



**Tutorial: Windows Mobile Application  
Development**

---

**Sybase Unwired Platform 2.1**

DOCUMENT ID: DC01285-01-0210-01

LAST REVISED: December 2011

Copyright © 2011 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 trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

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

<b>Sybase Unwired Platform Tutorials .....</b>	<b>1</b>
<b>Task Flow .....</b>	<b>3</b>
<b>Getting Started .....</b>	<b>5</b>
Installing Sybase Unwired Platform .....	5
Installing Microsoft Synchronization Software .....	6
Starting Sybase Unwired WorkSpace .....	7
Connecting to Sybase Control Center .....	7
Learning the Basics .....	8
<b>Developing a Windows Mobile Device Application .....</b>	<b>11</b>
Generating C# Object API Code .....	12
Creating the User Interface for the Windows Mobile Device Application .....	13
Creating the Customers Form .....	14
Creating the Customer Details Form .....	19
Creating the Main Program File .....	23
Deploying and Running the Device Application .....	25
<b>Learn More about Sybase Unwired Platform .....</b>	<b>31</b>
<b>Index .....</b>	<b>33</b>

# Contents

# Sybase Unwired Platform Tutorials

The Sybase® Unwired Platform tutorials demonstrate how to develop, deploy, and test mobile business objects, device applications, and mobile workflow packages. You can also use the tutorials to demonstrate system functionality and train users.

- Learn mobile business object (MBO) basics, and create a mobile device application:
  - *Tutorial: Mobile Business Object Development*
- Create native mobile device applications:
  - *Tutorial: BlackBerry Application Development*
  - *Tutorial: iOS Application Development*
  - *Tutorial: Windows Mobile Application Development*
- Create a mobile workflow package:
  - *Tutorial: Mobile Workflow Package Development*

The tutorials demonstrate a cross section of basic functionality, which includes creating MBOs, and using various Sybase Unwired WorkSpace development tools, independent development environments, and device types. Tutorial projects are available if you want the finished tutorial without going through the steps.



# Task Flow

Sybase Unwired WorkSpace Eclipse tutorials explain how to develop, deploy, and run a mobile application.

**Table 1. Eclipse tutorials**

Task	Goals	Steps required to complete the task
Getting Started	<ul style="list-style-type: none"> <li>• Install all required WorkSpace components and external resources.</li> <li>• Start Unwired Server, then use Sybase Control Center to connect to the server.</li> <li>• Open the Mobile Development perspective, and become familiar with the views of the perspective, and the Mobile Application Diagram.</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Installing Sybase Unwired Platform</i> on page 5</li> <li>• <i>Installing Microsoft Synchronization Software</i> on page 6</li> <li>• <i>Starting Sybase Unwired WorkSpace</i> on page 7</li> <li>• <i>Connecting to Sybase Control Center</i> on page 7</li> <li>• <i>Learning the Basics</i> on page 8</li> </ul> <hr/> <p><b>Note:</b> These steps are prerequisites for the rest of this tutorial. You need to perform them only once.</p>
Developing Database Mobile Business Objects	<ul style="list-style-type: none"> <li>• Create a mobile application project and a connection to the database.</li> <li>• Create two mobile business objects, and create a relationship between them.</li> <li>• Deploy the mobile business objects to Unwired Server.</li> </ul>	Complete the <i>Tutorial: Mobile Business Object Development</i> .
Developing a Windows Mobile Device Application	<ul style="list-style-type: none"> <li>• Generate code for the Windows Mobile platform, code a Windows Mobile device application, and run it on an emulator.</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Generating C# Object API Code</i> on page 12</li> <li>• <i>Creating the User Interface for the Windows Mobile Device Application</i> on page 13</li> <li>• <i>Deploying and Running the Device Application</i> on page 25</li> </ul>

## Task Flow



# Getting Started

Install and learn about Sybase Unwired Platform and its associated components.

Complete the following tasks for all tutorials, but you need to perform them only once.

**1. *Installing Sybase Unwired Platform***

Install Sybase Unwired Platform.

**2. *Installing Microsoft Synchronization Software***

Install and configure Microsoft synchronization software so you can deploy and run a mobile application on a Windows Mobile emulator.

**3. *Starting Sybase Unwired WorkSpace***

Start Unwired WorkSpace.

**4. *Connecting to Sybase Control Center***

Open the Web-based Sybase Control Center administration console to manage Unwired Server and its components.

**5. *Learning the Basics***

Learn about Sybase Unwired WorkSpace and how to access help (optional).

## Installing Sybase Unwired Platform

---

Install Sybase Unwired Platform.

Before starting this tutorial, be sure you have all the requisite Unwired Platform components installed. For complete installation instructions, see:

- *Release Bulletin for Sybase Mobile SDK*
- *Installation Guide for Sybase Mobile SDK*
- *Release Bulletin for Runtime*
- *Installation Guide for Runtime*
- Install Sybase Mobile SDK, which includes:
  - Development support for Native Object API applications, HTML5/JS Hybrid (Mobile Workflow) applications, and OData SDK applications.
  - Sybase Unwired WorkSpace, the Eclipse-based development environment for MBOs and Mobile Workflows.
- Install Unwired Platform Runtime:
  - Data Tier (included with single-server installation)
  - Unwired Server

## Installing Microsoft Synchronization Software

---

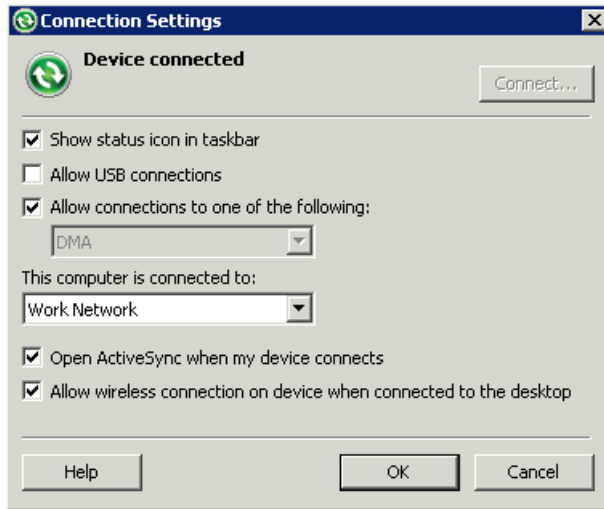
Install and configure Microsoft synchronization software so you can deploy and run a mobile application on a Windows Mobile emulator.

---

**Note:** This tutorial shows how to install Microsoft ActiveSync for Windows XP. If you are using Windows Vista, Windows 7, or Windows 2008, install Virtual PC 2007 SP1 and Windows Mobile Device Center to manage synchronization settings. Download the Windows Mobile Device Center from <http://www.microsoft.com/windowsmobile/en-us/downloads/microsoft/device-center-download.msp> and follow Microsoft instructions for installing and using that software instead of this procedure.

---

1. Download Microsoft ActiveSync:
  - a) In your Web browser, open the Windows Phone page at <http://www.microsoft.com/windowsmobile/en-us/help/synchronize/device-synch.msp>.
  - b) In the Windows Phone page, follow the instructions to select and download the sync software for your computer's operating system. Windows XP requires ActiveSync version 4.5.
  - c) In the Windows Phone downloads page, click the **ActiveSync** button.
  - d) In the ActiveSync page, download the ActiveSync install file and save it to your local system.
2. Run the downloaded install file.  
For example, double-click `setup.msi` in Windows Explorer.
3. When the installation is complete, restart your machine.
4. Start ActiveSync if it does not start automatically.  
For example, click **Start > Programs > ActiveSync**.
5. In ActiveSync, click **File > Connection Settings**.
6. Select **Allow connections to one of the following**, then select **DMA**.
7. For the option, **This computer is connected to**, select **Work Network**.



8. Click **OK**.

## Starting Sybase Unwired WorkSpace

---

Start Unwired WorkSpace.

Select **Start > Programs > Sybase > Unwired Platform > Unwired WorkSpace**.

The Sybase Unwired Workspace opens in the Mobile Development perspective. The Welcome page displays links to product information, and to the product.

### Next

To read more about Sybase Unwired WorkSpace concepts and tasks, select **Help > Help Contents** from the main menu.

## Connecting to Sybase Control Center

---

Open the Web-based Sybase Control Center administration console to manage Unwired Server and its components.

From Sybase Control Center, you can:

- View servers and their status
- Start and stop a server
- View server logs
- Deploy a mobile application package
- Register devices

## Getting Started

- Set role mappings

For information on configuring, managing, and monitoring Unwired Server, click **Help > Online Documentation**.

1. Click **Start > Programs > Sybase > Sybase Control Center**.

---

**Note:** If the Sybase Control Center service does not open, make sure that the Sybase Control Center service is started. See the *Installation Guide for Runtime*.

---

2. In Sybase Control Center, log in by entering the credentials set during installation. Logging in to Sybase Control Center gives you access to the Unwired Platform administration features that you are authorized to use.

## Learning the Basics

---

Learn about Sybase Unwired WorkSpace and how to access help (optional).

### Prerequisites

Start Unwired WorkSpace.

### Task

1. In the Welcome page, click any of the links to explore the Unwired WorkSpace environment.
2. To enter the Sybase Unwired WorkSpace development environment, click **Start Development** or close the Welcome tab.

The default Mobile Development perspective provides ready access to most of the tools you need to create, update, and manage mobile business objects (MBOs). This table describes the main windows and views of the Mobile Development perspective. Note that not all the views are open initially; some views become available only after you begin developing your MBOs:

View or Window	Description
WorkSpace Navigator	A view of mobile application projects. Each project folder includes resources and data source references to which the MBOs are bound, personalization keys, and so on.  Use this view to review and modify MBO-related properties.
Enterprise Explorer	A view of enterprise back-end resources, such as database servers, SAP® servers, and Sybase Unwired Server.

View or Window	Description
Mobile Application Diagram	<p>A graphical editor for designing mobile business objects. A Mobile Application Diagram is associated with each project.</p> <p>Use the Mobile Application Diagram to create MBOs (including attributes and operations), then define relationships with other MBOs. You can:</p> <ul style="list-style-type: none"> <li>• Create MBOs in the Mobile Application Diagram using Palette icons and menu selections. Either bind to a data source now or defer binding. For example, using a top-down approach, you might model your MBOs before creating the data sources to which they bind.</li> <li>• Drag items from Enterprise Explorer and drop them onto the Mobile Application Diagram to create the MBO – quickly creates the operations and attributes automatically based on the data source of the items.</li> </ul>
Palette	A view from which you can drag controls onto an open Mobile Application Diagram and define their attributes, operations, and relationships to your application.
Properties	A view that shows the properties of the object currently selected in the Mobile Application Diagram, and lets you edit them. You cannot create an MBO from the Properties view, but generally, most development and configuration is performed here.
Outline	An outline of the file that is currently open in an editor, listing structural elements. The contents are editor-specific.
Error Log	The error log captures Eclipse warnings and errors, including stack traces.
Problem	A view that displays problems, errors, or warnings.

3. To access the online help, click **Help > Help Contents** in the main menu bar.
4. Expand any of the documents that appear in the left pane.  
Some documents are for Sybase Unwired Platform, while others are for the Eclipse development environment.



# Developing a Windows Mobile Device Application

Generate code for the Windows Mobile platform based on the MBOs, add additional code to develop a Windows Mobile device application, and test its functionality.

## Prerequisites

- *Getting Started* on page 5
- Install Visual Studio 2008.
- Install Microsoft .NET Compact Framework 3.5 from <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=e3821449-3c6b-42f1-9fd9-0041345b3385&displaylang=en>, if it was not installed with Visual Studio.
- *Tutorial: Mobile Business Objects Development*, which provides the foundation tasks for this tutorial.
- In Unwired WorkSpace, open the SUP101 mobile application project. In WorkSpace Navigator, right-click the SUP101 folder and select **Open in Diagram Editor**.

## Task

The device application communicates with the database mobile business objects that are deployed to Unwired Server.

### 1. *Generating C# Object API Code*

Generate Object API code for Windows Mobile devices.

### 2. *Creating the User Interface for the Windows Mobile Device Application*

Import the SUP101 project in to Visual Studio 2008, configure the project, then download the tutorial code snippets from Product Documentation so you can create the application user interface.

### 3. *Deploying and Running the Device Application*

Deploy the device application to a Windows Mobile device emulator, and test its functionality.

## Generating C# Object API Code

---

Generate Object API code for Windows Mobile devices.

### Prerequisites

1. Connect to both the sampledb database and Unwired Server. Code generation fails if the server-side (run-time) enterprise information system (EIS) data sources referenced by the MBOs in the project are not running and available to connect to when you generate object API code.
2. Switch to the Advanced developer profile to see advanced options in Unwired WorkSpace.

You must be

### Task

1. Right-click the SUP101 Mobile Application Diagram and select **Generate Code**.
2. Select **Continue without a configuration**, and click **Next**.
3. Select the Customer MBO, then click **Next**.
4. Enter the information for these configuration options:

Option	Description
Language	Select <b>C#</b> .
Platform	Select <b>NET Compact Framework 3.5 for Windows Mobile</b> .
Unwired Server	Select <b>My Unwired Server</b> .
Server domain	Select <b>default</b> .
Page size	Leave blank.
Namespace	Enter SUP101.
Destination	Accept the default. The default location for generated code is \SUP101\Generated Code.
Replication-based	This tutorial uses <b>Replication-based</b> synchronization.

5. Click **Finish**.



## Creating the User Interface for the Windows Mobile Device Application

---

Import the SUP101 project in to Visual Studio 2008, configure the project, then download the tutorial code snippets from Product Documentation so you can create the application user interface.

### Prerequisites

- Install Windows Mobile 6 SDK from <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=06111a3a-a651-4745-88ef-3d48091a390b&displaylang=en>.
- Obtain the text files that contain the code snippets you need to build the user interface from the SUP\_WM\_Custom\_Dev\_Tutorial\_code.zip file. The code snippets are used to create the CustomerSample, CustomerList, and CustomerSampleScreen C# files.
  - If you are viewing this guide as a PDF, you can obtain the text files from the Sybase Product Documentation Web site at <http://sybooks.sybase.com/nav/summary.do?prod=1289&lang=en&submit=%A0Go%A0&prodName=Sybase+Unwired+Platform&archive=0>. Select the appropriate version, navigate to this topic in the tutorial, then click the link for the zip file to save the files locally.
  - If you are viewing this guide online from the Sybase Product Documentation Web site, click *SUP\_WM\_Custom\_Dev\_Tutorial\_code.zip* to access the text files.

### Task

If needed, install the following:

1. In Visual Studio 2008, open SUP101.csproj.
2. Edit the project properties.
  - a) In Solution Explorer, right-click SUP101, then select **Change Target Platform**. Select **Windows Mobile 6 Professional SDK**, then **OK**.
  - b) Right-click SUP101, then select **Properties**.
  - c) In the Application tab, select **Console Application** for **Output type**.
  - d) In the Devices tab, select **Windows Mobile 6 Professional Emulator** for the **Target device**.
  - e) Select **File > Save All**, then save the solution as SUP101.sln.

### See also

- *Deploying and Running the Device Application* on page 25

## Creating the Customers Form

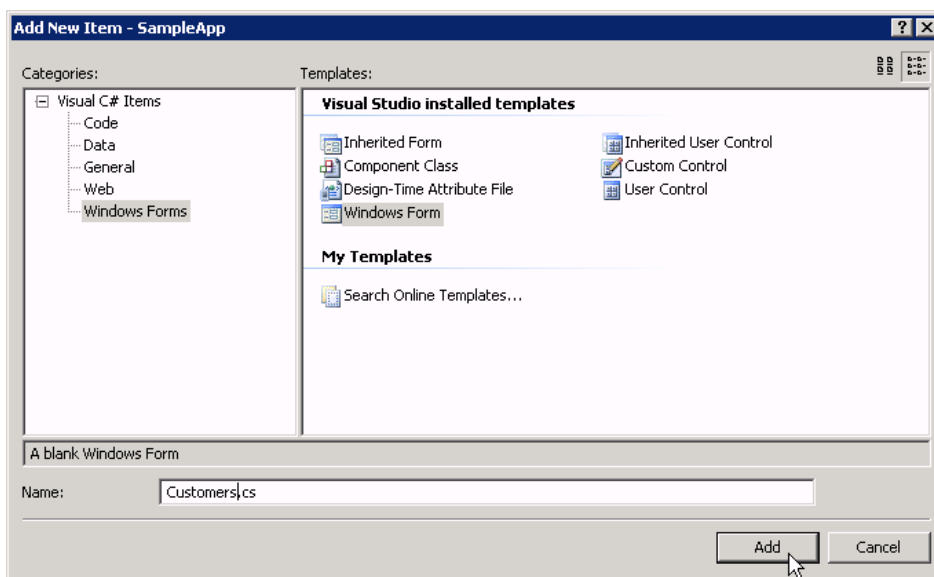
Create the user interface for the Customers form.

### Prerequisites

Delete `Form1.cs` from the `SampleApp` project.

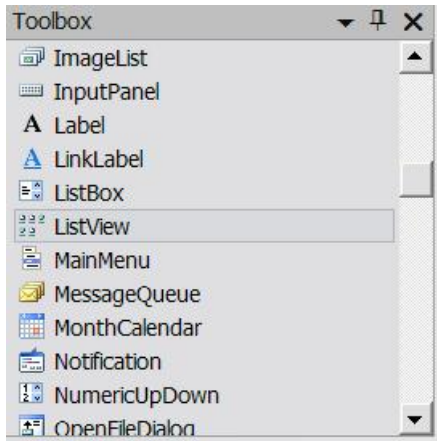
### Task

1. In Solution Explorer, right-click the `SUP101` project and select **Add > New Item**.
2. In the Add New Item dialog, select **Windows Form** from the Categories and Templates, enter `Customers` as the form name, and click **Add**.

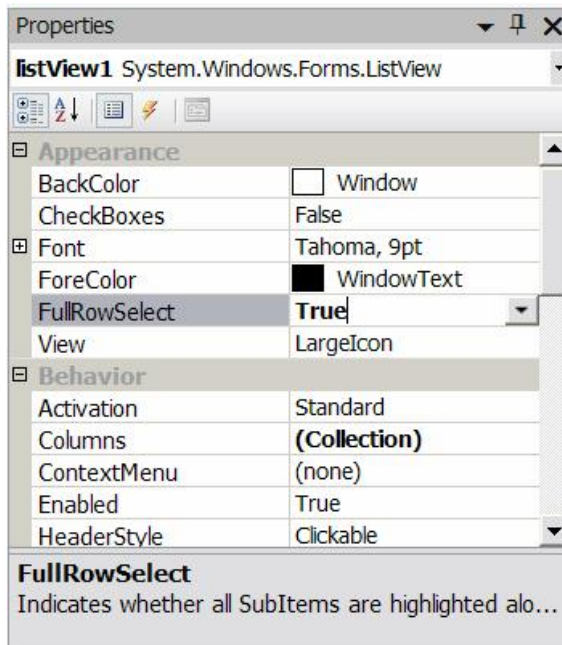


An empty form, **Customers**, displays on the **Customer.cs [Design]** tab.

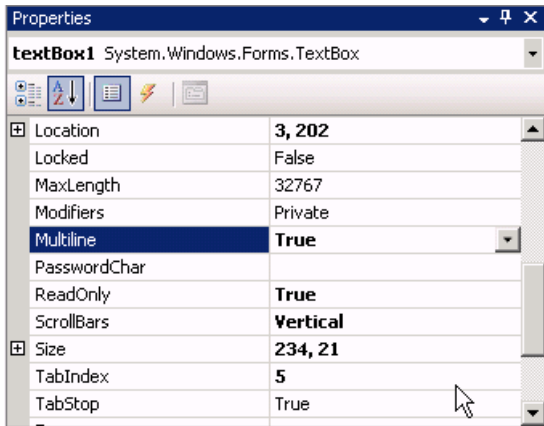
3. From the Toolbox, drag and drop three buttons on to the form.
4. Select each button, and in the Properties view, change the **Text** of the buttons to:
  - button1 – Sync
  - button2 – Refresh
  - button3 – Update
5. In the Toolbox, select **List View** from **Common Device Controls**, and drag and drop it to the Customers form.



6. In the Toolbox, select **Textbox** from **Common Device Controls**, and drag and drop it to the Customers form.
7. In the **Customers** form, click the **ListView**, then in the Properties pane, set **FullRowSelect** to **True**.



8. In **Customers**, select the **Textbox**, then in the Properties pane, set these properties:
  - Multi-line – True
  - Read-only – True
  - Scrollbars – Vertical



9. Arrange the controls on the Customers form so they look like this:



10. Save the `Customers.cs` form.
11. In Solution Explorer, in the `SUP101` project, right-click `Customers.cs`, then select **View Code**.
12. Replace the code with the source code from the `Customers.cs` file you downloaded from the Sybase Product Documentation Web site:

```

using System;
using System.Linq;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using SUP101;

namespace SUP101
{
    public partial class Customers : Form
    {
        public Customers()
        {
            InitializeComponent();

            //Click event for Sync button in Customers screen
            private void button1_Click(object sender, EventArgs e)
            {
                if (!SUP101DB.DatabaseExists())
                {
                    //This creates the database.
                    SUP101DB.CreateDatabase();
                }

                try
                {
                    //This calls SUP101, the package database, user name
                    and password to connect to the Unwired Server.
                    //If your login name and password is different, then
                    the changes must be reflected here.
                    //The default is used below.
                    SUP101DB.LoginToSync("supAdmin", "s3pAdmin");
                    SUP101DB.SubmitPendingOperations();
                    SUP101DB.Synchronize();
                    AddString("Synchronization completed.");
                }
                catch (Exception ex)
                {
                    List<LogRecordImpl> lrs =
                    LogRecordImpl.FindWithQuery(new Sybase.Persistenc.Query());
                    AddString("log record count=" + lrs.Count);
                    for (int i = 0; i < lrs.Count; i++)
                    {
                        AddString(lrs[i].Message);
                    }
                }
                AddListView();
            }

            private void AddString(String s)
            {
                textBox1.Text += s + "\r\n";
            }
        }
    }
}

```

```

        textBox1.SelectionStart = textBox1.Text.Length;
        textBox1.ScrollToCaret();
        textBox1.Refresh();
    }

    //Click event for Refresh button in Customers screen
    private void refresh_Click(object sender, EventArgs e)
    {
        //Check if "default" synchronization group has been
synchronized.
        //All MBOs not specifically put inside any synchronization
group in eclipse tooling, will be put inside "default"
synchronization group
        //In the example, Customer MBO is in default
synchronization group
        if (SUP101DB.IsSynchronized("default"))
        {
            Cursor.Current = Cursors.WaitCursor;
            AddString("Refresh data");
            AddDataToListView();
            Cursor.Current = Cursors.Default;
        }
        else
        {
            AddString("Please synchronize first.");
        }
    }

    private void AddListView()
    {
        this.listView1.Clear();
        listView1.Columns.Add("Id", listView1.Width / 4,
HorizontalAlignment.Left);
        listView1.Columns.Add("First Name", listView1.Width /
3, HorizontalAlignment.Center);
        listView1.Columns.Add("Last Name", listView1.Width / 3,
HorizontalAlignment.Right);
        listView1.View = View.Details;
        listView1.FullRowSelect = true;
    }

    private void AddDataToListView()
    {
        this.listView1.Clear();
        listView1.Columns.Add("Id", listView1.Width / 4,
HorizontalAlignment.Left);
        listView1.Columns.Add("First Name", listView1.Width /
3, HorizontalAlignment.Center);
        listView1.Columns.Add("Last Name", listView1.Width / 3,
HorizontalAlignment.Right);
        listView1.View = View.Details;
        listView1.FullRowSelect = true;

        List<Customer> c = Customer.FindAll();
        if (c.Count > 0)
        {

```

```

        for (int i = 0; i < c.Count; i++)
        {
            ListViewItem item = new
ListViewItem(c[i].Id.ToString());
            item.SubItems.Add(c[i].Fname);
            item.SubItems.Add(c[i].Lname);
            listView1.Items.Add(item);
        }
    }

    //Click event for Update button in Customers screen
    private void update_Click(object sender, EventArgs e)
    {
        if (listView1.FocusedItem != null)
        {
            Program.setCustomer(listView1.FocusedItem.Text);
            Program.getForm2().Visible = true;
            Program.getForm1().Visible = false;
        }
        else
            MessageBox.Show("Please select a row.");
    }
}

```

13. Be sure the bolded code line matches the Sybase Unwired Platform Admin login and password you indicated during installation.

```
SUP101DB.LoginToSync("supAdmin", "s3pAdmin");
```

14. Click the Customers.cs[Design] tab to go back to the Customers form design view to add event handlers to the buttons.

- a) Click the Sync button on the form.
- b) In the Properties view for the button, click the Event icon (lightning bolt),
- c) Next to the Click databinding, select **button1\_Click**.
- d) Repeat these steps for each button, selecting these events for each Click databinding.
  - Refresh – **refresh\_Click**
  - Update – **update\_Click**

15. Save your changes.

## Creating the Customer Details Form

Create the user interface for the Customer\_details form.

1. Add another Windows Form to the project, and name this one Customer\_details.
2. From the Toolbox, drag and drop three labels onto the Customer\_details form.
3. Align the labels on the left side of the form.

## Developing a Windows Mobile Device Application

In the Properties view, in the Text field, rename the labels **First Name**, **Last Name**, and **City**.

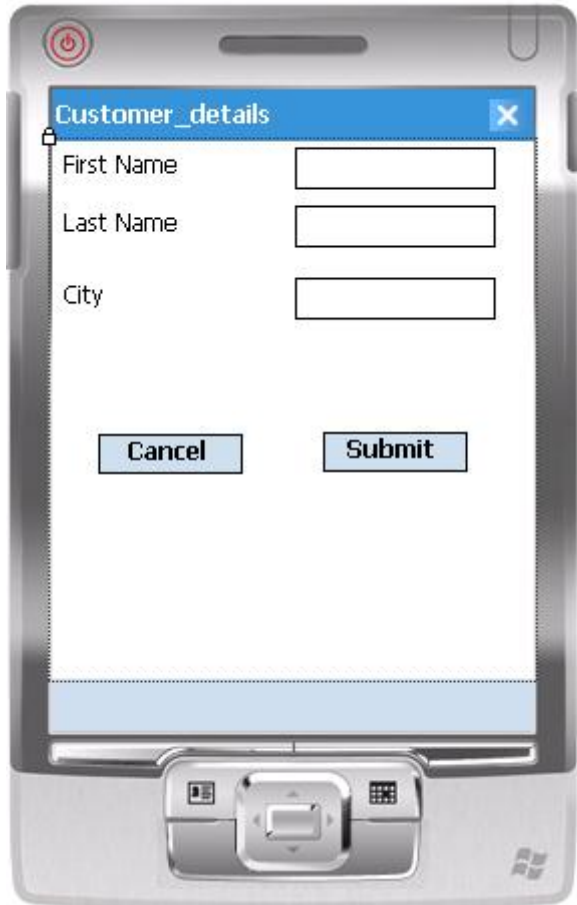
4. From the Toolbox, drag and drop three text boxes onto the Customer\_details form and align them to the right of each of the three labels.
5. Set the properties for each text box.
  - For First Name, use tbFName.
  - For Last Name, use tbLName.
  - For City, use tbCcity.



6. From the Toolbox, drag and drop two buttons from Common Device Controls to the Customer\_details form below the labels and text boxes.
  - a) In the Properties view, in the **Text** field, rename the button on the left to **Cancel**.
  - b) In the Properties view, in the **Text** field, rename the button on the right to **Submit**.

The form looks like this:





7. Save the `Customer_details.cs` form.
8. In Solution Explorer, in the SUP101 project, right-click `Customer_details.cs`, then select **View Code**.
9. Replace the existing code with the code from the `Customer_details.cs` file you downloaded from the Sybase Product Documentation Web site:

```
using System;
using System.Linq;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using SUP101;

namespace SUP101
```

```

{
    public partial class Customer_details : Form
    {
        Customer thisCustomer;
        public Customer_details()
        {
            InitializeComponent();
        }

        //Click event for Cancel button in Customer_detail screen
        private void Cancel_Click(object sender, EventArgs e)
        {
            Program.getForm1().Visible = true;
            Program.getForm2().Visible = false;
        }

        //Click event for Submit button in Customer_detail screen
        private void Submit_Click(object sender, EventArgs e)
        {
            thisCustomer.Fname = tbFName.Text;
            thisCustomer.Lname = tbLName.Text;
            thisCustomer.City = tbCity.Text;
            thisCustomer.Save();
            Program.getForm1().Visible = true;
            Program.getForm2().Visible = false;
        }

        private void AddDataToForm()
        {
            int id = Int32.Parse(Program.getCustomer());

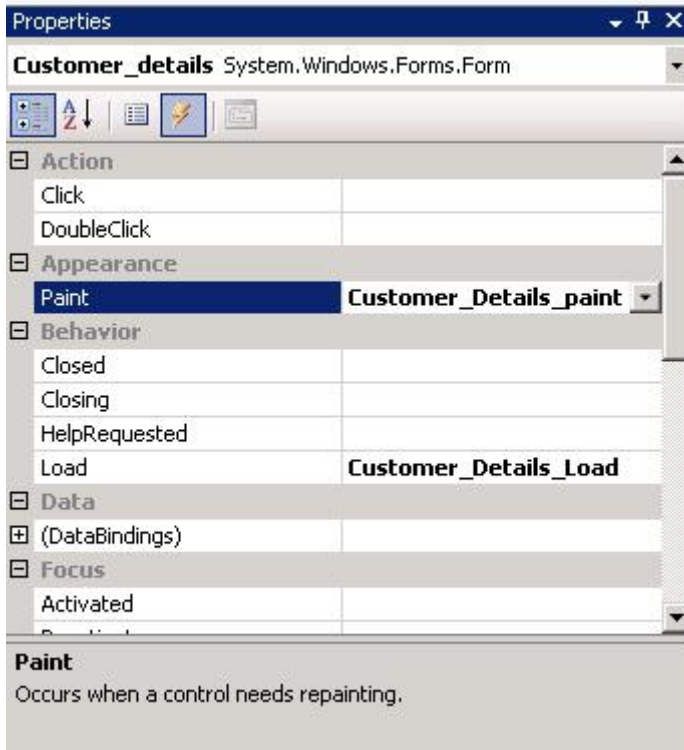
            //Retrieves data from the local database.
            thisCustomer = Customer.FindByPrimaryKey(id);
            tbFName.Text = thisCustomer.Fname;
            tbLName.Text = thisCustomer.Lname;
            tbCity.Text = thisCustomer.City;
        }

        //Load method event for Customer_detail
        private void Customer_Details_Load(object sender, EventArgs
e)
        {
            AddDataToForm();
        }

        //paint method event for Customer_detail
        private void Customer_Details_paint(object sender,
PaintEventArgs e)
        {
            AddDataToForm();
        }
    }
}

```

10. In the the Customer\_details.cs[Design] view, click the **Submit** button, then in the Properties view, add the Submit\_Click event to the Submit button, and the Cancel\_Click event to the Cancel button.
11. Add events to the Customer\_details.cs.
  - a) Click the Customer\_details.cs [Design] tab.
  - b) In Properties view, select **Customer\_details System.Windows.Forms.Form**.
  - c) Click the **Events** icon (lightning bolt).
  - d) In **Load**, add the **Customer\_Details\_Load** event.
  - e) In **Paint**, add the **Customer\_Details\_paint** event.



12. Save your changes.

## Creating the Main Program File

Create the Program.cs file, which is the main entry point for the application.

1. In the Solution Explorer, right-click the SUP101 project, then select **New Item**.
2. In Templates, select **Code File**.
3. Name the code file Program.cs, then click **Add**.

4. Right-click `Program.cs`, then select **View Code**.
5. Replace the existing code with the code from the `Program.cs` file you downloaded from the Sybase Product Documentation Web site:

```
using System;
using SUP101;
using System.Windows.Forms;

namespace SUP101
{
    static class Program
    {
        private static Customers _form1 = new Customers();
        private static Customer_details _form2 = new
Customer_details();
        private static string _custid;

        public static string getCustomer()
        {
            return _custid;
        }

        public static void setCustomer(string custid)
        {
            _custid = custid;
        }

        //Customer list screen
        public static Customers getForm1()
        {
            return _form1;
        }

        //Customer detail screen
        public static Customer_details getForm2()
        {
            return _form2;
        }

        static void Main(string[] args)
        {
            Application.Run(_form1);
        }
    }
}
```

6. Save your changes.
7. Build the project by pressing **Control+Shift+B**.

## Deploying and Running the Device Application

---

Deploy the device application to a Windows Mobile device emulator, and test its functionality.

1. In Visual Studio, choose **Tools > Device Emulator Manager**.
2. Scroll down to **Windows Mobile 6 Professional Emulator**, right-click, and select **Connect**.
3. Right-click **Windows Mobile 6 Professional Emulator** again and select **Cradle**.
4. In the Solution Explorer of Visual Studio, right-click the SUP101 project and select **Deploy**.
5. Then in the Windows Mobile emulator go to **Start > Programs > File Explorer**.
6. Navigate to the installed path by going in **Program Files > sup101**.
7. Click **SUP101** to start the application.
8. Click **Sync**.

A message displays in the output box below.



9. Click **Refresh** to populate Customer data in the list view.

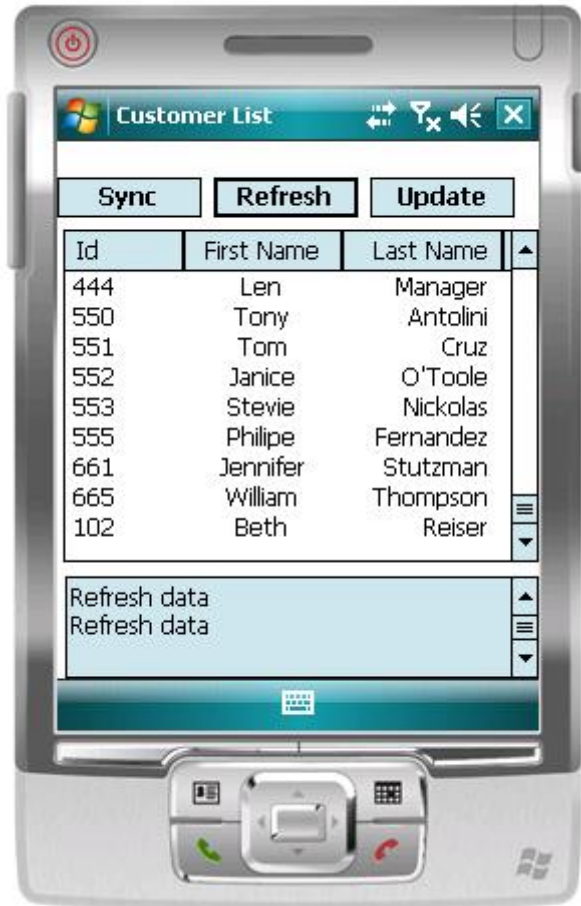


10. Highlight a customer record and then click **Update**.



11. Make changes, then click **Submit** to return to the Customers List screen.
12. Click **Sync**.  
The change is reflected in the backend database.
13. Click **Refresh** and scroll to the bottom of the list view to see the local cache of the changed data.





You have now successfully completed the replication-based sample tutorial for Windows Mobile.

**See also**

- *Creating the User Interface for the Windows Mobile Device Application* on page 13



# Learn More about Sybase Unwired Platform

Once you have finished, try some of the other samples or tutorials, or refer to other development documents in the Sybase Unwired Platform documentation set.

Check the Sybase Product Documentation Web site regularly for updates: access <http://sybooks.sybase.com/nav/summary.do?prod=1289>, then navigate to the most current version.

## *Tutorials*

Try out some of the other getting started tutorials available on Product Documentation to get a broad view of the development tools available to you.

## *Tutorial Projects*

Tutorial projects are available for download, if you want the finished tutorial without going through the steps. Download tutorial projects from: <http://www.sdn.sap.com/irj/sdn/mobile?rid=/webcontent/uuid/40ea4956-b95c-2e10-11b3-e68c73b2280e>.

## *Samples*

Sample applications are fully developed, working applications that demonstrate the features and capabilities of Sybase Unwired Platform.

Check the SAP® Development Network (SDN) Web site regularly for new and updated samples: <https://cw.sdn.sap.com/cw/groups/sup-apps>.

## *Online Help*

See the online help that is installed with the product, or the Product Documentation Web site.

## *Developer Guides*

Learn about using the API to create device applications:

- *Developer Guide: BlackBerry Native Applications*
- *Developer Guide: iOS Native Applications*
- *Developer Guide: Windows and Windows Mobile Native Applications*
- *Developer Guide: Mobile Workflow Packages*

Customize and automate:

- *Developer Guide for Unwired Server Management API* – customize and automate system administration features.
- *Developer Guide: Unwired Server* – customize and automate server-side implementations for device applications, and administration, such as data handling.

Javadoc and HeaderDoc are also available in the installation directory.

Learn More about Sybase Unwired Platform

# Index

## A

ActiveSync, installing and configuring 6  
articles 31

## B

basics, learning 8  
BlackBerry 1

## C

C# object API code generation 12  
Customer\_details form 19  
Customers form 14

## D

deploying  
    configuring ActiveSync for 6

## E

Eclipse Studio Edition  
    Sybase Unwired WorkSpace 7  
Enterprise Explorer, defined 8

## G

getting started  
    Sybase Unwired Platform 5  
    Sybase Unwired WorkSpace 8  
    tutorial projects 1  
    tutorials 1

## H

help, online 8

## I

installing  
    Microsoft ActiveSync 6  
    Sybase Unwired Platform 5

    synchronization software 6  
iOS 1

## M

Microsoft .NET Compact Framework 3.5 11  
Microsoft ActiveSync, installing and configuring 6  
Mobile Application Diagram, defined 8

## O

online help, accessing 8

## P

Palette, defined 8  
program.cs 23  
Properties view, defined 8

## S

samples 31  
starting  
    Sybase Unwired WorkSpace 7  
Sybase Control Center  
    connecting to 7  
Sybase Unwired Platform  
    getting started 5  
    installing 5  
    learning more about 31  
Sybase Unwired WorkSpace  
    getting started 8  
    starting 7  
synchronization software  
    installing 6

## T

tutorial projects 31  
tutorials 31

## V

Visual Studio 2008 11

**W**

Windows Mobile 1

Windows Mobile 6.5 Professional Developer Tool  
Kit 11

WorkSpace Navigator, defined 8