



i.MX31 SOM-LV Windows Embedded CE 6.0

Lab 3: Using the Device as a Web and File Server

BSP Documentation

Logic // Products
Published: June 2009

This document contains valuable proprietary and confidential information and the attached file contains source code, ideas, and techniques that are owned by Logic Product Development Company (collectively "Logic's Proprietary Information"). Logic's Proprietary Information may not be used by or disclosed to any third party except under written license from Logic Product Development Company.

Logic Product Development Company makes no representation or warranties of any nature or kind regarding Logic's Proprietary Information or any products offered by Logic Product Development Company. Logic's Proprietary Information is disclosed herein pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase or lease equipment. The only warranties made by Logic Product Development Company, if any, with respect to any products described in this document are set forth in such license or agreement. Logic Product Development Company shall have no liability of any kind, express or implied, arising out of the use of the Information in this document, including direct, indirect, special or consequential damages.

Logic Product Development Company may have patents, patent applications, trademarks, copyrights, trade secrets, or other intellectual property rights pertaining to Logic's Proprietary Information and products described in this document (collectively "Logic's Intellectual Property"). Except as expressly provided in any written license or agreement from Logic Product Development Company, this document and the information contained therein does not create any license to Logic's Intellectual Property.

The Information contained herein is subject to change without notice. Revisions may be issued regarding changes and/or additions.

© Copyright 2009, Logic Product Development Company. All Rights Reserved.

Revision History

REV	EDITOR	DESCRIPTION	APPROVAL	DATE
A	MH	Initial release	JCA	06/09/09

Table of Contents

1	Introduction.....	1
1.1	Learning Objectives	1
1.2	Prerequisites	1
1.3	Lab Setup.....	1
2	Creating the Platform Image.....	1
2.1	Create a new OS Design	1
2.2	Configuring Telnet.....	15
3	Download the Platform Image.....	17
3.1	Configure Download/Debug Transport	17
3.2	Download the Operating System	22
4	Setting up the File Server	22
4.1	Using the Remote Web Administration Interface	22
4.2	Create and Configure a Share	24
5	Setting up the Web Server.....	26
5.1	Create a Web directory	26
5.2	Log in to the Web Server's Configuration Interface	27
6	Setting up a Remote Display.....	32
6.1	Connect to the Telnet server.....	32
7	Creating an ISAPI Extension DLL	34
7.1	Export an SDK	34
7.2	Create the ISAPI DLL Project	35
7.3	Export Functions from the DLL	39
7.4	Using the ISAPI Extension.....	41
7.5	Modifying the ISAPI Extension.....	43
8	Summary	44

1 Introduction

1.1 Learning Objectives

- Learn how to enable the Web Server on the device
- Learn how to enable the File Server on the device
- Use the supplied WebAdmin tool to setup the Web Server
- Create a simple ASP web page and include it on the Web Server
- Use the supplied RemoteAdmin tool to setup the File Server

1.2 Prerequisites

Knowledge of the vocabulary used in OS design, Visual Studio, and the Platform Builder Plug-in for Visual Studio

Knowledge of C and C++ languages

1.3 Lab Setup

To complete this lab, you must have:

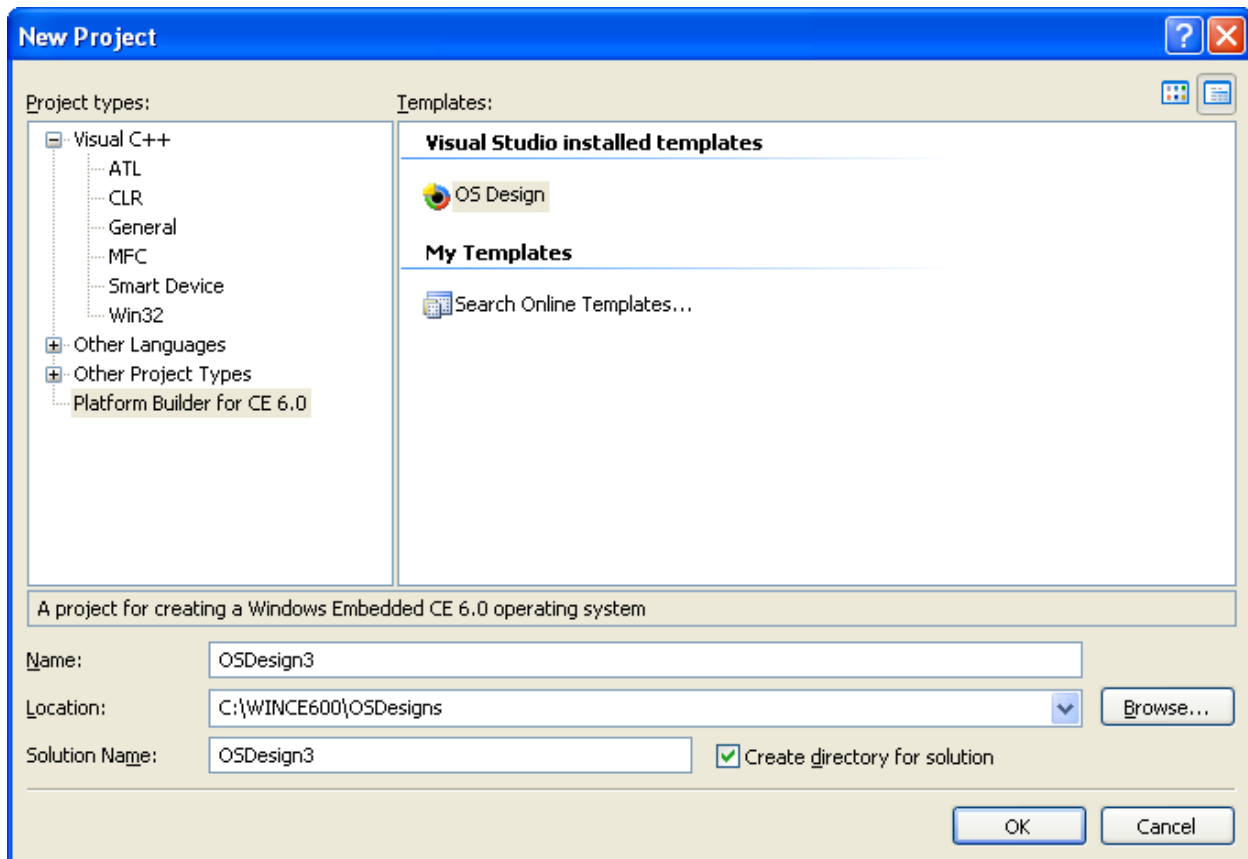
- A development workstation running Windows XP or Vista
- Visual Studio 2005 (Version 8)
- Service Pack 1 for Visual Studio 2005
- Service Pack 1 for Visual Studio 2005 on Vista (if you are running Vista)
- Windows CE 6.0 R2 Platform Builder plug-in with Service Pack 1
- A Zoom i.MX31 LITEKIT
- The i.MX31 SOM-LV Windows CE 6.0 BSP

2 Creating the Platform Image

2.1 Create a new OS Design

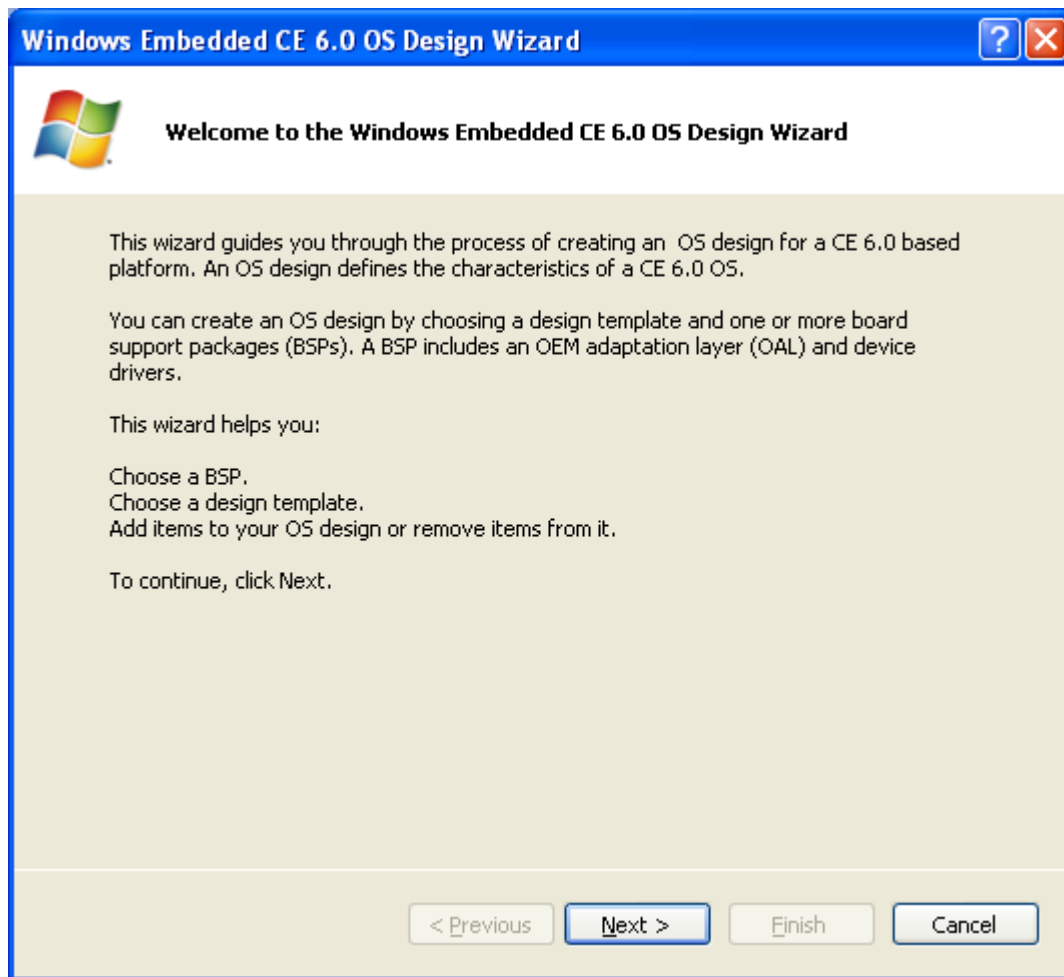
Launch the Visual Studio 2005 New Project dialog.

- Select **File** | **New...** | **Project...**



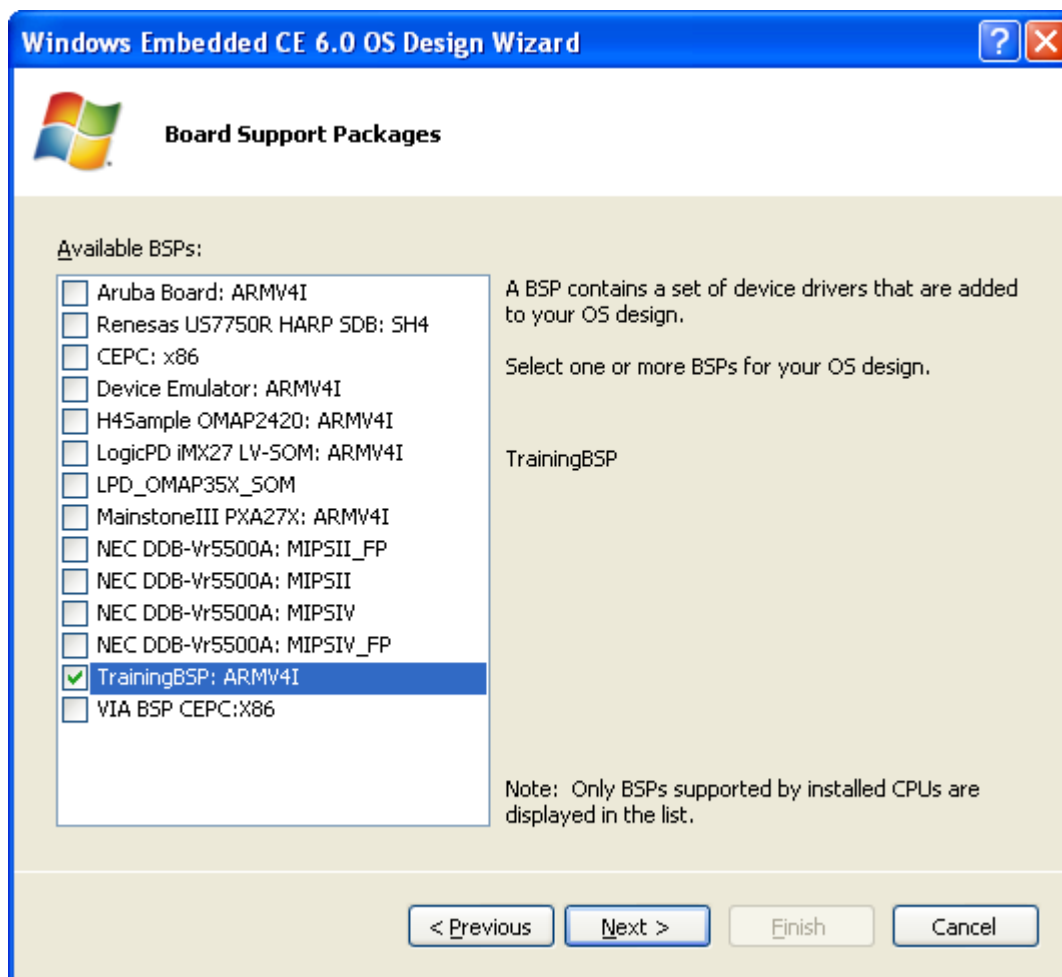
- Select the **Platform Builder for CE 6.0** project type.
- Choose the **OS Design** template
- Enter a name for your project, or go with the default name
- Click **OK**

The initial Dialog that is displayed below outlines the process of creating an OS Design. There are a number of steps involved and you will step through the wizard and make selections as appropriate.



We will build an operating system for the i.MX31 SOM-LV device.

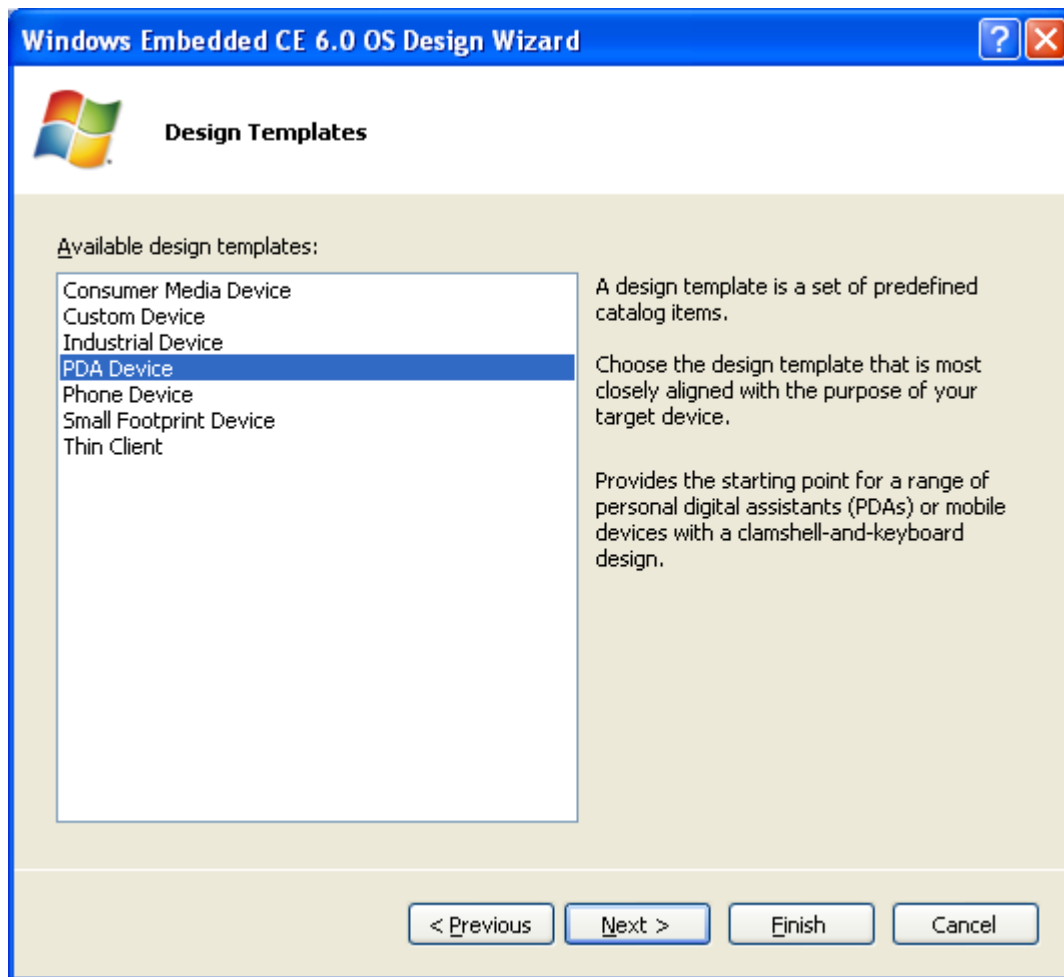
You could select more than one BSP (Board Support Package). For example, it may be useful to select the Device Emulator and a hardware reference platform; the emulation environment can be used by your internal/external application developers to build applications for your device while hardware is still being developed.



- Select **TrainingBSP: ARMV4I**
- Click **Next**

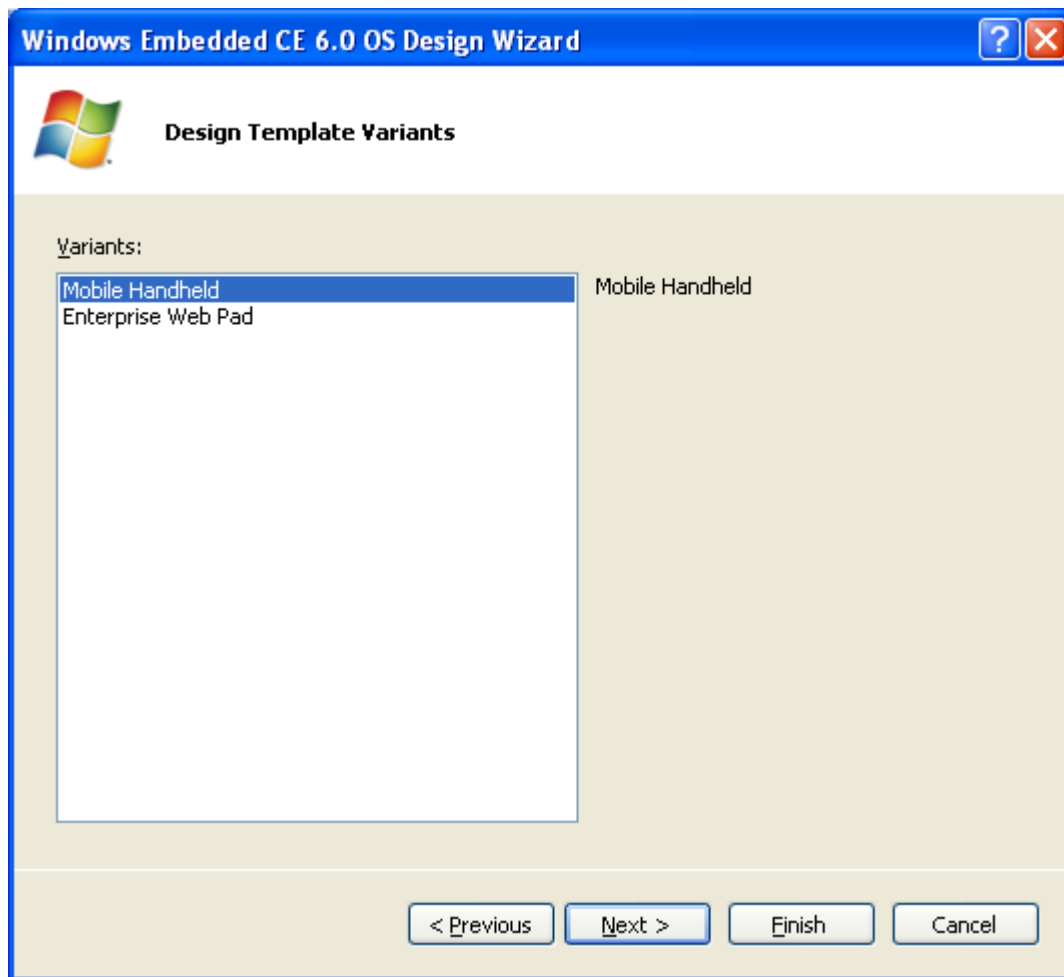
You are now given a number of Design Templates to choose from as the starting point for this project. If none of the options match your needs you can simply select '**Custom Device**' and build the image based on the components you select from the catalog.

For the purposes of this tutorial you will be using the **PDA Device** template.



- Select **PDA Device** from the list of Design Templates
- Click **Next**

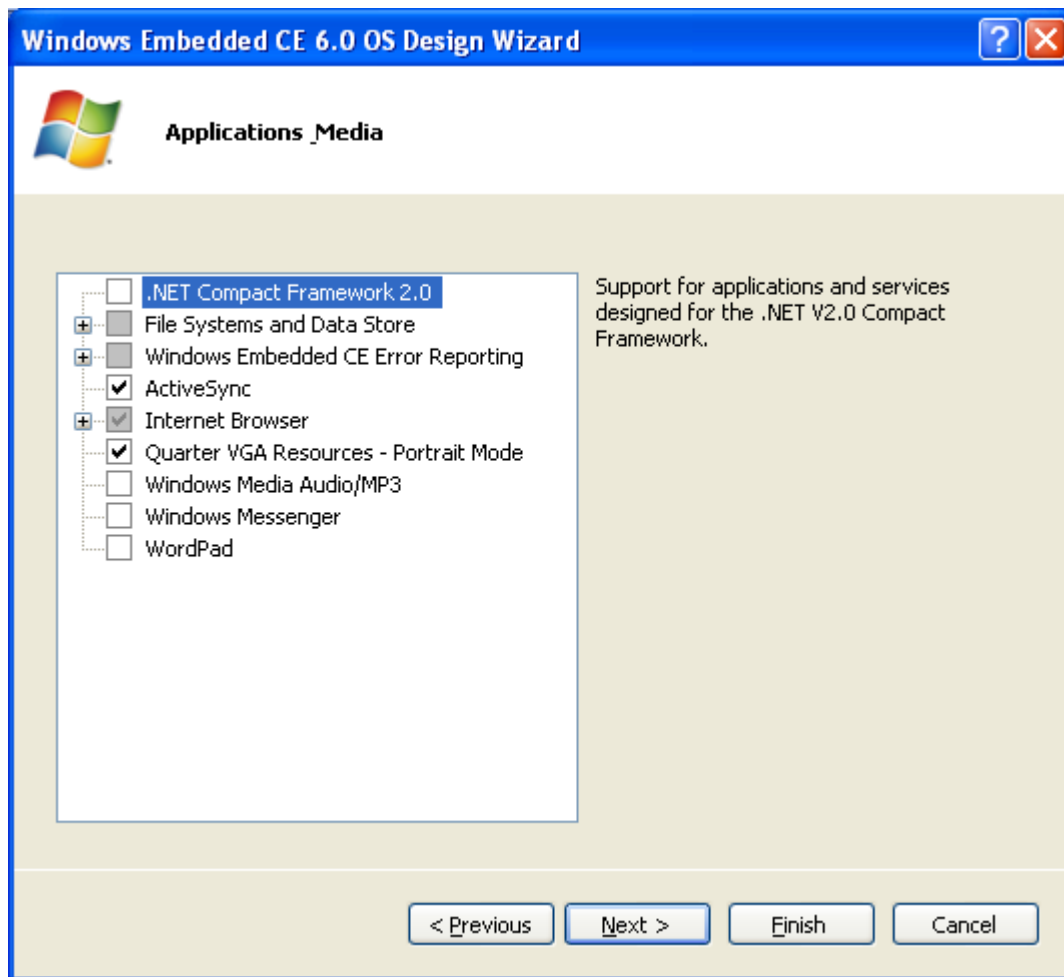
The Wizard now gives you the option of choosing a variant of the Design Template. For this tutorial you will be using the Mobile Handheld variant.



- Select **Mobile Handheld** from the list of Design Template Variants
- Click **Next**

There are a number of options which can be selected for each of the sample platforms, the OS Design Wizard will only show options that are relevant to the platform you are building. For example, it would not make sense to include Internet Explorer or WordPad in a headless device such as a Gateway. The Mobile Handheld device can include applications such as Internet Explorer, WordPad, and Windows Messenger.

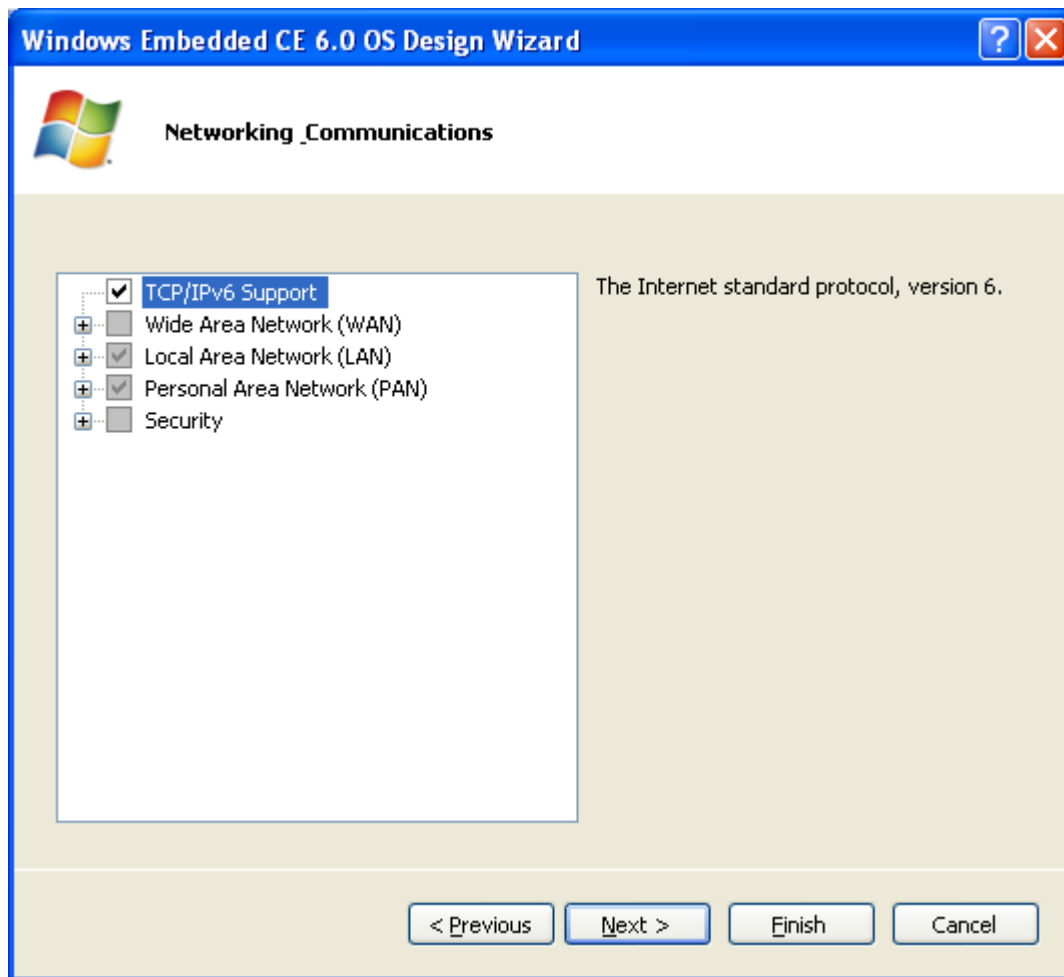
In this case we don't need the .NET Compact Framework. De-select this item.



- De-select the **.NET Compact Framework**
- Click **Next**

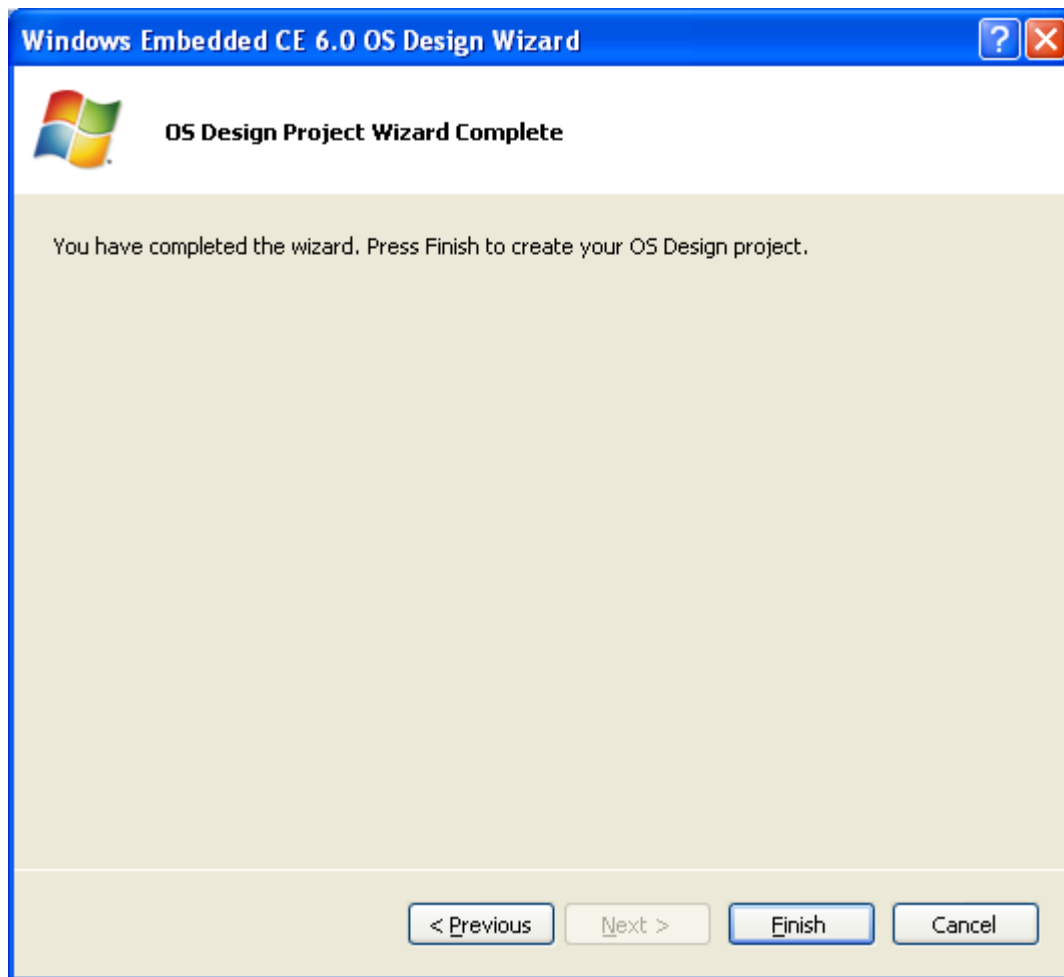
Next comes networking and communications, you can see that Windows CE provides support for Personal, local, and wide area networking through technologies such as Bluetooth, IrDA, Wired and Wireless networking, and VPN.

In our case the default options are fine.



■ Click **Next**

We're done! – The wizard is complete. You've configured the OS Design and can now further customize it by adding and/or removing components from the Catalog.



■ Click **Finish**

The Wizard will now prompt you with any security warnings or other notifications related to the components you selected. Acknowledge these notifications.

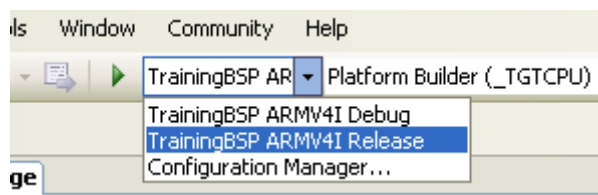


- Click **Acknowledge**

At this point you have an OS Design containing all of the OS components which have been selected from the Wizard.

Now we will set the project's build configuration to **Release**.

- Verify that the active build configuration is set to **TrainingBSP ARMV4I Release** using the pulldown menu in the main toolbar or by using **Build | Configuration Manager**

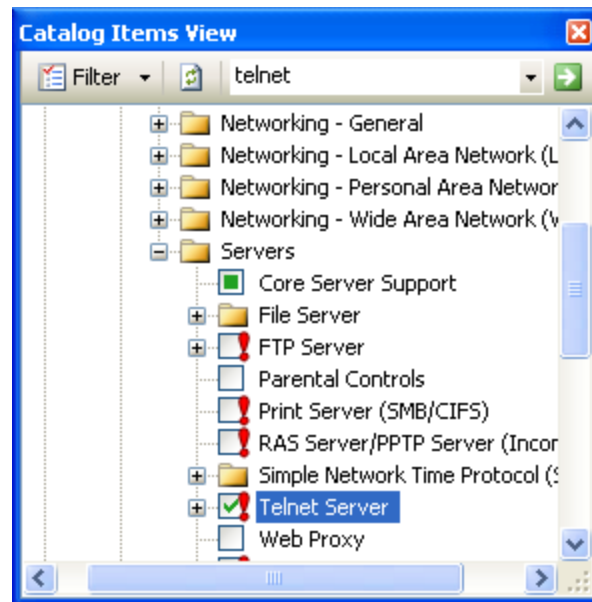


You will now make changes to the items included in your OS Design using the Catalog View.

First, ensure that the Catalog View is visible.

- Select the **View** menu
- Choose **Other Windows | Catalog Items View**

You want to ensure that Telnet is included in the run-time image that you build. Search for "Telnet" to locate the appropriate catalog item.



- Type **Telnet** in the search box
- Press the **Enter** key to search
- **Check** the box next to the **Telnet Server** item to include it in your OS Design
- You will get a security warning. Click “Acknowledge” to continue



- There are some specific Catalog Components that need to be selected in order for the device to support the Web and File server functionality.
- To find each of these components in the catalog, search for it using the search box at the top of the Catalog View panel, then click the checkbox to enable the component
- **Web Server (HTTPD)**: Enables the web server (hypertext transfer protocol daemon - "daemon" is a legacy term for a service)
 - **Active Server Pages (ASP) Support**: Enables ASP for pages on the web server, which allows for dynamic page generation server-side
 - **JScript 5.6**: Enables Microsoft J-Script in ASP pages
 - **VBScript 5.6**: Enables Microsoft VB script in ASP pages
 - **Web Server Administration ISAPI**: Enables the server/WebAdmin/ directory
- After selecting **Web Server (HTTPD)**, you will get a security warning:



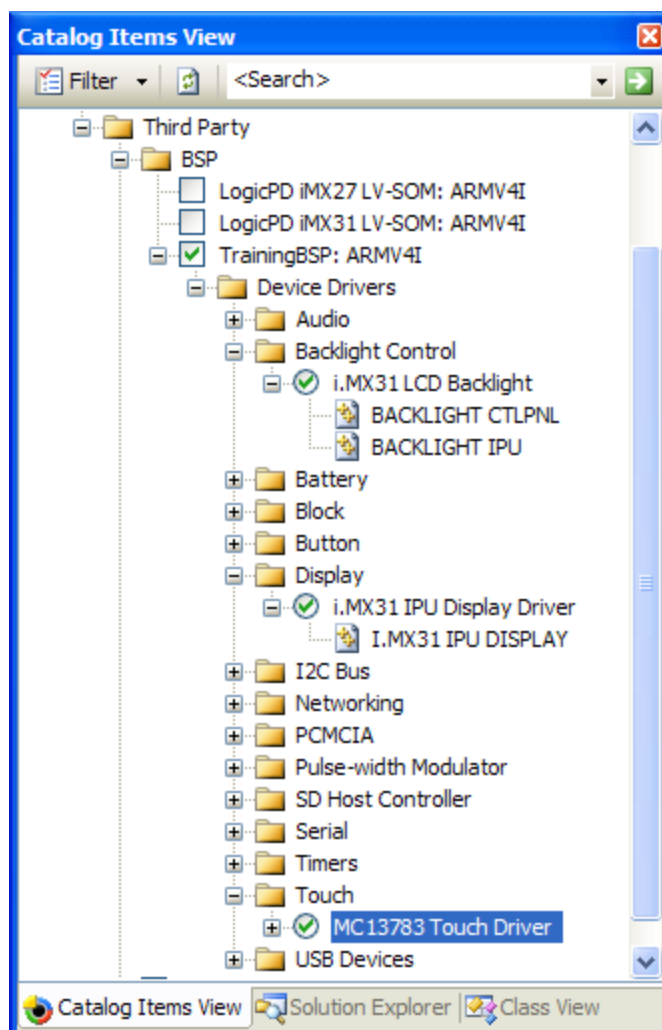
- Click **Acknowledge** to continue
- **File Server (SMB/CIFS)**: Enables the CIFS file server component. This is the protocol used to share files and folders over a Windows network when you right click the folder and select Sharing and Security or map a network drive
- After selecting **File Server (SMB/CIFS)**, you will get a security warning:



- Click **Acknowledge** to continue
- **File Server Customizable Web UI:** Enables web-based configuration of the file server component
- **Remote Display Application:** Enables a remote desktop server on the device

We need to select a few other components for our i.MX31 SOM-LV device.

- In the search box, type **i.MX31 LCD Backlight**
- Select the **i.MX31 LCD Backlight** component
- Note that there are other device drivers that are a part of this BSP.
- Under **Display**, select **i.MX31 IPU Display Driver**
- Under **Touch**, select **MC13783 Touch Driver**
- Your selections should match the following image:



The Catalog is an important tool for configuring your OS Design. If you choose to explore this view further, you will find advanced features for the analysis of dependencies between the various items.

2.2 Configuring Telnet

One easy way to communicate with the device and run console applications is to use Telnet. The Telnet Server catalog item has already been selected in the BSP. You can search for it in the Catalog to view it, if you wish. We will need to modify some registry settings to properly setup the Telnet device-side server. One of the main issues is that by default Telnet has user authentication enabled. We need to disable it.

To correct this issue, add the following registry settings to your project.reg file:

- To do this, first open the **Solution Explorer** by entering the **View** menu and selecting **Solution Explorer**
- Expand the **Parameter Files** section
- Expand **TrainingBSP: ARMV4I**
- Double click **project.reg**
- Select the **Source** button at the bottom of the registry editor view that pops up
- Add the following settings to your project.reg file:

```
[HKEY_LOCAL_MACHINE\Comm\TELNETD]

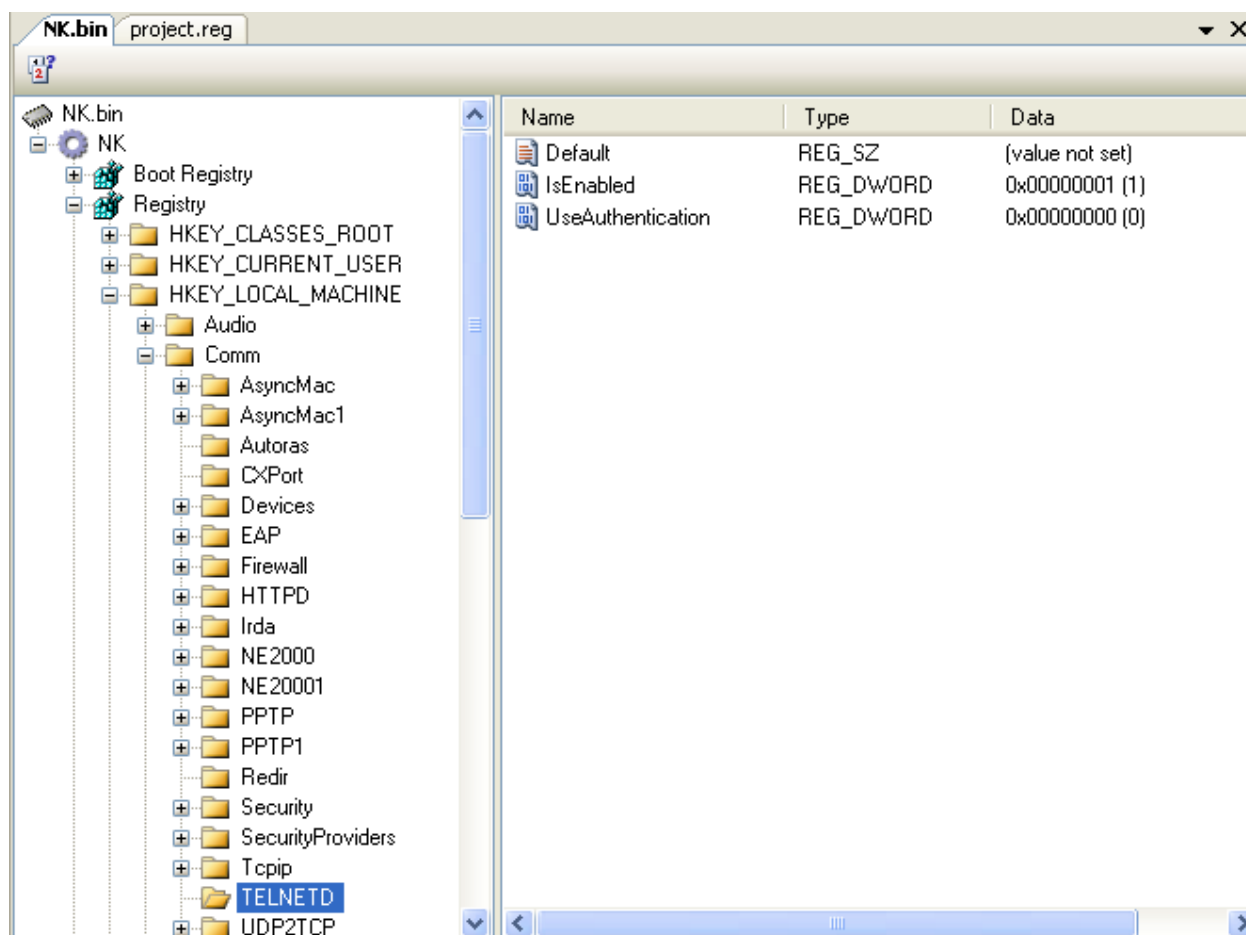
    "IsEnabled"=dword:1

    "UseAuthentication"=dword:0
```

- **Save** your changes to project.reg
- Build the BSP to include the project.reg additions **Build | Build Solution**

Next let's verify that the new registry settings were incorporated into the new run-time image by opening the NK.bin generated and looking at the registry. To do so:

- Select **File | Open | File**
- In the box, browse to or copy and paste this path:
C:\WINCE600\OSDesigns\OSDesign3\OSDesign3\RelDir\TrainingBSP_ARMV4I_Release\NK.bin
- Use the viewer to verify that the registry contains at least one of the newly added keys.

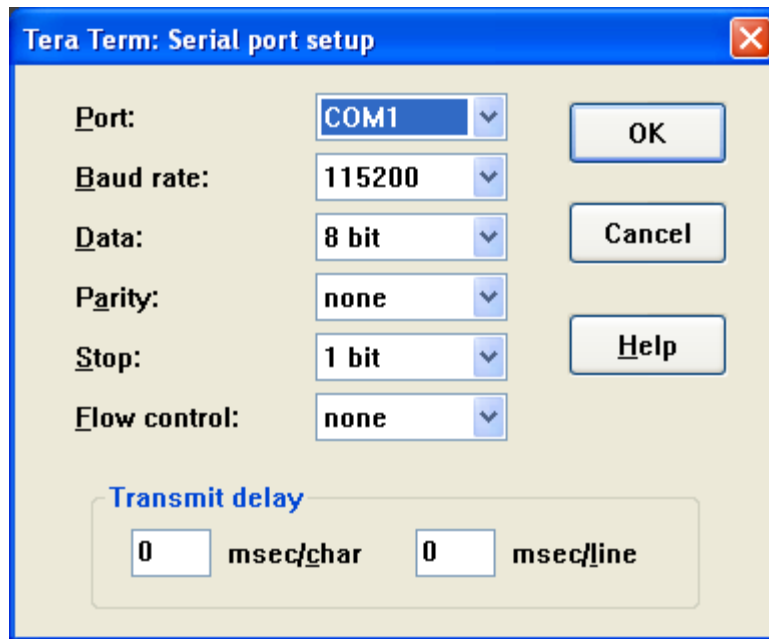


3 Download the Platform Image

3.1 Configure Download/Debug Transport

You are now ready to download and debug the Windows CE operating system image – before you download let's check to make sure the debug and kernel transports are configured and that we can view debug information over the serial terminal.

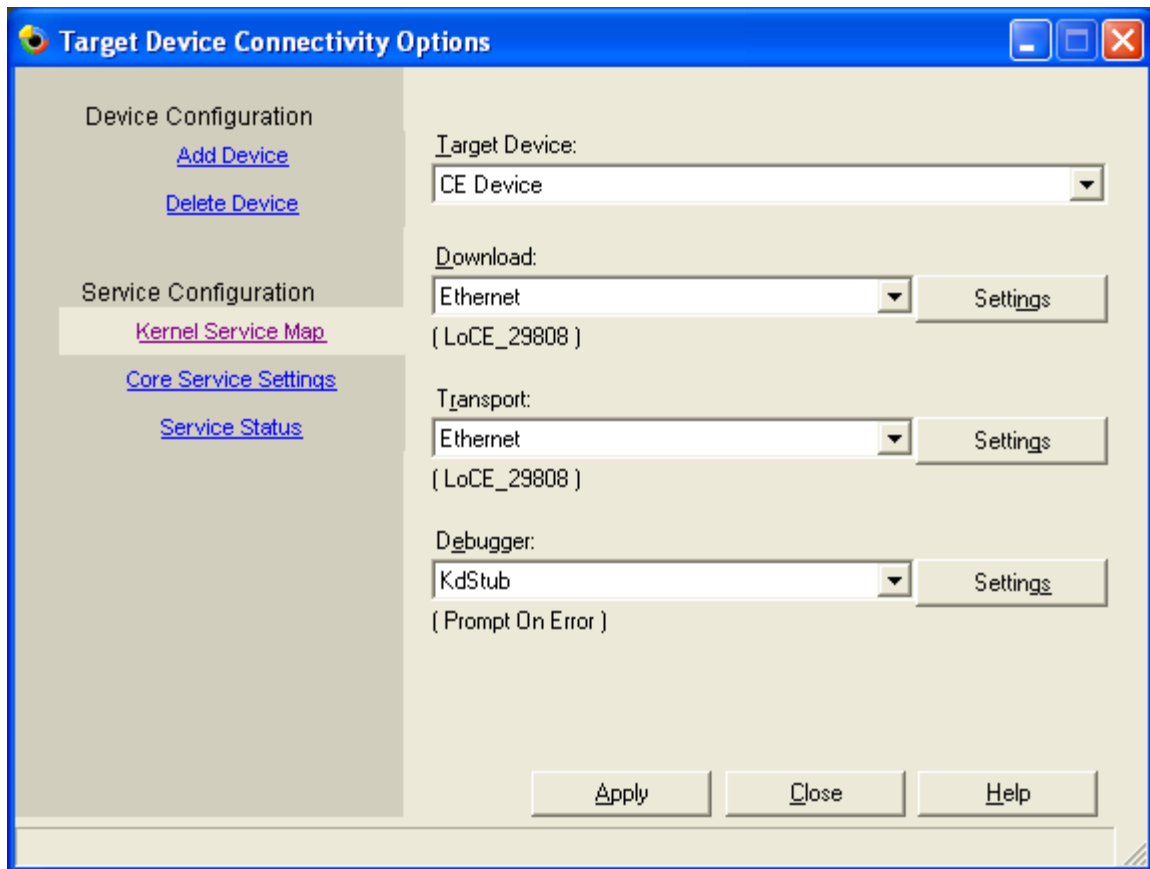
- Open **TeraTerm** by double clicking the icon on the desktop.
- Verify the serial port settings by clicking the **Setup | Serial Port** menu item (Your physical Port may differ):



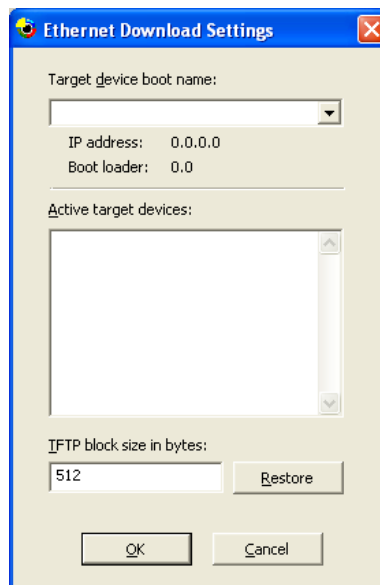
Ok now that you have configured your serial terminal we can continue to setup the device and Platform Builder. Switch back to your Visual Studio 2005 instance.

■ **Select Target | Connectivity Options**

The following dialog will be displayed:



- In the **Download** box select **Ethernet**
- In the **Transport** box select **Ethernet**
- In the **Debugger** box select **KdStub**
- Click on the **Settings** box next to the **Download:** dropdown box. A box like this will appear:



- Leave this box open and move on to the next step

Now we will switch to the **Tera Term** window to configure our device.

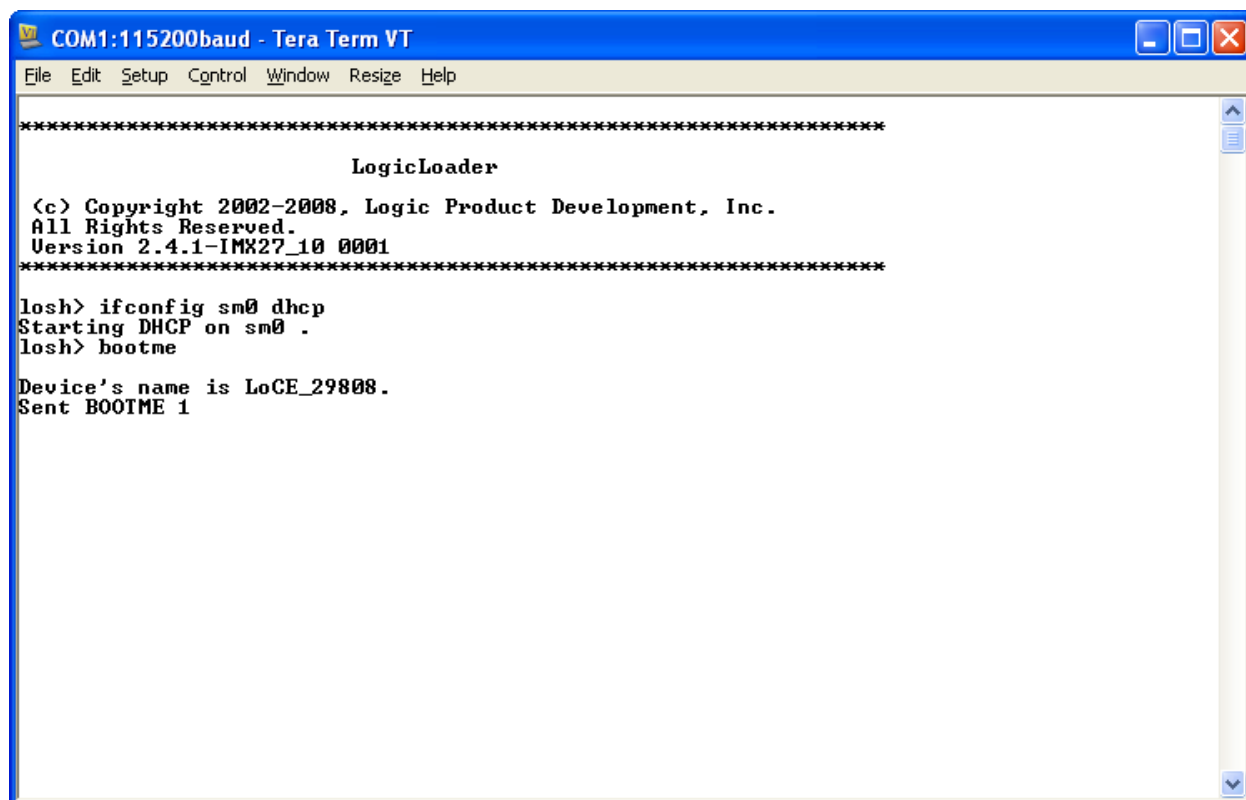
- Power on the device.
- You should see a **losh** prompt display.

```

COM1:115200baud - Tera Term VT
File Edit Setup Control Window Resize Help
*****
                LogicLoader
(c) Copyright 2002-2008, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.1-IMX27_10 0001
*****
losh> █

```

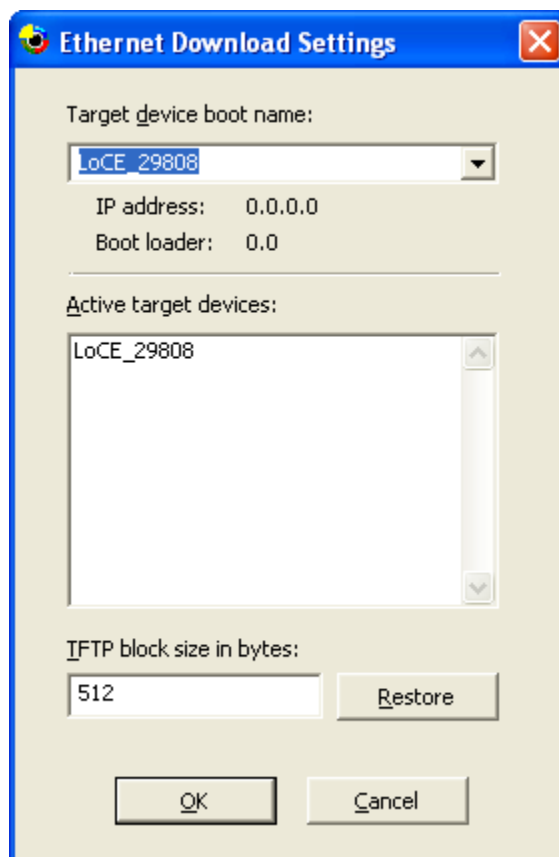
- Configure the network interface using either DHCP or a statically assigned address:
 - For DHCP, type: **ifconfig sm0 dhcp**
 - For a static IP address, use the following command: **ifconfig <interface> <ip> <netmask> <gw>**
 - Example: **ifconfig sm0 192.168.1.1 255.255.255.0 192.168.1.0**
- Type the command **bootme** and press the Enter key. This will allow visual studio to connect to this device. Note the name of the device that is displayed:



The screenshot shows a Tera Term VT window titled "COM1:115200baud - Tera Term VT". The window contains the following text:

```
*****  
                        LogicLoader  
  
<c> Copyright 2002-2008, Logic Product Development, Inc.  
All Rights Reserved.  
Version 2.4.1-IMX27_10 0001  
*****  
  
losh> ifconfig sm0 dhcp  
Starting DHCP on sm0 .  
losh> bootme  
  
Device's name is LoCE_29808.  
Sent BOOTME 1
```

- Switch back to Visual Studio and the box you left open should now be populated with a device. If there are multiple devices in the list, click on them and view the IP address. Make sure it matches up with the one that you wrote down.
- **Note:** If the device does not appear in Platform Builder's "Ethernet Download Settings" window, please check your firewall settings and anti-virus software. The "bootme" protocol uses raw UDP packets for communication between the device and the tools.



- Click **OK** in the small box, and then click **Apply**. Once you have done this, click the **Close** button.

3.2 Download the Operating System

- Select **Target | Attach Device**
- The device should still be executing the **bootme** command, but if it is not, type **bootme** at the **losh** prompt
- When the image is finished transferring (a **losh** prompt will display), type the following command to boot the image:
exec rtc:rtc_imx31_pmic:dbg_leds:1:disp_num:3:kitl:true:share_eth:1:

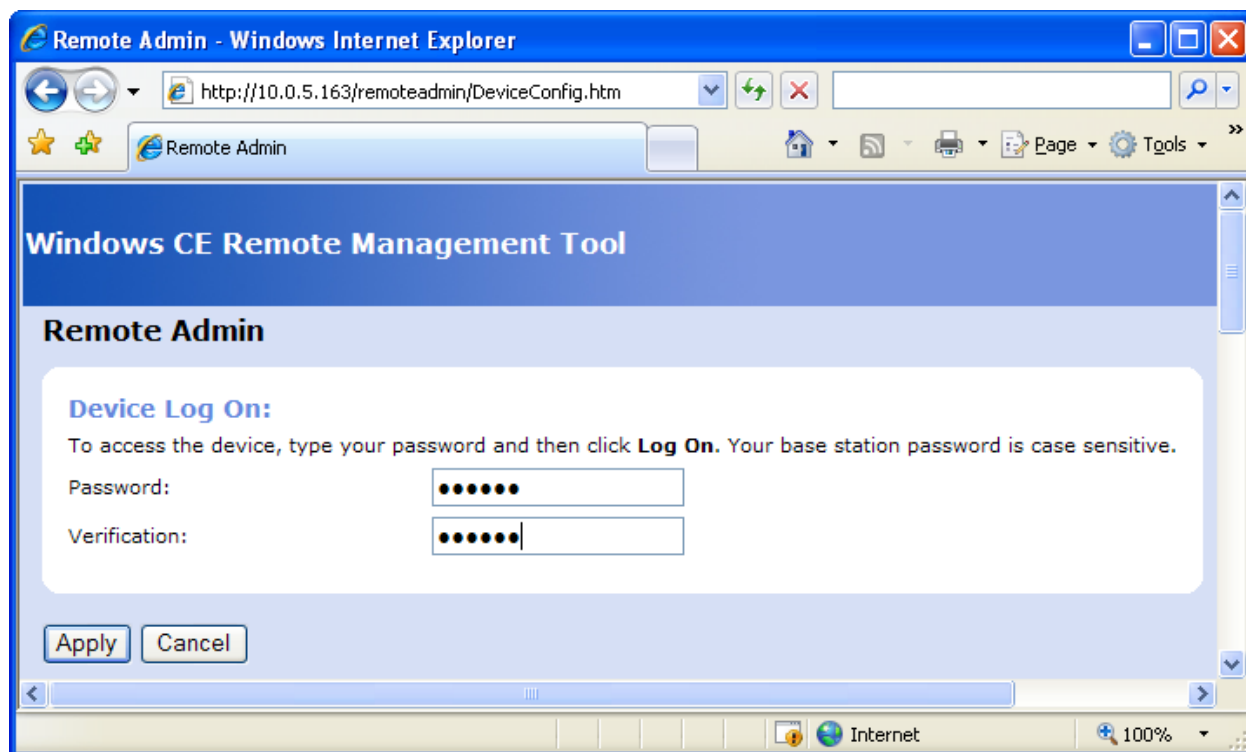
4 Setting up the File Server

Now we'll set up the file server on the device, so you can access files on it like with any other windows network share. To do this, we'll have to use the remote web-based administration tool.

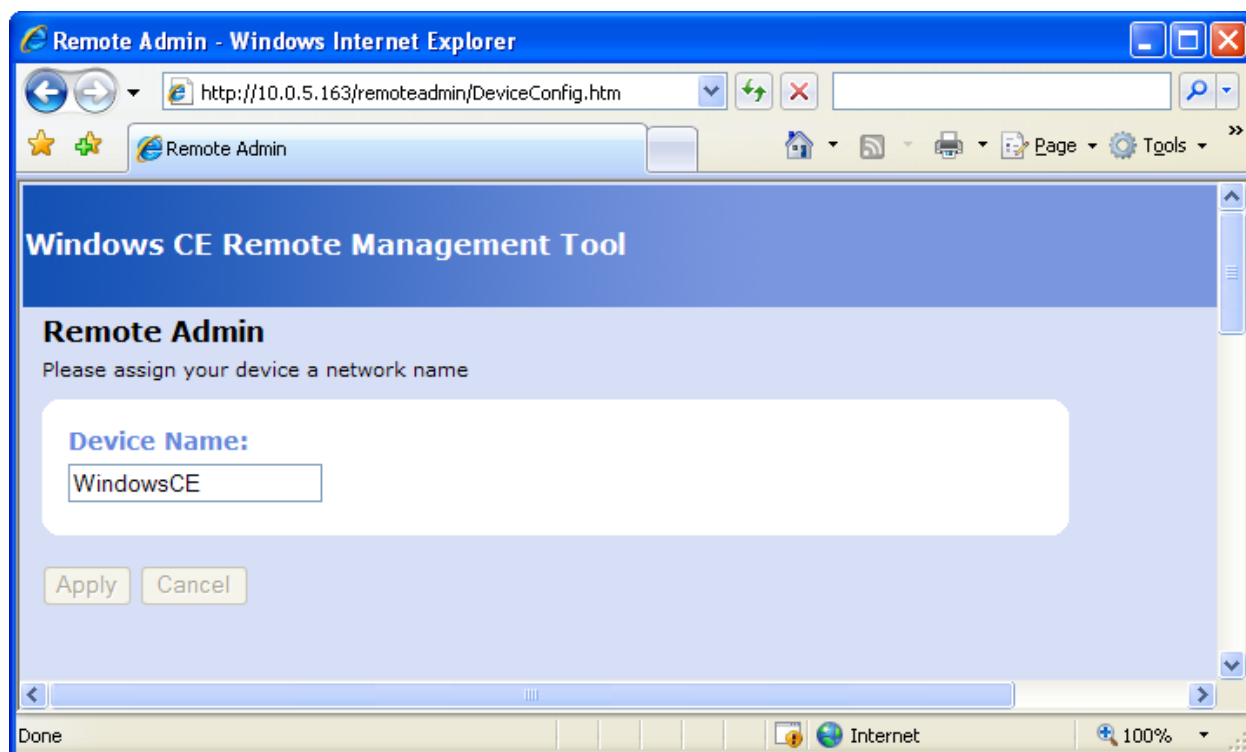
4.1 Using the Remote Web Administration Interface

Now we'll configure the file server from its web interface.

- Open a new web browser window and type in `http://(Your device's IP address)`. You should see a remote administration page. You'll prompted to set a password, so choose a good password and click apply

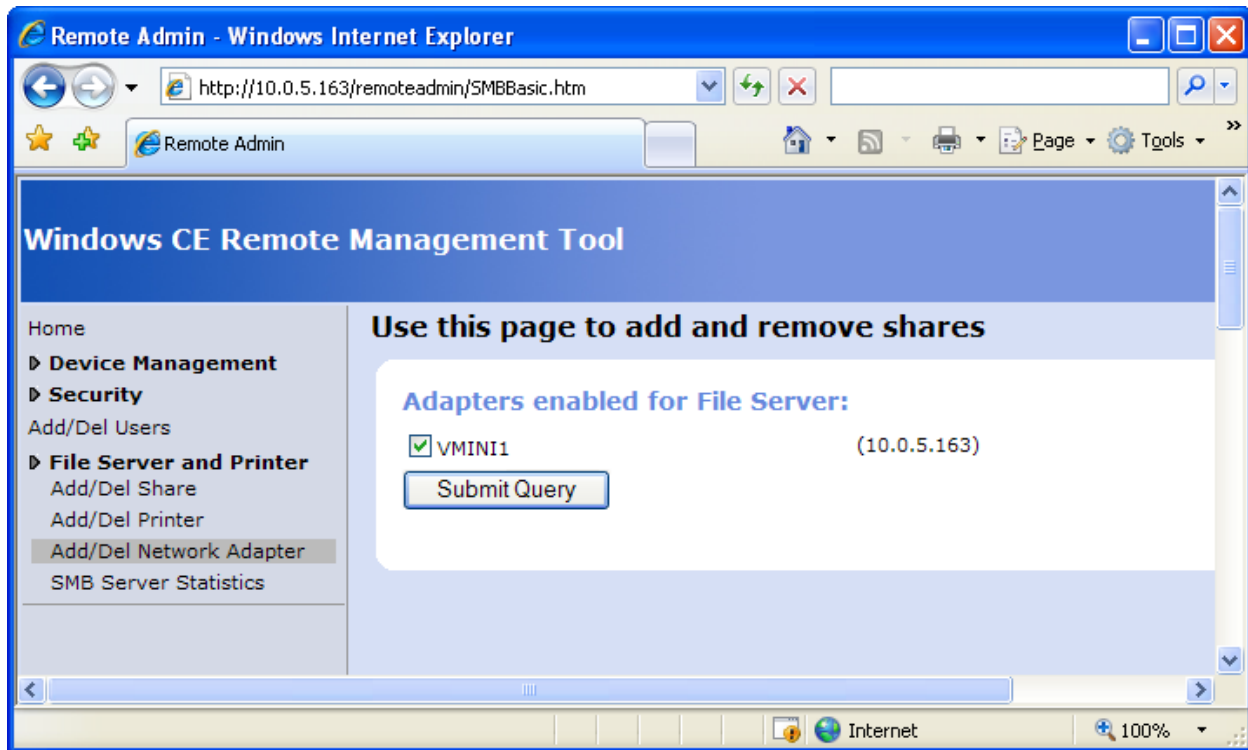


- Wait about 20-30 seconds to be redirected. If you weren't automatically redirected, go to /remoteadmin/DeviceConfig.htm. When prompted for a password, use the username **ADMIN** and the password that you just set. Then you should see a page like the following:



- Enter a unique name for your device, such as **IMX_YourLastName** and click Apply

- Afterward, you'll see an administration page with more options. Expand **File Server and Printer** and then select **Add/Del Network Adapter** to see the following screen:

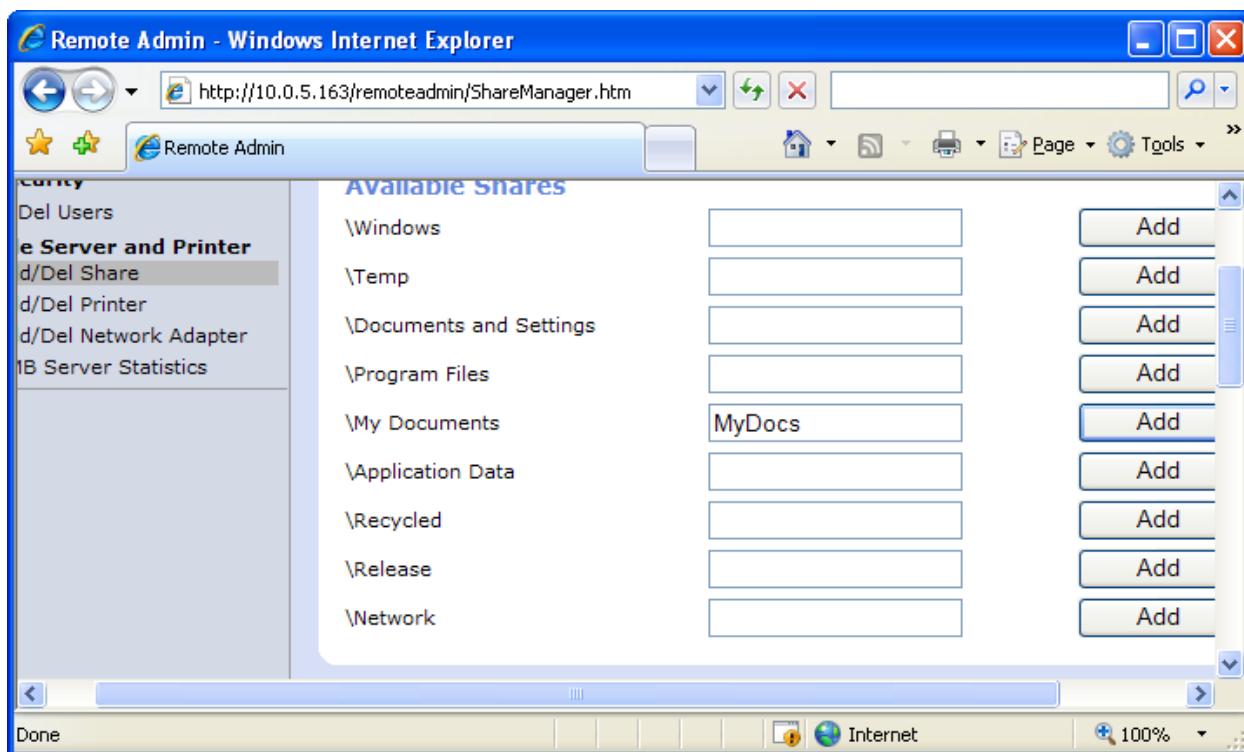


- Check the box next to the network adapter and press **Submit Query**. This will enable the file server on that network adapter

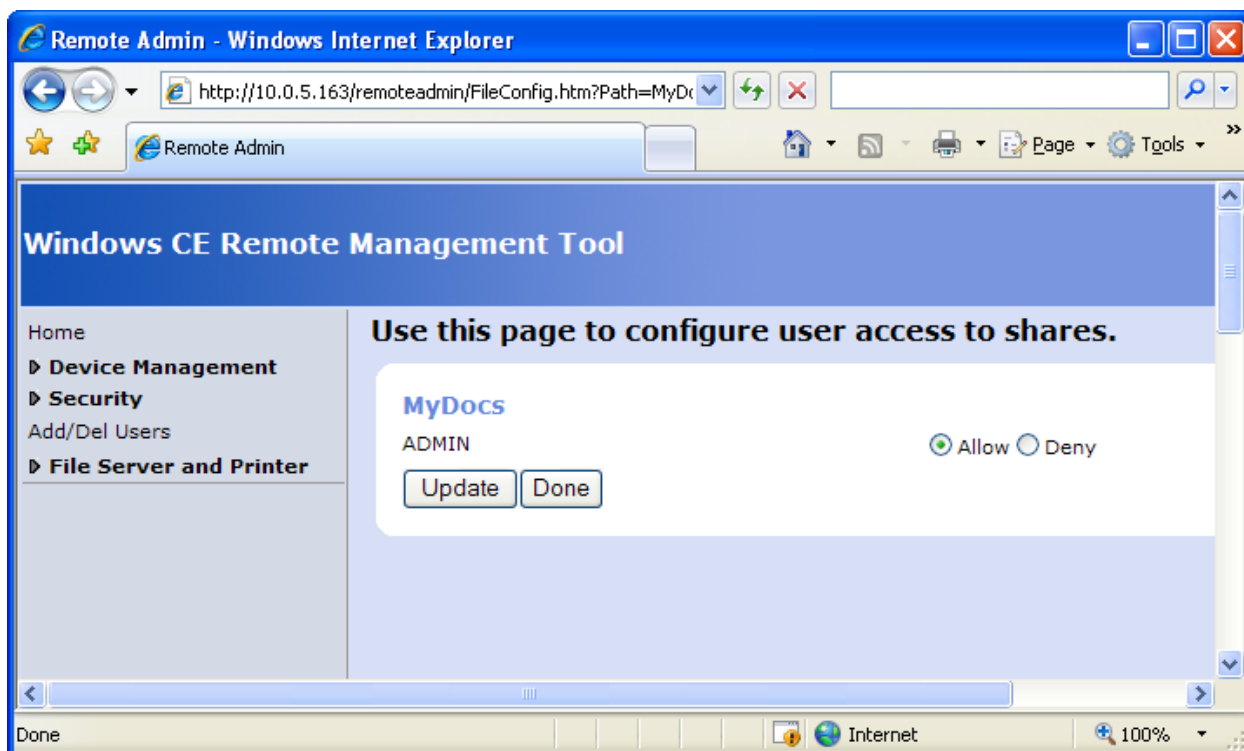
4.2 Create and Configure a Share

Now we'll add a share using the web interface.

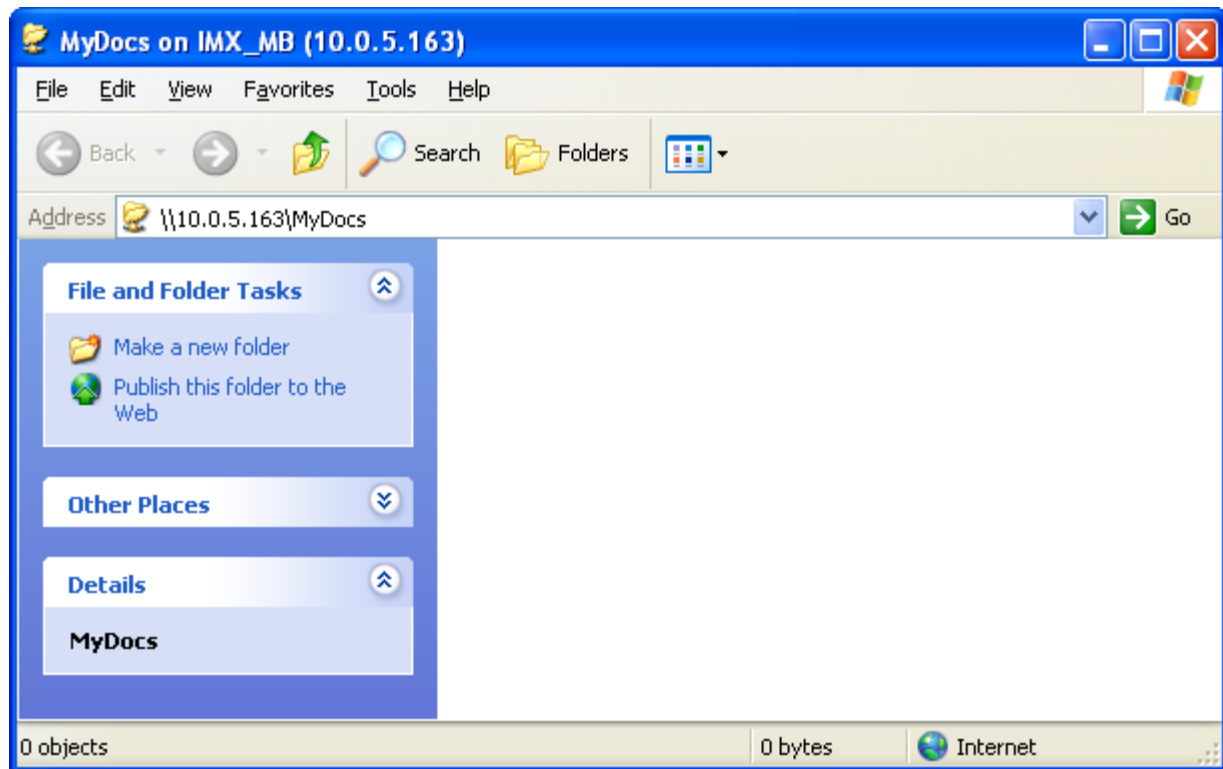
- In the web interface, select **Add/Del Share**, type in the name **MyDocs** (this is what the share will be called) next to the **My Documents** directory, then click **Add**



- Press **Permissions** next to HD after adding the share and at the Permissions screen select **Allow** for the ADMIN user and then press **Update**



- On your **development workstation**, Open a **Windows Explorer** window by double clicking any folder on your PC, or select **Start | Run** and type **\\(device IP)MyDocs**. When it prompts you for a login, log in as username **ADMIN**, and use the password you set on the device earlier



- This screen will likely be empty, because you do not have anything in that folder yet!

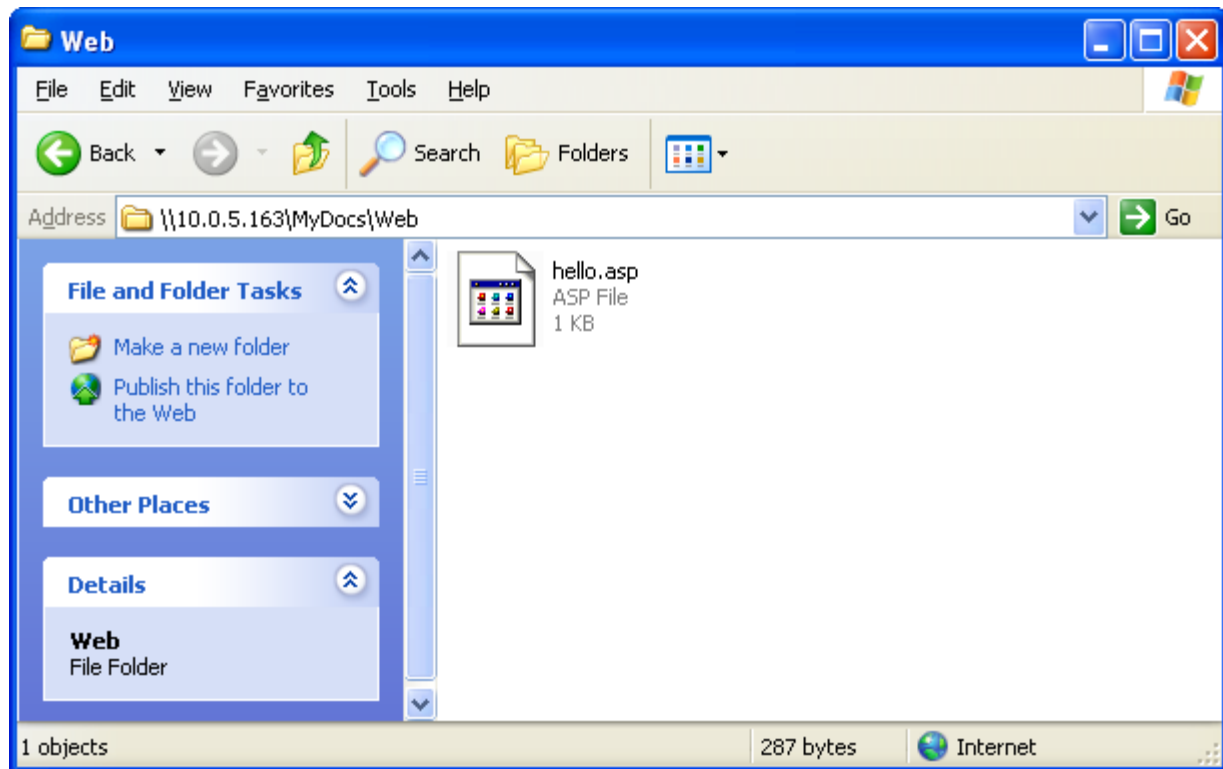
5 Setting up the Web Server

In this part we will set up the web server in a custom directory and have it run a simple VBScript ASP sample application

5.1 Create a Web directory

Using the Windows file server we set up in the previous step, we can now create a directory we'll use for the root of our web server

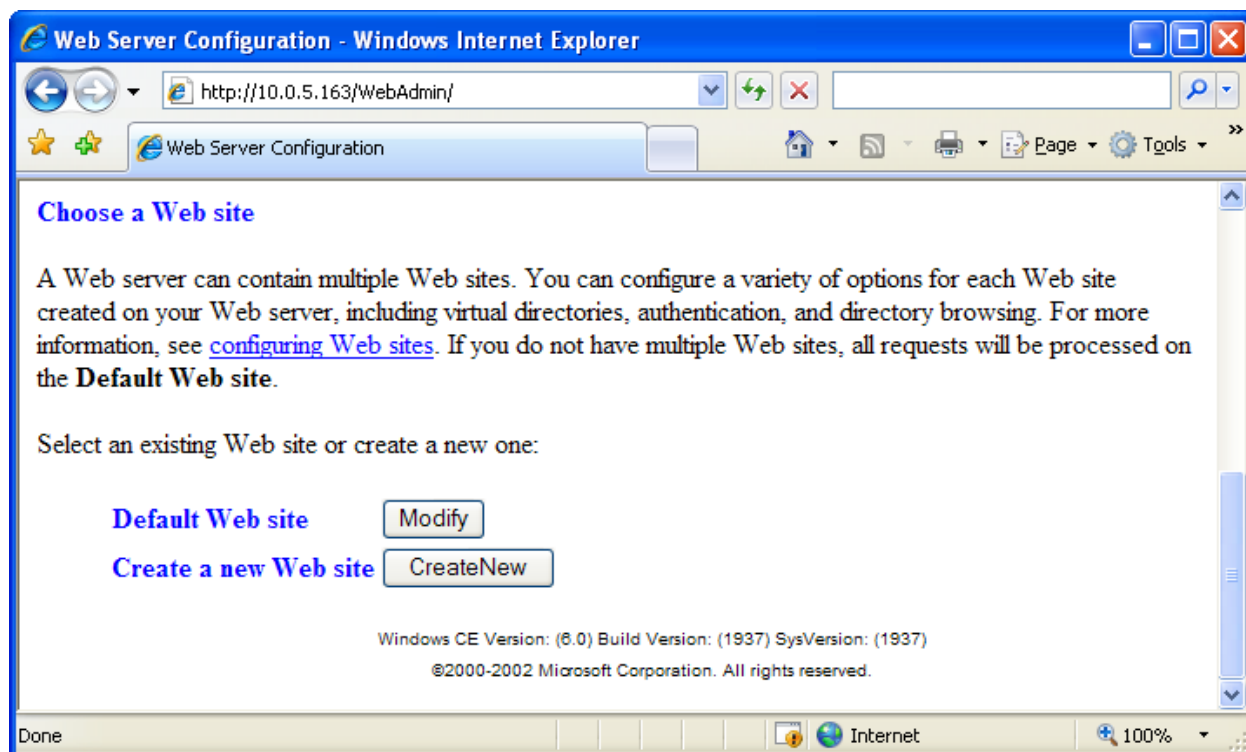
- In the **Windows Explorer** window you have open and connected to your device, create a folder called **Web**
- Open the directory you just created, and copy in **hello.asp** from the **LabFiles** directory, **C:\Labs\Lab3\LabFiles**



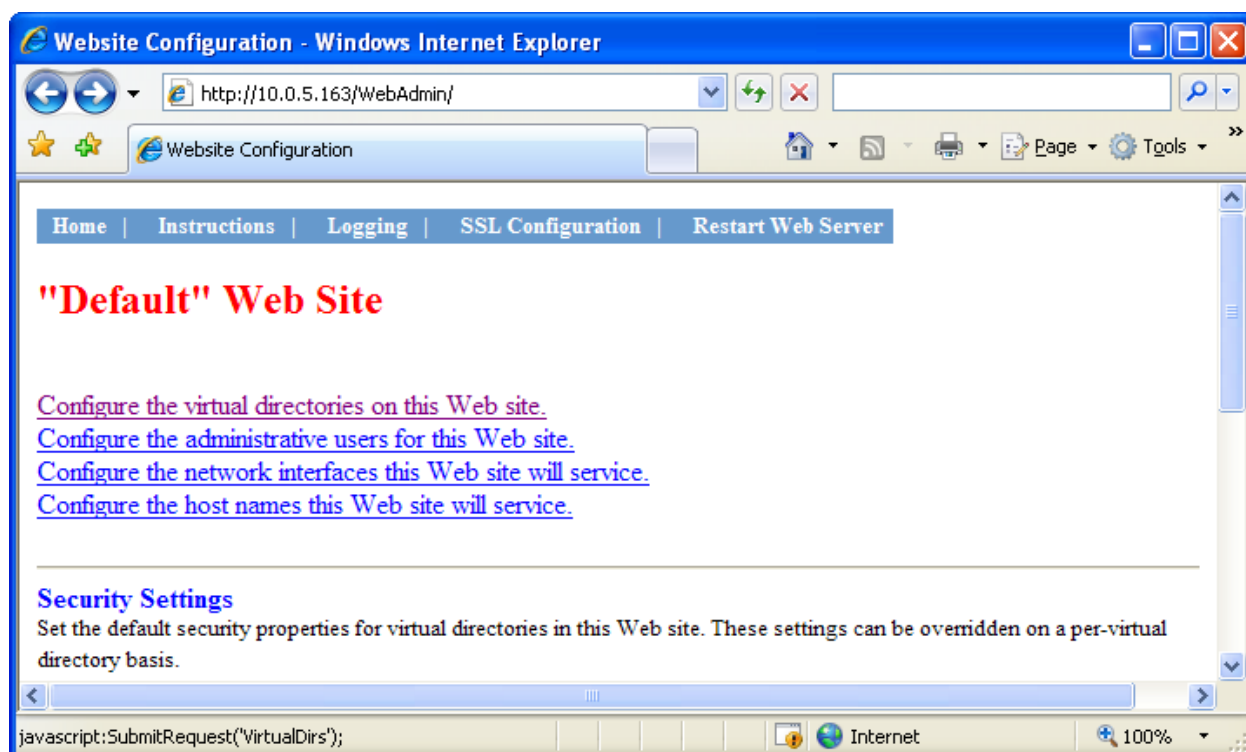
5.2 Log in to the Web Server's Configuration Interface

Like the Windows file server, the web server is configured through a web interface.

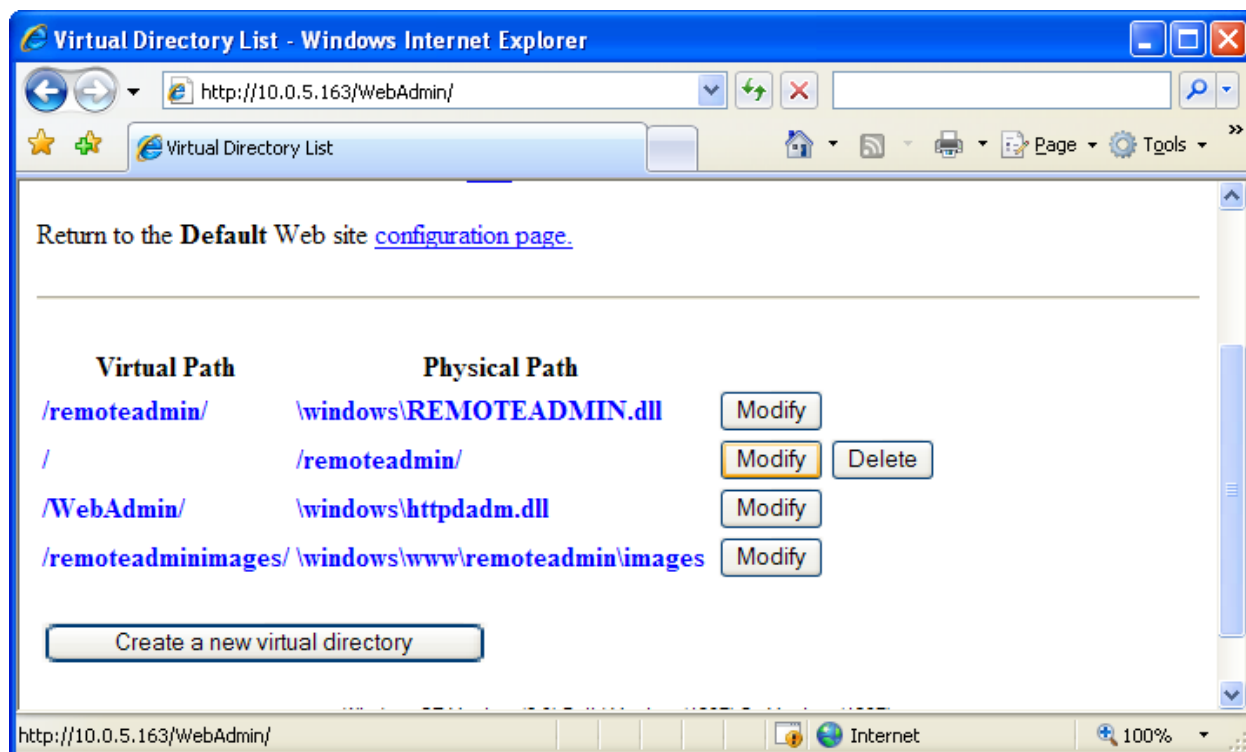
- Navigate your web browser to your device's IP address/WebAdmin/. You may be prompted to enter a username and password, if so enter ADMIN and your device's password



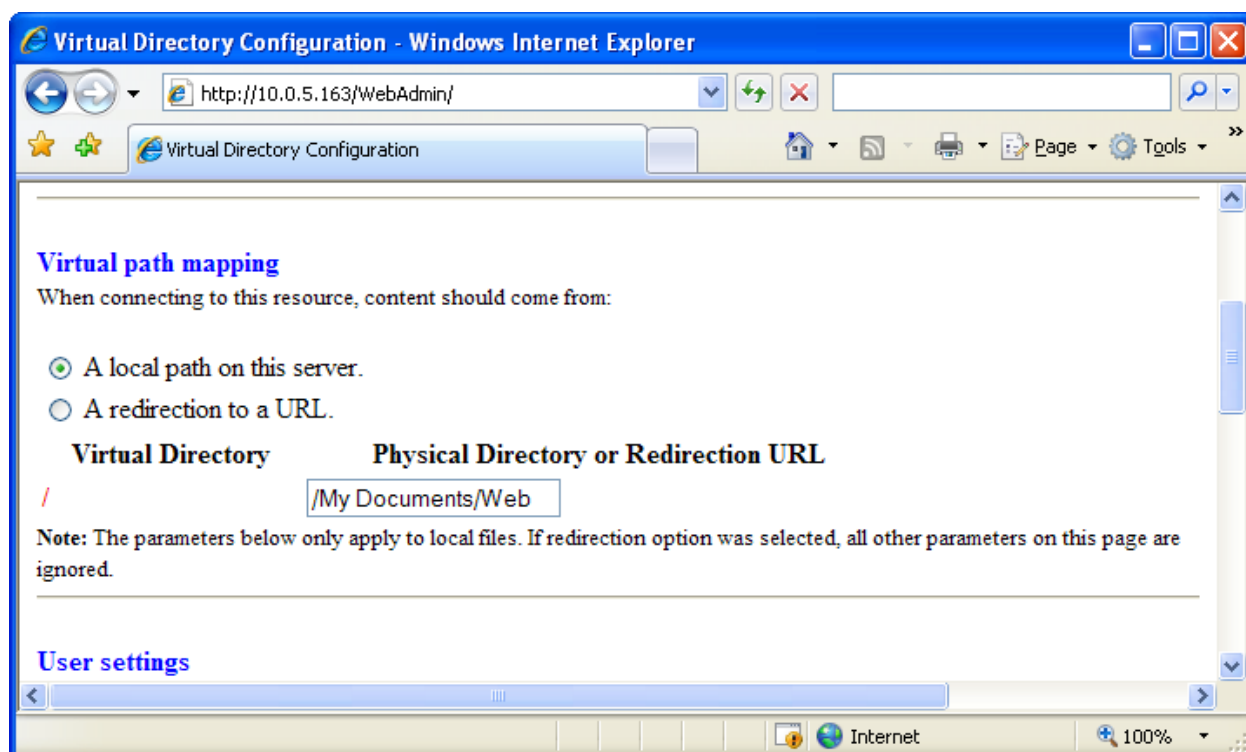
- Scroll down and select **Modify** next to **Default Web Site**
- Select **Configure the virtual directories on this Web site**



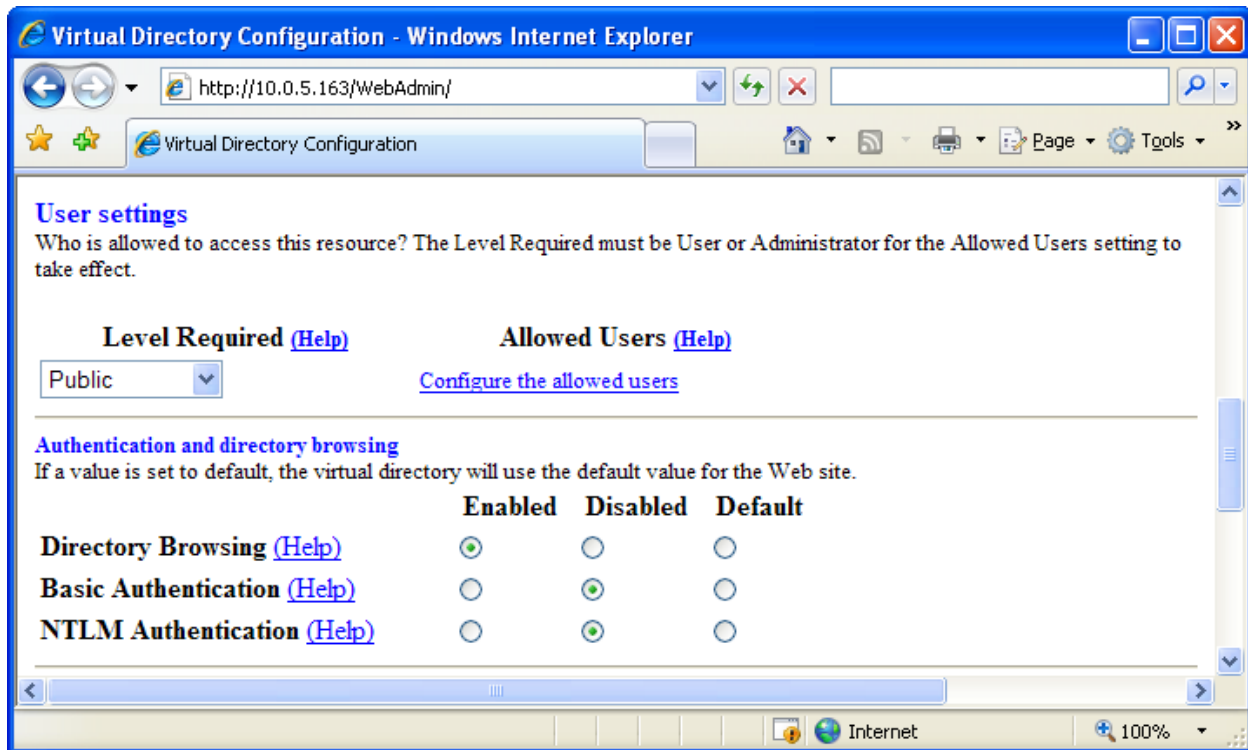
- Scroll down and select **Modify** next to / (which is the root directory)



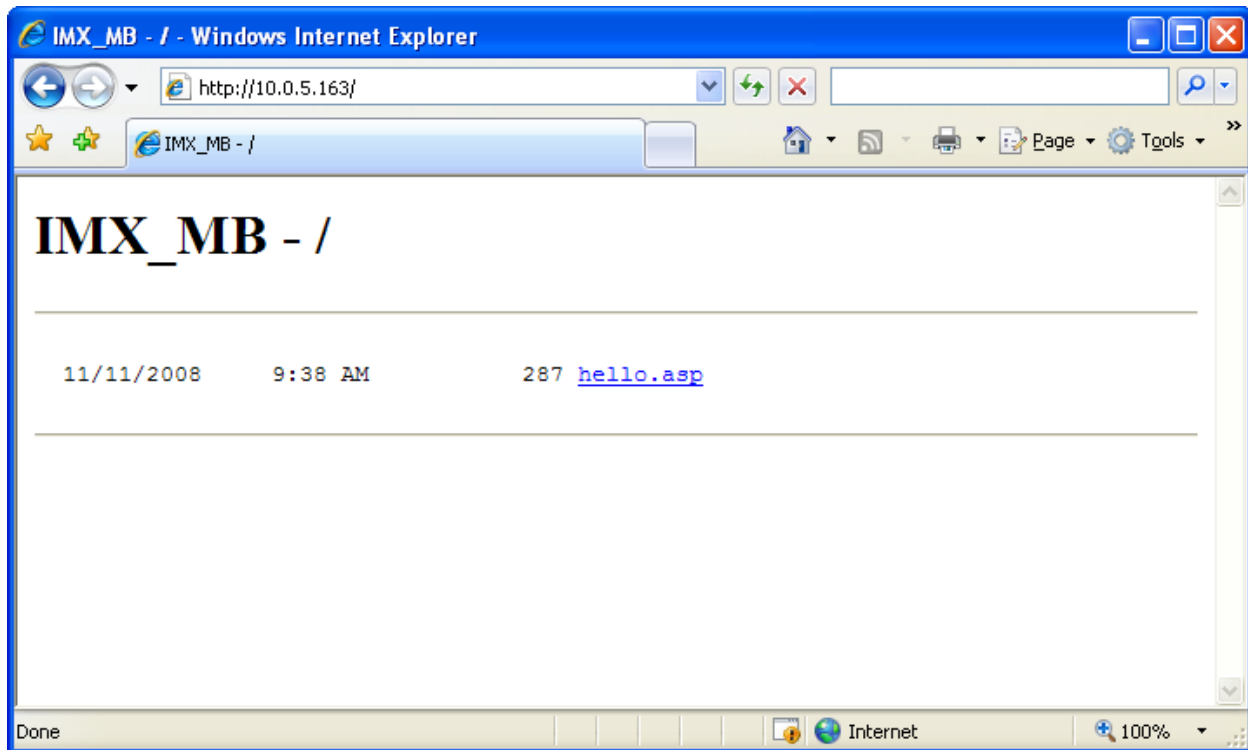
- Select **A Local Path on this server**
- Type **/My Documents/Web** into the directory box



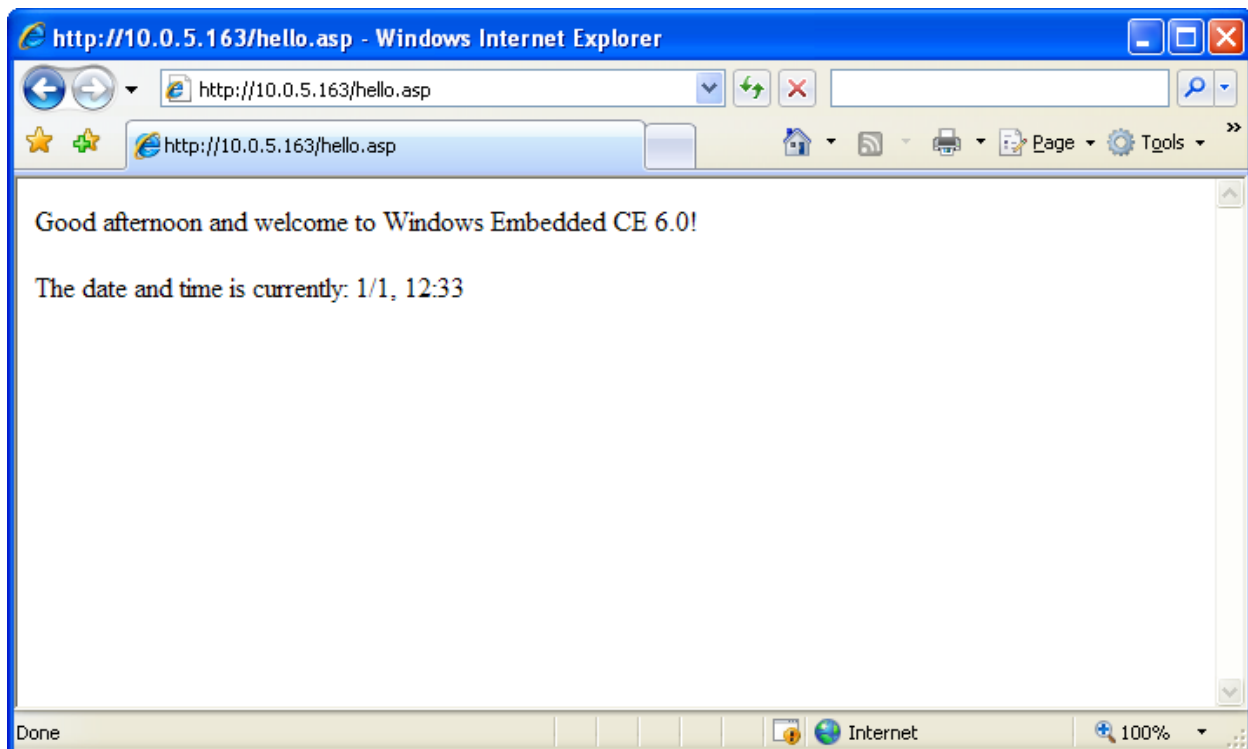
- Set the **Level Required** to **Public**
- Disable **NTLM** and **Basic Authentication**, and enable **Directory Browsing**



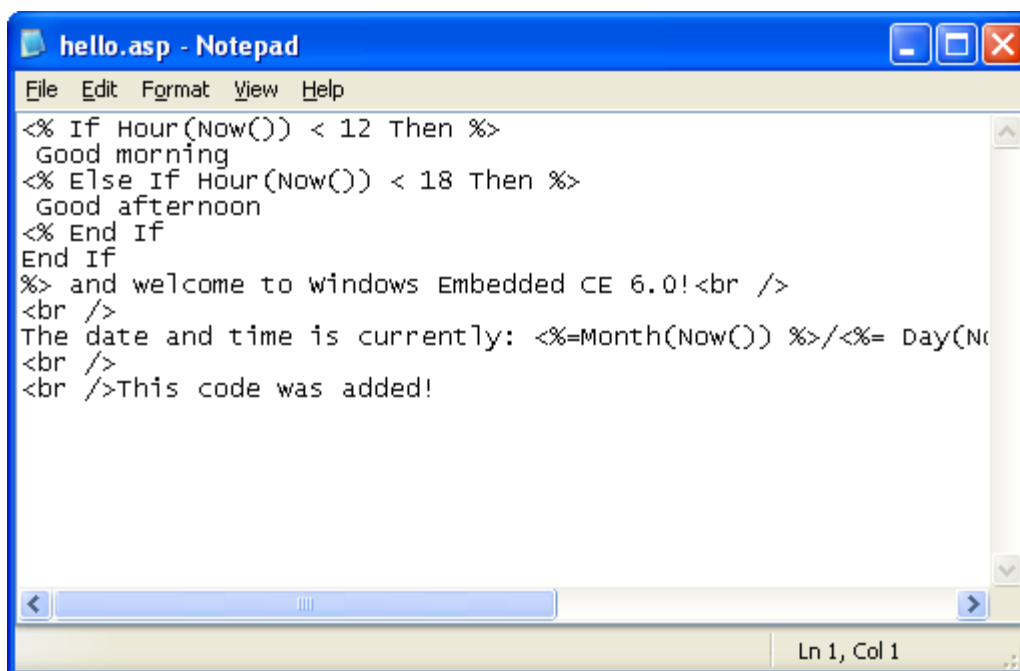
- Click **Update**
- You should see a message indicating that the settings were saved and telling you to restart the web server:
- Select **Restart Web Server** at the top right corner of the page to restart the web server on the CE device.
- Navigate your web browser to your **http://(your device's IP address)/**. You should see a directory listing containing **hello.asp**



- Click [hello.asp](#) to see a dynamically-generated webpage which prints out the current date and time



- From the **Windows Explorer** window you have navigated to your device over the file server, open `hello.asp` into your file editor of choice to edit this file
- Add some code, then save

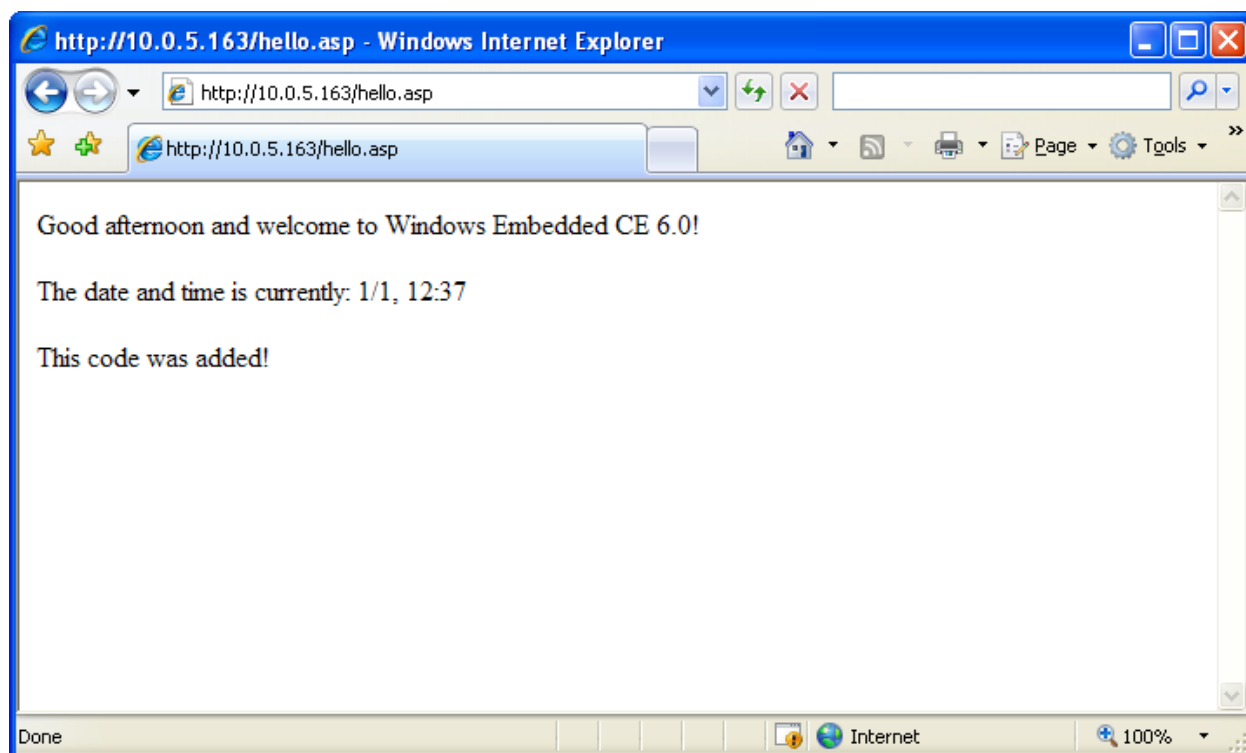


```

File Edit Format View Help
<% If Hour(Now()) < 12 Then %>
  Good morning
<% Else If Hour(Now()) < 18 Then %>
  Good afternoon
<% End If
End If
%> and welcome to windows Embedded CE 6.0!<br />
<br />
The date and time is currently: <%=Month(Now()) %>/<%= Day(Now()) %>
<br />
<br />This code was added!
Ln 1, Col 1

```

- Click **Refresh** in your web browser to view the changes



6 Setting up a Remote Display

6.1 Connect to the Telnet server

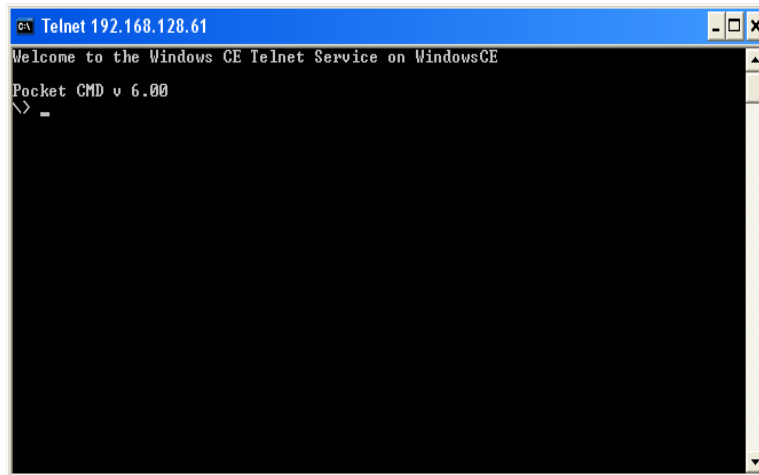
- Open a command prompt window by clicking “Run” in your start menu. Then type **cmd** and press the Enter key.

- In the terminal type:

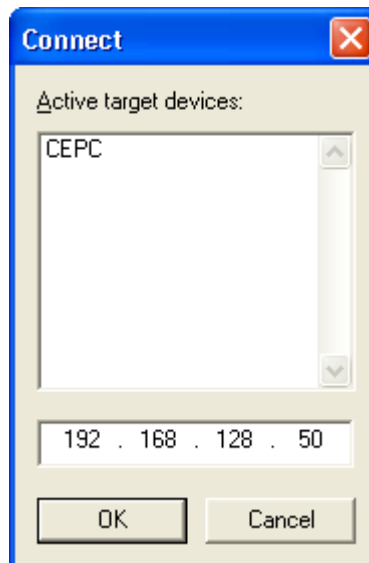
```
telnet xxx.xxx.xxx.xxx
```

(where the xxx.xxx.xxx.xxx should be replaced by your device's IP address)

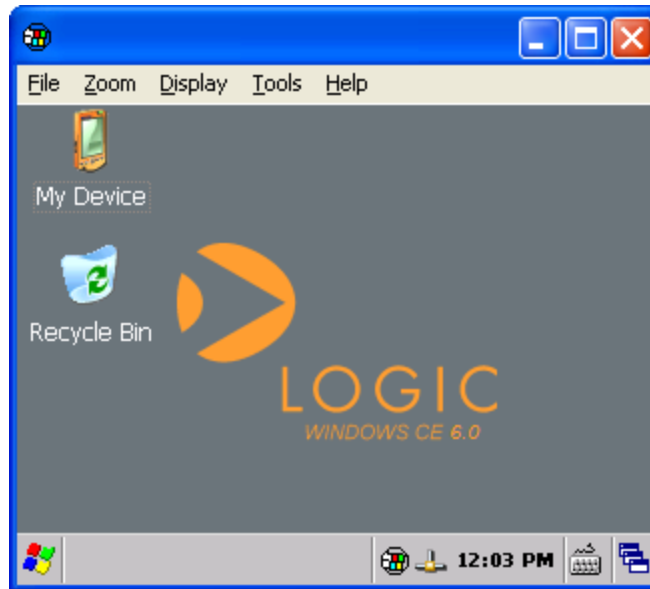
- Now press enter to login to the device



- The telnet interface should pop up. At the command prompt type **cerdisp.exe -c** to launch the remote display application and set it to automatically accept incoming connections.
 - It might be necessary to run **cd windows** first
- Run **cerhost** on your development workstation. It's located here:
C:\WINCE600\PUBLIC\COMMON\OAK\BIN\I386\cerhost.exe
- Select **File | Connect** and type in the IP address of your device, then click **OK** (Your device name and IP address will differ from the picture shown below)



- The remote display tool should start.



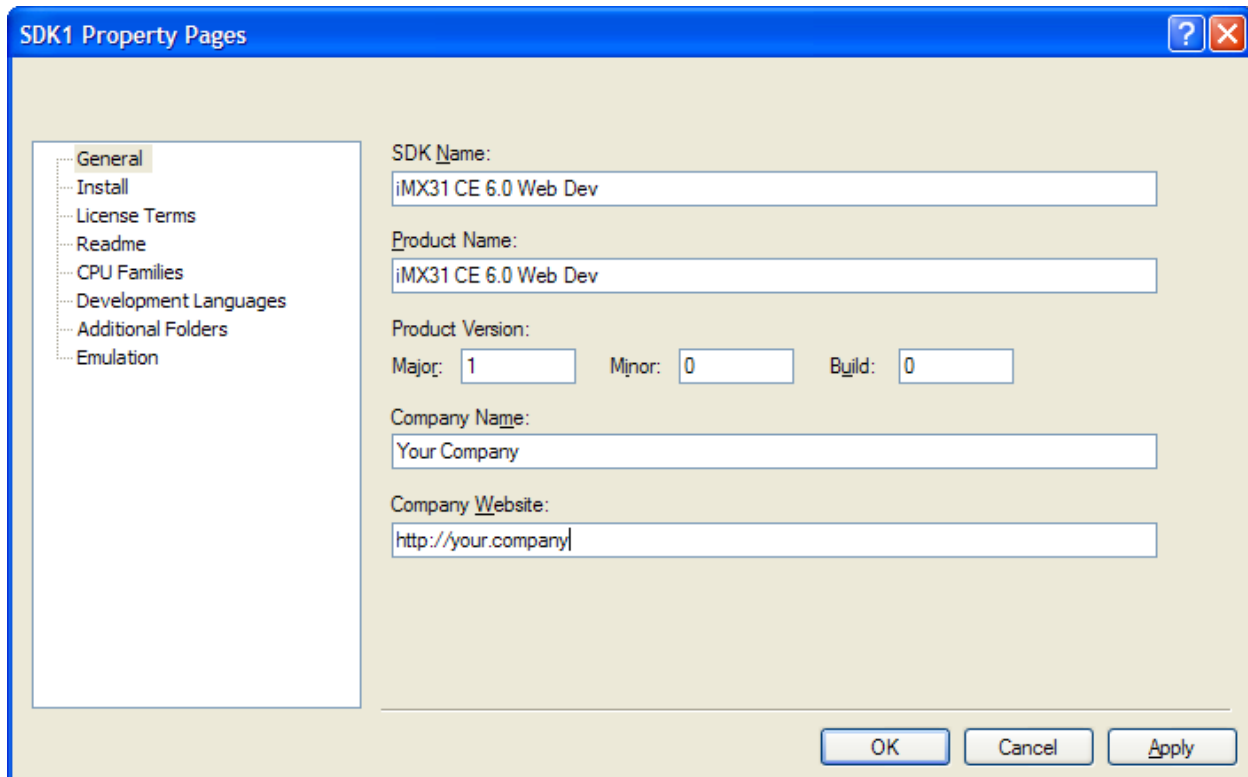
7 Creating an ISAPI Extension DLL

If you wish to add a web based frontend to an application on your board, the most straightforward way to do so is of course to write an ASP page. If ASP is too slow or not powerful enough for you and you're familiar with C/C++ programming, however, you can create an ISAPI Extension instead. You can write a custom DLL which exports several functions, and then map it to a virtual directory on your server. Then, whenever a client requests the page at that virtual directory, the web server will call a function in your DLL to generate the response.

7.1 Export an SDK

Currently, ISAPI extensions must be developed with native code. You'll have to export an SDK from your OS Design in order to develop code to run on it.

- Return to the instance of Visual Studio with your OS Design
- Open the **Solution Explorer** view by selecting **View | Solution Explorer**
- Right click **SDKs** and select **Add New SDK**
- Choose a descriptive name such as **iMX31 CE 6.0 Web Dev**

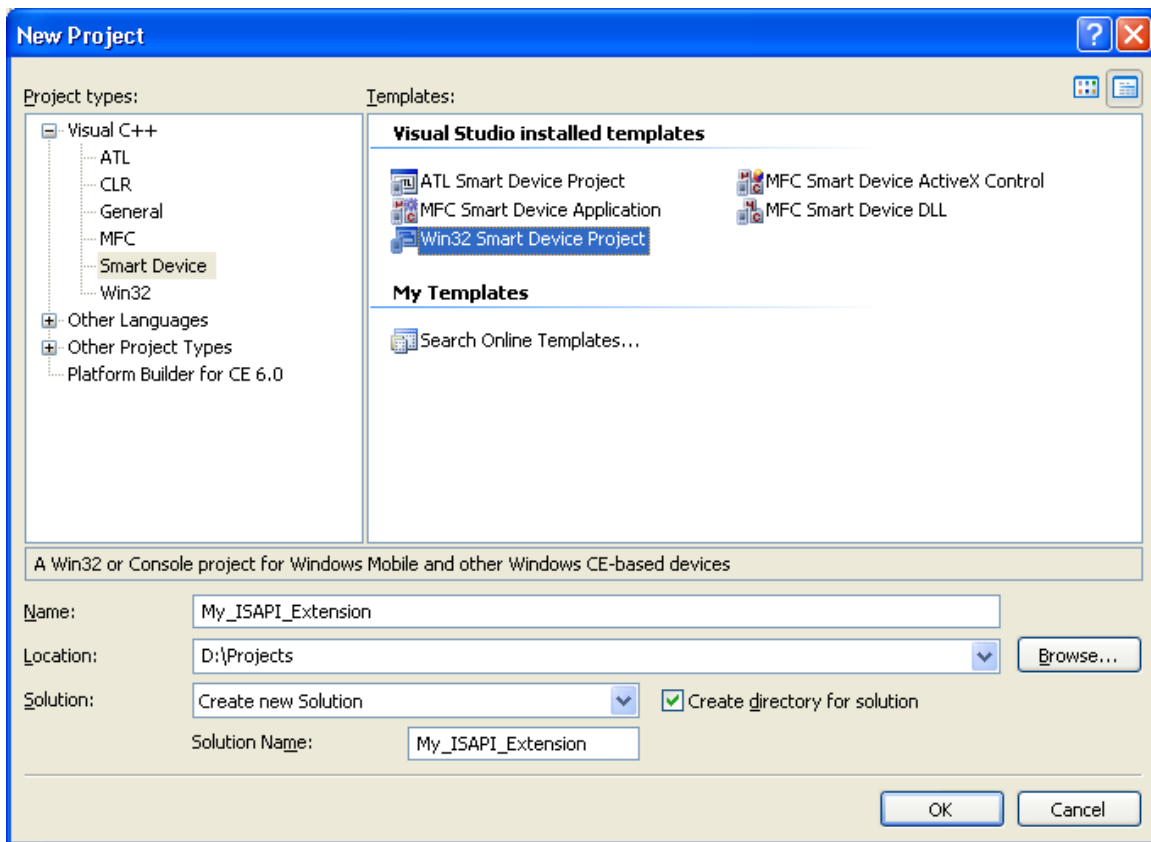


- Select the **Development Languages** tab on the left
- Verify that **Native Development Support** is checked
- Click **OK** to finish creating the SDK
- Verify that your OS Design is still set to build in **Release** mode (TrainingBSP ARMV4I Release)
- Right click the new SDK and select **Build**
- After the build, navigate to the new SDK's MSI installer (By default, it will be in C:\WINCE600\OSDesigns\OSDesign3\OSDesign3\SDKs\SDK1\MSI) and install the new MSI on your development workstation

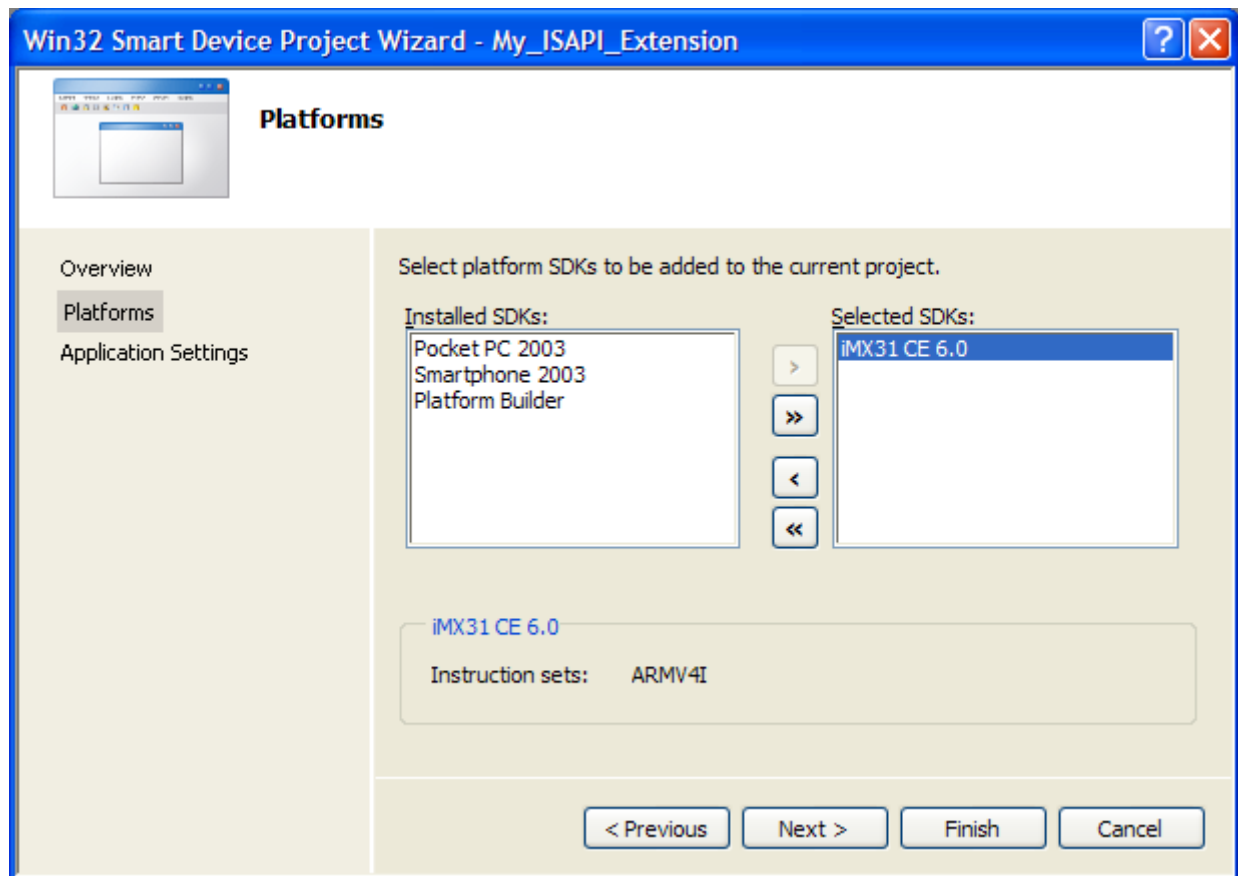
7.2 Create the ISAPI DLL Project

Now that we have an SDK built and installed, we can build a DLL that will run on our platform. ISAPI DLLs are just normal DLLs that export 2-3 specific functions

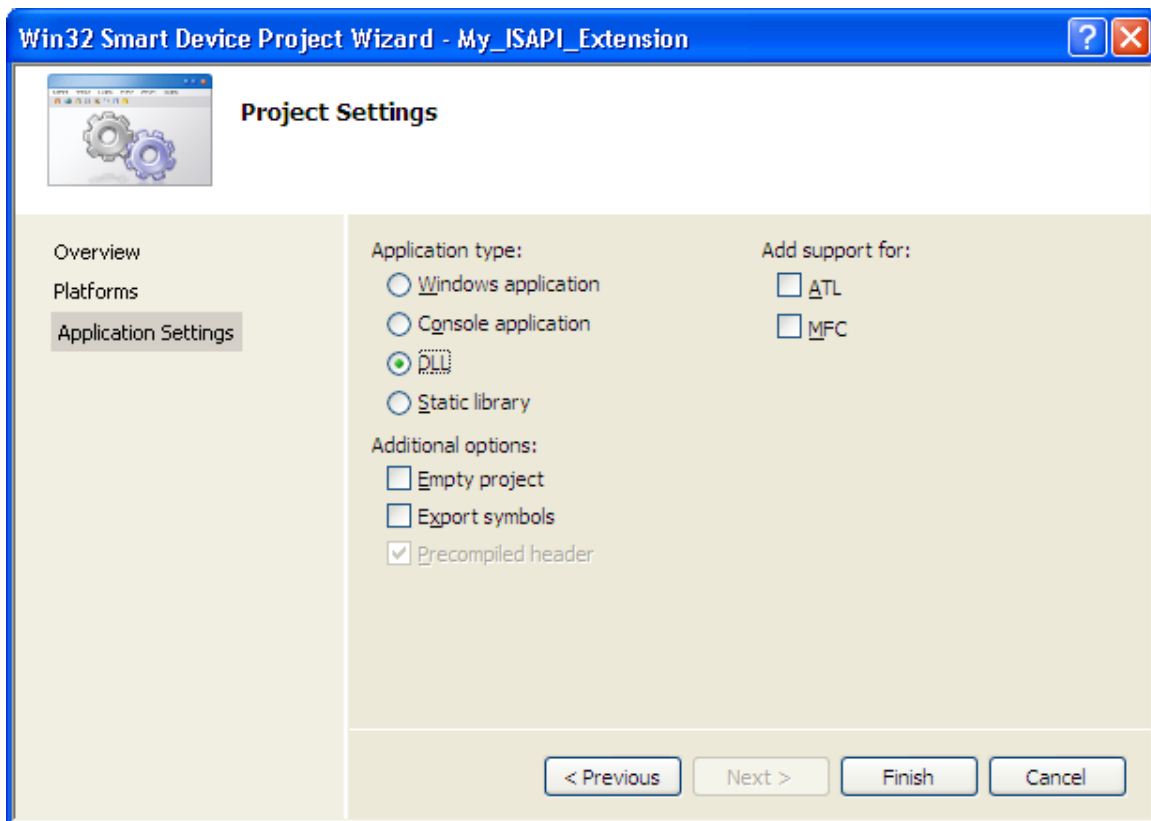
- Open a new instance of Visual Studio
- Create a new project by going to **File | New | Project**
- Select **Visual C++ | Smart Device** from the **Project Types** on the left, and **Win32 Smart Device Project** from the templates on the right



- Select **Next** when the wizard starts to reach the **Available Platforms** screen
- Select the newly installed SDK in the **Available Platforms** screen and click the right arrow to put it in the list of platforms for the project. Remove Pocket PC 2003 from the list on the right by selecting it and clicking the left arrow



- Select **Next**
- At the **Project Settings** screen select **DLL**

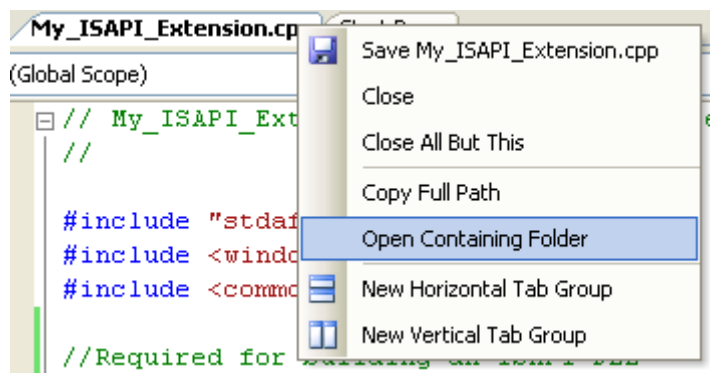


- Press **Finish** to create the project
- You should now have a basic DLL which exports no symbols and just has a mostly empty `DLLMain` function in a file called `My_ISAPI_Extension.cpp`
- To work as an ISAPI Extension, our DLL must export two functions:
 - **GetExtensionVersion** will be called by the web server when it first loads the DLL to obtain basic information about it
 - **HttpExtensionProc** will be called whenever a client accesses something which is mapped to the DLL. This is the main workhorse of the DLL
- Additionally, if it's available, **TerminateExtension** is called just before your extension is unloaded to give you a chance to free up any resources allocated
- Template code for an ISAPI Extension is available with the LabFiles for Lab 3, in `C:\Labs\Lab3\LabFiles`
- Add the contents of the file **ISAPI_Header.c** to your DLL's code above `DllMain`
- Add the code for the exported functions **HttpExtensionProc** and **GetExtensionVersion** below `DLLMain`. You can find this code in the LabFiles in the file **ISAPI_Functions.c**
- Set the DLL to build in **Release** using **Build | Configuration Manager** or just using the pulldown box in the main toolbar
- Select **Build | Build Solution** to build the DLL and verify there are no compilation errors. If there are, make sure the correct header is specified in the CPP file, and make sure it's part of the SDK that was installed. Note that we still haven't exported any functions from the DLL yet.

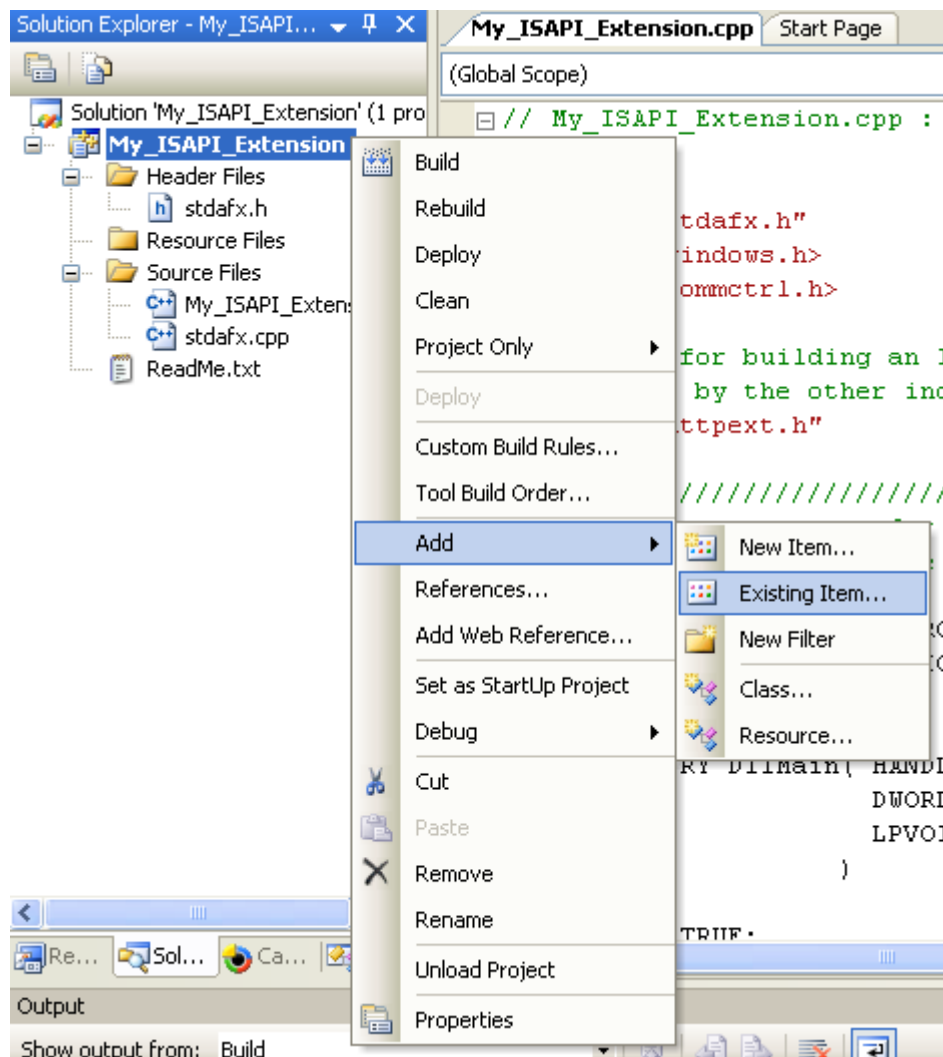
7.3 Export Functions from the DLL

Now that our DLL builds correctly, let's add a .def file to export functions from it, so the web server will be able to load and use the DLL

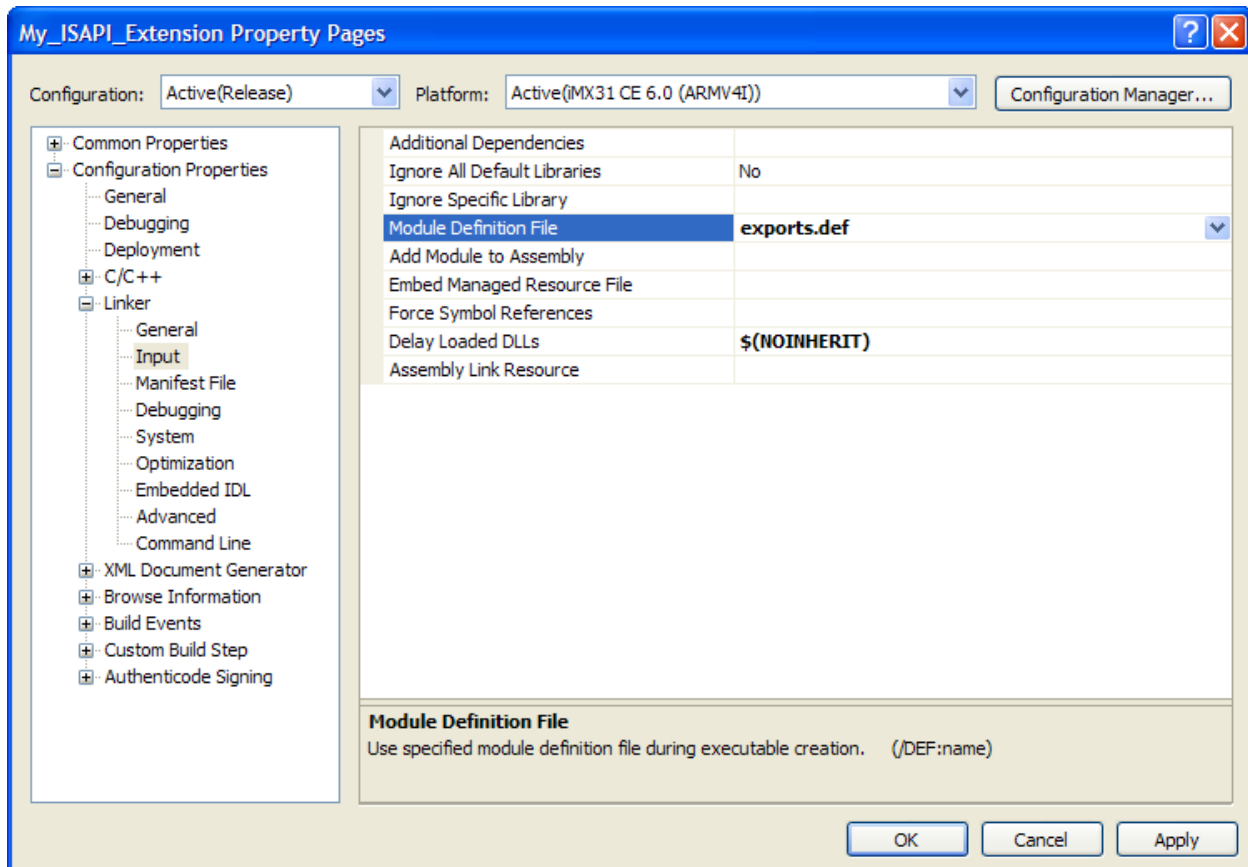
- Copy **exports.def** from the LabFiles directory into the same directory as your project. The quickest way to open a window at your project is to right click the name of the CPP file and select **Open Containing Folder**



- Add the **exports.def** file to the project by right clicking **My_ISAPI_Extension** in the Solution Explorer view and selecting **Add | Existing Item**



- Select **All Files** for the file type and double click **exports.def** to add it to the project
- Tell the linker to use this new def file by right clicking **My_ISAPI_Extension** again and selecting **Properties**, then expand **Linker** and select **Input**

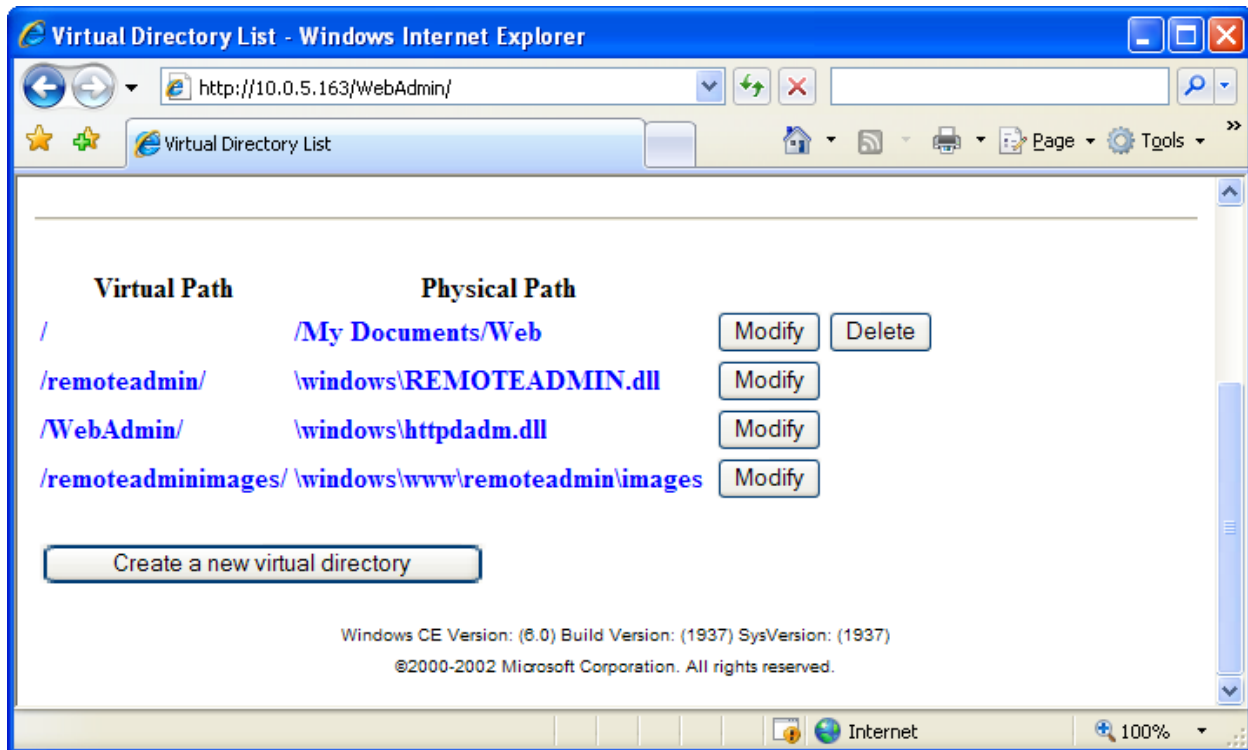


- Set the **Module Definition File** to **exports.def**
- Click **OK** at this window
- Select **Build | Rebuild Solution** to build a DLL that exports the necessary functions

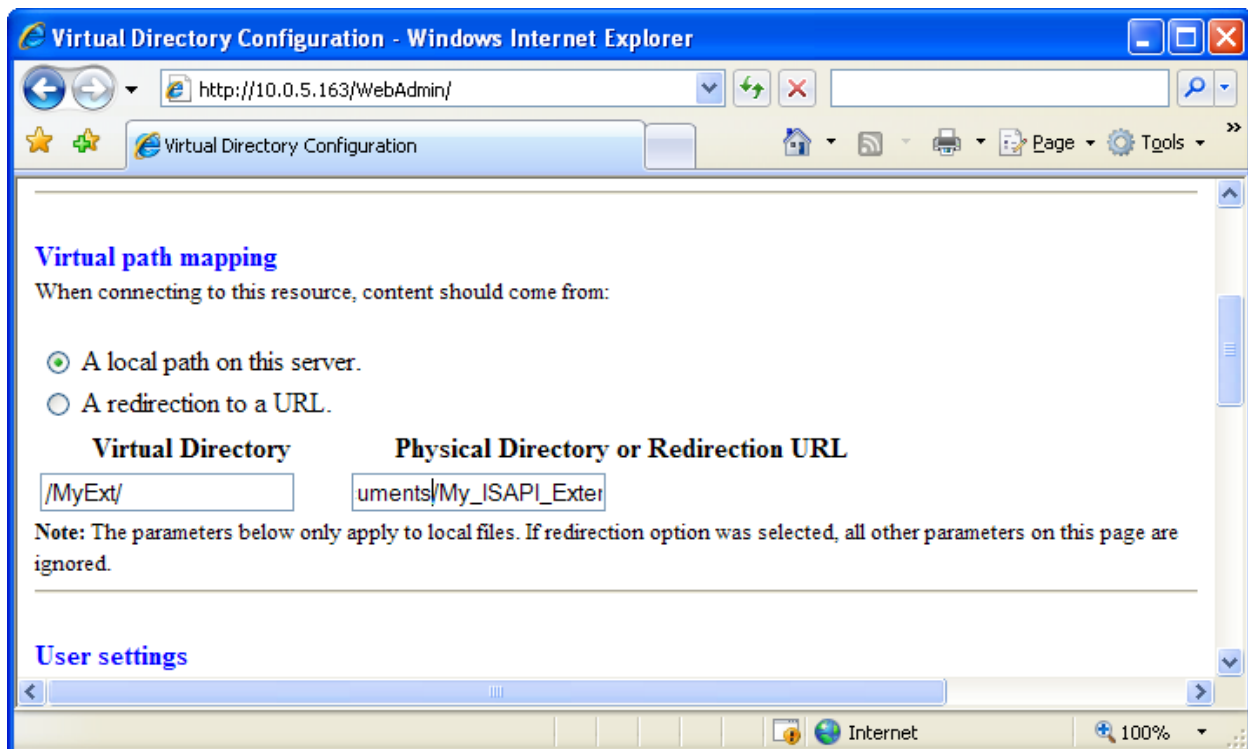
7.4 Using the ISAPI Extension

Now that we have a built version of our ISAPI Extension, let's put it on the device and set the device's web server to use it.

- Open your ISAPI Extension's release directory, which will be a subfolder of the folder you created the project in
- Open a window pointing to your device's MyDocs share
- Copy **My_ISAPI_Extension.dll** onto the device. For this example, we'll put it in the MyDocs share
- Open your web browser and point it to the WebAdmin page (at your device's IP address /WebAdmin/)
- You may have to log in as username ADMIN and the password you chose earlier
- Select **Modify**
- Select **Configure the virtual directories on this Web site**
- Select **Create a new virtual directory**

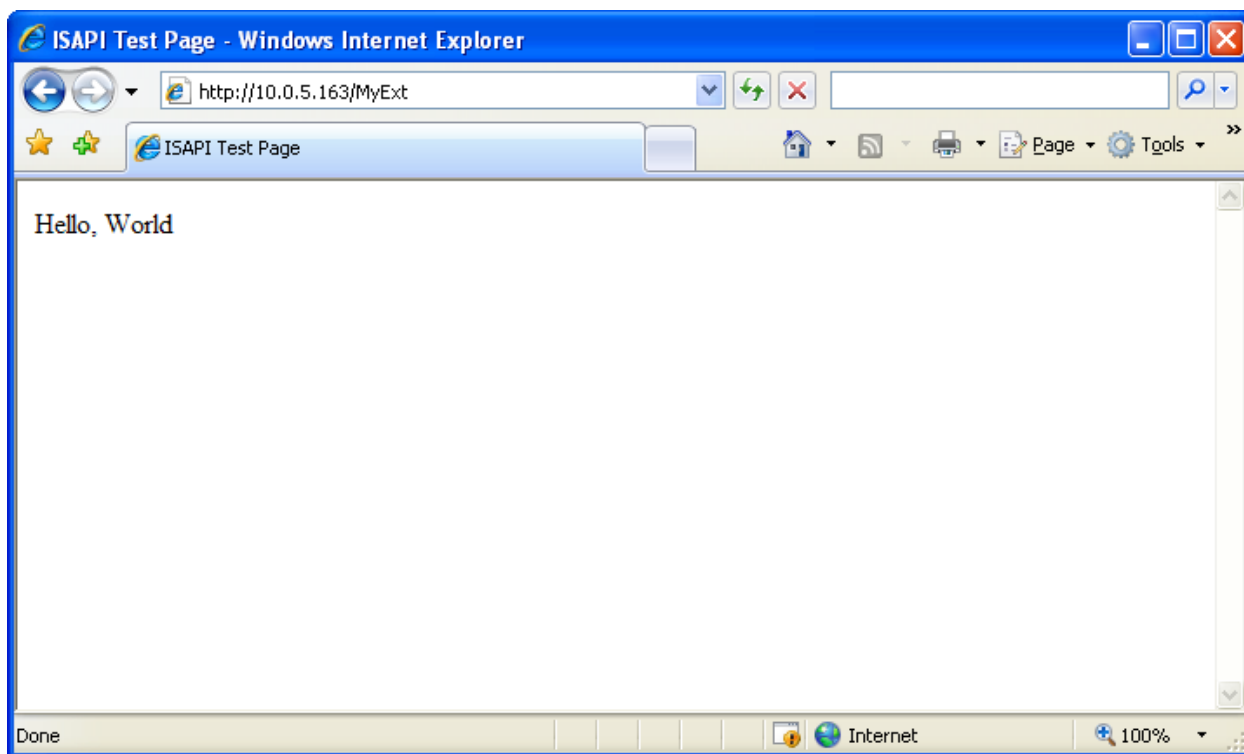


- At this screen, set the Virtual Directory to **/MyExt/** and the Physical Directory to **/My Documents/My_ISAPI_Extension.dll**



- Click **Update** at the bottom of the screen

- You'll see a message at the top of the screen indicating that you should restart the web server for your modifications to take effect. Click **Restart Web Server** at the top right to restart the web server.
- Navigate to `http://(your device's IP address)/MyExt/`



- The web server will load your new ISAPI extension and you'll see a "Hello, World" message in your browser

7.5 Modifying the ISAPI Extension

Now we'll demonstrate how to alter the new DLL and see the changes take effect.

- First we'll add some code to the DLL so we know it's using the updated version. In the function `HttpExtensionProc`, change "Hello, World" to a message of your choosing.
- Add `RETAILMSG` statements at the beginning and end of the function. `RETAILMSG` statements act like `wprintf`, except they only trigger if their first parameter evaluates to true, and they're automatically removed from SHIP builds.

```
DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pEcb)
{
    RETAILMSG(1, (L"My_ISAPI_Extension's work begins\r\n"));

    SendHTMLHeader(pEcb);

    Output(pEcb, "<html>");

    Output(pEcb, "<head><title>ISAPI Test Page</title></head>");
```

```

Output(pEcb, "<body bgcolor=#FFFFFF>");

Output(pEcb, "Hello again, World");

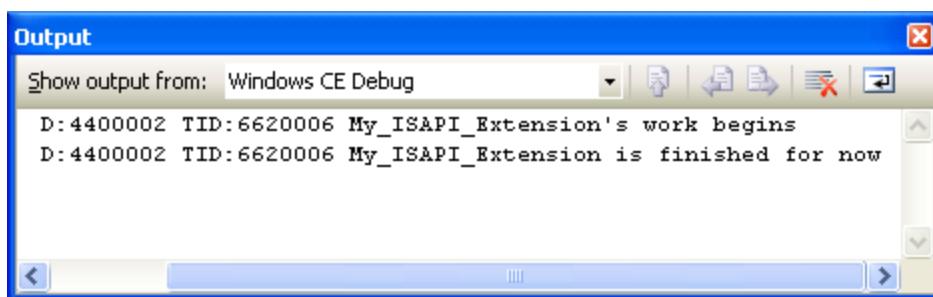
Output(pEcb, "</body></html>");

RETAILMSG(1, (L"My_ISAPI_Extension is finished for now\r\n"));

return HSE_STATUS_SUCCESS;
}

```

- Build the updated version of the DLL
- Restart the web server again through the WebAdmin interface or by returning to the telnet prompt and running “**services refresh http0:**”
- Delete the My_ISAPI_Extension.dll in the MyDocs folder on your device, then copy the new one into its place.
 - If you encounter an error, it may be that the web server loaded your DLL again. Restart the server again, or remove the virtual mapping to the DLL, restart, replace the DLL, and re-add the mapping. This shouldn't generally be necessary, however
- Load the URL pointing to your ISAPI Extension again. It should print the new message you replaced Hello, World with, and the Windows CE Debug output window should show the output of the RETAILMSG statements you added



8 Summary

You have now completed all the steps in this guide. Here's what we have covered:

- Adding and configuring a web server with ASP support
- Adding and configuring a Windows SMB/CIFS File Server
- Setting up a Telnet server
- Setting up and using the remote display application
- Creating an ISAPI DLL