

CC Pilot™ XS

Application developers guide



Table of Contents

INTRODUCTION	3
<i>Purpose</i>	3
<i>References</i>	3
<i>History</i>	3
CC PILOT XS – COMPUTER MANAGEMENT	4
<i>Copy files via USB</i>	4
<i>Using the FTP server</i>	4
<i>Using telnet to CC Pilot™ XS</i>	7
<i>Controlling the Video and CAN power output</i>	9
<i>Using the built in GPRS modem</i>	11
<i>Hide Windows CE desktop</i>	15
<i>Auto start applications</i>	19
SOFTWARE DEVELOPMENT WITH VISUAL STUDIO 2005	22
<i>Software requirements</i>	22
<i>Developing a .NET application (managed code)</i>	22
<i>Developing a C/C++ application (unmanaged code)</i>	24
<i>Connect to Device</i>	28
<i>Deployment and Debugging</i>	31

Introduction

Purpose

This document contains tips and information about “how to” developing applications for the CC Pilot™ XS.

References

-

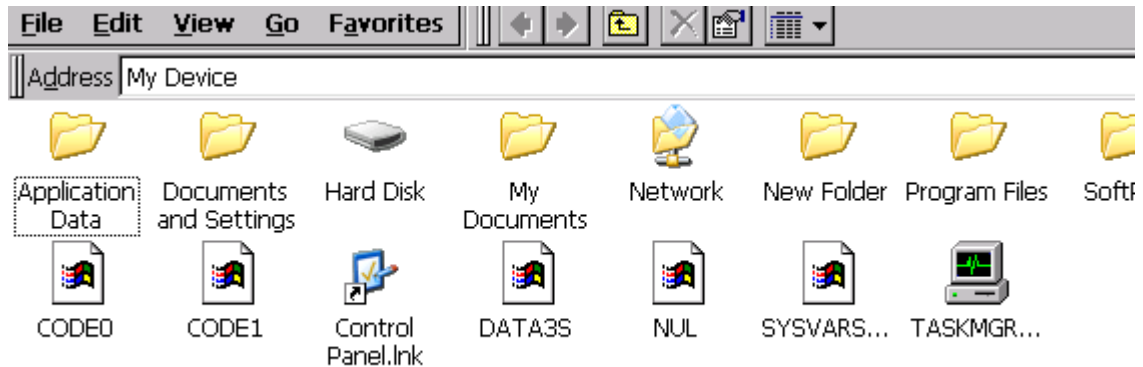
History

Rev	Date
1.0	2008-04-22
1.1	2008-06-26

CC Pilot XS – Computer management

Copy files via USB

An easy way to copy files to and from the CC PILOT™ XS is to use a USB memory. This has the advantage that the CC PILOT™ XS doesn't need to be connected to a network. When a USB memory is connected via the USB cable to the CC PILOT™ XS it will show up in Explorer under My Device and have the name Hard Disk. To copy simply copy paste the standard windows way.



Using the FTP server

The CC Pilot™ XS hosts a built-in FTP server that can be used for remotely upload or download information from the unit.

If the CC PILOT™ XS are connected to the development PC via Ethernet, files can be copied using FTP. This can for example be done via Explorer in Windows.

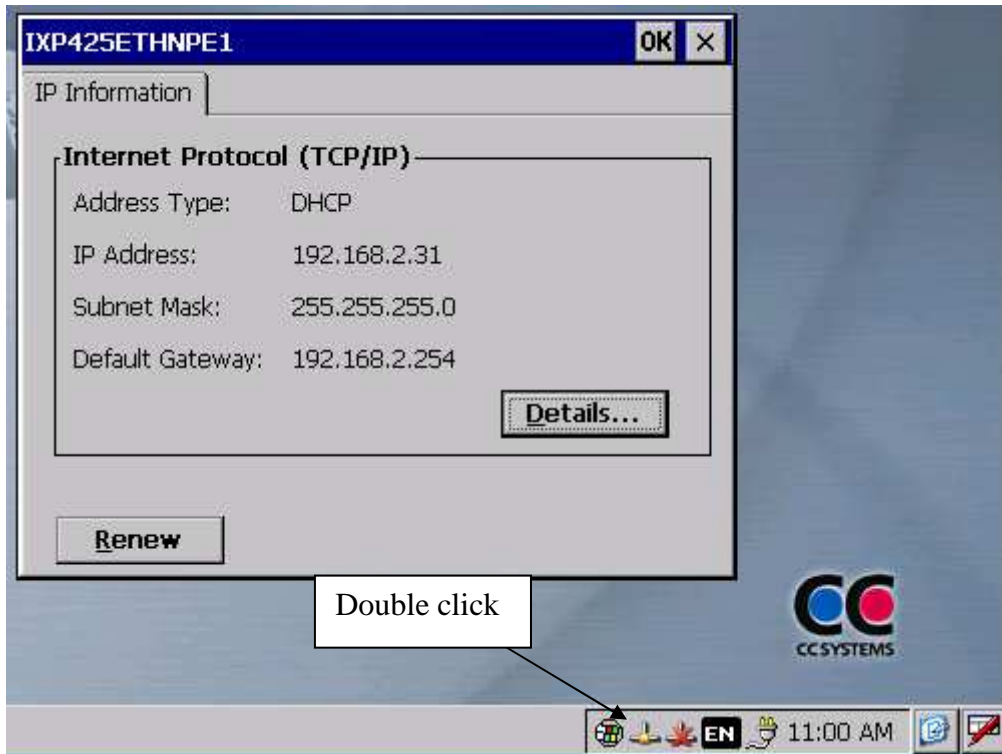
The default security settings for FTP is to allow anonymous access, which gives the connecting entity possibilities to upload files, delete files and rename files without authorization. Anyone can log in using the username "anonymous" and any password to gain access.



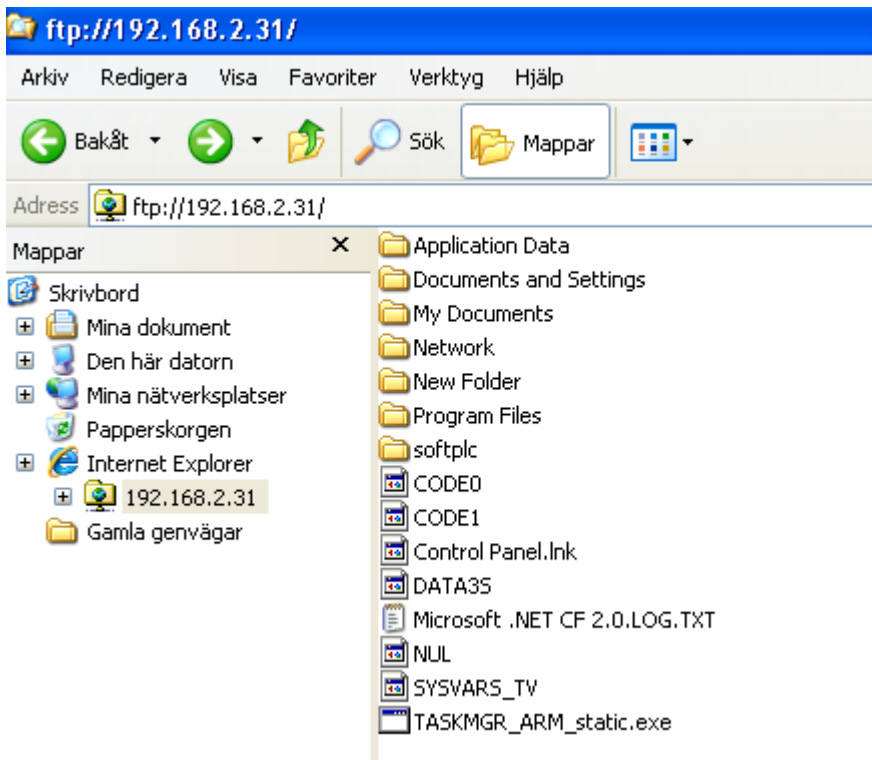
Be aware that when connecting CC Pilot™ XS to a network the FTP server is open for anyone to access and can be a security risk.

Example:

1. Find out the IP-address of the CC PILOT™ XS. This can be done by double clicking on the network icon in the lower right of the CC PILOT™ XS screen which brings up IP information.



2. Start explorer on your PC, in the address field of explorer, type ftp://<ip-address>, e.g. ftp://192.168.2.31 and press enter.



This brings up the file system on your CC PILOT™ XS.

Now it is possible to copy and past files to a location in the CC Pilot™ XS filesystem.

It is also possible to use other FTP clients for the communication within the CC Pilot™ XS.

Disable FTP Server

The built-in FTP server on the CC Pilot™ XS can be disabled by changing a registry key. The FTP server settings are located under the *HKEY_LOCAL_MACHINE\Comm\FTPD* registry key.

<i>IsEnabled</i> REG_DWORD	Default set to 1. Possible values are 0 (false) or 1 (true). Determines whether or not the server will accept incoming connections. 1 (true) – Enables FTP server 0 (false) – Disables FTP server
-------------------------------	--

Using telnet to CC Pilot™ XS

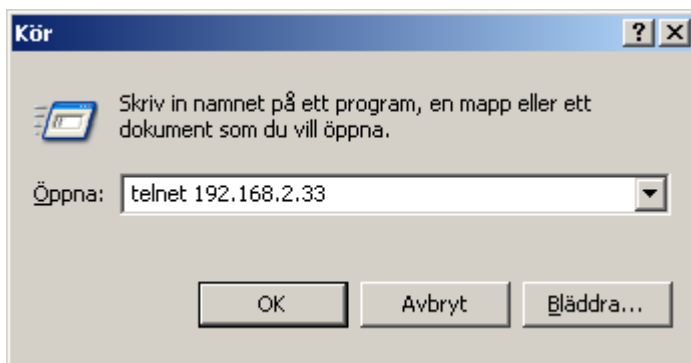
The CC Pilot™ XS hosts a built-in telnet server. This means that it is possible to execute operating system commands remotely on the CC Pilot™ XS.



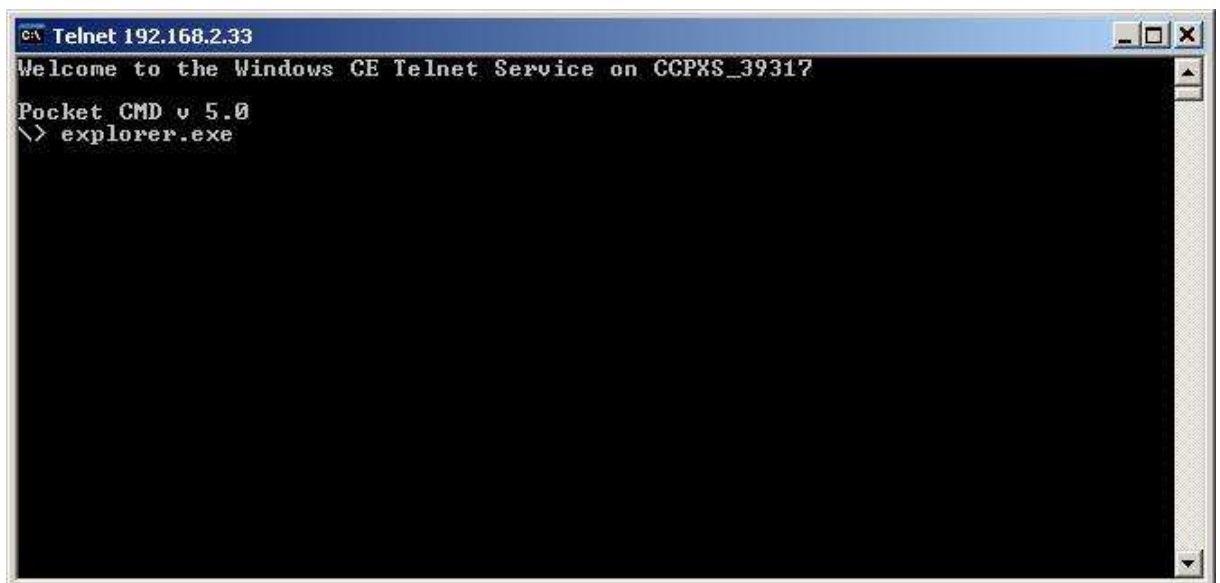
Be aware that when connecting CC Pilot™ XS to a network the telnet server is open for anyone to access and can be a security risk.

For example, start explorer on the CC Pilot™ XS from remote:

1. Press **Start -> Run** on your PC on your local PC.
2. Type **telnet <IP-address>** of the CC Pilot™ XS (example **telnet 192.168.2.33**) and press OK



3. Then type **explorer.exe** and hit **enter**



4. The Windows CE desktop will pop up on the CC Pilot™ XS.

Disable Telnet Server

The built-in telnet server on the CC Pilot™ XS can be disabled by changing a registry key.

The Telnet server settings are located under the *HKEY_LOCAL_MACHINE\Comm\TELNETD* registry key.

<i>IsEnabled</i> REG_DWORD	Indicates if the Telnet server is enabled. Set this value to a non-zero value to enable the server, and to 0 to disable the server. If the sample Telnet server is started and this value has not been set, it defaults to accepting connections.
-------------------------------	---

Telnet also verifies that *IsEnabled* has a non-zero value before servicing each request. However, if *IsEnabled* is set to 0 during the Telnet server's initialization, it will never service any requests, even if the registry key is set to a non-zero value after server start-up.

To start the Telnet server in this case, set *IsEnabled* to a non-zero value and restart the server.

Controlling the Video and CAN power output

The video and the CAN connectors contain power outputs. These power outputs can be controlled automatically by the operating system depending on system mode. How the power is controlled for each system mode, is configured in XS Settings. The control can be set to “don’t care”, in which case the power outlets can be controlled manually by the I/O interface in the Windows environment.

System mode settings

To modify the power output settings, open XS Settings from the Control Panel. Move to the Power tab and select a system mode to modify, in the drop down list. The power settings for the video and CAN power output, are set individually for each system mode.

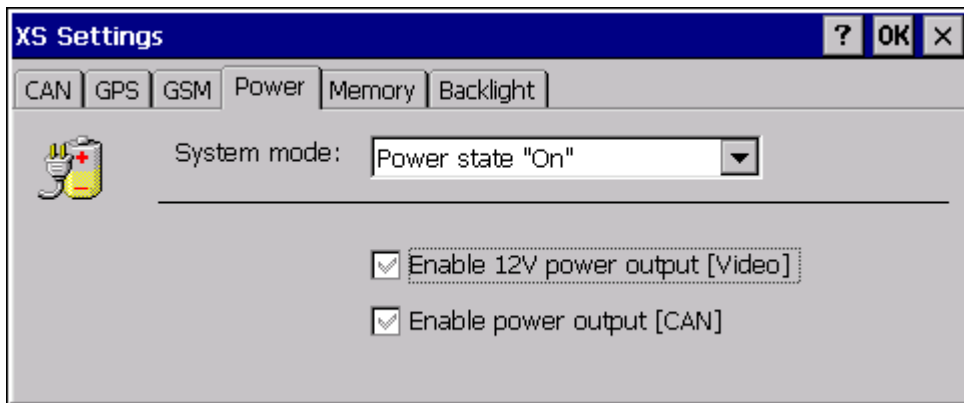


Figure 1 – power tab in XS Settings

The power output settings can be one of the following:

- | | | |
|-------------------------------------|------------|--|
| <input checked="" type="checkbox"/> | Don't care | No change will be made when going into the particular system mode. Use this option when other applications are controlling the power output. |
| <input checked="" type="checkbox"/> | On | The power output will automatically be turned on when entering the selected system mode |
| <input type="checkbox"/> | Off | The power output will automatically be turned off when entering the selected system mode |



Note that controlling the power outputs automatically depending on system mode, can interfere with other programs controlling the output by the I/O interface. Also be aware that putting a high load on the power outlets will result in a higher unit temperature.

Manual control via I/O interface in Windows

If the power output is set to “don’t care” for each system mode, then it will not be changed according to changes in system mode. It can then be controlled by the I/O interface.

To control the video power output from I/O interface, use offset = 0x006, mask = 0x0000 and value = 0x0001 for on or value = 0x0000 for off.

In example code:

```
IoctlIoWrite ioctlIoWrite = {0x0006, 0x0000, "value"};
```

To control the CAN power output from I/O interface, use offset = 0x006, mask = 0x0001 and value = 0x0001 for on or value = 0x0000 for off.

In example code:

```
IoctlIoWrite ioctlIoWrite = {0x0006, 0x0001, "value"};
```

For detailed information about how to use the I/O interface directly, see the SDK document *IO device interface description.pdf*.

Using the built in GPRS modem

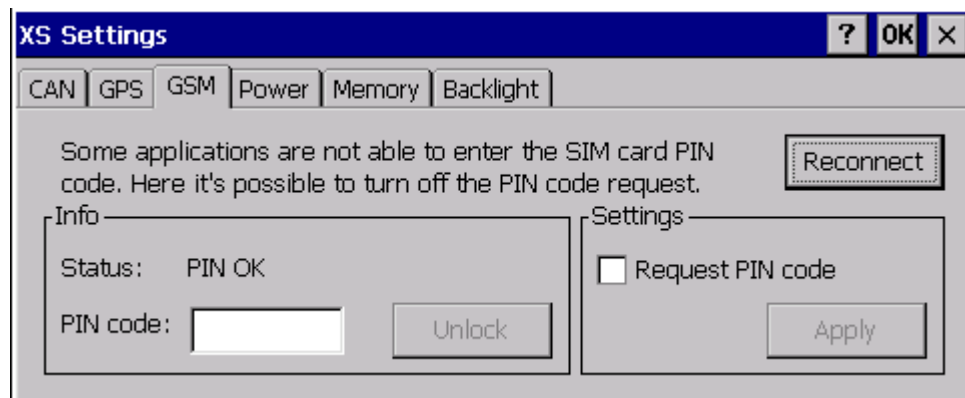
This section contains information about how to get started with the GPRS modem.

Preparing the CC Pilot™ XS

Before the GPRS function can be used, the SIM card must be prepared as follows:

Check that the modem can be accessed and that any SIM activation code is deactivated. Use the XS Settings application in the control panel and click the GSM tab.

Verify that the status says **PIN OK**.



If the application shows any error messages, the SIM activation code may need to be deactivated. This can be done by using the XS Settings application or by putting the SIM card in a mobile phone and deactivating the code from there. For instructions on using the XS Settings application, see the CC Pilot™ XS manual.

Mounting the antenna

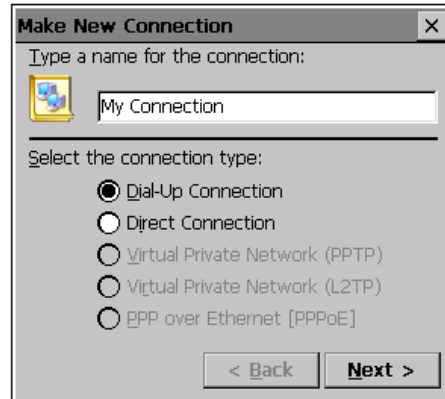
The CC Pilot™ XS can be equipped with both GPS and GPRS. The connectors for these devices are very similar. An easy rule to remember is that the GPRS antenna is always mounted in the connector closest to the SIM card holder.

Connection configuration

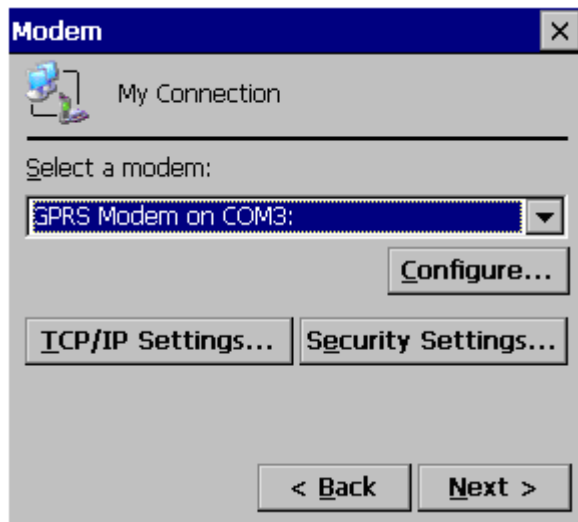
In the following description the settings information used, is for the Swedish operator (Telia). Settings information that needs to be changed to local operator settings is marked in the description.

When making a new dial up connection, most of the alternatives should be left unaltered. But there is some information that needs to be configured, such as:

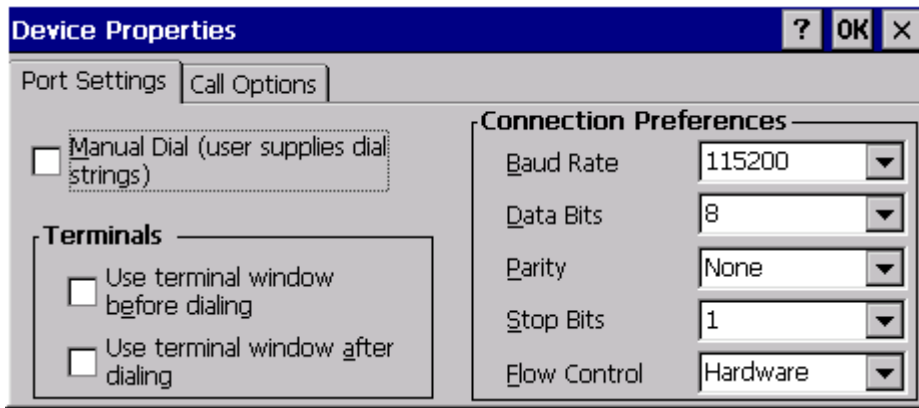
1. Press **Start -> Settings -> Network and Dial-up Connections**
2. Double click on **Make new Connection**



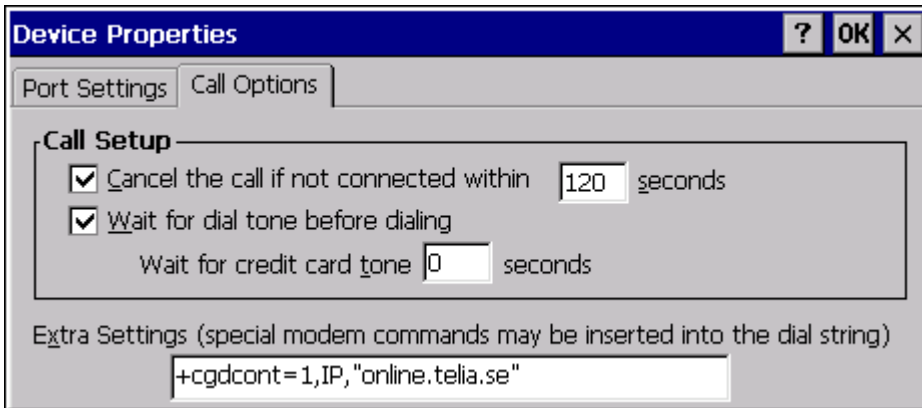
3. Press **Next**
4. Choose **GPRS Modem on COM3**



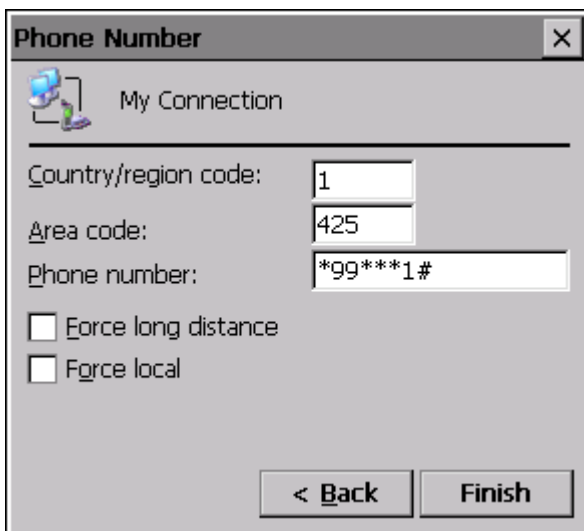
5. Choose to **configure** the modem.
6. Within the tab port settings, check the settings for
 - a. Baud Rate 115200
 - b. Databits 8
 - c. Parity None
 - d. Stop Bits 1
 - e. Flow Control Hardware



7. Within the tab Call options, enter the extra settings text **+cgdcont=1,IP,"online.telia.se"**
Note: online.telia.se should be replaced with the corresponding text for the GSM operator, for instance Deutsche Telecom.

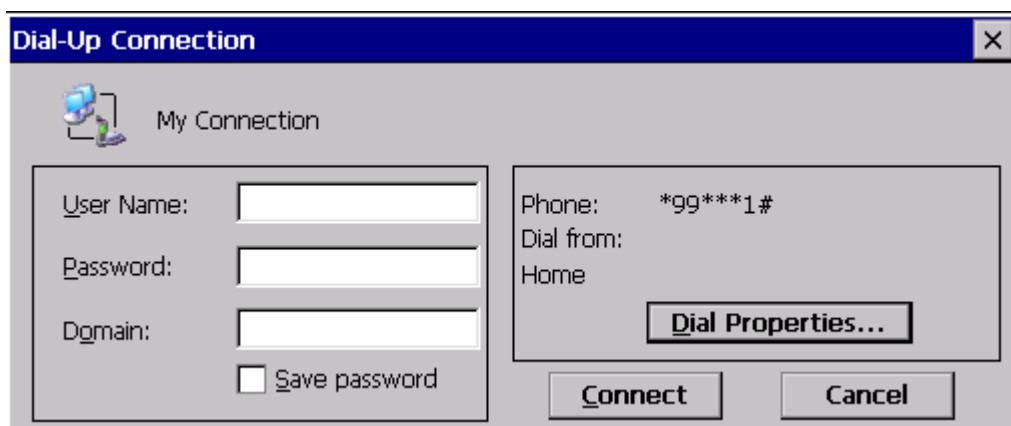


8. Press **OK** and then **Next**
9. Set the phone number to ***99***1#**
10. Keep country as 1 and area as 425 also make sure that force long and local calls are not checked.



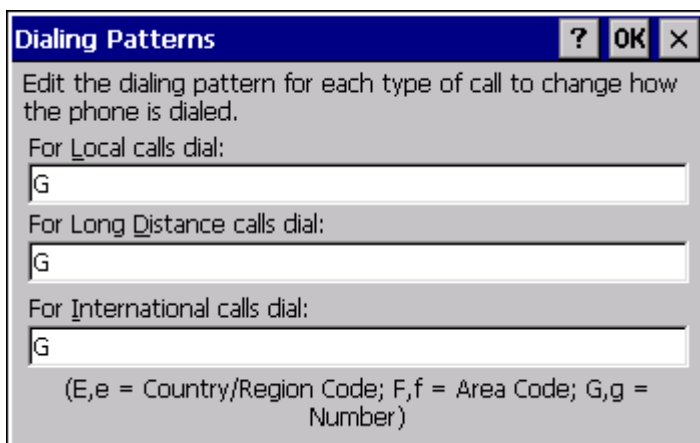
11. Press Finish
12. Open the new connection by double click on the icon

13. The following dialog should appear



14. Open the dial properties...

15. Go into dial patterns and change the content of the tree dial patterns so that it only says G in all three.



16. When going back to the Dial-Up Connection dialog verify that the phone number is *99***1#

Now everything is set up and should work.

Hide Windows CE desktop

What starts on the CC Pilot™ XS is determined by the init registry key. The init strings are launched from a script. This script launches executable files and sets dependencies between them, e.g in the script above device.exe depends on shell.exe, meaning it can't start before shell.exe has started.

The Windows CE desktop is loaded from the program explorer.exe. This section describes how to hide the Windows CE desktop and thereby hide the operating system interface from the user. Hiding the operating system means in practice to prevent Explorer.exe, the graphical shell of windows, from starting.

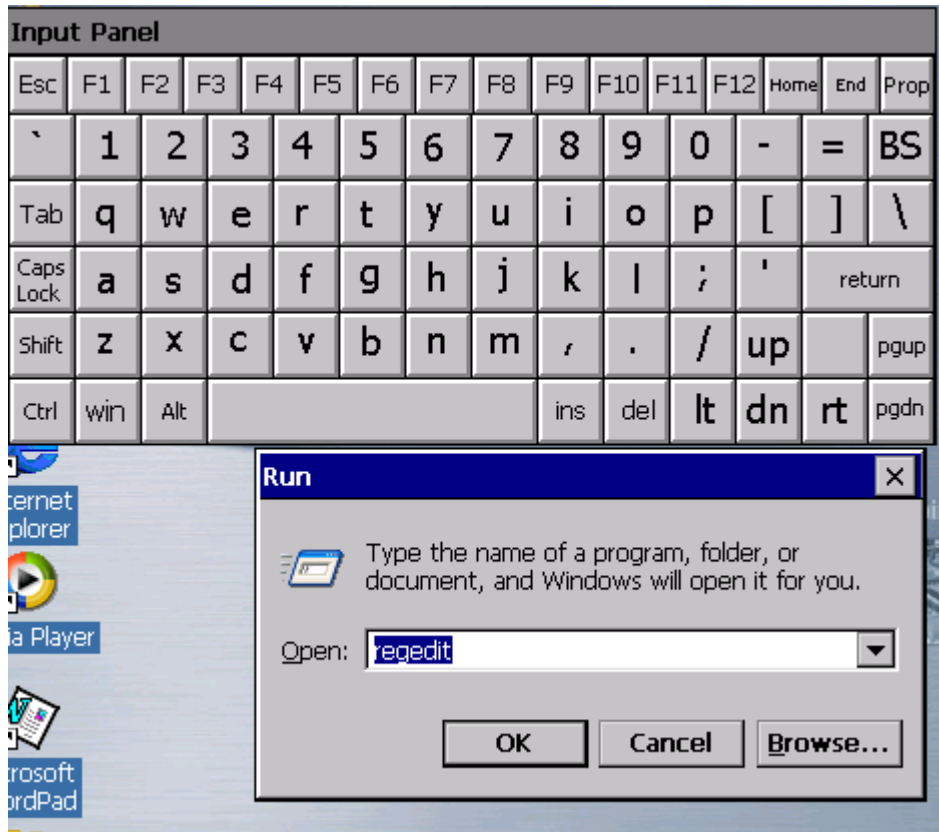
Remove explorer

To remove explorer, apply the following changes in the registry:

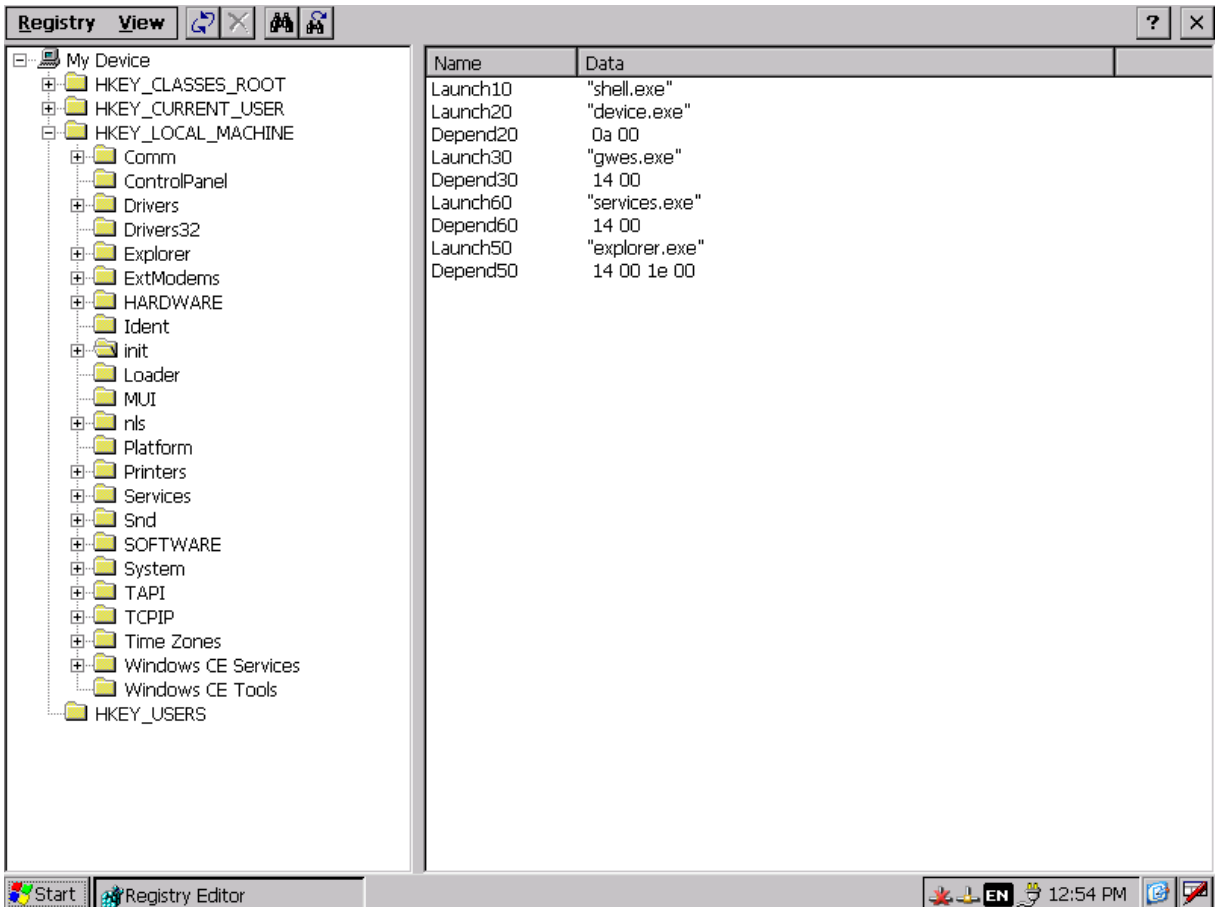
1. Press **Start -> Run** on the CC Pilot™ XS



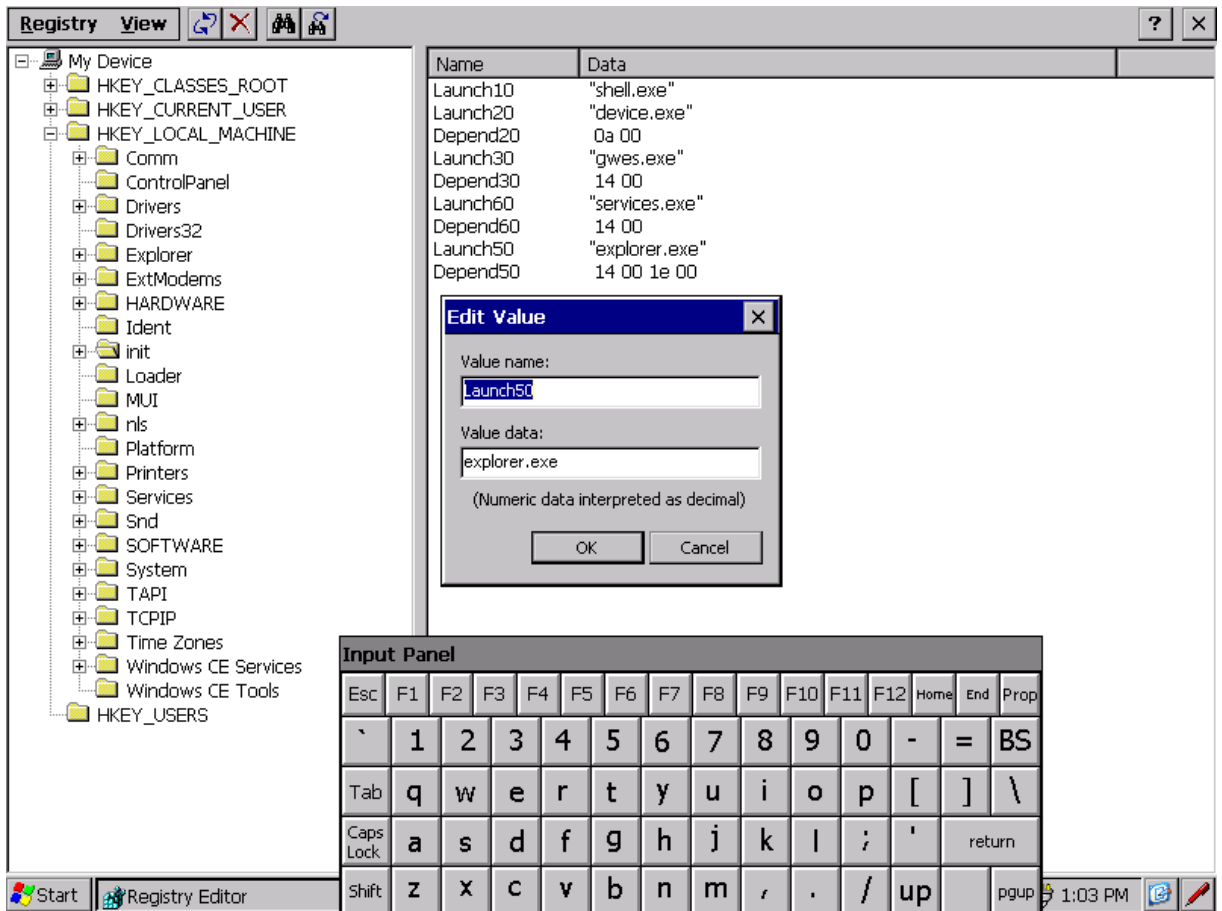
2. Type **regedit** and press OK



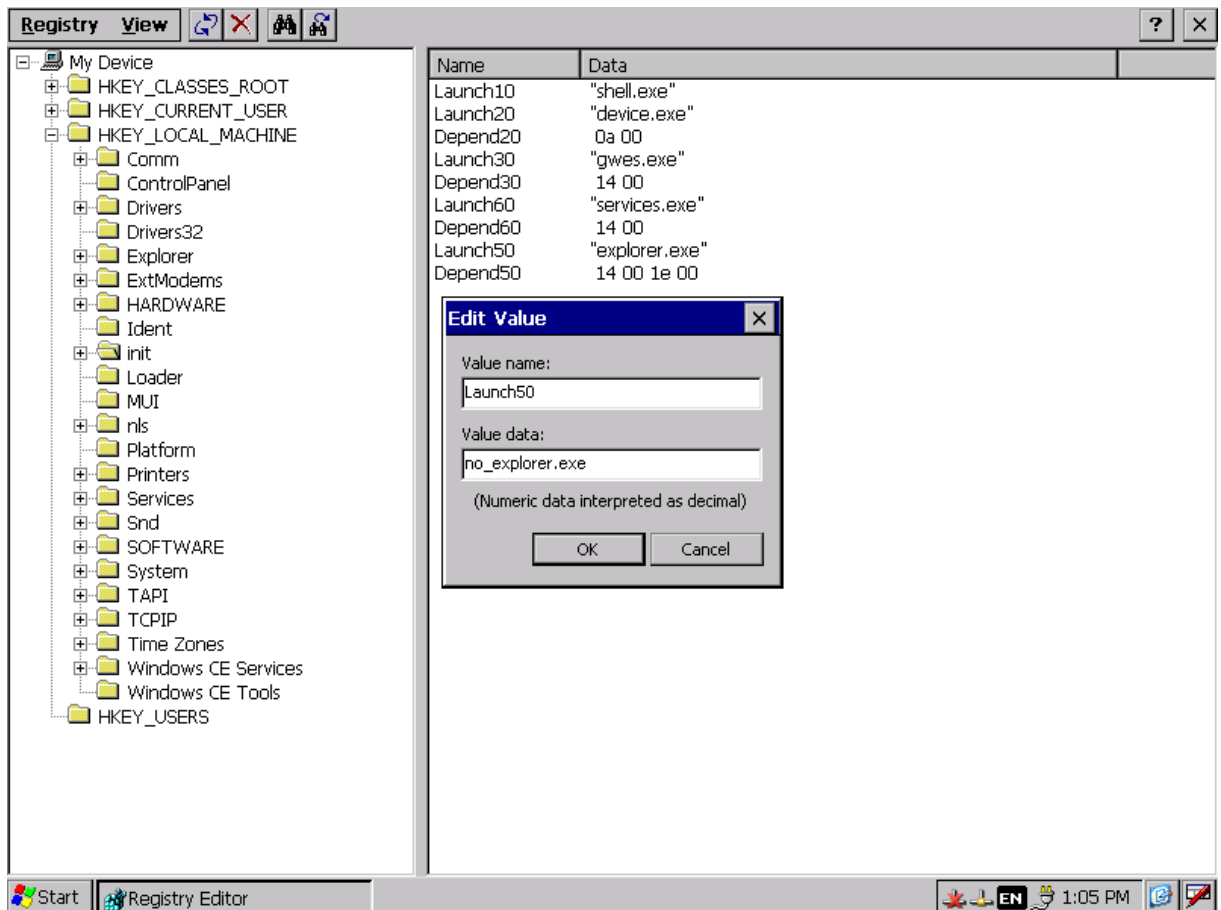
3. Expand the HKEY_LOCAL_MACHINE -> init



4. Double click on the Launch50 key



5. Edit the Value to no_explorer.exe



6. Press OK
7. Press Start -> Suspend

The next time the computer starts without the Windows CE desktop

Recover Windows CE desktop

One important issue is how to bring up explorer and interface the operating system again. This could be needed for service or software upgrade and so on.

Start Explorer from the user application

If e.g. you have buttons or images in your interface you can use them to close your application and bring up Explorer. A good idea might be to demand a certain combination of clicks before starting Explorer, just to make sure it doesn't happen unintentionally. To start explorer from e.g. a C# application can be done with the **Process.Start** method:

```
System.Diagnostics.Process.Start(\\windows\\explorer.exe, null);
```

The user application can simply be closed with **this.Close();**

Start Explorer via telnet

If connecting your development PC to the CC Pilot™ XS via Ethernet is an option, telnet can be used to start Explorer. If the IP-address of the CC Pilot™ XS is known, which is most probably achieved by setting it static. See section Using telnet to CC Pilot™ XS on how to use Telnet.

Auto start applications

This section describes how to make the application start automatically on the CC Pilot XS. Auto start is created by putting in init strings to the registry.

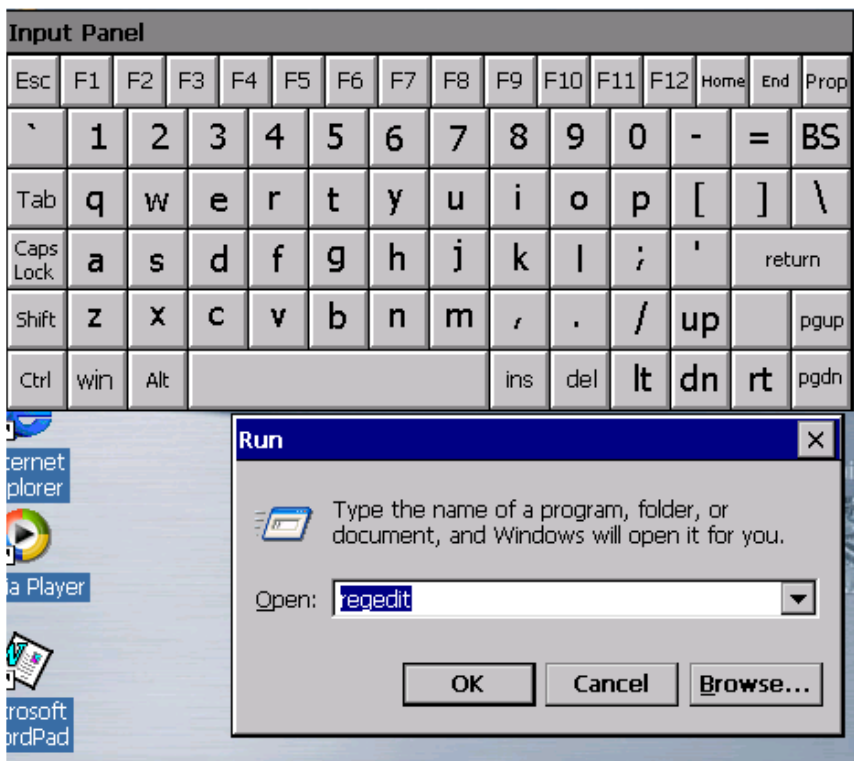
Auto start application

To auto start the example.exe, apply the following:

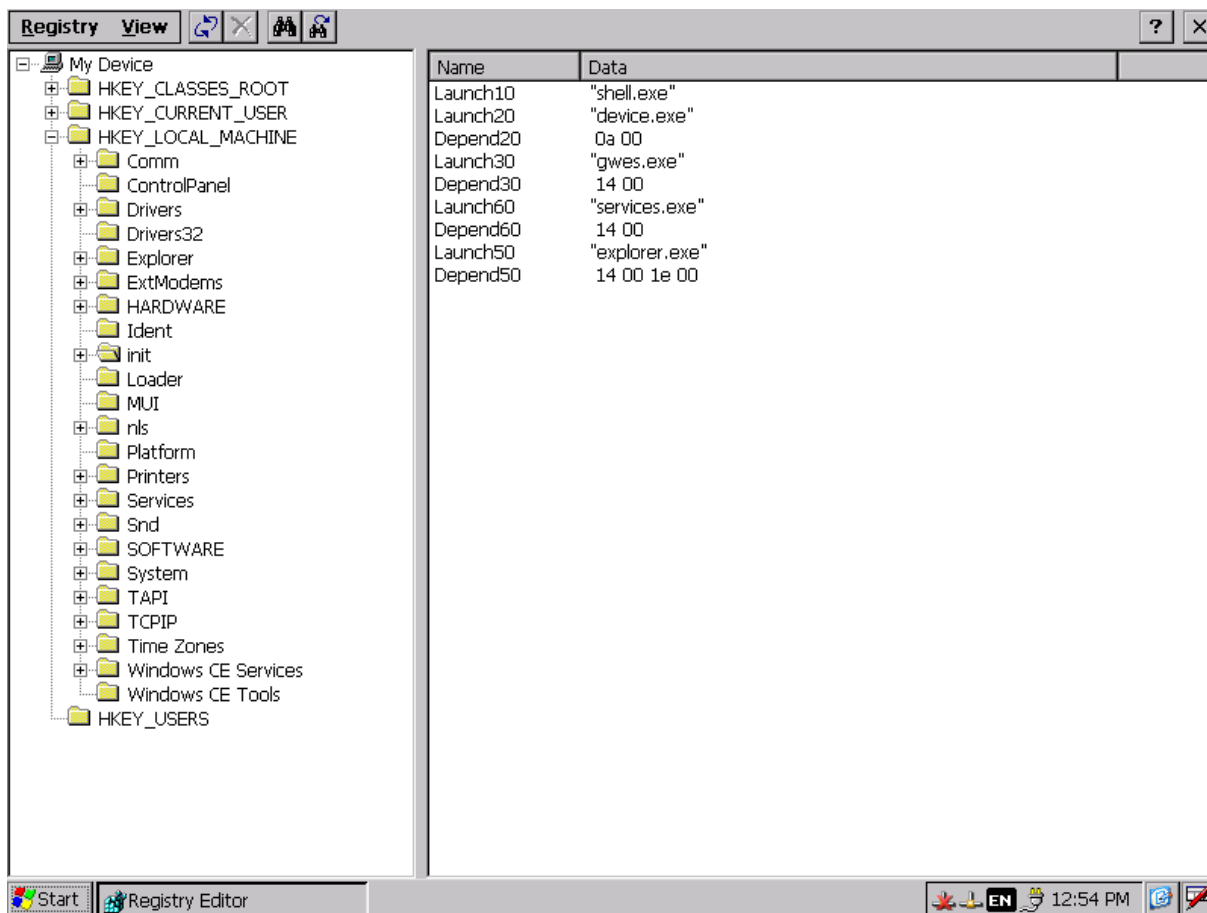
1. Press **Start -> Run** on the CC Pilot™ XS



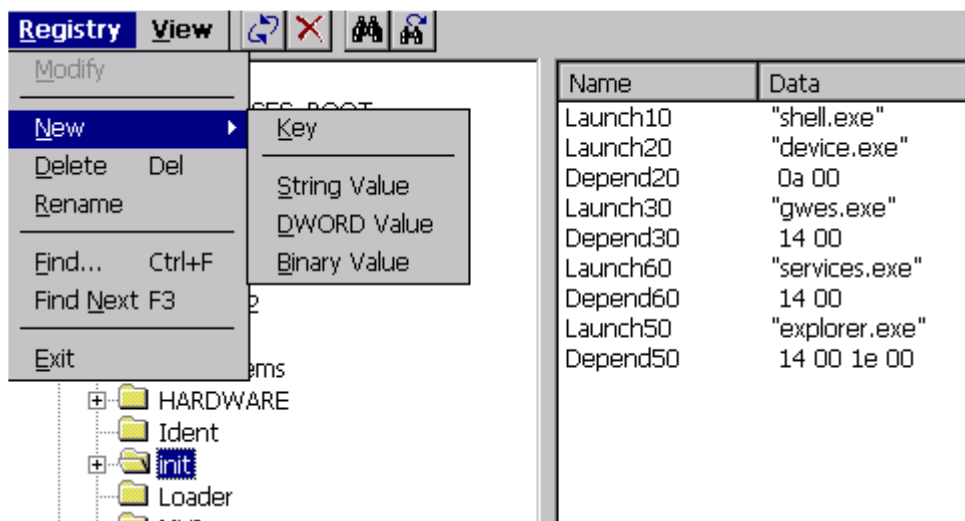
2. Type **regedit** and press OK



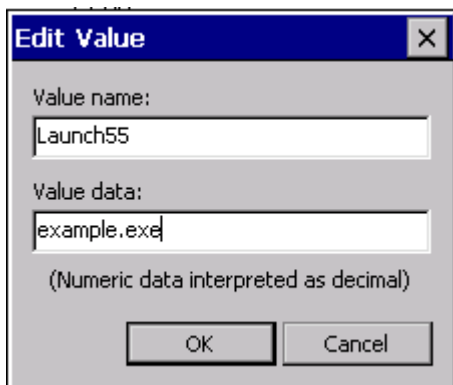
3. Expand the **HKEY_LOCAL_MACHINE -> init**



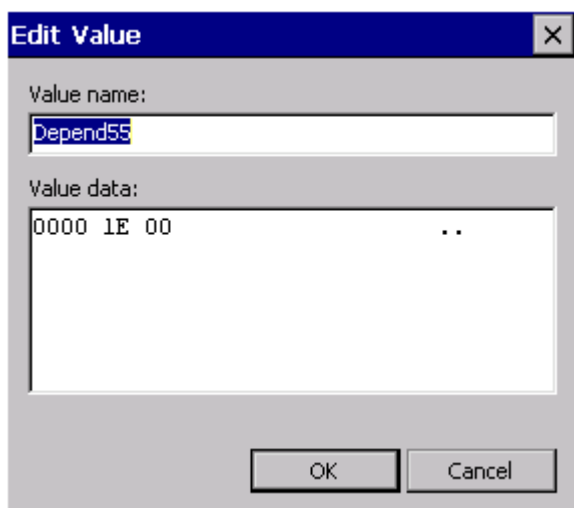
4. Add a new string value by choosing **Registry -> New -> String Value**.



5. Give the value name and value data in the dialog that comes up. An application called example.exe would be started by adding the string value with name Launch55 and value "example.exe."



6. Add the binary Depend value, this is important because it handles the start order of applications. In this case it crucial that gwes.exe which handles graphics start before all other applications. Set the value name and value data in the dialog that comes up. The value name for the example Launch55 should be Depend55 and the value data should be 1E 00.



7. This will start the application at boot up, just after gwes.exe has started.

Note: It isn't possible to start applications by putting them in the \Windows\StartUp folder unless Explorer.exe is started.

Software development with Visual Studio 2005

Software requirements

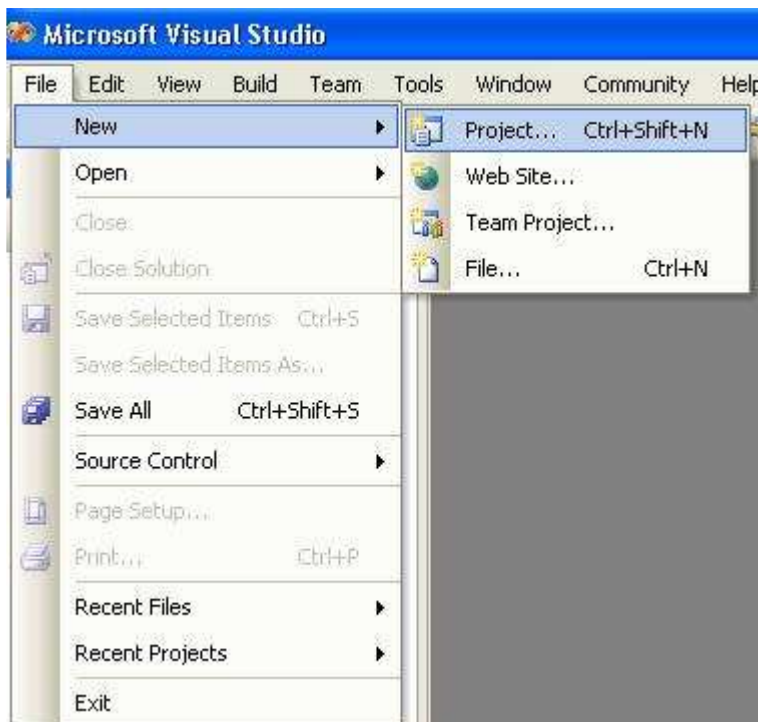
Visual Studio 2005 will be assumed.

Developing a .NET application (managed code)

This instruction will guide you through the steps required to make a C# application targeted for the CC Pilot™ XS.

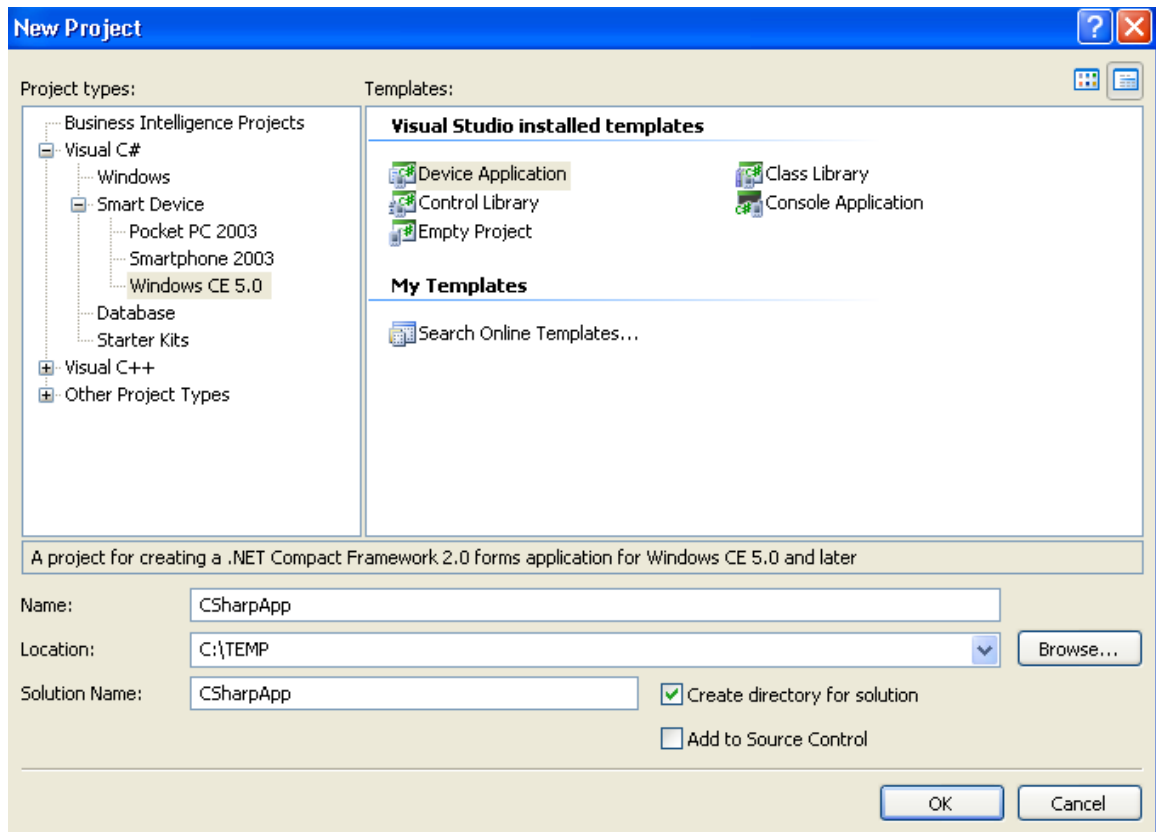
Start a new project

Choose **File -> new -> project**

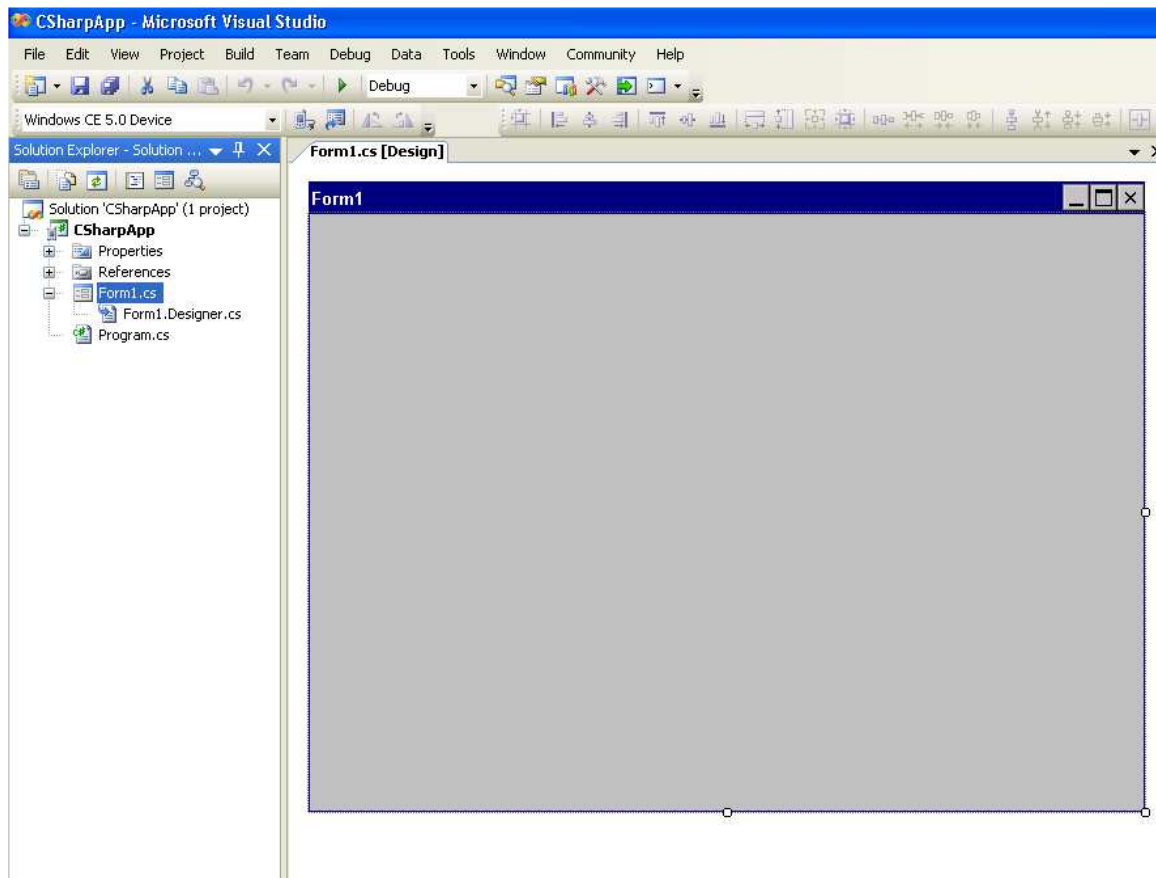


Under **Project types** choose **Visual C# -> Smart Device -> Windows CE 5.0** and under **Templates** choose the desired template e.g. **Device Application**.

Choose a path and name for the application, it might also be a good idea to create a directory for your solution. Accept the settings by pressing **OK**.



The following will be the automatically generated skeleton.



To generate an executable hit **Build -> Build Solution**. This generates the executable file CSharpApp.exe found in the build directory, in this case under **C:\TEMP\CSharpApp\CSharpApp\bin\Debug**.

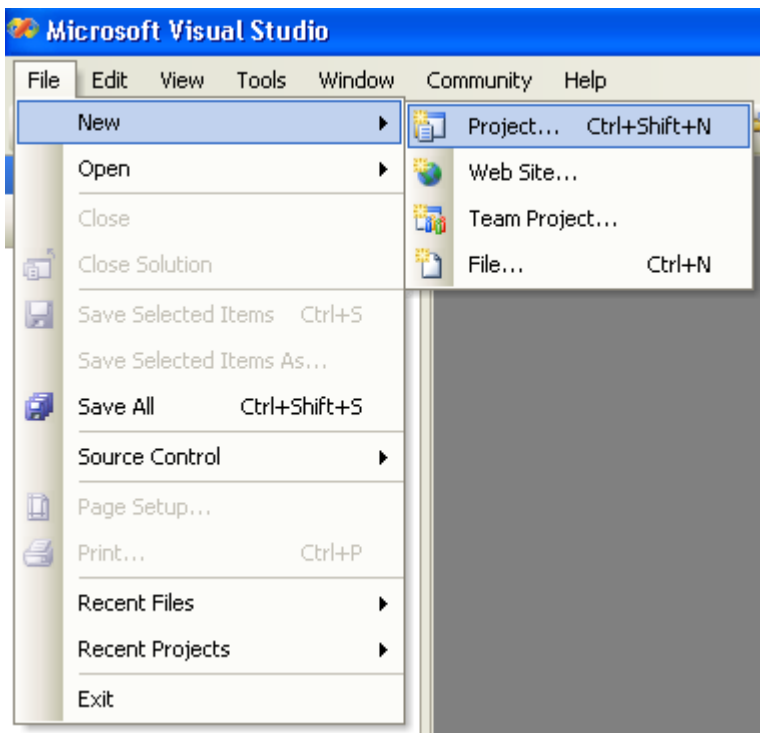
Developing a C/C++ application (unmanaged code)

SDK

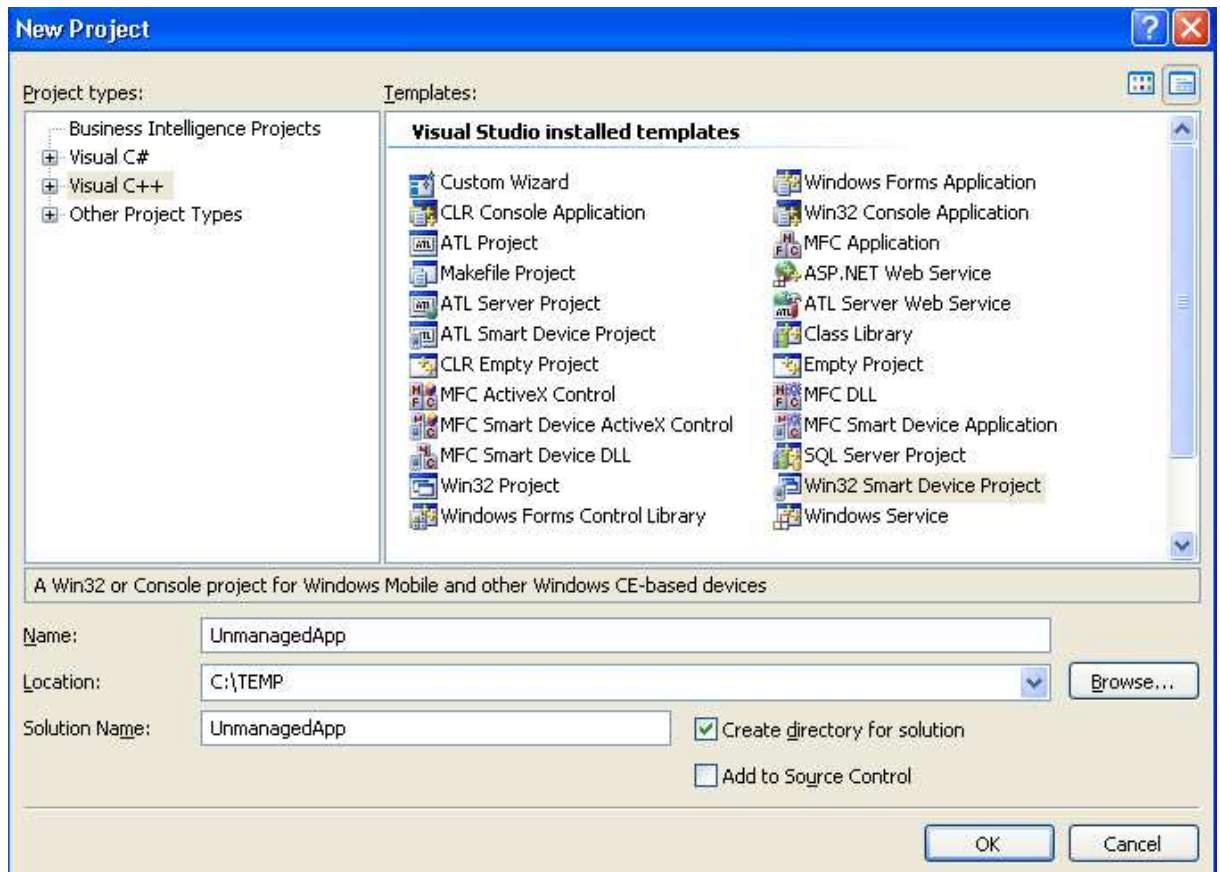
To be able to write unmanaged code for the CC PILOT™ XS you first need to install the software development kit **CCPXS_SDK.msi**.

Start a new project

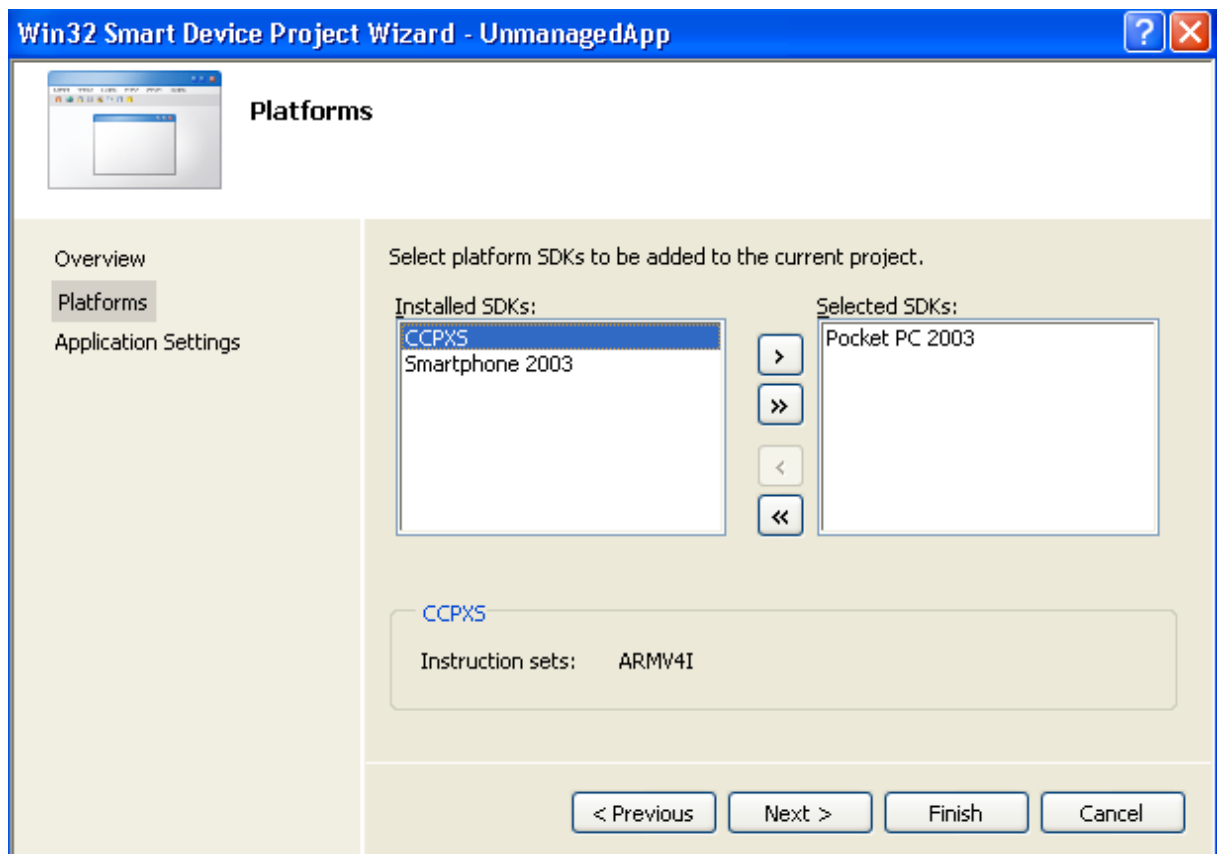
File -> new -> Project



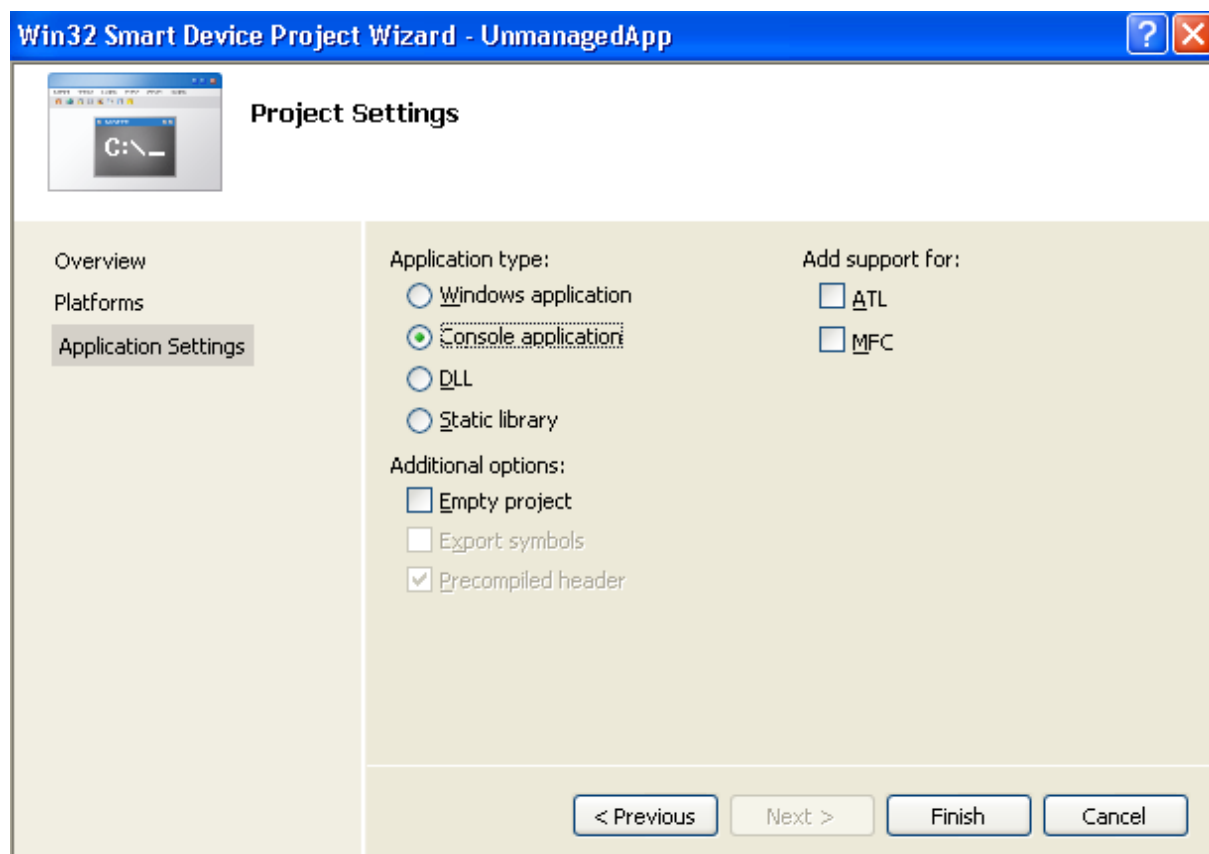
In the New Project dialog choose **Visual C++** as the project type and Win32 Smart Device Project as Template.



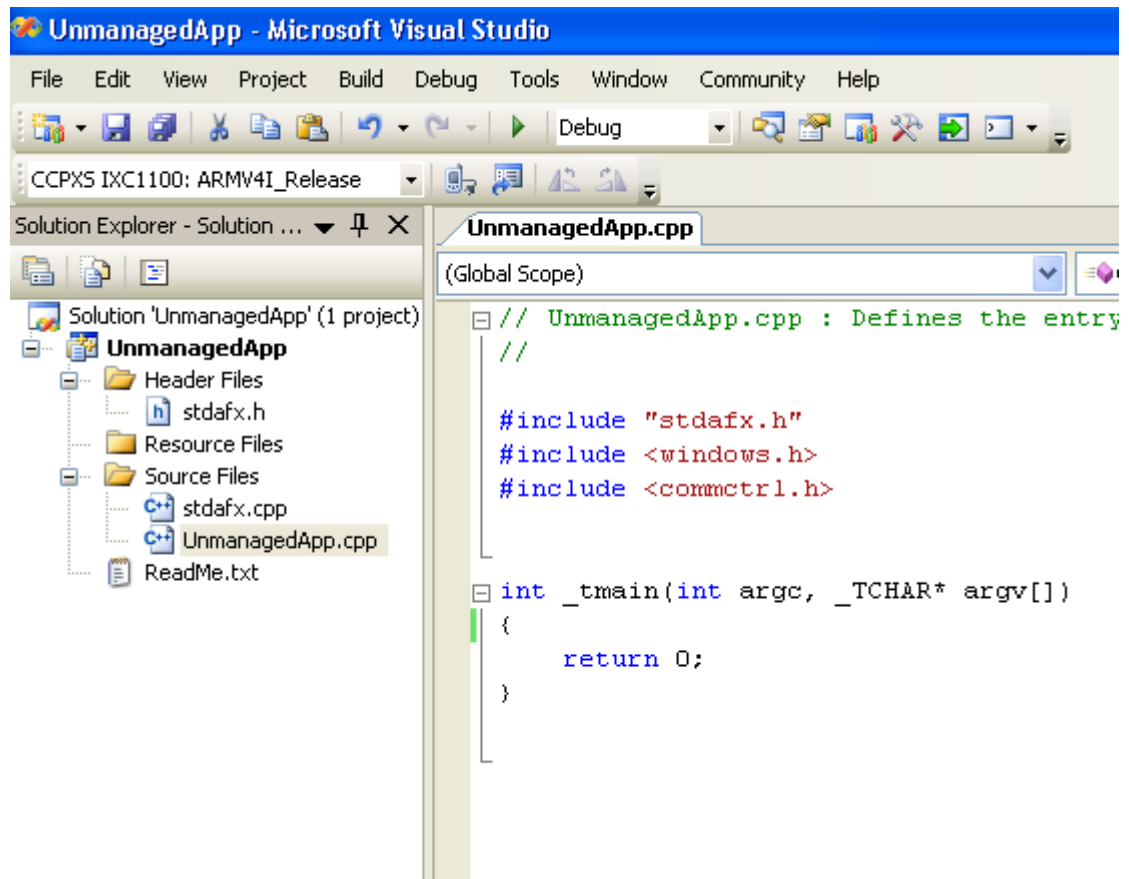
In the **Win32 Smart Device Project Wizard** choose **Platforms** and add the CCPXS SDK to the project.



Under **Application Settings** choose application type and hit **Finish**.



If you choose a console application the following will be created:



Connect to Device

To be able to connect to the CC PILOT™ XS from Visual Studio via Ethernet there are a few steps needed to be taken as can be read here:

[http://msdn2.microsoft.com/en-us/library/ms228708\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms228708(VS.80).aspx)

In short what has to be done is:

Copy files

Copy the following files from Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i to the windows folder in the CC PILOT™ XS:

Clientsshutdown.exe

CMAccept.exe

ConmanClient2.exe

DeviceDMA.dll

eDbgTL.dll

TcpConnectionA.dll

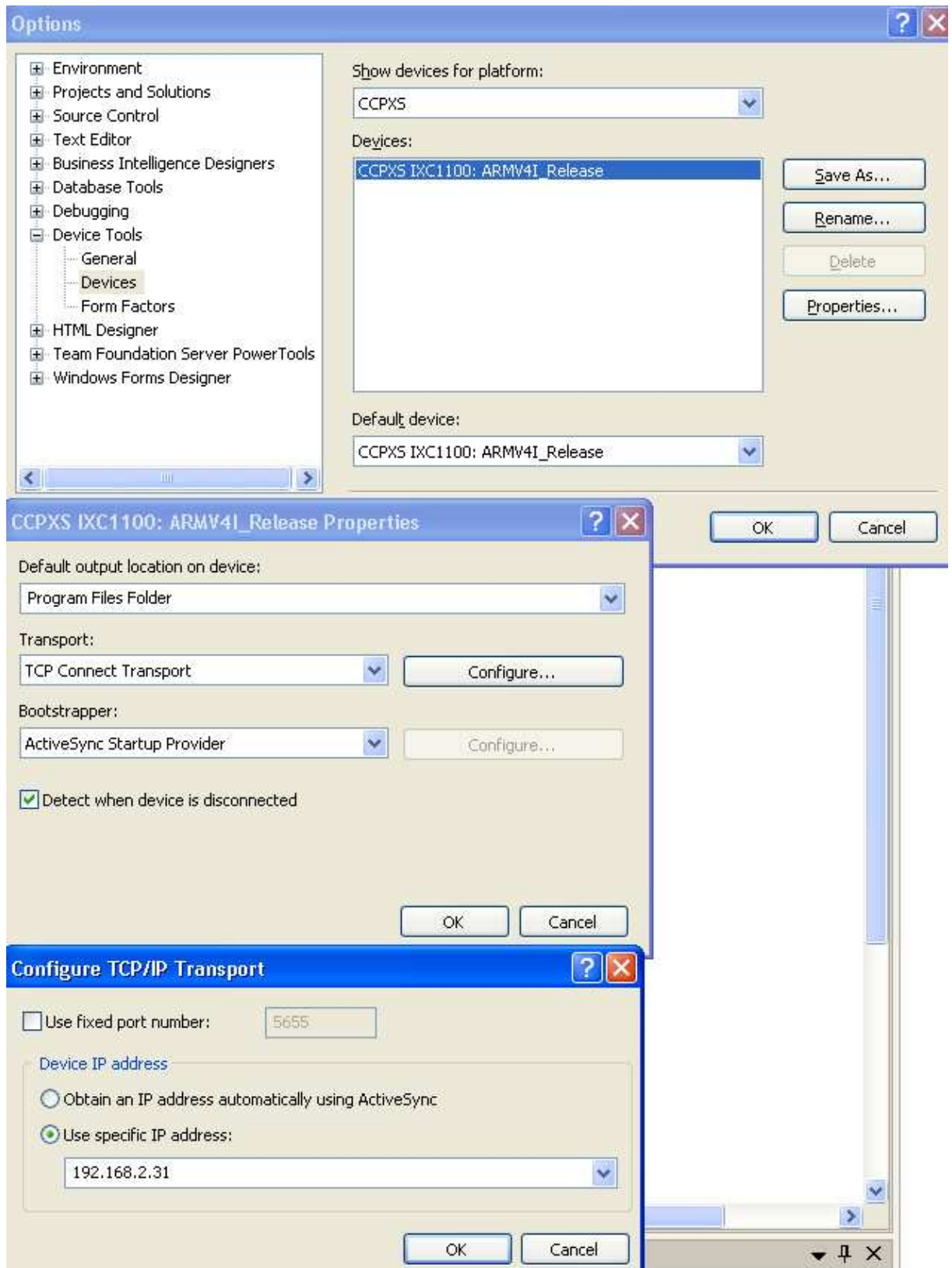
Note: If you have a Swedish installation of Windows the files are located under C:\Program\Delade filer\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i.

Set Target IP-address

In Visual Studio set the IP-address of the target device.

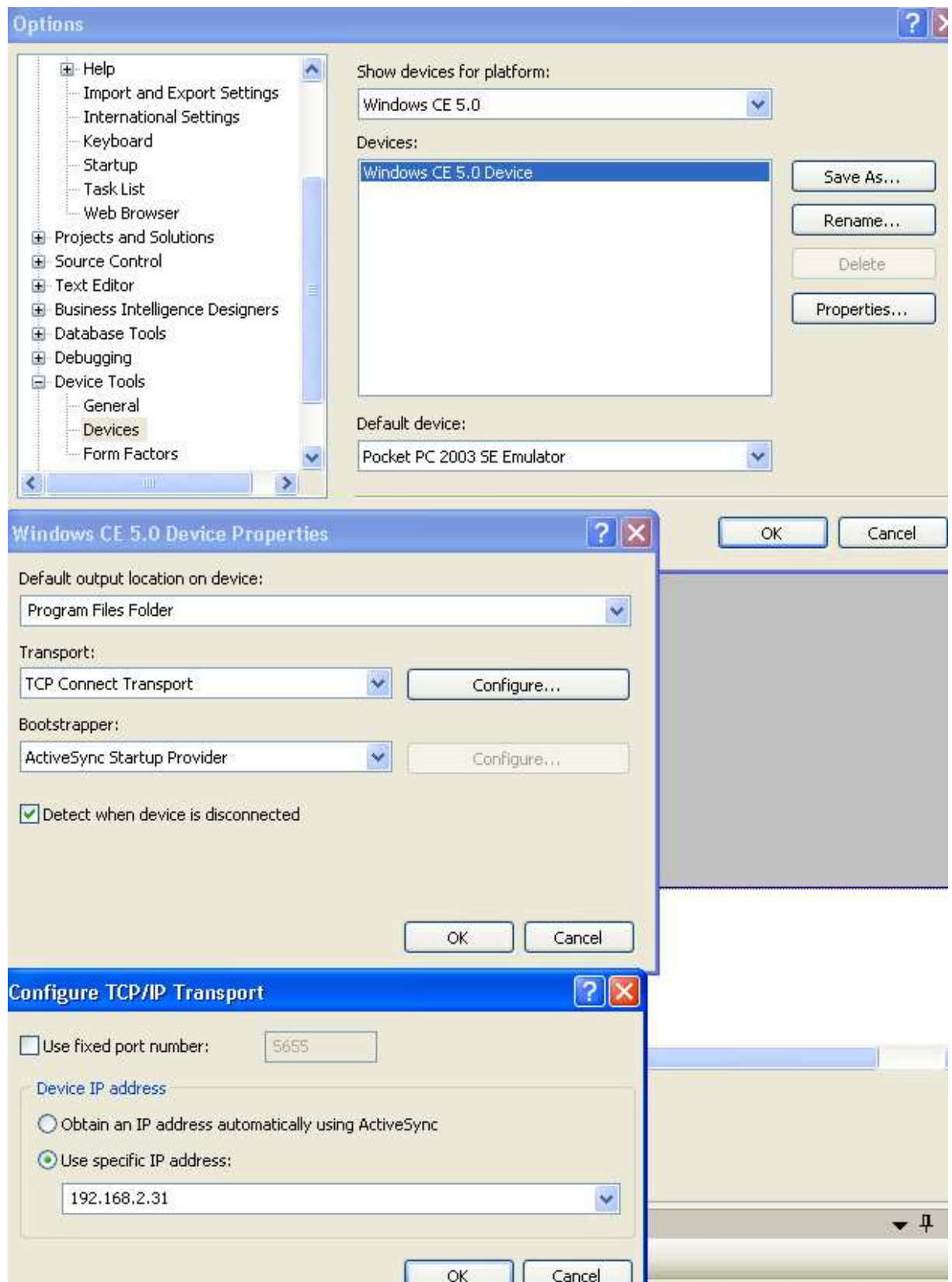
For a C/C++ Project:

Tools -> Options -> Device Tools -> Devices choose **CCPXS** under **Show Devices for platform** choose **CCPXS IXC100: ARMV4I_release** and hit **Properties**. For **TCP Connect Transport** hit **Configure** and set the IP-address of your CC PILOT™ XS.



For a C# Project:

In this case you will have to set the properties for a general Windows CE 5.0 Device, otherwise the procedure is the same.



Run file on the CC PILOT™ XS

Choose start -> Run on the CC PILOT™ XS and start conmanclient2.exe and cmaccept.exe in that order. You start by typing the filenames in the Run dialog and hit **OK**.



Within 3 minutes either connect to device, deploy or debug the application, the connection will be handled automatic from here.

Deployment and Debugging

Deployment

Manual copy

Perhaps the easiest way to deploy the application is to simply copy the executable file to the CC PILOT™ XS.

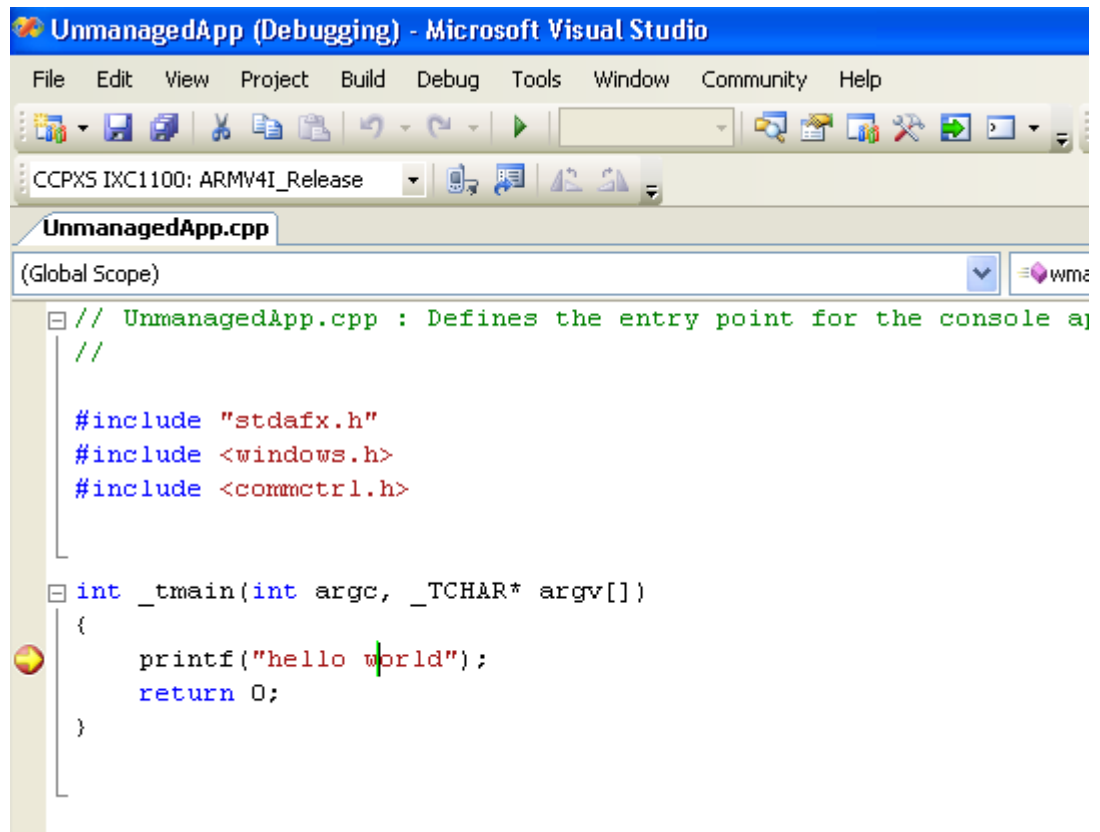
Deploy from Visual Studio

An alternative way to deploy would be using Visual Studio.

When you have started **conmanclient2.exe** and **cmaccept.exe** on the CC PILOT™ XS, choosing **Build -> Deploy Solution** will copy the executable and all dependent files to **Program Files** on the CC PILOT™ XS, this location can be changed in the project properties under **Devices -> Output file folder**. If you think deploying a C# application is slow you might want to uncheck **Deploy the latest version of the .NET Compact Framework** under project **properties -> Devices**.

Debug using visual studio

To start debugging first run **conmanclient2.exe** and **cmaccept.exe** as described under **Deploy from Visual Studio**. Then choose **Debug -> Start Debugging** or hit **F5**. Now you can set breakpoints and so on as for example in the following figure.



Executing this line by pressing F10 will make a console window to pop-up in the CC PILOT™ XS:

