Getting Started with Windows Mobile Development – Windows Mobile SDK – C#

Contents

Overview

This document demonstrates the process of designing a Windows Mobile application using the Link-OS Multiplatform SDK. The videos' contents demonstrate creating the user interface, adding code in C# to the project, running the project in a Windows Mobile Device Simulator, packaging the project and finally deploying to a real Windows Mobile device.

The sample code used in this guide is available on GitHub in the Zebra Technologies repository here.

For information on joining the ISV Program contact a Program manager in your area. Details here.

Target Audience

The information delivered in this document assumes the reader has reasonable technical competence covering Microsoft Windows and Mobile device environments, core programming concepts and rationales.

System Prerequisites

The following system prerequisites must be satisfied in order to complete this tutorial.

Java Runtime Environment
For more information including system prerequisites and download links for Java, click here.

Microsoft Visual Studio 2008 Professional Edition
Microsoft Visual Studio is the integrated development environment (IDE) used to develop applications for Windows Mobile.

Note: Visual Studio 2008 Express Edition is not compatible with the Microsoft Mobile Development Tool Kit which is a requirement for developing applications for Windows Mobile platform.

Microsoft .NET Compact Framework 3.5
The Microsoft .NET Compact Framework 3.5 includes everything you need to test .NET Compact Framework (Versions 1, 2 and 3.5) applications.

For more information including system prerequisites and download links for .NET Compact Framework, click here.

Microsoft Windows Mobile 6 Professional SDK
The toolkit is distributed free of charge by Microsoft. It adds sample code, header and library files, emulator images and tools to Visual Studio that let you build applications for Windows Mobile 6.

For more information including system prerequisites and download links, click here.

Microsoft Windows Mobile Development Tool Kit
The toolkit is distributed free of charge by Microsoft. It includes documentation, sample code, code library files and emulator images and tools required to develop applications for Windows Mobile.

For more information including system prerequisites and download links, click here.

Microsoft Windows Mobile Active Sync or Windows Vista Mobile Device Center
Activesync / Windows Mobile Device Center are required to deploy applications to a Windows Mobile device.

For more information including system prerequisites and download links for ActiveSync, click here.

Windows Mobile Device Center replaces ActiveSync when developing and deploying when using Windows Vista.

For more information including system prerequisites and download links for Windows Mobile Device Center, click here.

Zebra Multiplatform SDK
The Zebra Multiplatform SDK contains all the required components to develop applications for Zebra label printers. The SDK includes a Microsoft .Net library (.dll) which provides the means to scan for, connect to and print on Zebra label printers.

For more information, including system prerequisites and download links for Zebra Multiplatform SDK, click here.

Zebra Network-Enabled Label Printer
In order to fully test the application created during the course of this tutorial, a network-enabled Zebra printer capable of understanding ZPL or CPCL.

Bluetooth connectivity is available through the Zebra Multiplatform SDK and covered in other available tutorials.

For more information on Zebra label printers, click here.

Windows Mobile Device
Whilst the Windows Mobile Emulator included in the Windows Mobile Development Toolkit satisfies most of the anticipated requirements for developing against Windows Mobile, it is recommended that all development is fully tested against a physical device.

Note: For the list of printers and mobile devices that are supported by the Zebra Multiplatform SDK, click here.

Installations

All the items listed in the “System Prerequisites must be installed. The default installation options for all these prerequisites are satisfactory.

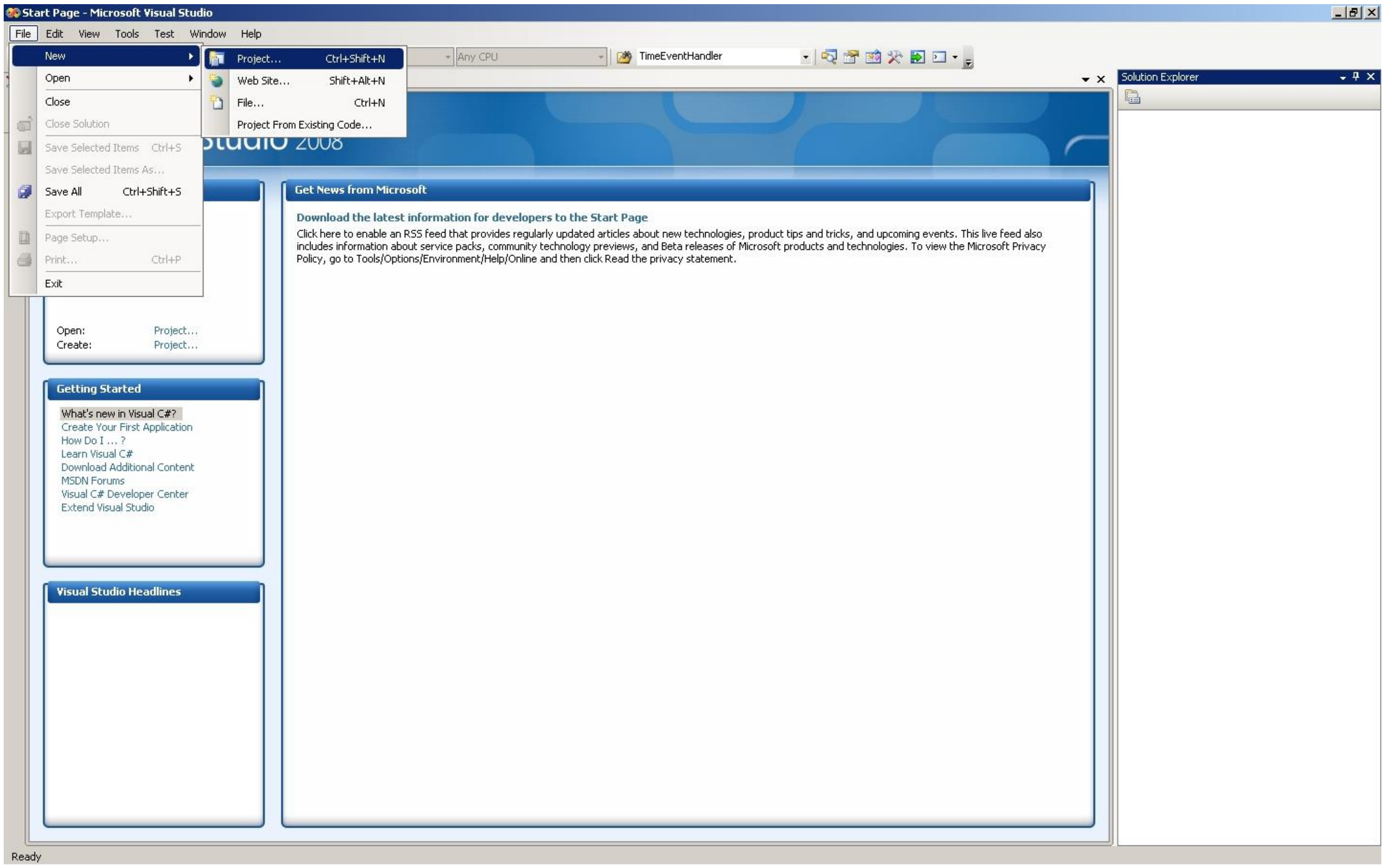
Creating Your First Zebra Mobile Application

You will be guided on how to create a mobile application that allows the handheld device to establish a TCP connection with a Zebra mobile printer.

1. **Launch Microsoft Visual Studio 2008.** Go to Start -> Programs -> Microsoft Visual Studio 2008 -> Microsoft Visual Studio 2008.

2. **Create a Smart Device Project.**

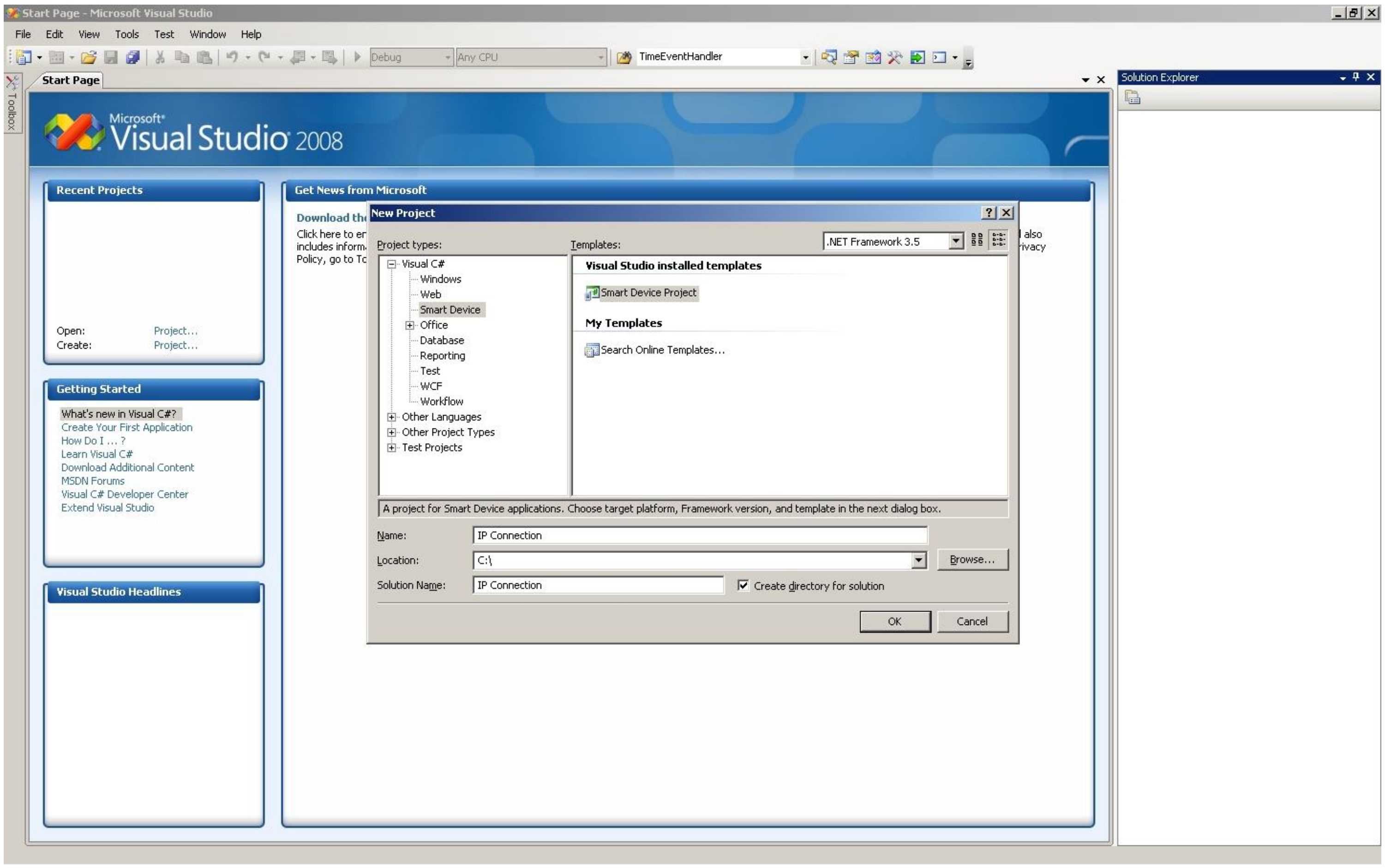
    a. Click on File -> New -> Project.



Download the latest information for developers to the Start Page

Click here to enable an RSS feed that provides regularly updated articles about new technologies, product tips and tricks, and upcoming events. This live feed also includes information about service packs, community technology previews, and Beta releases of Microsoft products and technologies. To view the Microsoft Privacy Policy, go to Tools/Options/Environment/Help/Online and then click Read the privacy statement.

    b. A New Project dialog appears. In the Project Types pane, expand Visual C# and select Smart Device, then select the Smart Device Project template.



A project for Smart Device applications. Choose target platform, Framework version, and template in the next dialog box.

Name: IP Connection

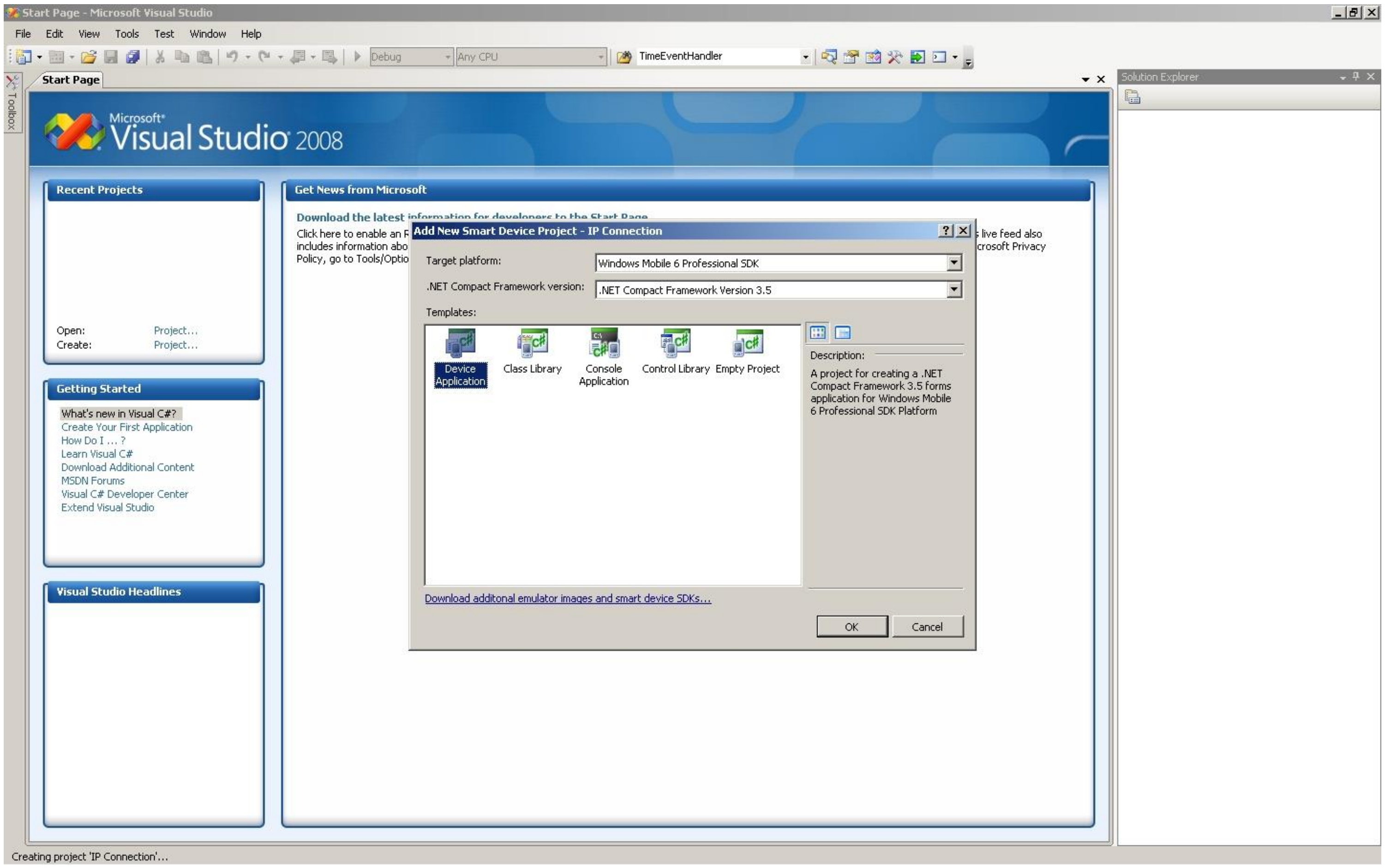
Location: C:\

Solution Name: IP Connection

☑ Create directory for solution

c. Enter “IP Connection” in the Name field. Choose your project location and make sure the Create directory for solution checkbox is checked. Click OK.

d. On the Add New Smart Device Project dialog, select Windows Mobile 6 Professional SDK, then select .NET Compact Framework Version 3.5. If these options are not available, refer to system prerequisites on instructions on how to install these options. Select Device Application template, and then click OK.



Add New Smart Device Project - IP Connection

Target platform: Windows Mobile 6 Professional SDK

.NET Compact Framework version: .NET Compact Framework Version 3.5

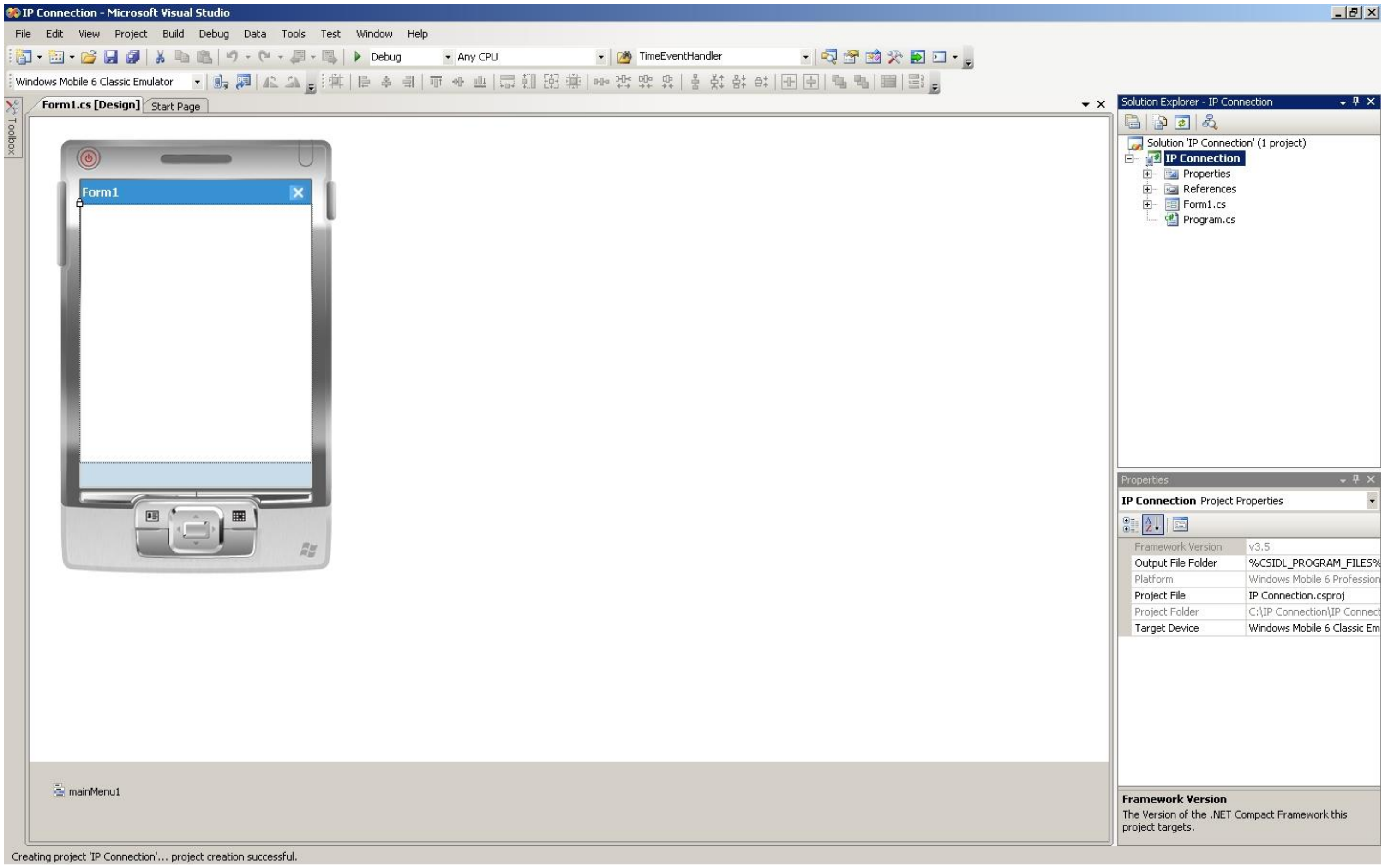
Templates: Device Application, Class Library, Console Application, Control Library, Empty Project

Description: A project for creating a .NET Compact Framework 3.5 forms application for Windows Mobile 6 Professional SDK Platform

Download additonal emulator images and smart device SDKs…

OK Cancel

e. You now have a project opened in the IDE. The Form Design is opened by default.


IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
TimeEventHandler
Windows Mobile 6 Classic Emulator
Toolbox
Form1.cs [Design]
Start Page
Form1
mainMenu1
Solution Explorer - IP Connection
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
Form1.cs
Program.cs
Properties
IP Connection Project Properties

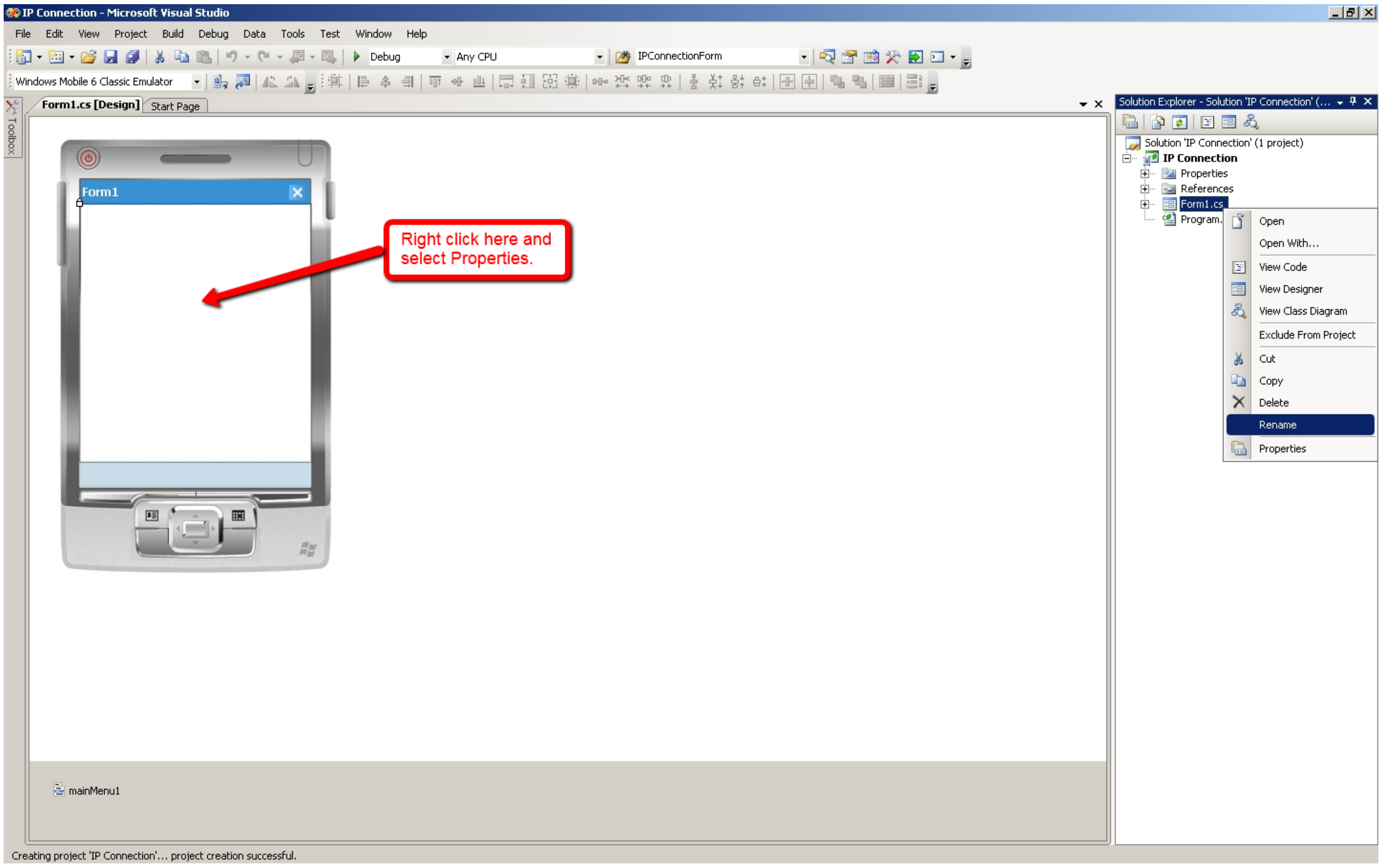
Framework Version	v3.5
Output File Folder	%CSIDL_PROGRAM_FILES%
Platform	Windows Mobile 6 Profession
Project File	IP Connection.csproj
Project Folder	C:\IP Connection\IP Connect
Target Device	Windows Mobile 6 Classic Em

Framework Version
The Version of the .NET Compact Framework this project targets.

Creating project 'IP Connection'... project creation successful.

3. **Design the Form Layout.**

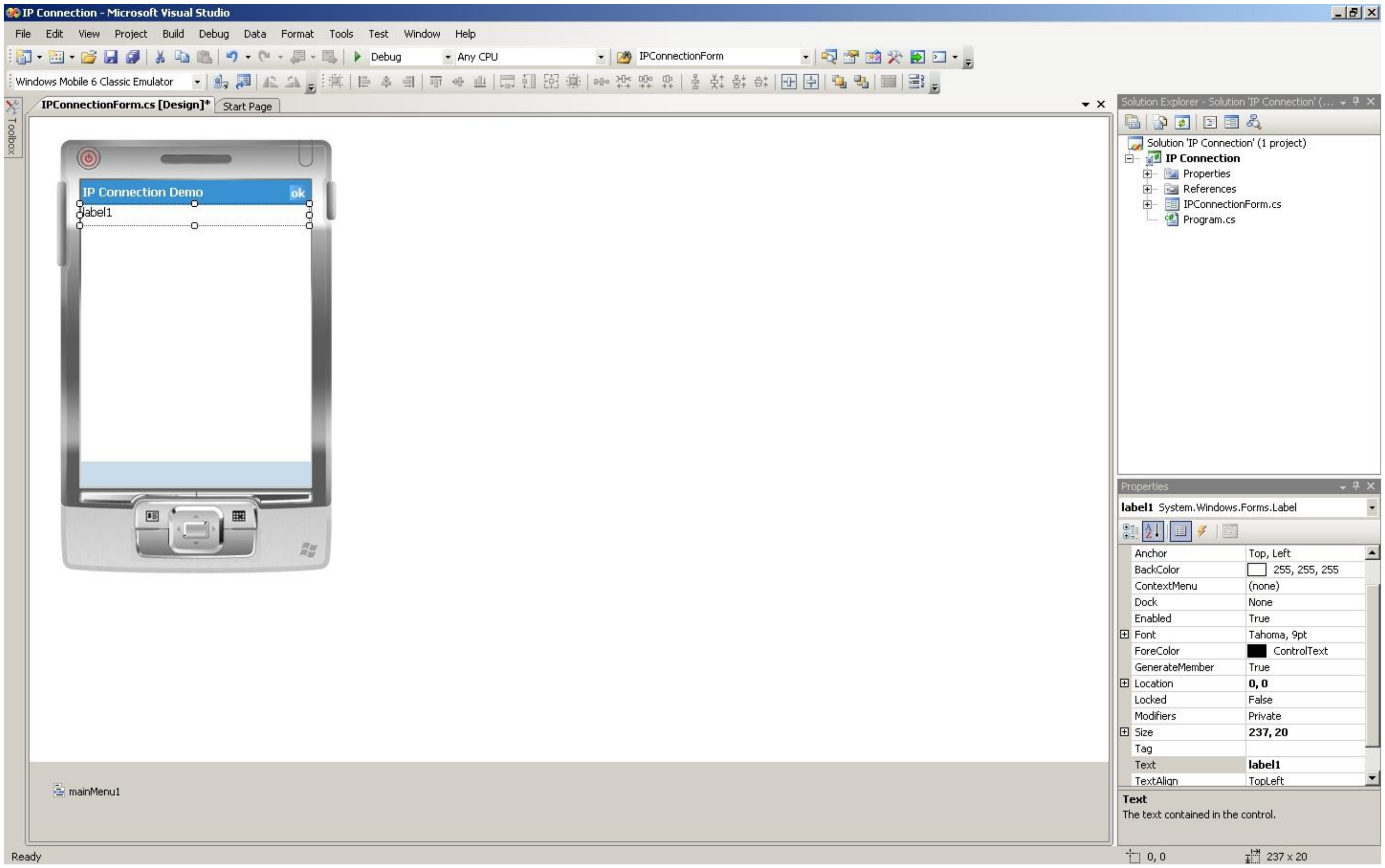
    a. Rename the form name in the solution explorer. Right-click Form1.cs and select Rename. Enter “IPConnectionForm.cs” and hit Enter. Click Yes when prompted to rename all references of “Form1”.


IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Form1.cs [Design]
Start Page
Form1
Right click here and select Properties.
mainMenu1
Solution Explorer - Solution 'IP Connection' (1 project)
IP Connection
Properties
References
Form1.cs
Program.cs
Open
Open With...
View Code
View Designer
View Class Diagram
Exclude From Project
Cut
Copy
Delete
Rename
Properties
Creating project 'IP Connection'... project creation successful.

b. Right-click anywhere on the form and select Properties. This brings up the Properties table on the bottom right panel. Enter “IP Connection Demo” in the Text field. This changes the text on the title bar.

c. On the Properties table, set MinimizeBox to False. This changes the “x” in the upper right corner of the form to “ok” which when clicked, closes the application instead of minimizing it.

d. We will use a Label to create a status bar for the TCP connection. Expand the Toolbox to the left, click and drag a Label onto the form. Position the label at the top and stretch it to fit the width of the form.


IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs [Design]*
Start Page
Toolbox
IP Connection Demo
ok
label1
mainMenu1
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Ready
0, 0
237 x 20

label1 System.Windows.Forms.Label

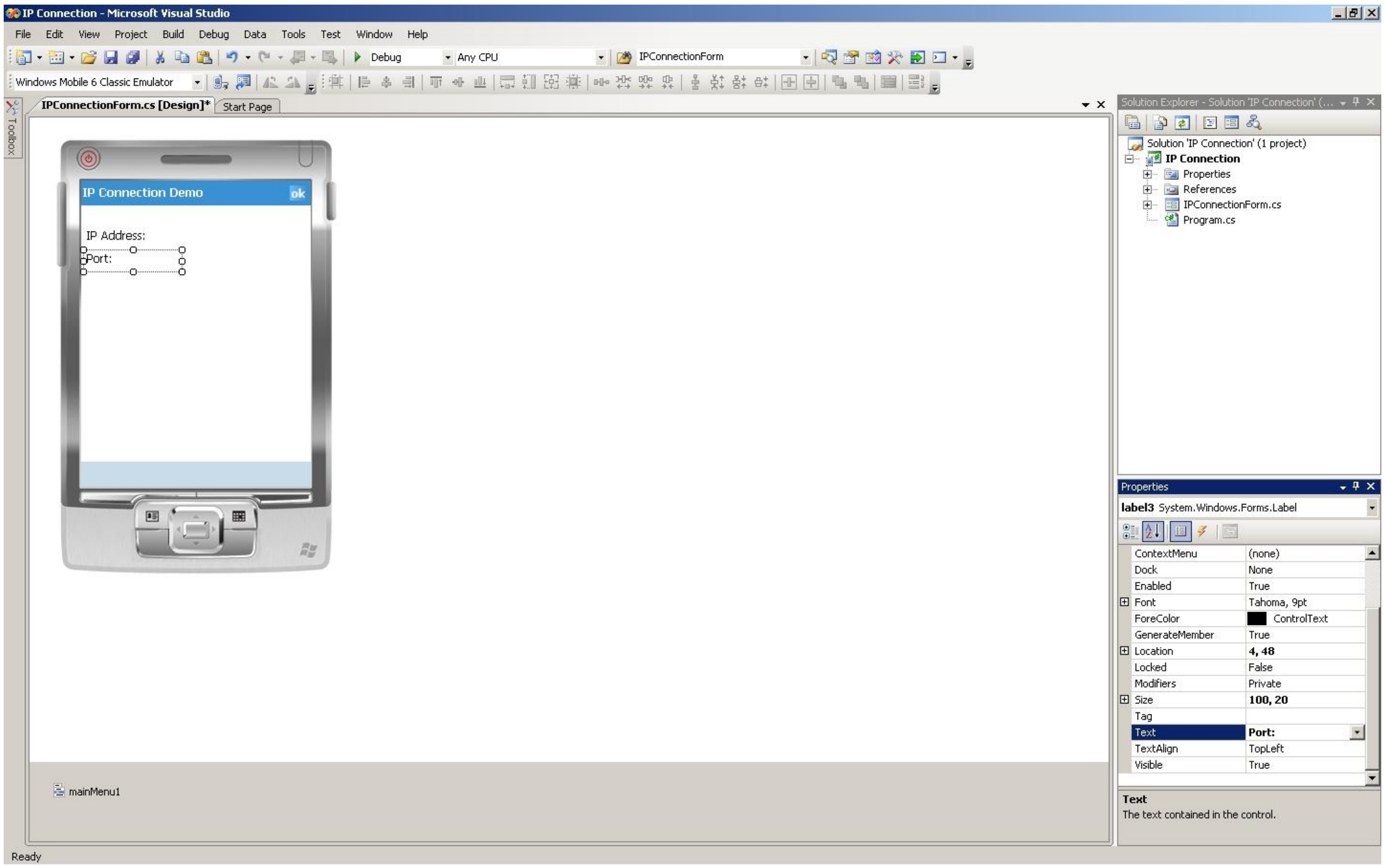
Anchor	Top, Left
BackColor	⬜ 255, 255, 255
ContextMenu	(none)
Dock	None
Enabled	True
Font	Tahoma, 9pt
ForeColor	⬛ ControlText
GenerateMember	True
Location	**0, 0**
Locked	False
Modifiers	Private
Size	**237, 20**
Tag	
Text	**label1**
TextAlign	TopLeft

Text
The text contained in the control.

e. In the Properties window, set Text to blank, and TextAlign to TopCenter.

f. Insert a Label to the bottom left of the status bar, and set its Text to “IP Address:”.

g. Insert another Label to the bottom of the “IP Address:” label, and set its Text to “Port:”. The form layout should now look like this.


IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs [Design]*
Start Page
Toolbox
IP Connection Demo
ok
IP Address:
Port:
mainMenu1
Ready

Solution Explorer - Solution 'IP Connection' (...

Solution 'IP Connection' (1 project)
- IP Connection
  - Properties
  - References
  - IPConnectionForm.cs
  - Program.cs

Properties

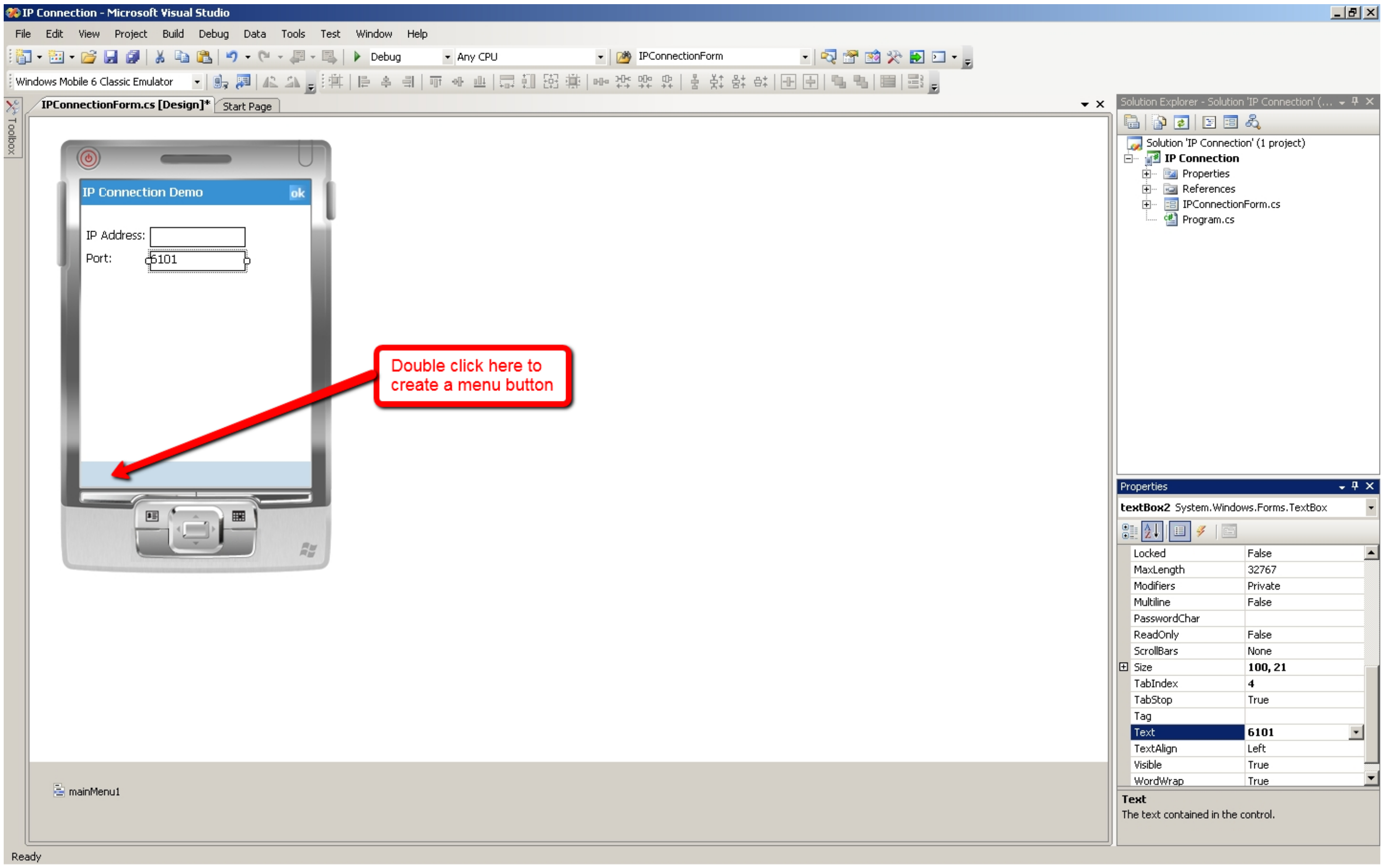
label3 System.Windows.Forms.Label

ContextMenu	(none)
Dock	None
Enabled	True
Font	Tahoma, 9pt
ForeColor	■ ControlText
GenerateMember	True
Location	**4, 48**
Locked	False
Modifiers	Private
Size	**100, 20**
Tag	
Text	**Port:**
TextAlign	TopLeft
Visible	True

Text
The text contained in the control.

h. Drag a TextBox from the Toolbox and position it to the right of the “IP Address:” label.

i. Add another TextBox to the right of the “Port:” label, and set its Text to “6101”. The form layout should now look like this.

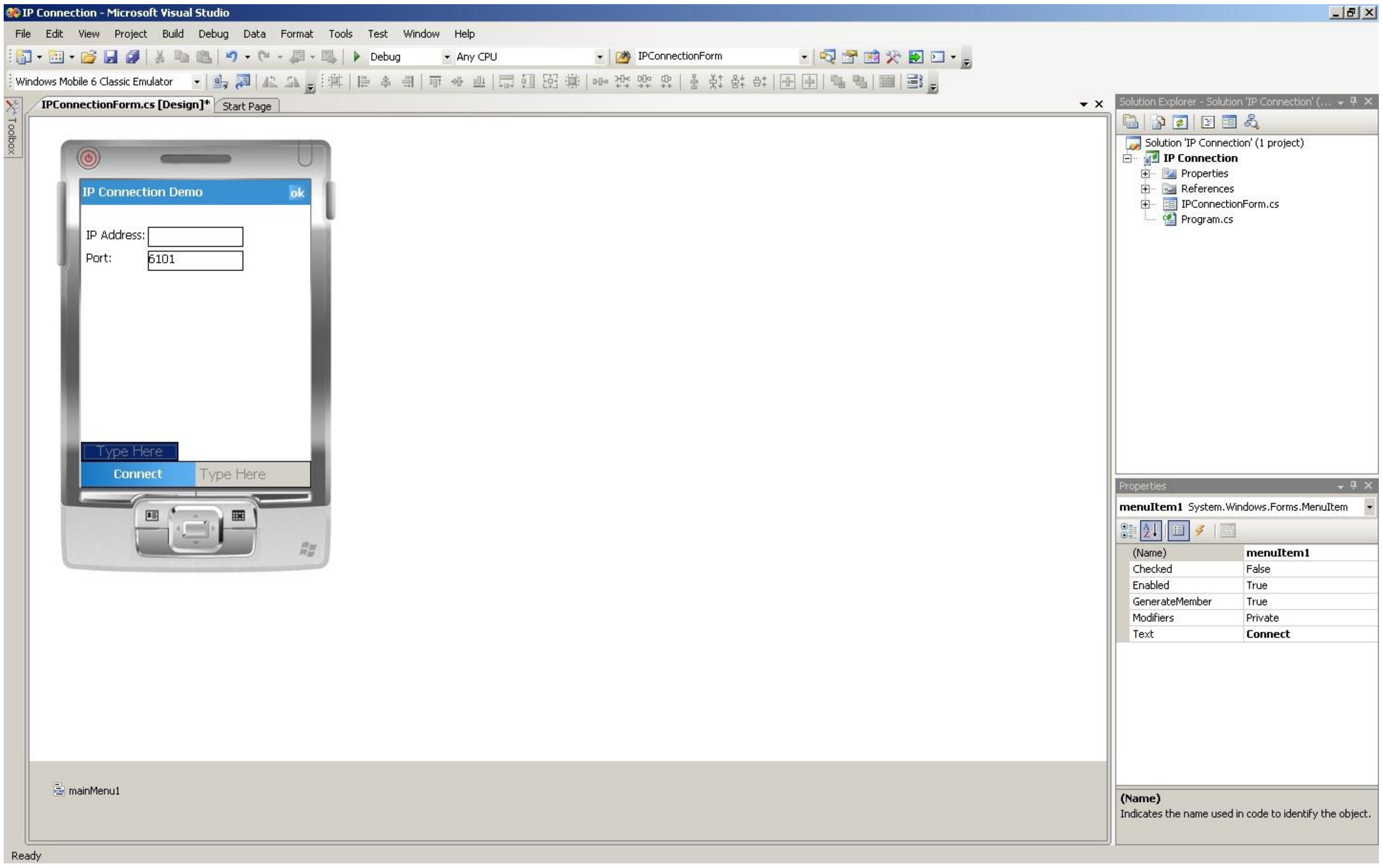

IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs [Design]*
Start Page
IP Connection Demo
ok
IP Address:
Port:
6101
Double click here to create a menu button
mainMenu1
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Ready

textBox2 System.Windows.Forms.TextBox

Locked	False
MaxLength	32767
Modifiers	Private
Multiline	False
PasswordChar	
ReadOnly	False
ScrollBars	None
Size	100, 21
TabIndex	4
TabStop	True
Tag	
Text	6101
TextAlign	Left
Visible	True
WordWrap	True

Text
The text contained in the control.

j. Double click the bottom left of the screen where the light blue bar is to create a menu button. Type “Connect” and hit Enter. The layout should look like this.

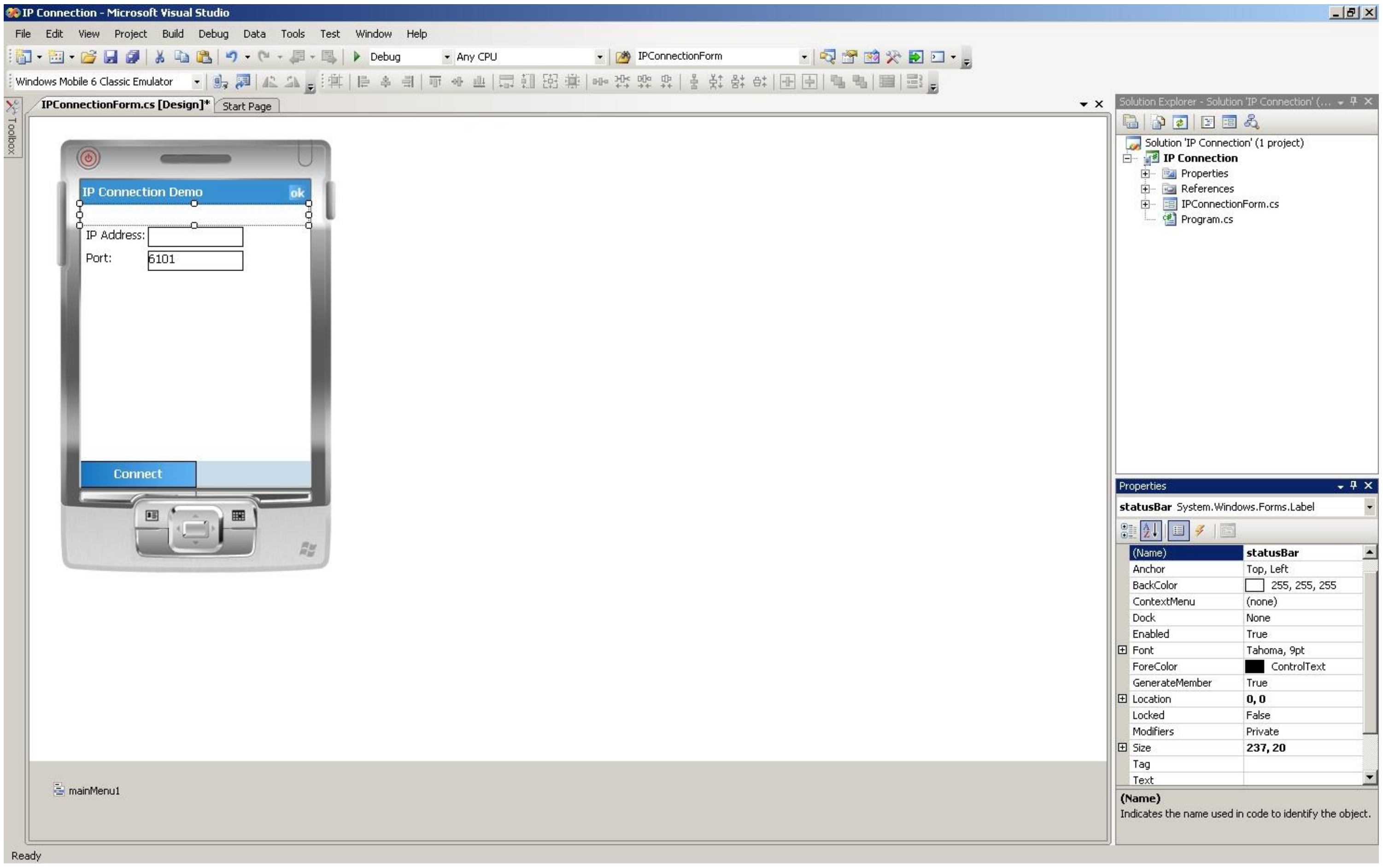

IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs [Design]*
Start Page
IP Connection Demo
ok
IP Address:
Port:
6101
Type Here
Connect
Type Here
mainMenu1
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
menuItem1 System.Windows.Forms.MenuItem
Ready

(Name)	menuItem1
Checked	False
Enabled	True
GenerateMember	True
Modifiers	Private
Text	Connect

(Name)
Indicates the name used in code to identify the object.

4. **Putting Them Altogether.**

   a. Rename the status bar label added in step 3d. Click on the label. On the properties page at the bottom right, enter "statusBar" in the Name field.

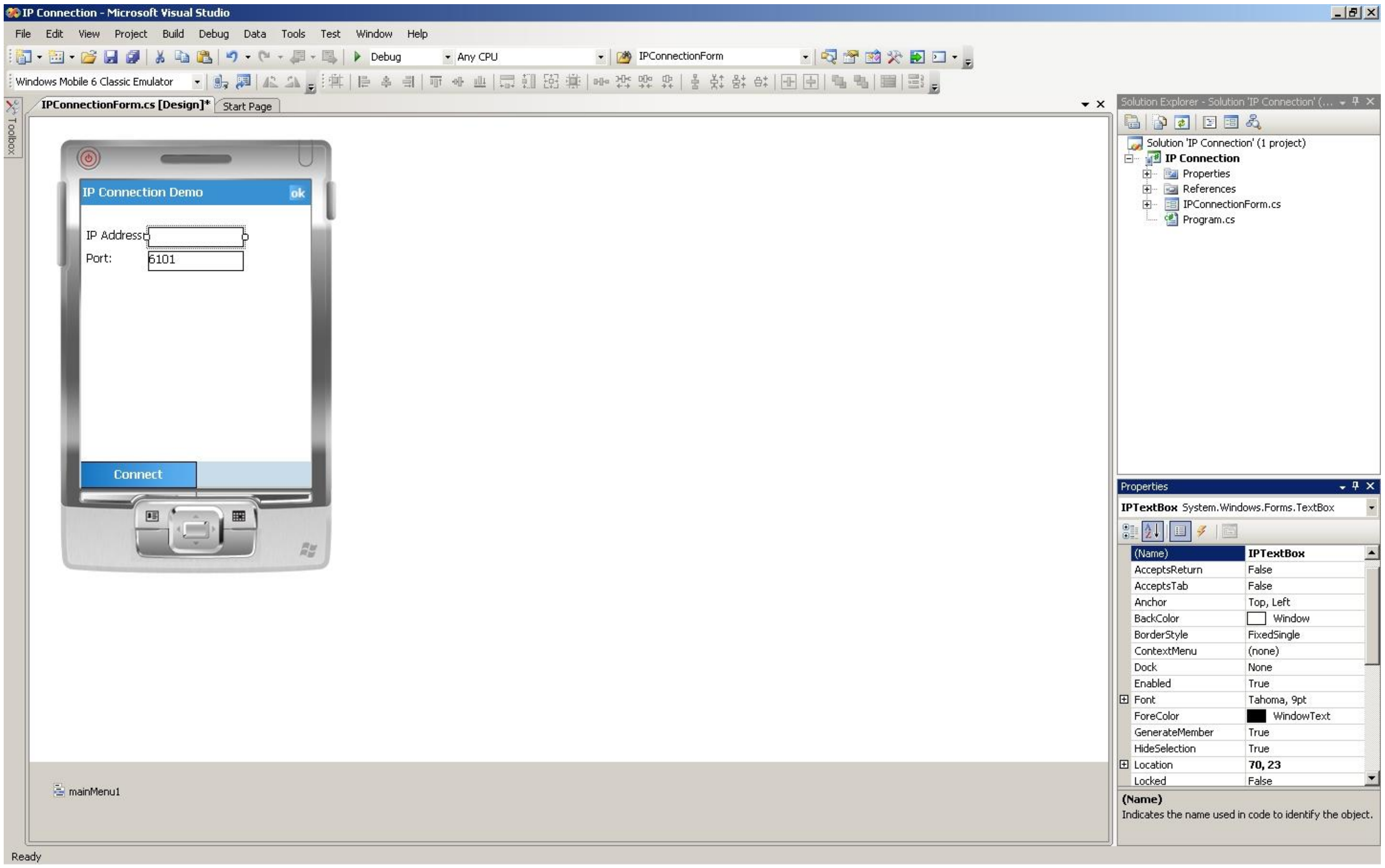

IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs [Design]*
Start Page
Toolbox
IP Connection Demo
ok
IP Address:
Port:
6101
Connect
mainMenu1
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Ready

statusBar System.Windows.Forms.Label

(Name)	statusBar
Anchor	Top, Left
BackColor	255, 255, 255
ContextMenu	(none)
Dock	None
Enabled	True
Font	Tahoma, 9pt
ForeColor	ControlText
GenerateMember	True
Location	0, 0
Locked	False
Modifiers	Private
Size	237, 20
Tag	
Text	

(Name)
Indicates the name used in code to identify the object.

b. Select the text box to the right of the “IP Address:” label. Change its Name to IPTextBox.


IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs [Design]*
Start Page
Toolbox
IP Connection Demo
ok
Connect
mainMenu1
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Ready

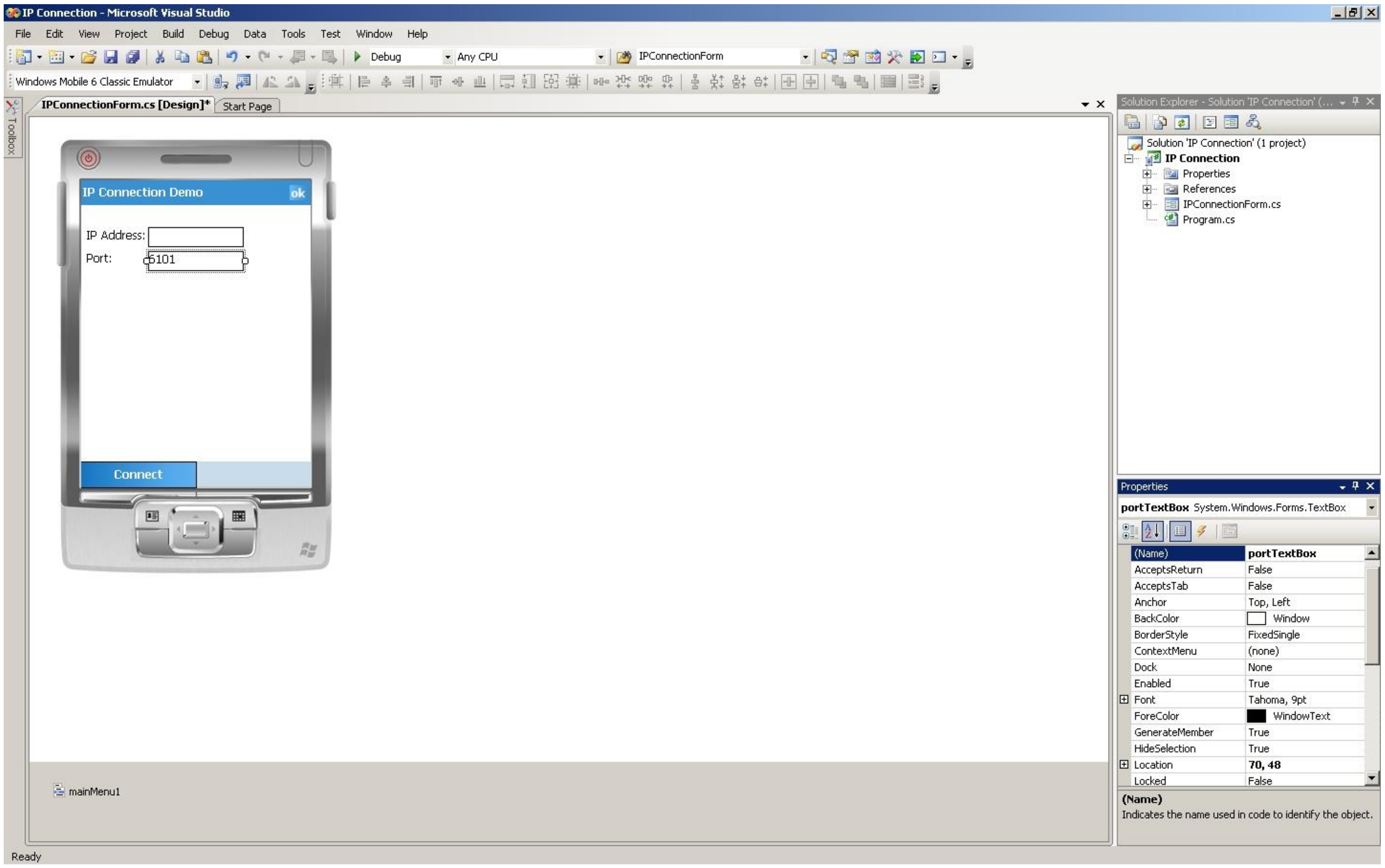
IP Address:
Port: 6101

IPTextBox System.Windows.Forms.TextBox

(Name)	**IPTextBox**
AcceptsReturn	False
AcceptsTab	False
Anchor	Top, Left
BackColor	Window
BorderStyle	FixedSingle
ContextMenu	(none)
Dock	None
Enabled	True
Font	Tahoma, 9pt
ForeColor	WindowText
GenerateMember	True
HideSelection	True
Location	**70, 23**
Locked	False

(Name)
Indicates the name used in code to identify the object.

c. Change the Name of the textbox that is next to the “Port:” label to “portTextBox”.

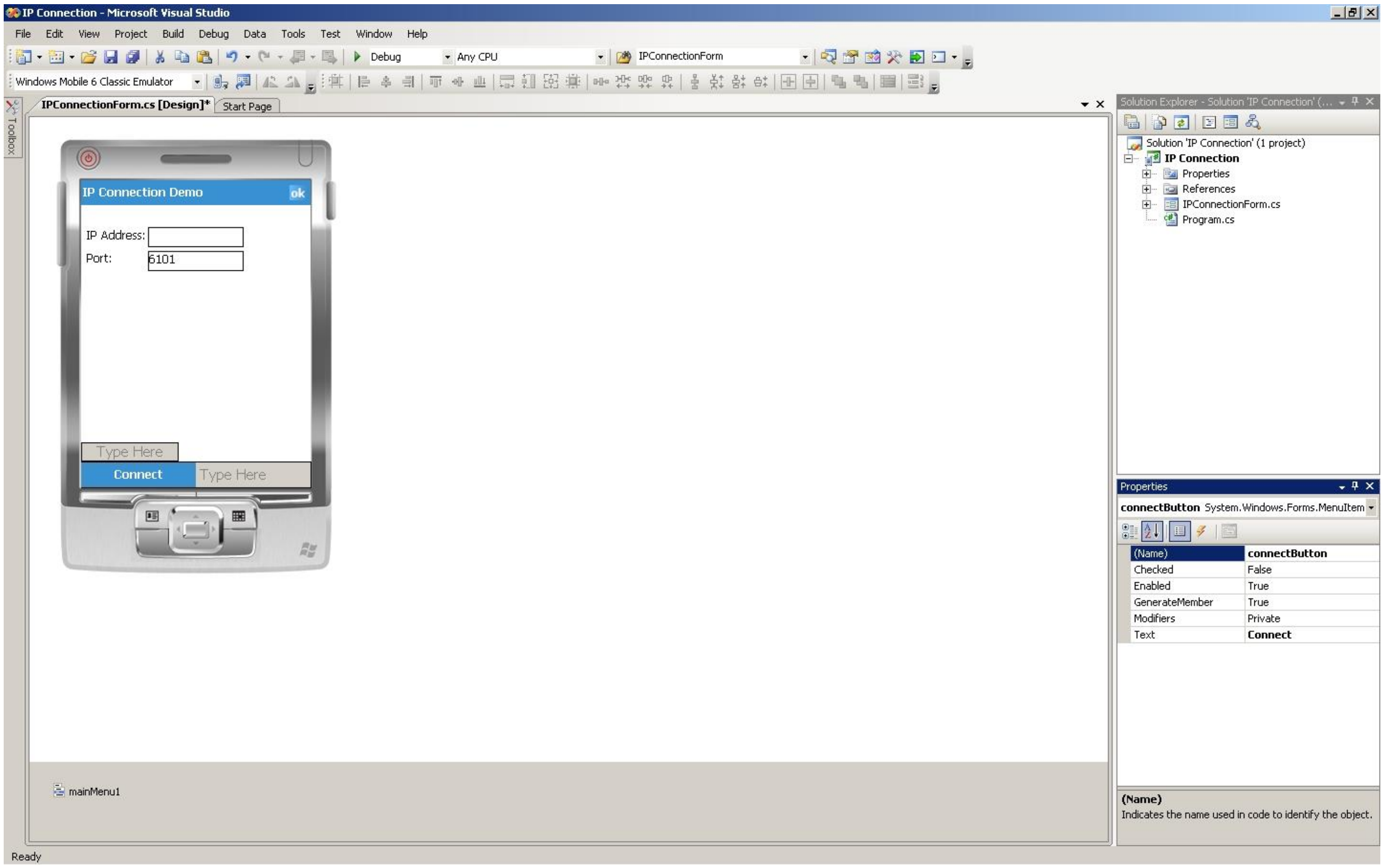

IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs [Design]*
Start Page
IP Connection Demo
ok
IP Address:
Port:
5101
Connect
mainMenu1
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Ready

portTextBox System.Windows.Forms.TextBox

(Name)	portTextBox
AcceptsReturn	False
AcceptsTab	False
Anchor	Top, Left
BackColor	Window
BorderStyle	FixedSingle
ContextMenu	(none)
Dock	None
Enabled	True
Font	Tahoma, 9pt
ForeColor	WindowText
GenerateMember	True
HideSelection	True
Location	70, 48
Locked	False

(Name)
Indicates the name used in code to identify the object.

d. Change the Name of the Connect button to “connectButton”.

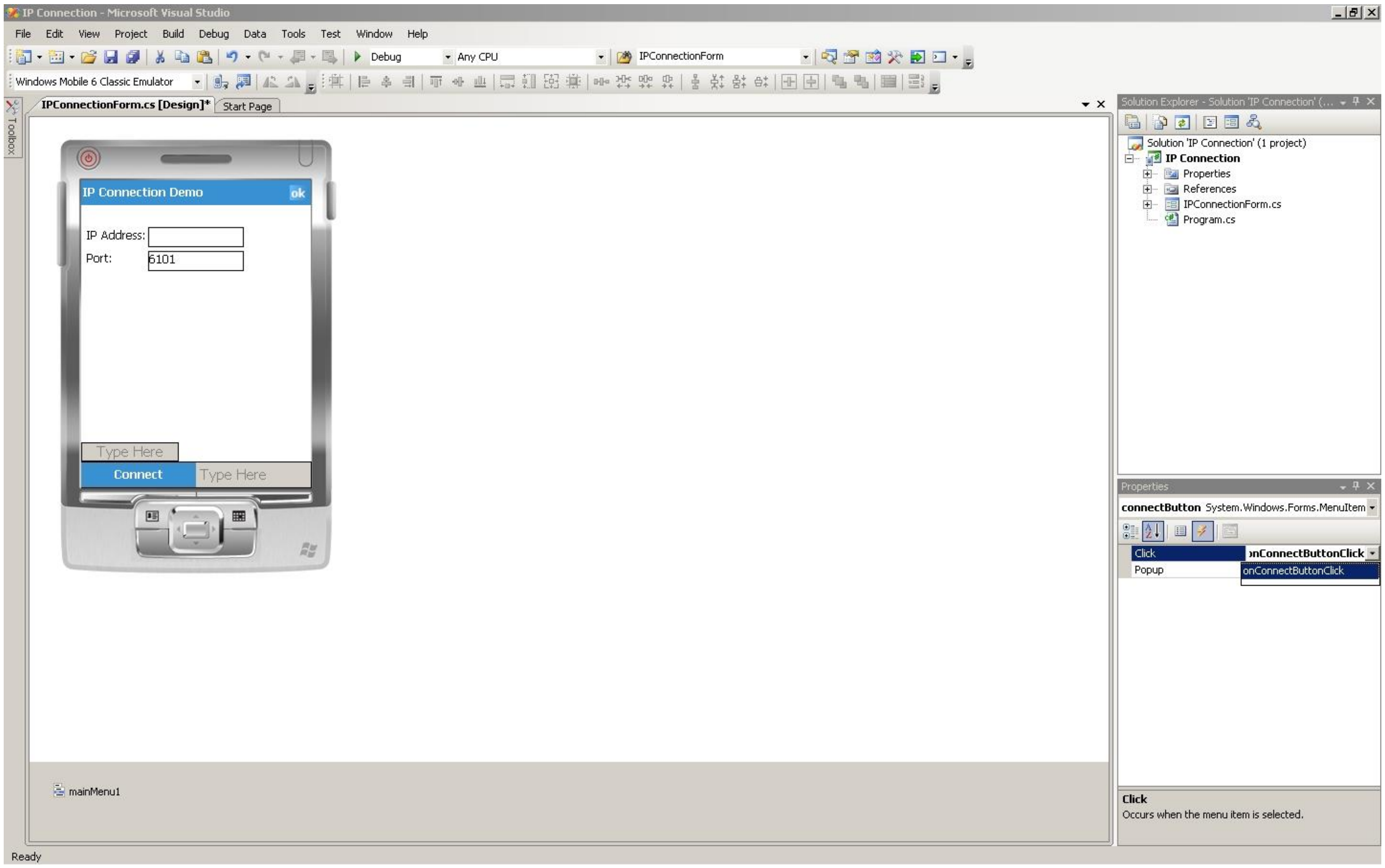

IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs [Design]*
Start Page
Toolbox
IP Connection Demo
ok
IP Address:
Port:
6101
Type Here
Connect
Type Here
mainMenu1
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Ready

connectButton System.Windows.Forms.MenuItem

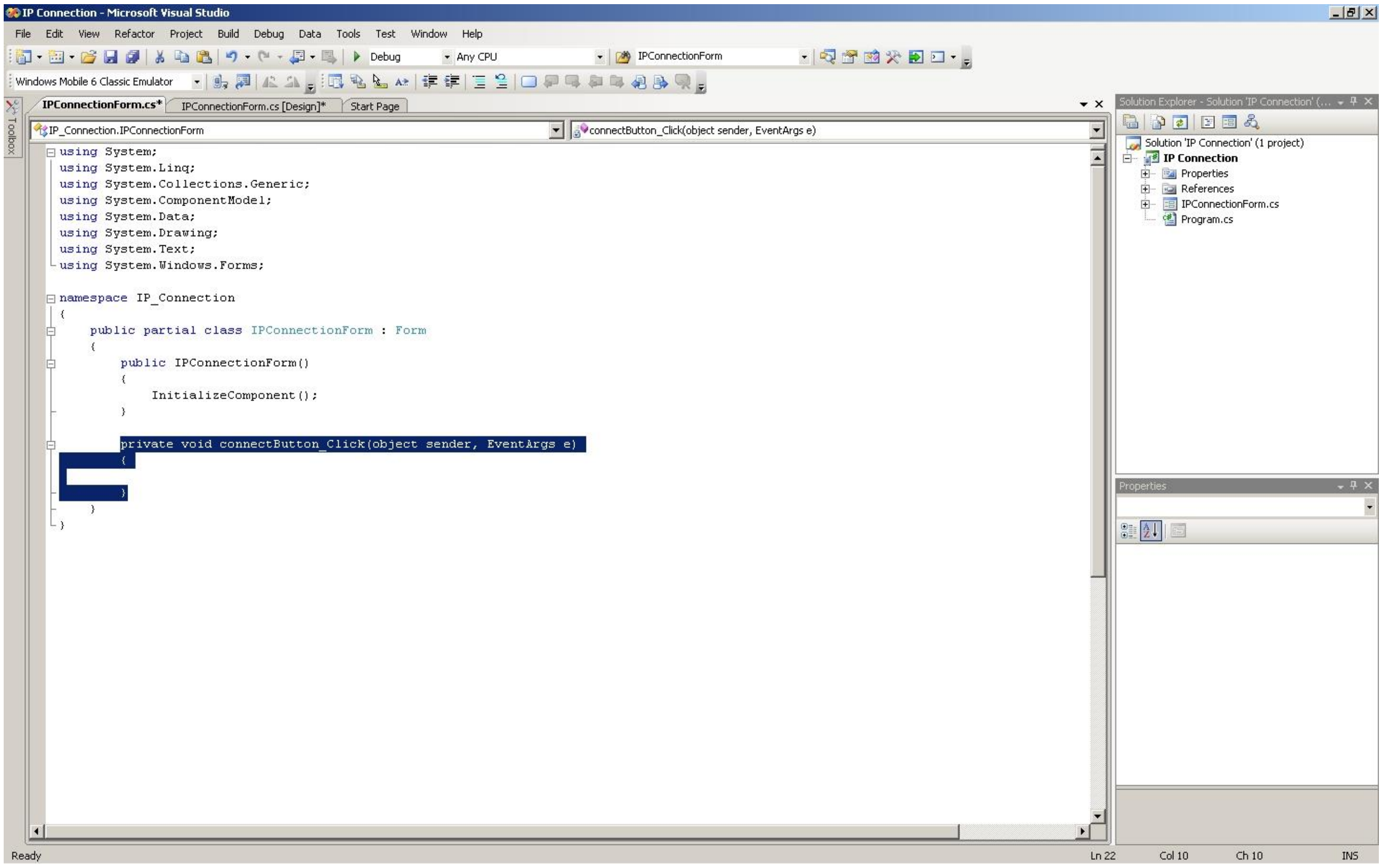
(Name)	connectButton
Checked	False
Enabled	True
GenerateMember	True
Modifiers	Private
Text	Connect

(Name)
Indicates the name used in code to identify the object.

e. With the Connect button still selected, click on the Lightning icon on the Properties panel to display the Events for the button. Click the “Click” event and type “onConnectButtonClick” and hit enter. The source code for the form will be displayed.



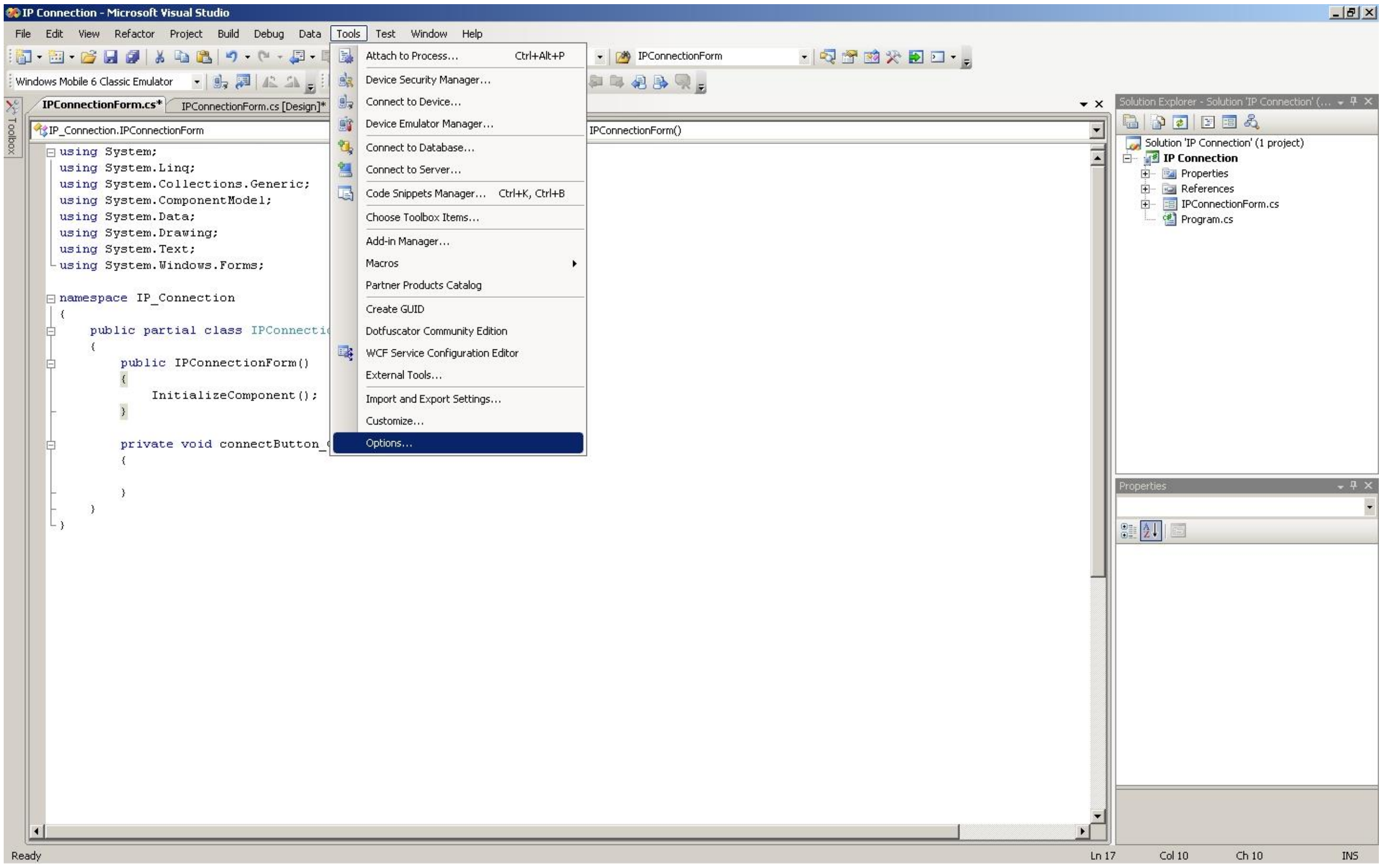
IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs [Design]*
Start Page
Toolbox
IP Connection Demo
ok
IP Address:
Port:
6101
Type Here
Connect
Type Here
mainMenu1
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
connectButton System.Windows.Forms.MenuItem
Click
onConnectButtonClick
Popup
onConnectButtonClick
Click
Occurs when the menu item is selected.
Ready


IP Connection - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs*
IPConnectionForm.cs [Design]*
Start Page
IP_Connection.IPConnectionForm
connectButton_Click(object sender, EventArgs e)
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Ready
Ln 22
Col 10
Ch 10
INS

```
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;

namespace IP_Connection
{
    public partial class IPConnectionForm : Form
    {
        public IPConnectionForm()
        {
            InitializeComponent();
        }

        private void connectButton_Click(object sender, EventArgs e)
        {

        }
    }
}
```

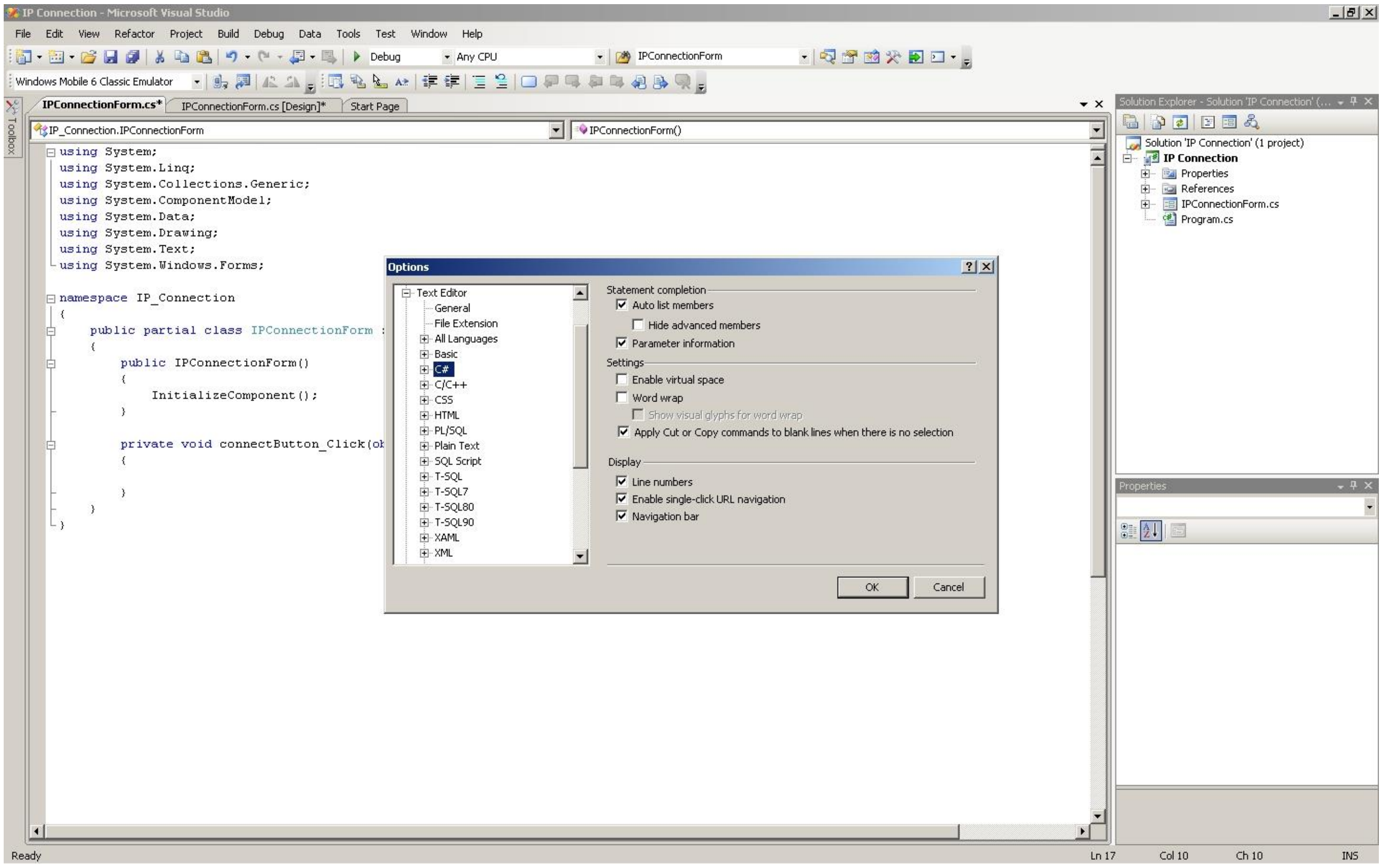
f. At this point, you might want to enable the line number before you insert our sample source code later. To do so, click on Tools -> Options.

The right asset. In the right place. At the right time.


IP Connection - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Window Help
Windows Mobile 6 Classic Emulator
IPConnectionForm
IPConnectionForm.cs*
IPConnectionForm.cs [Design]*
IP_Connection.IPConnectionForm
IPConnectionForm()
Toolbox
Attach to Process... Ctrl+Alt+P
Device Security Manager...
Connect to Device...
Device Emulator Manager...
Connect to Database...
Connect to Server...
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
Add-in Manager...
Macros
Partner Products Catalog
Create GUID
Dotfuscator Community Edition
WCF Service Configuration Editor
External Tools...
Import and Export Settings...
Customize...
Options...
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Ready
Ln 17
Col 10
Ch 10
INS

```
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;

namespace IP_Connection
{
    public partial class IPConnecti
    {
        public IPConnectionForm()
        {
            InitializeComponent();
        }

        private void connectButton_(
        {

        }
    }
}
```

g. An Options dialog is displayed. Expand Text Editor on the left panel, click on C#, and on the right, check the Line numbers option.



```
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;

namespace IP_Connection
{
    public partial class IPConnectionForm :
    {
        public IPConnectionForm()
        {
            InitializeComponent();
        }

        private void connectButton_Click(ob
        {

        }
    }
}
```

Statement completion
- [x] Auto list members
- [ ] Hide advanced members
- [x] Parameter information

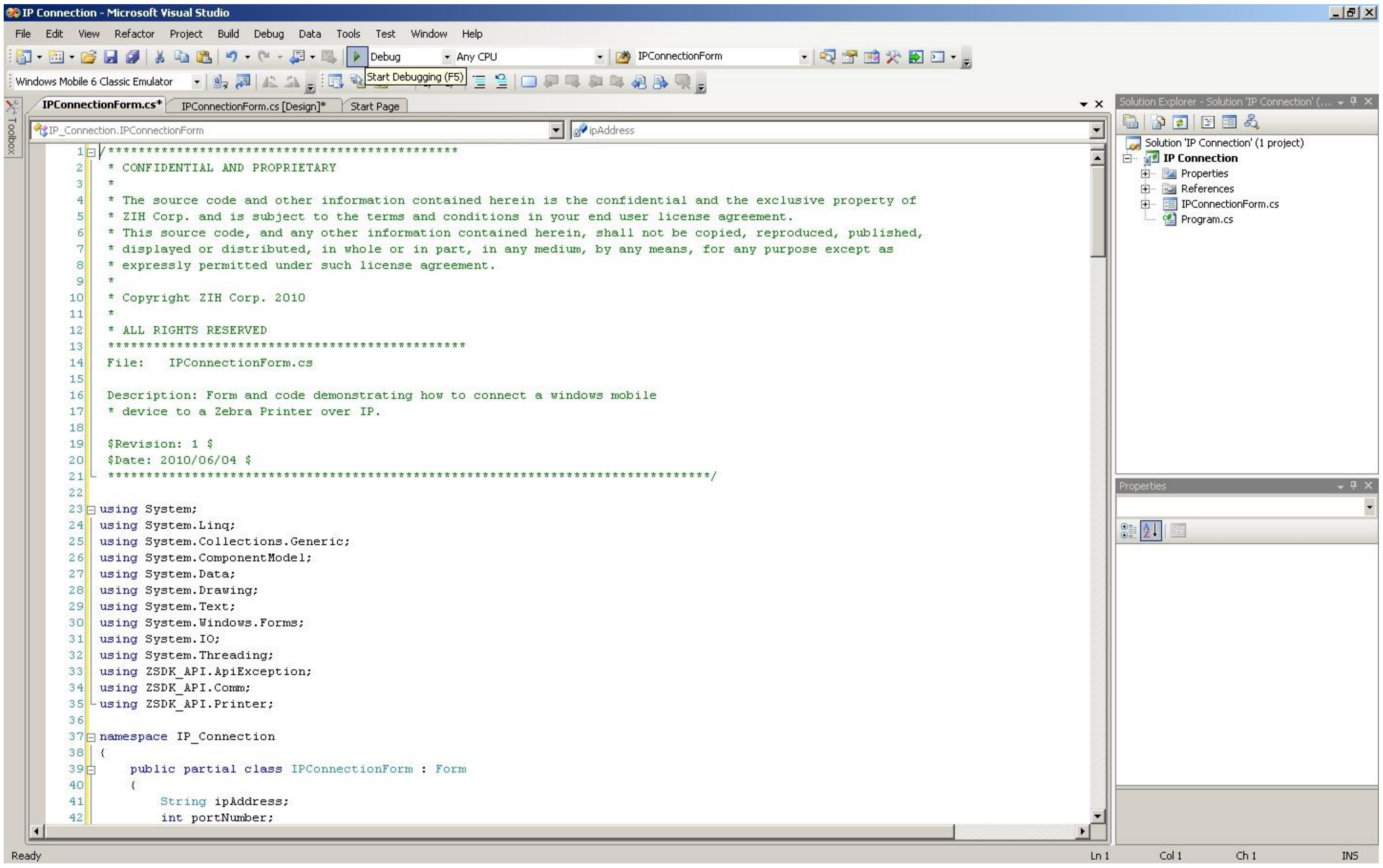
Settings
- [ ] Enable virtual space
- [ ] Word wrap
- [ ] Show visual glyphs for word wrap
- [x] Apply Cut or Copy commands to blank lines when there is no selection

Display
- [x] Line numbers
- [x] Enable single-click URL navigation
- [x] Navigation bar

h. Replace the entire source code in IPConnectionForm.cs with the contents of the IPConnectionForm.cs file from the "Connecting Over IP – Windows Mobile SDK – C#" sample code.

The sample code used in this guide is available on GitHub in the Zebra Technologies repository here. For information on joining the ISV Program contact a Program manager in your area. Details here.

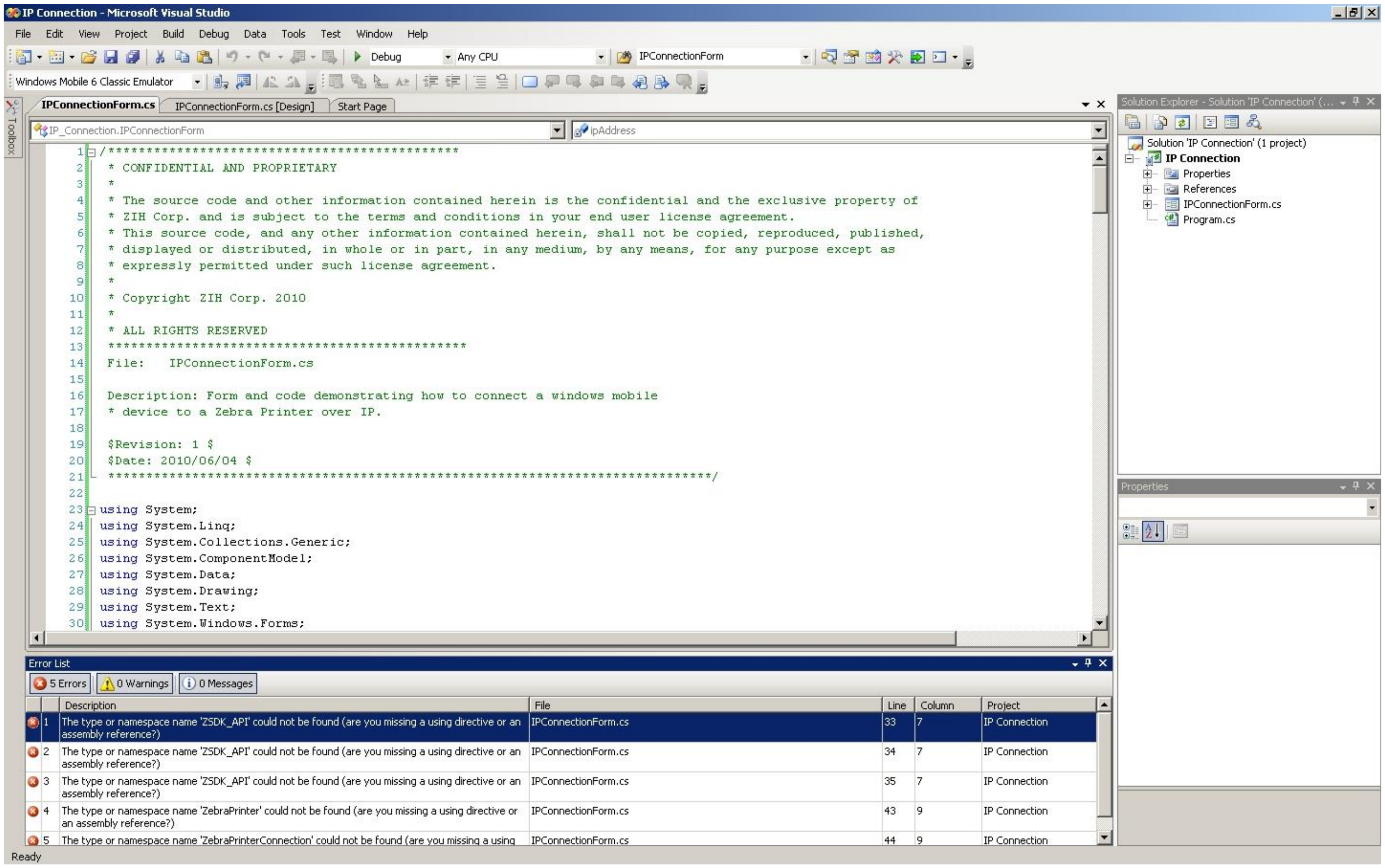
i. Click on the Start Debugging button to build and run the project.


IP Connection - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
Start Debugging (F5)
IPConnectionForm.cs*
IPConnectionForm.cs [Design]*
Start Page
IP_Connection.IPConnectionForm
ipAddress
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Ready
Ln 1
Col 1
Ch 1
INS

```
/**********************************************
 * CONFIDENTIAL AND PROPRIETARY
 *
 * The source code and other information contained herein is the confidential and the exclusive property of
 * ZIH Corp. and is subject to the terms and conditions in your end user license agreement.
 * This source code, and any other information contained herein, shall not be copied, reproduced, published,
 * displayed or distributed, in whole or in part, in any medium, by any means, for any purpose except as
 * expressly permitted under such license agreement.
 *
 * Copyright ZIH Corp. 2010
 *
 * ALL RIGHTS RESERVED
 ***********************************************
 File:   IPConnectionForm.cs

 Description: Form and code demonstrating how to connect a windows mobile
 * device to a Zebra Printer over IP.

 $Revision: 1 $
 $Date: 2010/06/04 $
 ********************************************************************************/

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 System.IO;
using System.Threading;
using ZSDK_API.ApiException;
using ZSDK_API.Comm;
using ZSDK_API.Printer;

namespace IP_Connection
{
    public partial class IPConnectionForm : Form
    {
        String ipAddress;
        int portNumber;
```

j. You will see a list of errors caused by missing dependencies.

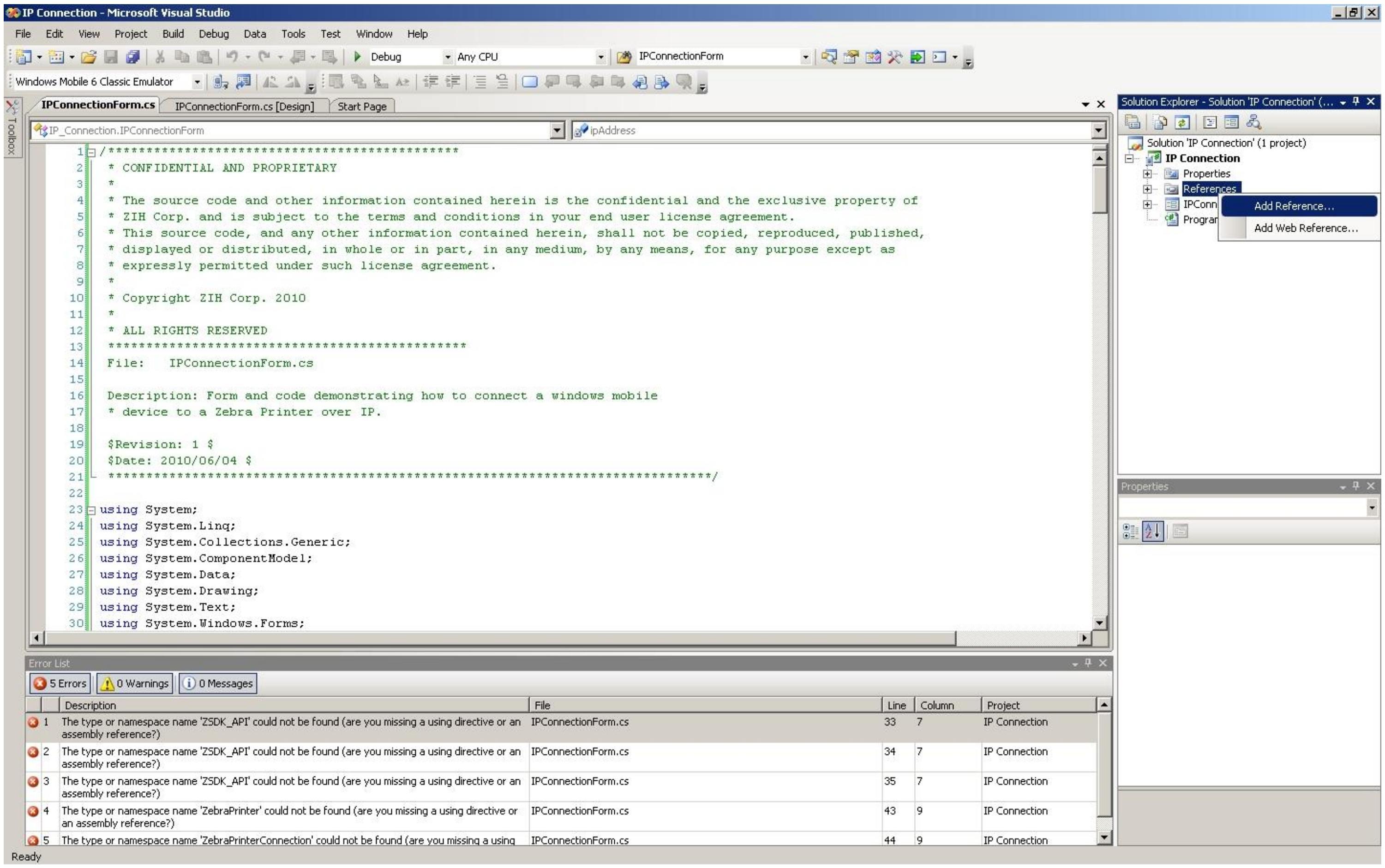
The right asset. In the right place. At the right time.


IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
Toolbox
IPConnectionForm.cs
IPConnectionForm.cs [Design]
Start Page
IP_Connection.IPConnectionForm
ipAddress
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Error List
5 Errors
0 Warnings
0 Messages
Ready

```
/*********************************************
 * CONFIDENTIAL AND PROPRIETARY
 *
 * The source code and other information contained herein is the confidential and the exclusive property of
 * ZIH Corp. and is subject to the terms and conditions in your end user license agreement.
 * This source code, and any other information contained herein, shall not be copied, reproduced, published,
 * displayed or distributed, in whole or in part, in any medium, by any means, for any purpose except as
 * expressly permitted under such license agreement.
 *
 * Copyright ZIH Corp. 2010
 *
 * ALL RIGHTS RESERVED
 *********************************************
 File:   IPConnectionForm.cs

 Description: Form and code demonstrating how to connect a windows mobile
 * device to a Zebra Printer over IP.

 $Revision: 1 $
 $Date: 2010/06/04 $
 ******************************************************************************/

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;
```

	Description	File	Line	Column	Project
1	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	33	7	IP Connection
2	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	34	7	IP Connection
3	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	35	7	IP Connection
4	The type or namespace name 'ZebraPrinter' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	43	9	IP Connection
5	The type or namespace name 'ZebraPrinterConnection' could not be found (are you missing a using	IPConnectionForm.cs	44	9	IP Connection

k. To add the dependencies, on the Solution Explorer on the right, expand the IP Connection project, right-click References, and Add Reference to bring up the Add Reference dialog.



IP Connection - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Classic Emulator
IPConnectionForm.cs
IPConnectionForm.cs [Design]
Start Page
IP_Connection.IPConnectionForm
ipAddress
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConn
Progran
Add Reference...
Add Web Reference...
Properties
Toolbox
Ready

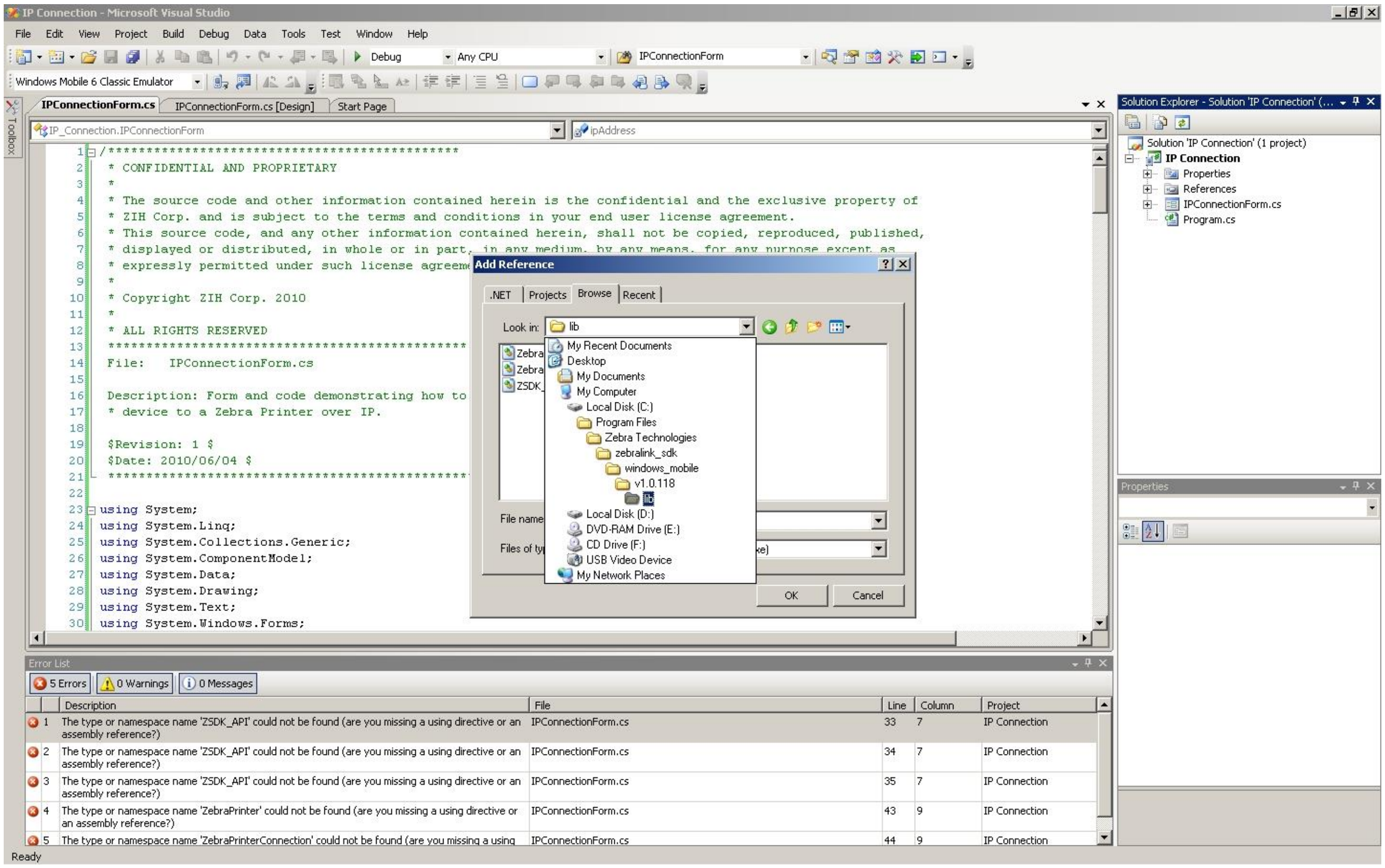
```
/**********************************************
 * CONFIDENTIAL AND PROPRIETARY
 *
 * The source code and other information contained herein is the confidential and the exclusive property of
 * ZIH Corp. and is subject to the terms and conditions in your end user license agreement.
 * This source code, and any other information contained herein, shall not be copied, reproduced, published,
 * displayed or distributed, in whole or in part, in any medium, by any means, for any purpose except as
 * expressly permitted under such license agreement.
 *
 * Copyright ZIH Corp. 2010
 *
 * ALL RIGHTS RESERVED
 **********************************************
 File:   IPConnectionForm.cs

 Description: Form and code demonstrating how to connect a windows mobile
 * device to a Zebra Printer over IP.

 $Revision: 1 $
 $Date: 2010/06/04 $
 *******************************************************************************/

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;
```

Error List

5 Errors | 0 Warnings | 0 Messages

	Description	File	Line	Column	Project
1	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	33	7	IP Connection
2	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	34	7	IP Connection
3	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	35	7	IP Connection
4	The type or namespace name 'ZebraPrinter' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	43	9	IP Connection
5	The type or namespace name 'ZebraPrinterConnection' could not be found (are you missing a using	IPConnectionForm.cs	44	9	IP Connection

l. Select the “Browse” tab. Assuming you have the Link-OS Multiplatform SDK installed at the default location, browse to C:\Program Files\Zebra Technologies\zebralink_sdk\windows_mobile\v1.0.118\lib.



```
/***********************************************
 * CONFIDENTIAL AND PROPRIETARY
 *
 * The source code and other information contained herein is the confidential and the exclusive property of
 * ZIH Corp. and is subject to the terms and conditions in your end user license agreement.
 * This source code, and any other information contained herein, shall not be copied, reproduced, published,
 * displayed or distributed, in whole or in part, in any medium, by any means, for any purpose except as
 * expressly permitted under such license agreeme
 *
 * Copyright ZIH Corp. 2010
 *
 * ALL RIGHTS RESERVED
 ***********************************************
 File:   IPConnectionForm.cs

 Description: Form and code demonstrating how to
 * device to a Zebra Printer over IP.

 $Revision: 1 $
 $Date: 2010/06/04 $
 ***********************************************

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;
```

Error List

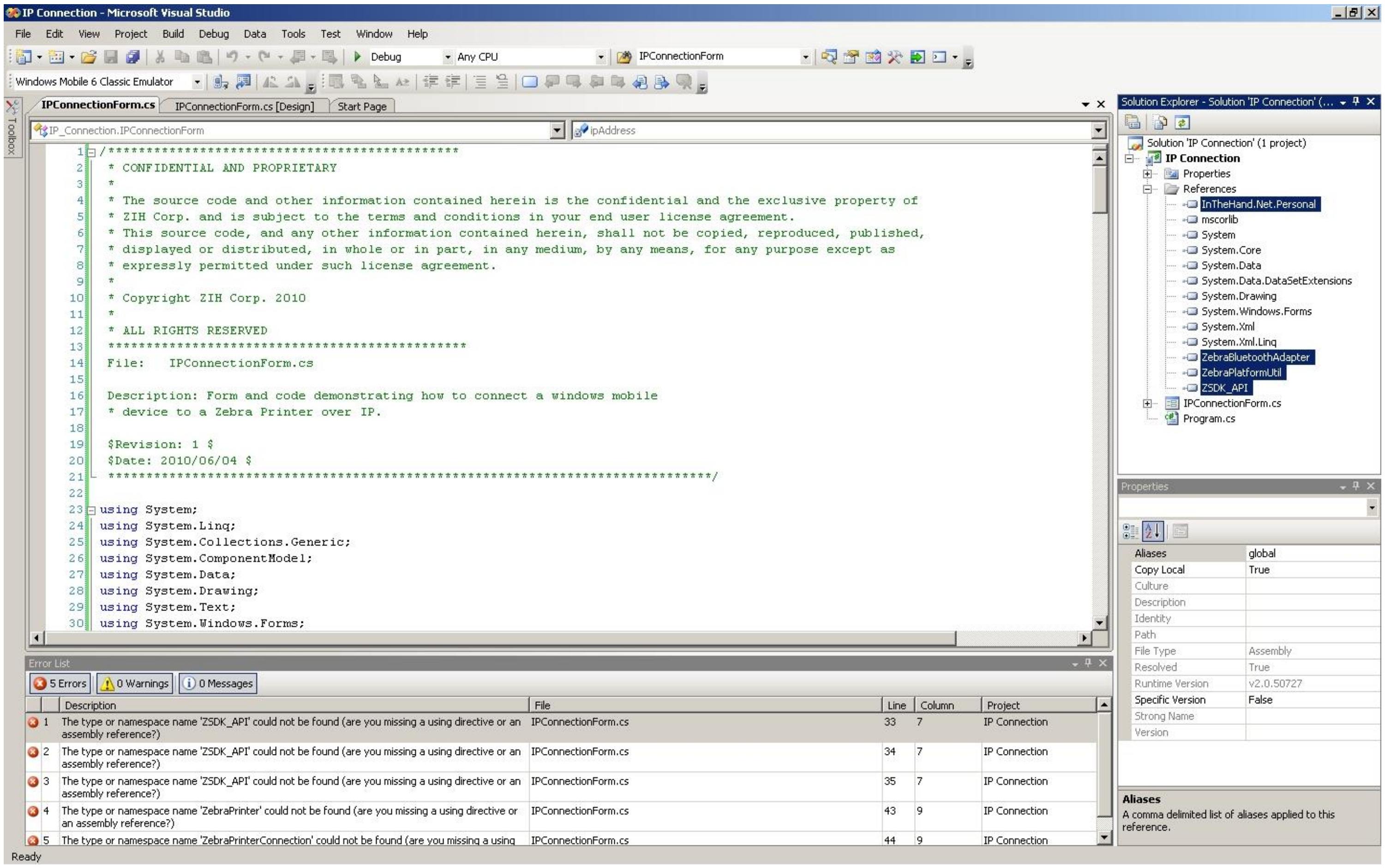
5 Errors | 0 Warnings | 0 Messages

	Description	File	Line	Column	Project
1	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	33	7	IP Connection
2	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	34	7	IP Connection
3	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	35	7	IP Connection
4	The type or namespace name 'ZebraPrinter' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	43	9	IP Connection
5	The type or namespace name 'ZebraPrinterConnection' could not be found (are you missing a using	IPConnectionForm.cs	44	9	IP Connection

m. Select all three dll files – ZebraBluetoothAdapter.dll, ZebraPlatformUtil.dll and ZSDK_API.dll – and click ok.

n. Add another reference. This time, browse to C:\Program Files\Zebra Technologies\zebralink_sdk\windows_mobile\v1.0.118\third_party, select InTheHand.Net.Personal.dll and click ok.

o. On the Solution Explorer, you should now see four new dll files added under References.



IP Connection - Microsoft Visual Studio

```
/***********************************************
 * CONFIDENTIAL AND PROPRIETARY
 *
 * The source code and other information contained herein is the confidential and the exclusive property of
 * ZIH Corp. and is subject to the terms and conditions in your end user license agreement.
 * This source code, and any other information contained herein, shall not be copied, reproduced, published,
 * displayed or distributed, in whole or in part, in any medium, by any means, for any purpose except as
 * expressly permitted under such license agreement.
 *
 * Copyright ZIH Corp. 2010
 *
 * ALL RIGHTS RESERVED
 ***********************************************
 File:   IPConnectionForm.cs

 Description: Form and code demonstrating how to connect a windows mobile
 * device to a Zebra Printer over IP.

 $Revision: 1 $
 $Date: 2010/06/04 $
 ***********************************************************************************/

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;
```

Solution 'IP Connection' (1 project)
- IP Connection
  - Properties
  - References
    - InTheHand.Net.Personal
    - mscorlib
    - System
    - System.Core
    - System.Data
    - System.Data.DataSetExtensions
    - System.Drawing
    - System.Windows.Forms
    - System.Xml
    - System.Xml.Linq
    - ZebraBluetoothAdapter
    - ZebraPlatformUtil
    - ZSDK_API
  - IPConnectionForm.cs
  - Program.cs

Aliases	global
Copy Local	True
Culture	
Description	
Identity	
Path	
File Type	Assembly
Resolved	True
Runtime Version	v2.0.50727
Specific Version	False
Strong Name	
Version	

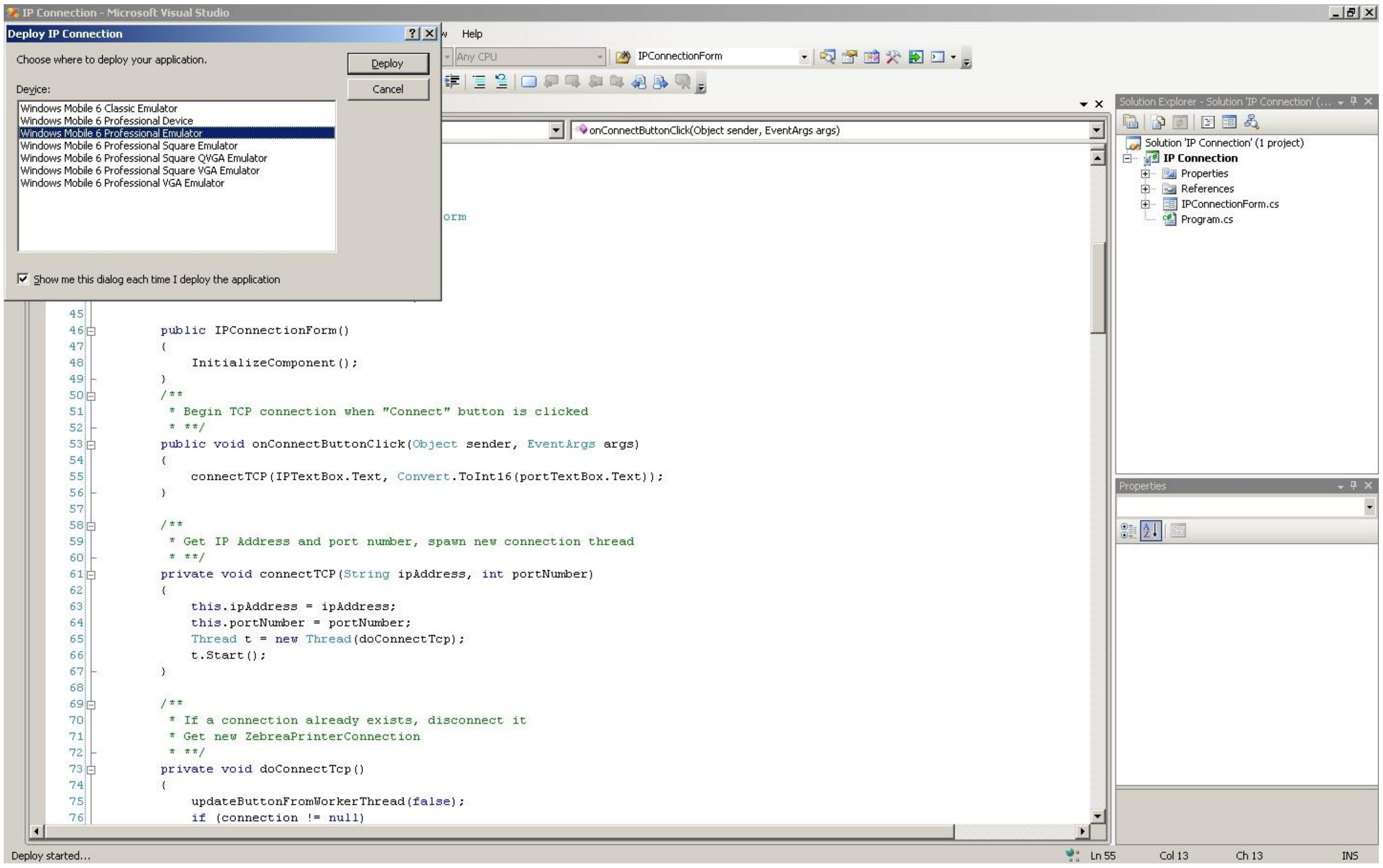
Aliases
A comma delimited list of aliases applied to this reference.

Error List: 5 Errors | 0 Warnings | 0 Messages

	Description	File	Line	Column	Project
1	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	33	7	IP Connection
2	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	34	7	IP Connection
3	The type or namespace name 'ZSDK_API' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	35	7	IP Connection
4	The type or namespace name 'ZebraPrinter' could not be found (are you missing a using directive or an assembly reference?)	IPConnectionForm.cs	43	9	IP Connection
5	The type or namespace name 'ZebraPrinterConnection' could not be found (are you missing a using	IPConnectionForm.cs	44	9	IP Connection

Ready

p. Click on the Start Debugging button to build and run the project. There should be no errors reported. A Deploy IP Connection dialog should appear.


IP Connection - Microsoft Visual Studio
Help
Any CPU
IPConnectionForm
onConnectButtonClick(Object sender, EventArgs args)
Solution Explorer - Solution 'IP Connection' (...
Solution 'IP Connection' (1 project)
IP Connection
Properties
References
IPConnectionForm.cs
Program.cs
Properties
Deploy
Cancel

Deploy IP Connection

Choose where to deploy your application.

Device:

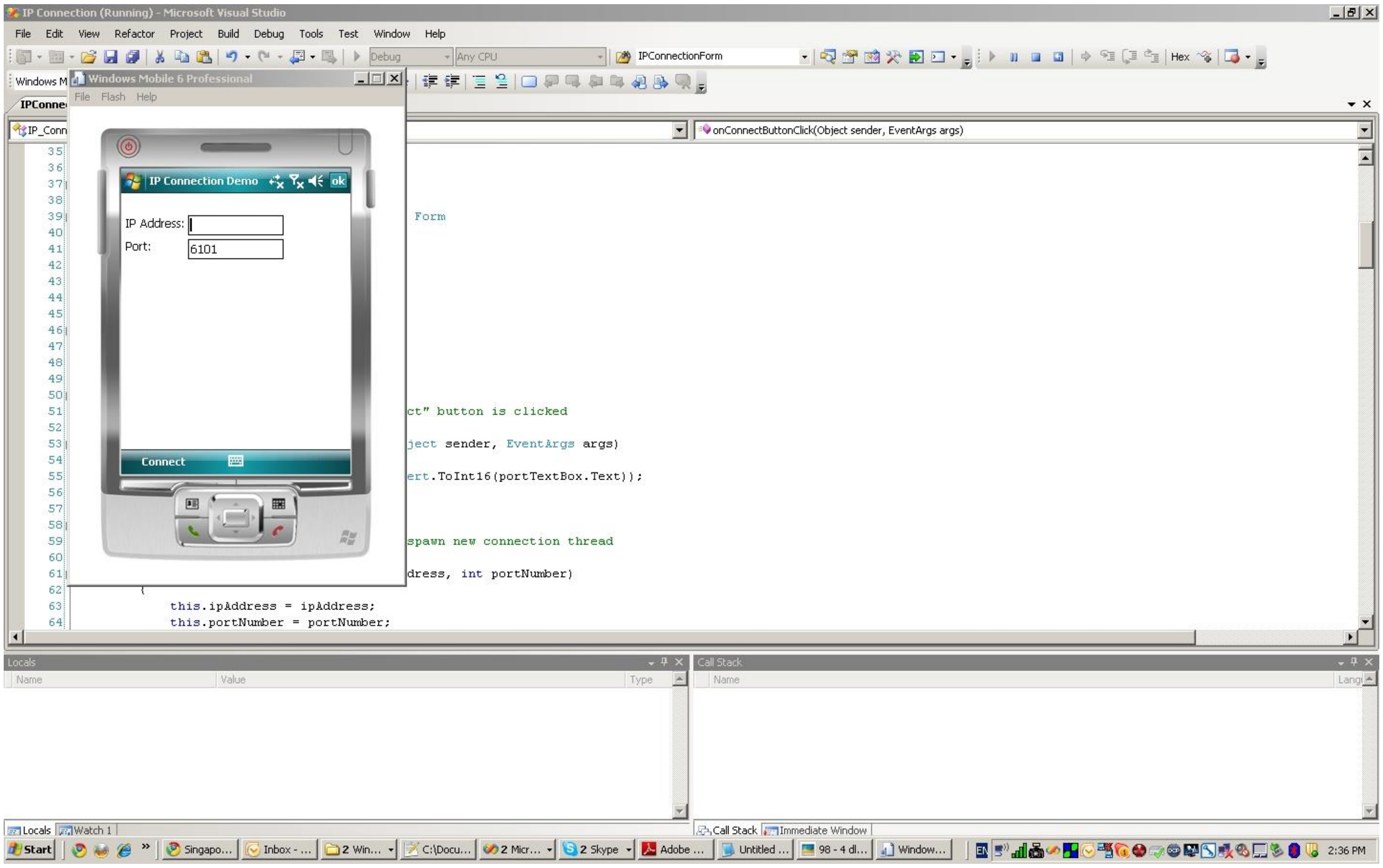
```
Windows Mobile 6 Classic Emulator
Windows Mobile 6 Professional Device
Windows Mobile 6 Professional Emulator
Windows Mobile 6 Professional Square Emulator
Windows Mobile 6 Professional Square QVGA Emulator
Windows Mobile 6 Professional Square VGA Emulator
Windows Mobile 6 Professional VGA Emulator
```

[✓] Show me this dialog each time I deploy the application

```
        public IPConnectionForm()
        {
            InitializeComponent();
        }
        /**
         * Begin TCP connection when "Connect" button is clicked
         * **/
        public void onConnectButtonClick(Object sender, EventArgs args)
        {
            connectTCP(IPTextBox.Text, Convert.ToInt16(portTextBox.Text));
        }

        /**
         * Get IP Address and port number, spawn new connection thread
         * **/
        private void connectTCP(String ipAddress, int portNumber)
        {
            this.ipAddress = ipAddress;
            this.portNumber = portNumber;
            Thread t = new Thread(doConnectTcp);
            t.Start();
        }

        /**
         * If a connection already exists, disconnect it
         * Get new ZebreaPrinterConnection
         * **/
        private void doConnectTcp()
        {
            updateButtonFromWorkerThread(false);
            if (connection != null)
```

Deploy started...   Ln 55   Col 13   Ch 13   INS

q. Select Windows Mobile 6 Professional Emulator, and click Deploy. This launches the application in the emulator. It will take a while for the emulated device to fully start up and load the application.


IP Connection (Running) - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Tools Test Window Help
Debug
Any CPU
IPConnectionForm
Windows Mobile 6 Professional
File Flash Help
IP Connection Demo
ok
Connect
onConnectButtonClick(Object sender, EventArgs args)
Locals
Call Stack
Immediate Window

IP Address:
Port: 6101

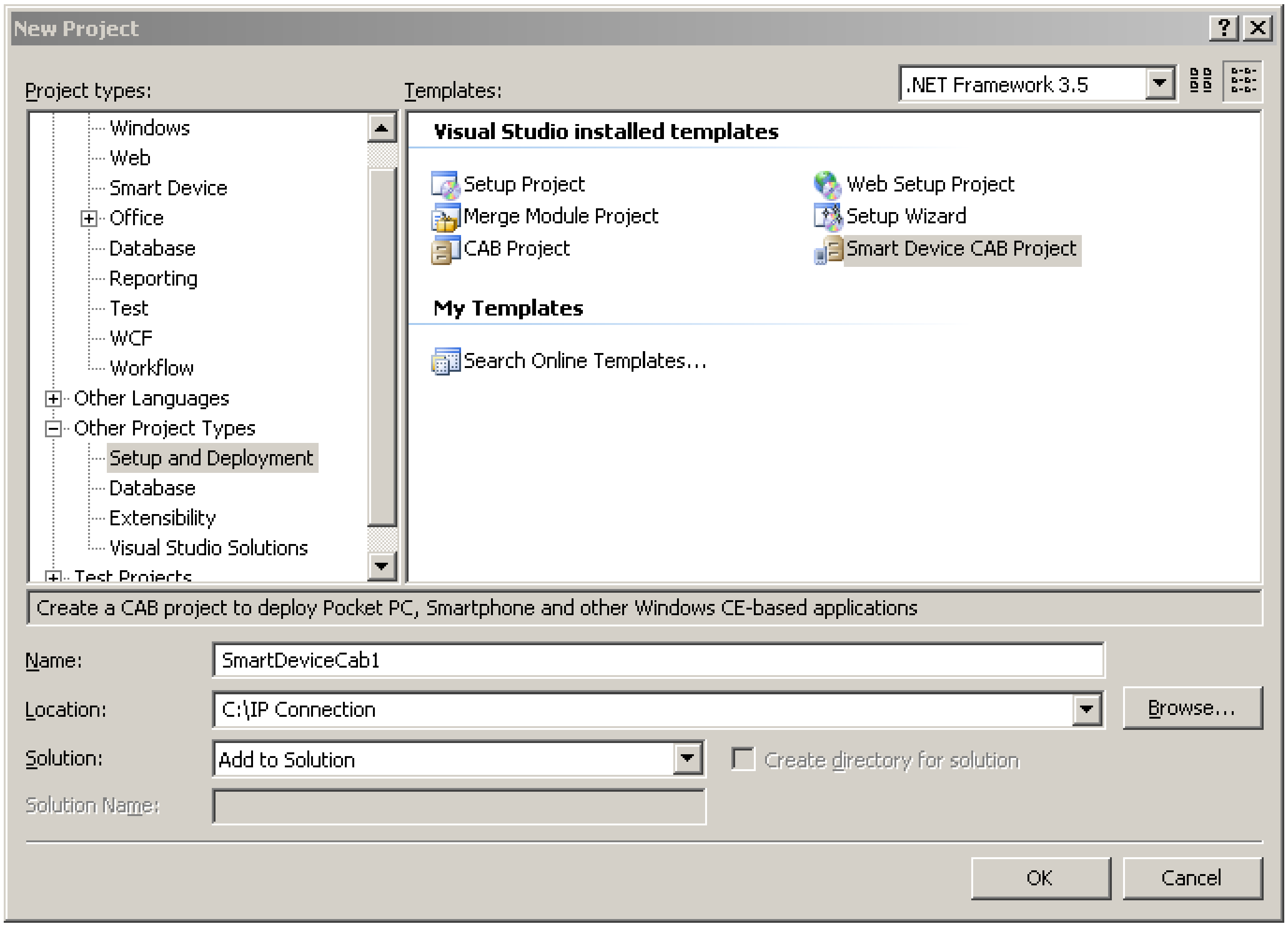
```
                                            Form

                                            ct" button is clicked

                                            ject sender, EventArgs args)

                                            ert.ToInt16(portTextBox.Text));

                                            spawn new connection thread

                                            dress, int portNumber)
        {
            this.ipAddress = ipAddress;
            this.portNumber = portNumber;
```

r. Congratulations. You now have your first Zebra mobile application running on an emulator! To exit the application, simply go to File -> Exit. When prompted to save emulator state, click No.

Deploying Your Mobile Application

After your Smart Device project is built and completed, it may be packaged into a CAB file for deployment on a Smart Device.

1. While your current Smart Device project is open, add a new Smart Device CAB Project to it. Select File -> New Project -> Other Project Types -> Setup and Deployment -> Smart Device CAB Project.



Create a CAB project to deploy Pocket PC, Smartphone and other Windows CE-based applications

Name: SmartDeviceCab1

Location: C:\IP Connection

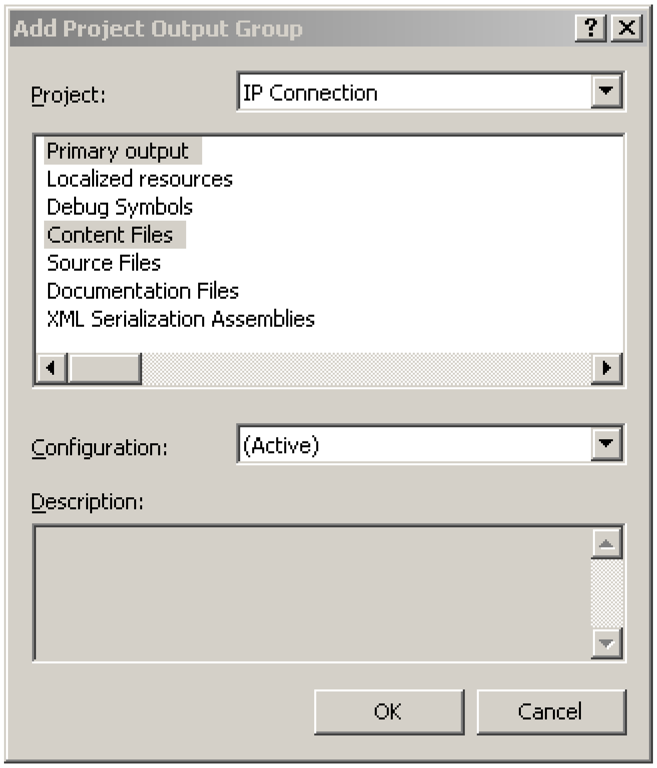
Solution: Add to Solution

Solution Name:

2. Provide a name for your CAB file.

3. Next to Solution, select ”Add to Solution”. This will create a CAB project in your current workspace.

4. Once the CAB project has been created, right click the CAB project and select Add -> Project Output. The following dialog will appear:



Project: IP Connection

Primary output
Localized resources
Debug Symbols
Content Files
Source Files
Documentation Files
XML Serialization Assemblies

Configuration: (Active)

5. Select "Content Files". This will add all your project dependencies under 'Content Files'.

6. Select "Primary output". This will add your target application to the CAB file.

7. Right click your CAB project then select "Build" or "Rebuild". This will build your CAB file.

8. Copy this file to your device and run it to install your application and related dependencies.

See Also

- For further information on ZPL, see the ZPL Programming Guide on the Zebra website
- For further information on CPCL, see the CPCL Programming Guide on the Zebra website.
- For any further information, sample code and solutions or to request further content, visit the Zebra Developer Portal.

Document Control

Version	Date	Description
1.0	9-Nov-10	Initial Release
1.1	28-Feb-11	See Also section updates
1.2	23-Jun-11	Updated disclaimer comments
2.0	01-Jul-12	Updated Chinese link into document
3.0	16-Sep-12	Updated Spanish and Portuguese link into document
4.0	2nd October 2017	Updated URLs

All links and information correct at time of writing
Created for the Zebra Global ISV Program by Zebra Development Services