



Building an UltraLite Palm Application Using MobileBuilder

A whitepaper from Sybase, Inc.

Table of Contents

Introduction	1
Lesson 1: Getting started.....	2
Lesson 2: Create a MobileBuilder project	3
Lesson 3: Add the UltraLite Component to your project	3
Lesson 4: Define your UltraLite database schema	5
Lesson 5: Configure synchronization.....	7
Lesson 6: Design a MobileBuilder form.....	8
Lesson 7: Build and run your application.....	12
Lesson 8: Complete the application	13
Restore the sample database.....	15

Introduction

This paper provides a tutorial that guides you through the process of developing an UltraLite application for the Palm Computing Platform using MobileBuilder. It describes how to build a very simple application, and how to add synchronization to your application.

- ◆ How to create an UltraLite database.
- ◆ How to add the UltraLite database control to your MobileBuilder project.
- ◆ How to configure the UltraLite database control for your application.
- ◆ How to bind a user interface component to a column of an UltraLite table.
- ◆ How to use the MobileBuilder Code Assistant to add functionality to your application.
- ◆ How to build and preview your application.

The application carries out the following tasks.

- 1 Connects to an UltraLite database, consisting of a single table. The table is the *ULProduct* table of the UltraLite Sample database.
- 2 Synchronizes with the consolidated database.
- 3 Displays some of the data in text fields.

Lessons

The lessons in this tutorial carry out the following steps:

- 1 Create a directory to hold the tutorial files.
- 2 Create a MobileBuilder project.
- 3 Add the UltraLite component to your MobileBuilder project.
- 4 Design the UltraLite database for the application in an Adaptive Server Anywhere reference database.
- 5 Configure the UltraLite component.
- 6 Design a form and bind controls to the UltraLite database.
- 7 Compile, link, and run the application.

Lesson 1: Getting started

In this lesson you prepare for the remainder of the tutorial. You make a directory to hold the files you will be using during the tutorial. In addition, you should make a copy of the UltraLite sample database as a backup copy so that you can easily restore it to its original state for future use.

During the tutorial, we use the original UltraLite 7.0 Sample database. At the end of the tutorial you can copy the untouched version from the *Tutorial* directory back into place.

❖ Ensure you have the required software installed:

- ◆ To carry out this tutorial, you need the following software:
- ◆ SQL Anywhere Studio 7.0.1 or later.
- ◆ PenRight! MobileBuilder 2.0 or later. Build 889 or higher is required.
- ◆ Palm OS SDK version 3.1. Version 3.5 is not supported by MobileBuilder.
- ◆ Palm emulator (included with MobileBuilder).
- ◆ The PRC Tools development suite for the Palm Computing Platform.

You need only the **cygwin** user tools, not the full PRC Tools install. You do not need to install PilRC as stated, because MobileBuilder includes a version of PilRC.

When you have installed PRC Tools, you can confirm that MobileBuilder has detected the tools. In MobileBuilder, choose Tools►Options►Build Tools. Highlight PRC-Tools (do not check the check box) and click Detect.

If the check box becomes checked, MobileBuilder has successfully detected PRC-Tools.

For more information on PRC Tools and the Palm emulator, see the Palm developer zone at <http://www.palmos.com/dev/>.

❖ Prepare a tutorial directory:

- ◆ From a command prompt, enter the following command:

```
mkdir c:\Tutorial
```

This creates a directory to hold the tutorial files. You can choose your own name for the directory, but in the remainder of the tutorial, we assume that this directory is *c:\Tutorial*.

❖ **Copy the sample database:**

- ◆ From a command prompt, change to the *UltraLite\Samples\CustDB* subdirectory of your SQL Anywhere directory. This directory holds the UltraLite 7.0 Sample database.
- ◆ Enter the following command:

```
copy CustDB.db c:\Tutorial
```

This makes a copy of the UltraLite 7.0 Sample database in the tutorial directory.

Lesson 2: Create a MobileBuilder project

In this lesson, you start building your MobileBuilder application. Here, you create a new MobileBuilder project and add the UltraLite database component to it.

❖ **Create a MobileBuilder project:**

- 1 Start MobileBuilder.
- 2 Create the project.
 - ◆ From the File menu, choose New Project. The New Project dialog is displayed.
 - ◆ Enter *Tutorial* as your project name
 - ◆ Enter your tutorial directory (*c:\Tutorial*) as the location.
 - ◆ Choose **C** as the language.
 - ◆ Choose **Palm** as the target.
 - ◆ Click OK to create the project. A Project Manager Panel is displayed on the left side of your MobileBuilder window and a Palm menu is added to the MobileBuilder interface.

You have created a standard MobileBuilder project. The next lesson describes how to add UltraLite features to this project.

Lesson 3: Add the UltraLite Component to your project

The UltraLite database component manages all the UltraLite-specific tasks within your MobileBuilder project. This lesson describes how to add the

component to your project and configure it for use with your reference database.

The UltraLite database component is a global component. It is not visible, and you do not need to add it to any form in your project. You need only one instance of the component in any application, and the component holds application-wide properties.

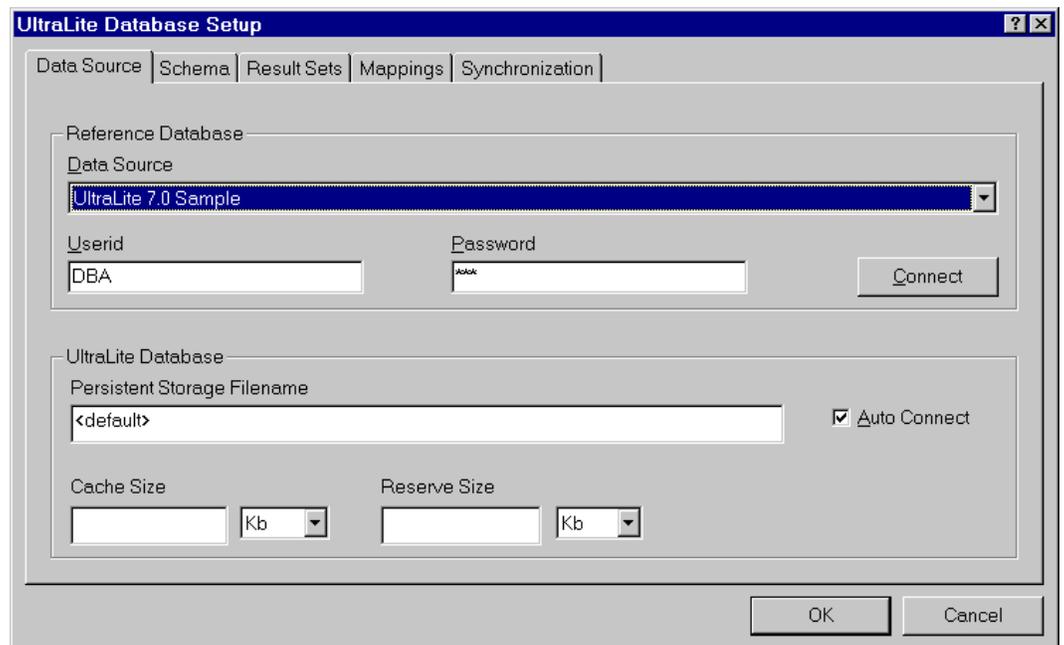
❖ **Add the UltraLite database component to your project:**

1 Add the UltraLite database component to your project.

- ◆ On the Component Palette at the top of your MobileBuilder window, click the UltraLite tab.
- ◆ Double-click the ULDatabase component. An UltraLite database component appears in the Global Components folder of your project. It has the name **ULDatabase1**.

2 Set the data source to be your reference database.

- ◆ In the Project Management Panel, open the Global Components folder and double-click ULDatabase1. The Properties tab for this component appears.
- ◆ Click the button next to the DataSource property. The UltraLite database setup dialog appears, open at the Data Source tab.



- ◆ From the Data Source drop-down list, choose **UltraLite 7.0 Sample**. This identifies the UltraLite sample database as your reference database.
- ◆ Enter **DBA** as user ID, and **SQL** as password. These entries are used during connection, and are also used by other pieces of the UltraLite database component.
- ◆ Click **Connect** to establish a connection to the reference database. An Adaptive Server Anywhere personal database server starts, and the UltraLite component connects to it.
- ◆ Leave the UltraLite database properties at their default settings. Ensure that **Auto Connect** is selected.
- ◆ Click **OK** to save the settings.

You must save the Data Source settings the first time you add them, before setting other UltraLite properties.

Lesson 4: Define your UltraLite database schema

Any UltraLite application requires a database. You design your UltraLite database in a **reference database**: an Adaptive Server Anywhere database that holds the tables your UltraLite application needs.

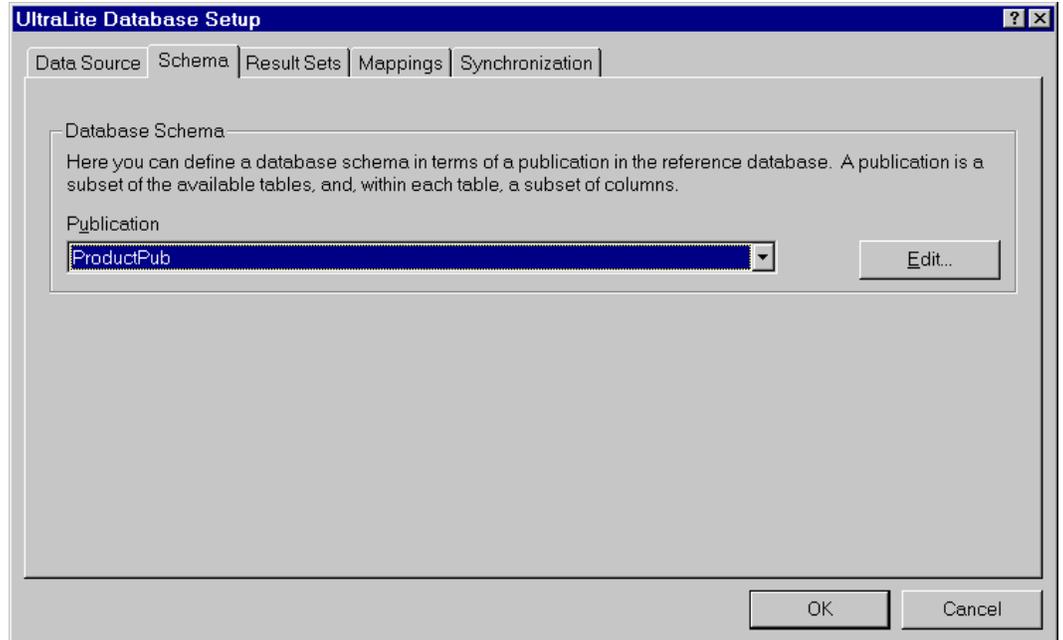
In this tutorial, you create an object called a **publication** in your reference database. The publication is a convenient device for assembling tables and column-based subsets of tables.

You can also define your UltraLite database by adding SQL statements to the reference database. SQL statements allow you to include joins and more advanced features in your UltraLite application.

❖ Create your UltraLite database schema:

1 Prepare to define the schema.

- ◆ In the Properties tab for the ULDatabase1 component, click the button by the **Schema** property. The UltraLite database setup dialog appears, open at the Schema tab.



2 Create a publication that reflects the data you wish to include in your UltraLite database.

- ◆ Click the Edit button. The Create Publication dialog appears.

The Table.Column list displays all columns of tables owned by the user ID you used in your connection.

- ◆ In the Publication Name field, enter the name **ProductPub**.
- ◆ Double-click the following columns to include them in the publication:
 - ◆ **ULProduct.prod_id**
 - ◆ **ULProduct.prod_name**
 - ◆ **ULProduct.price**
- ◆ Click OK to create the publication.
- ◆ In the UltraLite Database Setup dialog, ensure that **Product Pub** is selected in the Publication list, and click OK to complete the definition. Save your work.

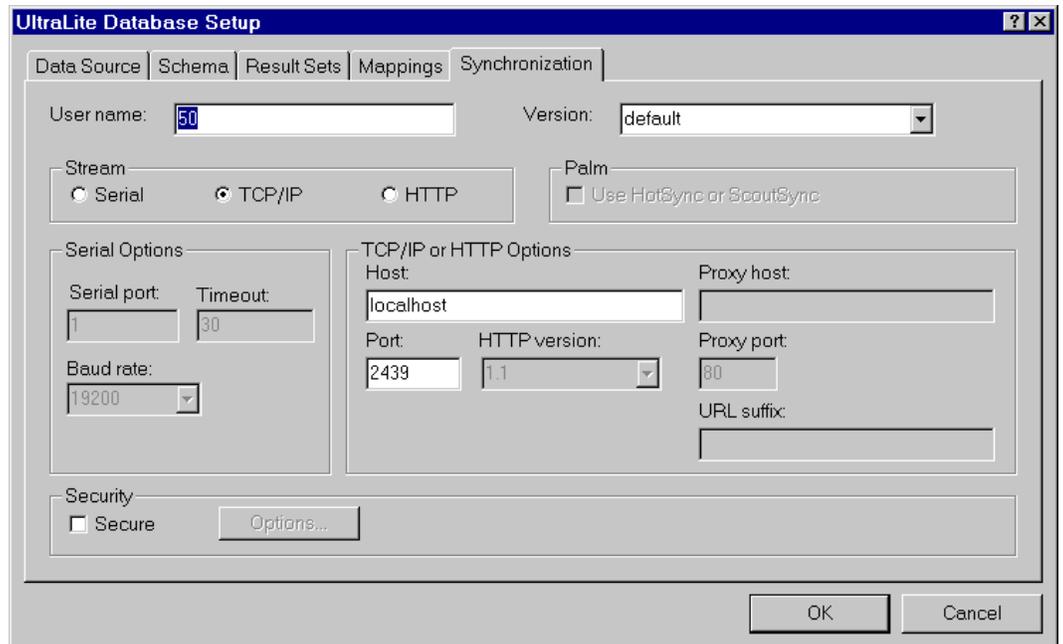
You have now finished designing your UltraLite database; in this case, just a single table. In the next lesson, you set up synchronization.

Lesson 5: Configure synchronization

There are several parameters that need to be selected in order to synchronize successfully. The meaning of the individual parameters is not described here. Instead, the section says which settings to use, and gives cross references for more information.

❖ Configure the synchronization parameters:

- 1 In the `ULDatabase1` property tab, click the button next to the Synchronization property. The UltraLite Database Setup dialog appears, open at the Synchronization tab.



- 2 In the User name field, enter 50. This value identifies the user for synchronization.
- 3 In the version field, choose default. This identifies the particular instructions (already stored in the UltraLite 7.0 Sample database) to be used for synchronization.
- 4 Leave the Stream setting at TCP/IP, the Host setting at localhost, and the Port setting at 2439. Leave the Security option unchecked.

It is more common to use HotSync or ScoutSync synchronization than TCP/IP for Palm devices, but here we use TCP/IP so that the application can be tested from the emulator without requiring a serial cable, which is needed for HotSync synchronization from the emulator.

For HotSync synchronization you would select HotSync, but leave the same information in the TCP/IP options. In this case the TCP/IP options direct the communication between the MobiLink HotSync conduit and the MobiLink synchronization server.

- 5 Click OK to complete the settings. Save your work.

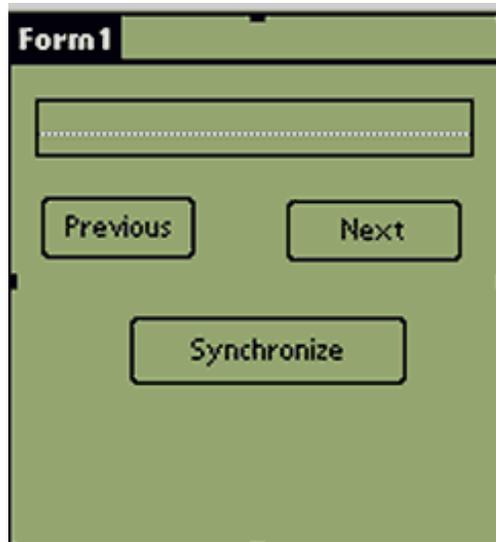
You have now configured the UltraLite database component. In this tutorial, you use default synchronization options. You set the Mappings properties in the next lesson.

Lesson 6: Design a MobileBuilder form

In this lesson you design a MobileBuilder form, and bind a list control on the form to the UltraLite database component.

❖ Design the MobileBuilder form:

- 1 Open the Project tab of the Project Manager Panel, and open the Forms folder.
- 2 Add a text field to the form. This field is used to display the Product name of the current product as you scroll through a list of products.
 - ◆ In the Project Management pane, double-click Form1 to display it in the right pane.
 - ◆ On the Component Palette, click Standard.
 - ◆ Click the text field icon (TField component), and draw a text field on the form. The text field can take up most of the space on the top half of the form. By default, the text field is named Field1. The following diagram shows all the visual components you will be adding to the form.



- 3 Add a Previous button to the form. This button is used to scroll backwards through a list of products.
 - ◆ With Form1 displayed, and the Standard toolbar displayed, click the button icon (TButton component), and draw a button underneath the text field on the left side of the form. This will be the back scrolling button. This button is called Button1
 - ◆ Right-click the button and select Properties from the drop down list. The Property sheet for the button is displayed in the left pane.
 - ◆ Set the caption of the button to Previous.
- 4 Add a Next button to the form. This button is used to scroll forward through a list of products.
 - ◆ Repeat the previous step. This time, draw the button on the right side of the form, underneath the text field. This will be the forward scrolling button. This button is called Button2
 - ◆ Set the caption to Next.
- 5 Add a Synchronize button to the form. This button is used to synchronize the application with the MobiLink synchronization server.
 - ◆ Draw a button underneath the Previous and Next buttons. This button is called Button3
 - ◆ Set the caption to Synchronize.
- 6 Bind the Text field to a column of the UltraLite database.

You bind controls to UltraLite objects from the UltraLite database component.

- ◆ In the Project Management pane, double-click the UltraLite database control.
 - ◆ Click the button beside the Mappings item. The UltraLite database Setup dialog appears, open at the Mappings tab.
 - ◆ Ensure that **Form1** is selected in the Forms list, and that **Field1** is selected in the Controls list.
 - ◆ In the Fields From Publication list, click **ULProduct.prod_name**. Click Map to add this field to the mappings list.
 - ◆ Click OK to complete the mapping.
- 7 Use the MobileBuilder Code Assistant to add code that synchronizes data to the Synchronize button.
- ◆ In the MobileBuilder right-hand pane, right-click the form and choose Code Assistant from the popup menu.
 - ◆ In the Code Assistant, make the following settings:

Field	Value
When This Component	Form1
Drop down box next to Form1	Button3
fires This Event	CtlSelect
then, with this component	Application
Drop down box next to Application	ULDatabase1
Do the following	Synchronize the connection

- ◆ Click Add Code, and in the dialog that is displayed, choose **ULProduct**.
 - ◆ Click OK to add the code. Leave the Code Assistant open.
- 8 Use the MobileBuilder Code Assistant to add code that opens the table for use after synchronizing.
- ◆ In the Code Assistant, make the following settings:

Field	Value
When this component	Form1
Drop down box next to Form1	Empty
fires this event	Frmlnit

Field	Value
then, with this component	Application
Drop down box next to Application	ULDatabase1
do the following	Open a table or result set

- ◆ Click Add Code, and in the dialog that appears, choose ULProduct.
- ◆ Click OK to finish. Leave the Code Assistant open.

9 Use the MobileBuilder Code Assistant to add code that scrolls through the data to the Previous and Next buttons.

- ◆ In the Code Assistant, make the following settings:

Field	Value
When This Component	Form1
Drop down box next to Form1	Button1
fires This Event	CtlSelect
then, with this component	Application
Drop down box next to Application	ULDatabase1
Do the following	Move to previous record

- ◆ Click Add Code, and in the dialog that appears, choose ULProduct.
- ◆ Click OK to finish. Leave the Code Assistant open.
- ◆ In the Code Assistant, make the following settings:

Field	Value
When This Component	Form1
Drop down box next to Form1	Button2
fires This Event	CtlSelect
then, with this component	Application
Drop down box next to Application	ULDatabase1
Do the following	Move to next record

- ◆ Click Add Code, and in the dialog that appears, choose ULProduct.
- ◆ Click OK to finish. Leave the Code Assistant open.

10 Click Close to close the Code Assistant.

You have now completed the application, which displays the product names in the database. You can look at the generated code in the code tab of the MobileBuilder right pane. Next, you compile and run your application.

Lesson 7: Build and run your application

This section describes how to compile and link your application using the PRC Tools.

❖ Build and run your application:

1 Save the application.

From the File menu, choose Save. Accept the default name and click Save.

2 From the MobileBuilder Project menu, choose Build.

When compiling is complete, the Output Panel at the bottom of the MobileBuilder workspace should display a line stating the following:

```
Tutorial.exe - 0 error(s), 0 warning(s)
```

Compiler failure This tutorial assumes that you have a working installation of the PRC Tool chain for Palm development. If the build fails, you should check that your PRC Tools installation is correct.

3 Start the MobiLink synchronization server so that a connection can be established.

From the Windows Start button, choose Programs ► Sybase SQL Anywhere 7 ► Synchronization ► MobiLink Synchronization Server Sample.

The MobiLink synchronization server starts, and displays that it is ready to handle requests. In this case we are synchronizing with the UltraLite 7.0 Sample database, which is serving as both reference and consolidated databases. In general, the reference and consolidated database are separate objects.

4 Start the application.

From the MobileBuilder Project menu, choose Run.

The Palm OS Emulator appears and displays the form you designed.

5 Prepare the emulator for TCP/IP synchronization.

- ◆ Right click the emulator, and choose Settings ► Properties from the popup menu.
- ◆ Check Redirect NetLib calls to host TCP/IP. This option sets the Emulator to use TCP/IP communications with applications on the current device.

- ◆ Leave other items in the dialog unchanged and click OK.

6 Synchronize your application.

- ◆ Click the Synchronize button.

Inspect the MobiLink synchronization server window. It should display a set of messages confirming that Synchronization is complete. You can scroll back through the messages to see that a set of rows were added to the UltraLite database running in the Emulator.

The current position is left before the first item of the result set, and so no item is yet displayed in the text field.

- ◆ Click the Next button.

The first product name from the database appears in the text field.

You have now built and run a simple UltraLite application using MobileBuilder. The application is still incomplete, however. To see this, click Applications in the Emulator to switch away from your application, and then click App to switch back to it. Although the user interface displays, you will see that the application no longer functions. The next lesson describes how to fix that problem.

You can close MobileBuilder and the MobiLink synchronization server.

Lesson 8: Complete the application

In this lesson you add the application-switching features needed to handle switching away from and back to your application.

To add these features, which are specific to the Palm Computing Platform, you need to write code for both the **AppExit** and **AppEnter** events.

Each time the user switches away from an application, the MobileBuilder **AppExit** event is invoked. You need to add code for this event to close the table. The current position is lost during the switch away from and back to the application, but the data is kept.

- ❖ **To close tables when the user exits the application:**

1 Open the Code Assistant.

In the MobileBuilder right-hand pane, right-click the form and choose Code Assistant from the popup menu.

- 2 In the Code Assistant, make the following settings:

Field	Value
When This Component	Application
Drop down box next to Form1	Empty
fires This Event	AppExit
then, with this component	Application
Drop down box next to Application	ULDatabase1
Do the following	Close a table or result set

- 3 Click Add Code, and in the dialog that is displayed, choose **ULProduct**.
- 4 Click OK to add the code. Leave the Code Assistant open.

Each time the user switches back to an application, the MobileBuilder **AppEnter** event is invoked. You need to add code for this event to open the table. The operation is the same whether or not the user is switching to the application for the first time or at a subsequent time.

❖ **To open tables when the user enters the application:**

- 1 In the Code Assistant, make the following settings:

Field	Value
When This Component	Application
Drop down box next to Form1	Empty
fires This Event	AppEnter
then, with this component	Application
Drop down box next to Application	ULDatabase1
Do the following	Open a table or result set

- 2 Click Add Code, and in the dialog that is displayed, choose **ULProduct**.
- 3 Click OK to add the code. Leave the Code Assistant open.
- 4 Compile and run your application to test it.

Result sets and tables

This tutorial used a table object to hold the data. You can also use result set objects to hold more complex query result sets. The two objects have different behavior for application switching.

- ◆ Tables must be closed using **ULBClose** before exiting the application, and opened using **ULBOpen** when the user enters the application.
- ◆ Result sets must *not* be closed before exiting the application. You do not need to take any specific action for result sets when exiting the application. When the user enters the application for the first time, the result set must be opened using **ULBOpen**. When the user enters the application in subsequent times, the result set must be reopened using **ULBReopen**.

The following code provides an example handler for the **AppEnter** event. The **ULDatabase1LaunchCode** function tests whether the application is being entered for the first time or not. The result set is opened on first entry, but is reopened on subsequent entries.

```
static BOOLEAN OnAppEnter(EVENTTYPE *pEvent, WORD *pError)
{
    // perform default processing
    DefaultHandler(pEvent, pError);
    switch( ULDatabase1LaunchCode() ){
    case LAUNCH_SUCCESS_FIRST:
        if( !ULBOpen( ResultSet1,
                    ULDatabase1connection) ){
            // initialization failed
            DisplayNotice ( "LAUNCH_SUCCESS_FIRST error",
                           "OK", "Cancel" );
            break;
        }
        break;
    case LAUNCH_SUCCESS:
        if( !ULBReopen(ResultSet1,
                      ULDatabase1connection) ){
            DisplayNotice ( "LAUNCH_SUCCESS error",
                           "OK", "Cancel" );
            break;
        }
        break;
    case LAUNCH_FAIL:
        // error
        break;
    }
    // return TRUE to continue event routing {{MB1}}
    return(TRUE);
}
```

The above code illustrates how to use a **DisplayNotice** function for rudimentary debugging.

Restore the sample database

Now that you have completed the tutorial, you should restore the sample database so that it can be used again. You created a backup copy of the UltraLite 7.0 Sample database at the beginning of this tutorial. You can now

replace the modified version of *custdb.db* with the backup copy, overwriting any changes.

❖ **Restore the sample database:**

- 1 Delete the database file *custdb.db*, and the transaction log file *custdb.log* in the *UltraLite\Samples\CustDB* subdirectory of your SQL Anywhere directory.
- 2 From a command prompt, change to the *UltraLite\Samples\CustDB* subdirectory of your SQL Anywhere directory and enter the following command:

```
dberase -y custdb.db
```

- 2 Copy the *custdb.db* file from your tutorial directory to the *UltraLite\Samples\CustDB* subdirectory of your SQL Anywhere directory.

Your sample database is now restored to its original state.

This completes the tutorial.



Sybase, Inc. Worldwide Headquarters 6475 Christie Avenue, Emeryville, CA 94608 USA
Phone: 1-800-8-SYBASE (in US and Canada); Fax: 1-510-922-3210.

World Wide Web: <http://www.sybase.com>

Copyright © 2000 Sybase, Inc. All rights reserved. Sybase, the Sybase logo, Adaptive Server, MobiLink, and SQL Anywhere are trademarks of Sybase, Inc. All other trademarks are property of their respective owners. ® indicates registration in the United States. Specifications are subject to change without notice.