options			
Target Directory:	Program Files			
🔽 Deploy After	Build			
Launch Appl	cation			
Add Shortcu Deploy Toda	: to "Start Menu" v Item			
	/			
CAB File Pack	aipa			
CAB File Pack	aging File for Distribution			
CAB File Pack	nging File for Distribution			
CAB File Pack	nging For Distribution			
CAB File Pack.	File for Distribution			
CAB File Pack. Create GAB Version Company Name: Product Name:	ging File for Distribution MyCompany emplist_tutorial			
CAB File Pack. CAB File Pack. Coreate CAB Version Company Name: Product Name: Description:	rile for Distribution %/Company emplist_tutorial %/ Application Description			
CAB File Pack. CAB File Pack. Create CAB Version Company Name: Product Name: Description: Copyright:	ging File for Distribution MyCompany emplist_tutorial My Application Description			
Capari, Dole CAB File Pack. Create CAB Version Company Name: Product Name: Description: Copyright: Yersion:	eging File for Distribution MyCompany emplist_tutorial My Application Description 1.0.0.1			
Create GAB Create GAB Version Company Name: Product Name: Description: Copyright: Yersion:	ile for Distribution MyCompany emplist_tutorial MyApplication Description [1.0.0.1	PKD Resource File N	ane	

The build platform is PocketPC Device (ARM) specified by the wizard.

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 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 on the device

Where you are

Create a workspace and target Create and preview a new DataWindow object Save the DataWindow object Attach the DataWindow object to a DataWindow control Modify the window properties Code the open event Run the application on the desktop Create a DSN for ASA 9 Copy the DSN and the database to the Pocket PC 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 emplist_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 emplist_tutorial.

If the start line of your DSN ends with a -q, ASA starts without displaying the log screen and then the Employees application starts.

If the start line of your DSN does not include -q, ASA starts and displays the log screen and then the Employees application starts.

Pocket_PC		. 🗆 ×
File Zoom Tools	Help	35 o k
First Name	Last Name	
Fran	Whitney	=
Matthew	Cobb	
Philip	Chin	
Julie	Jordan	111
Robert	Breault	1
Melissa	Espinoza	1
Jeannette	Bertrand	1
Marc	Dill	\Box
		•
ile Help		

- 3 Scroll through the employee list using the DataWindow scroll bars.
- 4 Tap the circular *ok* button (the one under the menu at the far right).

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

5 Tap the Close button.

You return to the Pocket PC Start menu.

Creating a Sales Application

In this lesson, you complete a skeleton sales application this is provided for you. The application uses MobiLink synchronization to synchronize the data in the Pocket PC and server databases. The databases are Adaptive Server Anywhere (ASA) 9 databases.

Using the SQL Anywhere 9 (developer version)

This lesson includes synchronization, so you must use the ASA 9 (developer version) included in the PocketBuilder installation. In the ASA setup program (that starts after the PocketBuilder setup completes), select the check box to install ASA for Windows CE if it is not already selected. If you did not install ASA initially to a Pocket PC device, you can rerun the ASA for Windows CE Setup program separately by running the setup.exe file in the SQL Anywhere 9\ce\install directory.

In this lesson, you:

- Set up the ASA 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 ASA SalesDB databases

Where you are

> Set up the ASA 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 ASA databases.

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

2 Run MakeDB.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.

Use ASA 9

The MakeDB.cmd file automatically detects the version of ASA running on your machine. If you have old versions of ASA installed, it uses the most recent version. Make sure the new version is at the beginning of the system path. You should use ASA 9 for this lesson. If you just installed SQL Anywhere Studio and the 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>Sybase>SQL Anywhere 9> Sybase Central.

- 4 Select Tools>Connect and then select Adaptive Server Anywhere 9 and click OK, or right-click Adaptive Server Anywhere 9 and click Connect in the popup menu.
- 5 Select the ODBC Data Source name radio button. Click Browse and select SalesDB, and then click OK.

Connect
Identification Database Advanced
The following values are used to identify yourself to the database User ID: Password: You can use default connection values stored in a profile None None
ODBC Data Source name
SalesDB Browse
C ODBC Data Source file
OK Cancel Help

6 Click the Advanced tab and make sure the iAnywhere JDBC driver radio button is selected. Click OK. 7 Verify that the SalesDB tables are created and sample data is provided by opening the Tables node, selecting a table, and clicking the Data tab.

Sybase Central						-03
Customer (DBA)	⇒ f	리 콜	9 B P	X 🖻 🛍 🗙 🗠		
	-		1	1		
Olders X	Linic	olumns	Foreign K	eys Refere Constraints Indexes	Triggers	Data
Sybase Central	- Onic	de consi		Consciance Indexes	mggors	L
W Adaptive Server Anywhere 9		_cust_id	cust_name	last_modified		
⊟–aggs ballesUb	1	2000	Apple St. Builders	2004-10-21 10:10:10.355		4
E- SalesDB - DBA	2	2001	Art's Renovations	2004-10-21 10:10:10.355		
Tables	3	2002	Awnings R Us	2004-10-21 10:10:10.355		
Customer (DBA)	4	2003	Al's Interior Design	2004-10-21 10:10:10.355		
CustomerIDPool (DBA)	5	2004	Alpha Hardware	2004-10-21 10:10:10.355		
- EmpCust (DBA)	6	2005	Ace Properties	2004-10-21 10:10:10.355		
Employee (DBA)	7	2006	A1 Contracting	2004-10-21 10:10:10.355		
- IdentifyEmployee_nosyr	8	2007	Archibald Inc.	2004-10-21 10:10:10.355		
NewOrders (DBA)	9	2008	Acme Construction	2004-10-21 10:10:10.355		
- OldOrders (DBA)	10	2009	ABCXYZ Inc.	2004-10-21 10:10:10.355		
- OrderIDPool (DBA)	11	2010	Buy It Co.	2004-10-21 10:10:10.355		
- Orders (DBA)	12	2011	Bill's Cages	2004-10-21 10:10:10.355		
Product (DBA)	13	2012	Build-It Co.	2004-10-21 10:10:10.355		
+- Views	14	2013	Bass Interiors	2004-10-21 10:10:10.355		
庄 📄 Indexes	15	2014	Burger Franchise	2004-10-21 10:10:10.355		
+- 📄 Triggers	16	2015	Bia City Builders	2004-10-21 10:10:10.355		
🕀 📄 System Triggers	17	2016	Bob's Renovations	2004-10-21 10:10:10.355		
Procedures & Functions	18	2017	Basements R LIs	2004-10-21 10:10:10.355		
Events	19	2018	BB Interior Design	2004-10-21 10:10:10.355		
- Domains	20	2019	Bond Hardware	2004-10-21 10:10:10.355		
- Users & Groups	21	2020	Cat Properties	2004-10-21 10:10:10.355		
- 🗋 Integrated Logins 🚽	22	2020	C & C Contracting	2004-10-21 10:10:10:355		
	23	2021	Closey Inc.	2004 10 21 10:10:10:00		-

8 Repeat steps 4-7 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 contains the synchronization scripts that are used by MobiLink. Now you verify that the scripts are created correctly.

9 In Sybase Central, select Tools>Connect. Select MobiLink Synchronization 9 and click OK.

New Connection
Choose a plug-in for the new connection:
Adaptive Server Anywhere 9
MobiLink Synchronization 9
I
OK Cancel Help

- 10 Select the ODBC Data Source name radio button and select SalesDB. Select the Advanced tab and make sure the iAnywhere JDBC driver radio button is selected. Click OK.
- 11 Expand the Synchronized Tables node and select Customer.



12 Double-click the salesdb in the Version column next to the download_cursor event.

The script should read: SELECT cust_id, cust_name FROM Customer WHERE last_modified > ?

This query downloads data that has changed since the last synchronization from the consolidated database to the remote database.

For the upload stream, there are three events: upload_insert, upload_update, and upload_delete.

13 Double-click the salesdb in the Version column next to upload_insert.

The script should read: INSERT INTO Customer (cust_id, cust_name) VALUES (?,?)

This query inserts any new customers from the remote database into the consolidated database.

14 Repeat the previous step for the upload_update event. Repeat it again for the upload_delete event. View the related SQL synchronization scripts.

How orders are handled

The synchronization scripts include a call to a stored procedure that handles orders that have been modified in the remote database. Every approved order is removed from the remote database upon synchronization.

Exploring how these scripts are made is beyond the scope of this lesson. For more information about synchronization scripts, see the MobiLink documentation.

15 Close Sybase Central.

Begin modifying the SalesDB application

Where you are

Set up the ASA 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 and then click New. Type SalesDB_remote for the Profile Name. Select SalesDB_remote for the Data Source. 3 Uncheck User ID and Password (since they are provided by the DSN file on the Pocket PC device).

Click OK to accept all other defaults for the SaleDB_remote database profile.

Database Profile S	etup - ODBC		×
Network	Opt	ions	Preview
Connection	System	Transaction	1 Syntax
Profile Name:	SalesDB_remo	te	
Connect Information	tion		
Data Source:	SalesDB_remo	te	▼
User ID:			
Password:			
Driver-Specific Pa	rameters:		
Other			
Isolation Level:	(Default Driver	Behavior)	-
Commit on Dis	Aode 🔲 i sconnect 🔲	Prompt for Data Generate Trace	abase Information
ОК	Cancel	Apply	Help

4 In the Database Profiles painter, expand ODB ODBC. Select SalesDB_remote and click Connect.

The ASA 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.

5 Select File>New. In the Workspace tab, select Workspace and click OK.

Ne	W						x
Ne	Workspace	Target PB Object	DataWindow	Database Pro	oject Tool	1	
-	Farget: (Not Applicable)			_	ОК	Cancel

6 Save the new workspace as salesdb_tutorial in the \PocketBuilder 2.0 \Tutorial\SalesDB directory.

New Workspa	ce		? ×
Save in: 🔂	SalesDB	- t t	∷
₫ф			
File name:	salesdb_tutorial		Save
Save as type:	Workspace Files (*.pkw)	•	Cancel

7 Select File>New and then in the Target tab, select Existing Application and click OK.

Workspace	Target	PB Object D	ataWindow Da	atabase Project			
Applicat	ion	PocketPC Tempk Application	ate SmartPh Template	one Existing	Application	Import Deskto CE	p to

8 Locate and expand salesdb_tutorial.pkl in the \Tutorial\Salesdb directory.



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

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

11 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 a new f_conn function.

- Select (None) at the bottom of the drop-down list for the return type. Type f_conn for the function name. Leave Pass By set to value and Argument Type as integer. Do not include an argument name or a Throws statement.
- 13 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 ASA 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:

🙀 f_conn () * (salesdb_t	utorial) - Function		_ 🗆 ×
Script - f_conn () returns (non	e)		
f_conn	•		
Access	Return Type	Function Name	-
public	(None)	▼ f_conn	
Pass By	Argument Type	Argument Name	
1			
sqlca.dbms='odb'			<u></u>
// establish DB connect using sq	connection lca;		ı ت
			•

- 14 Select File>Save to save the function. Click OK. Close the Function painter.
- 15 Close the Function painter.

16 Expand salesdb_tutorial.pkl in the System Tree in the file.

You can see the new 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 ASA 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. Now you sort the result by order_id.

5 Click the Sort tab (at the bottom) 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 checked.

If you do not see the tabs at the bottom

Select View>Layouts>(Default) from the menu bar.

III Select - ODBC.SalesDB_remote.dba		
🕼 Table Layout		
Selection List:	_id cust_name prod_name quant price disc status notes	
customer Type cust id integer cust_name varchar(30)	orders Type order Idinteger cust_Id Integer prod_id Integer emp Id Integer disc Integer quant Integer quant Integer notes varchar(50) status varchar(20)	
•	• •	
Sort		
Drag and drop columns in the order	r that you want them	
"orders"."order_id"	"orders"."order_id"	
"customer"."cust_name"		
"product"."prod_name"		
"orders", "quant"		
"product"."price"		
"orders"."disc"		
"orders"."status"		
"orders"."notes"		
1		
Sort Where Group Having Compute	⟩∫Syntax/	

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. Now we manipulate the location 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. 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. Press the Delete key.

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

10 Select the *Cust Name:* label and in the Properties view, change the Text property at the bottom to *Customer:*.

Properties - customer_cust_na		
General Position Font		
Name		
customer_cust_name_t		
Tag		
🔽 Visible 💼		
Border		
NoBorder (0) 🗾 🝙		
Alignment		
■ Right (1)		
Text		
Customer:		
<u>.</u>		

11 Also 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.

🖽 d_orders * (salesdb_tutorial) - DataWindow		
😰 Design - d_orders		
Header †	▲	
Customer:	customer_cust_name	
Product:	product_prod_name	
Quantity:	orders_quan	
Price:	product_pric	
Discount:	orders_disc	
Status:	orders_status	
Notes:	orders_notes	
	_	
•		

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 Format>Align from the menu bar, and you can equalize the space between controls by selecting them with Ctrl/click and then selecting Format>Space from the menu bar. For information, see the *User's Guide* in the PocketBuilder online Help.

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 ASA 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 tree view of the menu, 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.

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:



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 ASA 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.

Desktop Deployment Connection			
	To simplify the use of this wizard, please connect to your ASA remote database using a pre-defined database profile. This profile will also be used to test the MobiLink synchronization using the desktop deployment feature. PocketBuilder database profile SalesDB_remote Proceed without database connection		
Disconnected	< <u>B</u> ack Next > Cancel		

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.

Desktop Deployment Conn	ection
	To simplify the use of this wizard, please connect to your ASA remote database using a pre-defined database profile. This profile will also be used to test the MobiLink synchronization using the desktop deployment feature. PocketBuilder database profile SalesDB_remote Proceed without database connection Test Connection
Disconnected	< <u>B</u> ack Next > Cancel

6 Click Next to display the wizard page that shows the publications in the ASA remote database.

Select the salesapi publication. Click Next.

MobiLink Client Publicati	on (SalesDB_remo	te)	
	Choose the MobiLink	. Publication(s).	
	Publication Name:	salesapi	
·		< <u>B</u> ack Next >	Cancel

- 7 Click Next to accept the default names for the generated MobiLink objects.
- 8 In the Mobilink Client Display Options window, keep the default option so the wizard will generate a PocketBuilder window to show synchronization status.

9 Click Next.

The Optional Runtime Configuration Objects wizard lets you specify objects that allow users to change synchronization settings at runtime. For end-users, you should usually disable this option or deploy a minimal subset of the generated windows.

Optional Runtime Configuration Objects		
	If you wish to generate objects that will prompt the user for a password and other runtime changes, name these objects below.	
	Prompt user for password and runtime changes Configure function gf_salesdb_tutorial_configure_sync	
	Options window w_salesdb_tutorial_sync_options Parm structure s_salesdb_tutorial_sync_parms	
	C Do not allow runtime overrides to the synchronization	
	< <u>B</u> ack Next > Cancel	

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. 10 To demonstrate the full functionality of the wizard, click Next to accept the default selections and generate the synchronization options objects.

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.

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

The MobiLink Client Logging Options page displays.

12 Check the Show all except -vc and -vp (-v+) check box.

MobiLink Client Logging	Dptions	
	Choose logging options below:	
	Show all except -vc and -vp (-v+)	
	Show connect string (-vc)	
	Show upload/download row counts (-vn)	
	☞ Show command line and extended options (-vo)	
Alter States	🔲 Show MobiLink password (-vp)	
	🔽 Show upload/dowload row values (-vr)	
	🔽 Show hook script information (-vs)	
	🔽 Show upload stream information (-vu)	
	Output log file	
	File Name:	
	< <u>B</u> ack Next > Cancel	

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.

13 Click Next.

The MobiLink Client Additional Options page allows you to include additional MobiLink command line options or extended options and assign the MobiLink host and port.

Table-locking in ASA 9 significantly reduces data transfer times, so for your application to function properly, you set the LockTable (lt) options to *share*.

14 In the Extended Options text box, type the value *lt*=share.

MobiLink Client Addition	al Options		
	Please enter any additiona	al MobiLink client options.	
	Additional Command Line	Options:	Usage
2	Extended Options in the fo It=share	orm: name=value;name=value;	Usage
	MobiLink Server Host		8
	MobiLink Server Port		
		< <u>B</u> ack Next >	Cancel

15 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.

16 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.

17 Double-click m_salesdb in the System Tree to open the Menu painter. In the tree view of the menu, expand the File menu and double-click the Sync Options... menu item. 18 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.

19 Type the following code in script view:



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.

- 20 In the Menu painter's 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.
- 21 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)
```



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

Create the main application window

Where you are

Set up the ASA 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.



Window property	Value
Title	SalesDB tutorial
MenuName	m_salesdb
Show SIP Button	checked
Close (OK)	checked
HScrollBar	checked
VScrollBar	checked

2 Change the window properties to these values:

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 Select File>Save to save the window. Type w_orders as the name of the window. Click OK.
- 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.
- 5 Change the DataWindow control properties to these values:

DataWindow control property	Value
Name	dw_orders
DataObject	d_orders
HScrollBar	checked
VScrollBar	checked
HScrollBar VScrollBar	checked checked

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

6 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.</p> 7 Repeat the preceding step with the following settings:

Command button	
name	Value
cb_next	Next >
cb_approve	Approve
cb_deny	Deny

8 Rearrange the position of the DataWindow control and buttons so the resulting window looks like this:

S Layout		(
Customer		-
Draduat:		
Product.		
Quanity:		
Price:		
Discount:		
Status:		
Notes:	-	
•		
< Previous	Next >	
Approve	Deny	
•	······································	

Next you associate events with the buttons.

9 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.

10 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)

11 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.

12 In the System Tree, double-click the salesdb_tutorial application. In the top drop-down for 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 // 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 */. You uncomment the code by deleting the /* and the */ lines that surround the code. You leave the comments that start with // unchanged.

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



- 13 In the top drop-down, 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 // unchanged.) Select File>Save to save the changes. Close the Application painter.
- 14 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 // unchanged.) Select File>Save to save the changes. Close the Menu painter.

- 15 In the System Tree, double-click the f_scroll function. Uncomment the section following the text: Uncomment the following section after creating w_orders. Select File -> Save to save the changes. Close the Function painter.
- 16 Repeat step 15 for f_scroll_last, f_approve_deny, f_refresh_orders and f_set_dir_btn_enabled.
- 17 Take some time to examine the code in the functions that are provided for you.

You will find many embedded SQL calls and MobiLink related operations.

The SalesDB application is ready to be tested.

Test the application on the desktop

Where you are

Set up the ASA 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.

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.

- 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 ODBC connection string, type DSN=SalesDB.

MobiLink Synchronization Server Options		
Required information		
MobiLink version: Ada	ptive Server Anywhere 9.0 💌	
ODBC connection string: DSN	I=SalesDB	
Diagnostic information		
Log messages to a file		
Log file:	lbmlsrv.txt	
Display all messages in	server console	
Verbose level:		
C None 🖲 Minimal C	Full C Custom:	
Other options		
Automatic script generation	(requires SendColumnName)	
Automatic addition of users		
Other command line options:	Usage	
ОК	Cancel Help	

4 Click OK.

The MobiLink Synchronization Server runs and is ready to accept requests.

[
I. 10/22 11:29:56. (Main:: DDBC DBMS Version: 09.00.0001 I. 10/22 11:29:56. (Main:: DDBC DBMS Driver Version: 09.00.0001 I. 10/22 11:29:56. (Main:: DDBC Version supported by the driver: 3.52 I. 10/22 11:29:56. (Main:: Collation sequence of the consolidated database is 'cp1252' I. 10/22 11:29:56. (Main:: Collation set to: Read Committed I. 10/22 11:29:56. (Main:: Collation set to: Ready to handle requests I. 10/22 11:29:56. (thread 1.2): Ready to handle requests I. 10/22 11:29:56. (thread 1.3): Ready to handle requests I. 10/22 11:29:56. (thread 1.4): Ready to handle requests I. 10/22 11:29:56. (thread 1.5): Ready to handle requests
Shutdown

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.

These values will be used to generate command line arguments for the MobiLink client.	
Publication(s): salesapi	
MLUser: 50	
MLPassword:	
Remember password	
Subscriptions ML Server Settings	
OK Cancel	

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.

8 Click OK to synchronize.

First the synchronization completes.

Begin download. End download. Insert/Update rows Delete rows = 0 Disconnect MobiLi End Synchronizati Synchronization co	= 91 nk server. on. Status = 0 ompleted. ▼
OK	Cancel

Then the application runs.

18 B	5alesDB tut	orial			×
File	Order G	o Help			
	Customer	: Apple	St. Build	ders 📤	
	Product	4x8 E)rywall x1	00	
	Quanity		25000		
	Price		400		
	Discount		20	_	
	Status	:		_	
4				Ľ	1
	< Previous		Next	>	
	Approve		Deny		
•					Ľ

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.

9 Explore the layout of the application. Click the Order menu item and add a new order.

Add New Order
Customer:
Product:
·
Quantity: Price:
Discount (%):
Ok Cancel

The application appropriately handles orders that have been modified in the remote database. Every order that has been approved is removed from the remote database when synchronization occurs so that storage space on the device is conserved.

10 Click the Next, Approve, and Deny buttons.

Here the Approve button is clicked and the Good Work note is selected.

📲 SalesDB tutor	ial 📃 🗶
File Order Go	Help
Customer: /	Note:
Product: •	Good work!
Quanity:	
Price:	Ok Cancel
Discount:	
Status:	_
4	
< Previous	Next >
Approve	Deny
•	

11 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 ASA 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.

Application			
Executable File Nar	ne: salesdb_tutorial.exe		
<u>R</u> esource File Name			
Sign the Appli	ation		Application ⊆ertificate
Select Platform to Build Packet PC Device (ARM)			Select Platform
Build Directory:	(end)		
Delete Tempo	ary Files After Build		
The temporar	y files include any PKDs and the EXE	E which may be deplo	yed to the device.
Deploy Target	& Options		
Target Directory:	Program Files		
Deploy Offer I	sula		
I* Depicy Mittel i	Julia		
Launch Applic	ation		
Add Shortcut	ation :o "Start Menu"		
Launch Applic Add Shortcut Deploy Ioday	ation io "Start Menu" Item		
Launch Applic Add Shortcut Deploy Ioday CAB File Packa	stion stion io "Start Menu" Item jing		
CAB File Packar	tion tio"Start Menu" Item jing le for Distribution		
CAB File Packa Create CAB	uid ition io "Start Menu" Item ing e for Distribution		CAB <u>C</u> ertificate
Launch Applic Add Shortcut Deploy Ioday CAB File Packae Create CAB Fi Sign the CAB	unu tisin to "Start Menu" Item for Distribution		CAB <u>C</u> ertificate
CAB File Packa Create CAB File Sign the CAB Version	tion on "Start Menu" Item e for Distribution		CAB Certificate
Capacity And Applic Applic Add Shortcut Deploy Ioday CAB File Packae Create CAB Fi Sign the CAB Sign the CAB Version Company Name: Draduct Name:	of Start Menu" to "Start Menu" Item e for Distribution MyCompany Evaluation: https://distribution		CAB <u>C</u> ertificate
Launch Applic Launch Applic Add Shortcut Deploy Inday CAB File Packae Create CAB File Sign the CAB Sign the CAB Version Company Name: Product Name:	and Stat Menu" totion e for Distribution MyCompany Salesdb_tutorial		CAB <u>C</u> ertificate
Launch Applic Launch Applic Add Shortcut Deploy Taday CAB File Packa Greate CAB File Packa Greate CAB File Packa Sign the CAB Sign the CAB Sign the CAB Proglut Name: Description: Comwidth:	alia titon o "Start Menu" Item le for Distribution My-Company palesdb_tutorial My Application Description		CAB Certificate
Legeby Actor Launch Applic Add Shortcut Deploy Inday CAB File Packa Greate GAB File Sign the CAB Sign the CAB Version Company Name: Description: Copyright: Horder	Start Mend" tem ing for Distribution MyCompany salesb_tutorial My Application Description		CAB <u>C</u> ertificate
Launch Applic Launch Applic Add Shortcut Deploy Ioday CAB File Packat Greate CAB File Sign the CAB Version Company Name: Proglut Name: Description: Copyright: Yersion:	and the second s		CAB <u>C</u> ertificate
Launch Applic Launch Applic Add Shortcut Deploy Iodsy CAB File Packa Create GAB Fi Sign the CAB Gynty Share: Product Name: Description: Copyright: Yersion:	alla dita di di anti d	KD Resource File Na	CAB <u>C</u> ertificate

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.

Deploying to an emulator

To deploy to an emulator, you click Select Platform and then select one of the emulators you previously installed.

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.

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.
- 3 Open ActiveSync while a connection to the device is established.

Sync	_ 🗆 ×
elp	
tails Explore Options	
	6
Status	
Synchronized	
	Sync elp tails Explore Options Status Synchronized

4 Click Explore. Double-click My Pocket PC.

My Pocket PC is the root directory.

- 5 Copy SalesDB_remote.DSN from your PocketBuilder 2.0\Tutorial\SalesDB directory to the My Pocket PC directory.
- 6 On your device, go to the \Programs\Sybase\ASA 9 directory . Locate the \Tutorial\SalesDB\db\fresh directory on your desktop.
- 7 Copy SalesDB_remote.db and SalesDB_remote.log to the \Programs\Sybase\ASA 9 directory on your device.

The following instructions provide you with deployment information for both the Pocket PC 2002 emulator and the Pocket PC 2003 emulator.

If you are using an emulator

In PocketBuilder, you can also deploy an application to an emulator. Here are the instructions for copying the database and log files to an emulator.

- 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 emulator.
- 3 Run the Windows CE file viewer, cefilevw.exe, on the desktop.

For the Pocket PC 2002 emulator, you can find cefilevw.exe by default in C:\Program Files\Windows CE Tools\Common Files\Platman\bin.

For the Pocket PC 2003 emulator, you can find cefilevw.exe by default in C:\Program Files\Common Files\Microsoft Shared\Windows CE Tools \Platman\bin.

- 4 Using the Windows CE file viewer, copy the SalesDB_remote.dsn from your desktop to the root directory of the emulator file system (\).
- 5 If the SalesDB_remote ASA server on the PC is running, stop the server.

When the server is running, an icon in the System Tray displays. The icon disappears when you stop the server.

6 Locate the \Tutorial\SalesDB\db\fresh directory on your desktop and copy the SalesDB_remote.db and SalesDB_remote.log files to the \Program Files\Sybase\ASA 9 directory on your emulator.

Path to ASA for a device is different from the path for an emulator The SalesDB_remote.dsn file sets the DSN on the device and contains the path to the remote database. However, the path to ASA for a device is different from the path for an emulator. Emulator users must edit the DSN file manually (as described in the previous steps) by removing \Sybase from the paths. With this setup, the SalesDB_remote database starts automatically on the Pocket PC when a connection is initiated and terminates automatically when all connections are closed.

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 165 of this lesson.

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

.

•

🗊 p_salesdb_tutorial (salesdb_tutorial) - Project Application Executable File Name: salesdb_tutorial.exe Resource File Name: Sign the Application Application Certificate Select Platform to Build 2 Pocket PC Device (ARM) Select Platform Build Directory: V Delete Temporary Files After Build The temporary files include any PKDs and the EXE which may be deployed to the device. Deploy Target & Options Target Directory: Program Files 3 🔽 Deploy After Build Launch Application
 Add Shortcut to "Start Menu"
 Deploy Ioday Item CAB File Packaging 4 Version Company Name: MyCompany 5 salesdb_tutorial Product Name: Description: My Application Description Copyright: 1.0.0.1 Version: PKD | Resource File Name c:\Program Files\Sybase\PocketBuilder 2.0\Tutorial(SalesDB)

The Project painter opens.

2 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

Where you are

Set up the ASA 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 run the SalesDB application.

About troubleshooting the application

If you have problems running the application, read the troubleshooting information in "Troubleshoot the application" on page 173.

- 1 Tap the Start Menu on the Pocket PC, and in the Programs menu, select File Explorer.
- 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 MakeDB.cmd batch file are intended for single-user devices only. Adaptive Server 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.

🔊 SalesDB tutorial	4 € 6:1	1 🚳
Customer : Apple	e St. Builders	≞ ≜
Product: 4x8 D	rywall x100	ΗĦ
Quanity:	25000	=
Price:	400	
Discount:	20	۱۱ ۲
Status:		
Notes:		-
◀ Ⅲ	•	-
< Previous	Next >]
Approve	Deny]
File Order Go Help		

- 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.

🖅 Add New Order 🛛 📢 2:15
Customer:
•
Product:

Quantity: Price:
Discount (%):
Ok Cancel

Troubleshoot the application

Where you are

Set up the ASA 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	This is most likely due to an incorrect DSN
application, the Connect to	file. Check the SalesDB_remote.DSN
Adaptive Server dialog	located at the root directory of the device or
displays. The application cannot	emulator and make sure that the database
establish a connection to the	file and start properties point to the correct
database.	locations.
When I initialize synchronization	Check the version of the ASA engine on the
from the device or emulator, the	device or emulator and the version of the
MobiLink window shows the	MobiLink server. Both components need to
error Error: Protocol version	be version 9.0 or higher. ASA 9.0 is
mismatch.	bundled with PocketBuilder.
When I initialize synchronization	Make sure the MobiLink server is running
from the device or emulator, the	and check the MobiLink server log for
MobiLink window shows the	more details.
error Communication error	
occurred while receiving data	
from the MobiLink server.	

Question	Answer
When I initialize synchronization from the device, MobiLink stalls while displaying the message <i>Connecting to MobiLink server at</i>	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 true to superprise again
When I initialize synchronization from the device or emulator, an ASA MobiLink Synchronization dialog displays with the message <i>Error in command near "-pd"</i> .	Make sure ASA 9.0.0 build 1108 or later is installed on the device or emulator. Previous builds do not support this switch. ASA 9.0.0 build 1108 is bundled with PocketBuilder.

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

For more information, see the chapter on ASA 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/. For information on how to install the Online Books, see the *Installation Guide*.

Index

Α

Adaptive Server Anywhere about 7 characteristics of 8 database setup 126 features 8 Application object definition 21 icon 47, 50 Open event 66 application platforms 3 applications building 125 deploying to devices 167 deploying to emulators 168 development tasks 29 running on a device 122, 171 running on the desktop 54, 59 testing 160 troubleshooting 173 ASA Demo DB database, setting up 31 ASA. See Adaptive Server Anywhere audience for this book vii

В

building applications 57, 120, 170

С

CAB files, building for distribution 16 class user objects 27 Clip window 17 coding the open event 112 comments in DataWindows 105 conventions x creating a workspace 36, 90 creating targets 39, 91

D

data synchronization 6 database connectivity 15, 78 Database painter, using 84 database profiles 78 databases connecting to 15 extended attributes 87 integratiion within PocketBuilder 6 retrieving, presenting, and manipulating data 23 synchronization 11 table definitions in 84 DataWindow objects about 77, 89 106 attaching to DataWindow controls color and border settings 101 comments 105 creating 99 creation of 138 data source 99 overview 23 saving 104 technology 5 DataWindow painter, bands in - 98 deploying applications to devices 57, 120, 167, 170 to emulators 168 detail band in DataWindow painter 98 development environment and object orientation 4 using 13 29 development tasks

devices supported 4 docking views 70 drop-down menus 25 DSN copying to ASA 117 creating 114

Ε

events, about 14 executable file, and application icon 47 extended attributes 87

F

features 4 floating toolbars 74 views 70 footer band in DataWindow painter 98 functions about 14 global 26 object-level 26

G

global functions 26 structures 26

Η

header band in DataWindow painter 98

I

icons, for applications 47 inheritance, and object-oriented programming 14

L

libraries, about 27

Μ

main window, size 50 manipulating views 68 menu bars 25 menu items 25 Menu painter about 25 adding menu items 144 using 143 MobiLink synchronization features 11 MobiLink Synchronization wizard database connection 146 using 145

0

object orientation 14 object-level functions 26 structures 26 object-oriented development 4 odbc.ini 78 open event, coding of 112 Output window 17

Ρ

PainterBar pop-up menus 74 using 18 painters 17 pinning views 70 PKL. See PocketBuilder Library (PKL) platforms 3 PocketBuilder about 3 Adaptive Server Anywhere 7

data synchronization - 6 databases 6 DataWindow technology 5 development environment 13 features 4 MobiLink synchronization 10 object-oriented development 4 SQL Anywhere Studio integration 7 synchronization technologies 7 target application platforms 3 UltraLite - 9 wizards 6 workspaces and targets 5 PocketBuilder Library (PKL) 27 pop-up menus about 74 for toolbars 74 PowerBar 18 PowerScript 14 PowerTips 18 presentation styles DataWindow object 101 Tabular 101 Project painter 57, 120, 166, 170

Q

queries 26

R

right mouse button, and pop-up menus 74 running applications on a device 171

S

scripts, about 14 setting up for the tutorial 32 SIP. *See* soft input panel size, of main window 50 soft input panel system functions 14 using 110, 155 SQL Anywhere Studio, and integration in PocketBuilder SQL Select, using 99 26 structures StyleBar 18 summary band in DataWindow objects 98 supported platforms 4 Sybase Central, and use of in PocketBuilder 7 synchronization of data 6 synchronization technologies in PocketBuilder 7 System Tree 17

Т

Tabular presentation style 101 target application platforms 3 targets creating 39, 91 what they provide 5 working with 16 testing applications 160 To-Do List 46 toolbars, showing text on 73 troubleshooting applications 173 tutorials database connection 77 development environment customization 63 employee list application 89 files 32 hello world application 35 sales application 125 setup 32 typographical conventions х

U

UltraLite features 9 use of in PocketBuilder 7 user objects about 26 class 27 types of 27 visual 27

۷

views docking 70 floating 70 manipulating 68 pinning 70 saving layout schemes 72 stacks 71 views, types of Design (DataWindow painter) 98 HTML Preview (DataWindow painter) 103 Object Details (Database painter) 87 Object Layout (Database painter) 86 Objects (Database painter) 84 Preview (DataWindow painter) 103 visual user objects 27

W

window properties, modifying 110, 154 windows about 22 creating 154 size 50 Windows CE Start menu 15 wizards for DataWindow creation 99 using for development 6 workspaces creating 36,90 in PocketBuilder environment 16 what they provide 5