



**OpenNETCF Consulting, LLC**

**[www.OpenNETCF.com](http://www.OpenNETCF.com)**

**Microsoft Visual Studio 2005 Debugging for  
Windows CE 5.0 and 6.0 Devices without ActiveSync**

**Last Revised: March 21, 2007**

## Introduction

If you've tried to debug a native or managed application with Microsoft Visual Studio 2005 on any device other than a Windows Mobile device, you're likely very aware of how painful it can be to just get connected. Typically it's a process requiring the proper alignment of planets, holding your tongue just right and often a visit to Stonehenge.

We hope that this short white paper along with CeDbgSetup.exe will take a lot of the pain out of the process.

## Requirements and Assumptions

Before proceeding, make sure you have the following:

- Visual Studio 2005, Standard Edition or better with Service Pack 1 (SP1) installed. If you don't have SP1, CeDbgSetup.exe will deploy the wrong CoreCon files for your environment and Studio will only laugh at you with a very unhelpful "Cannot Connect" message.
- CeDbgSetup.exe. This is a free, shared source tool that we've written to do a lot of the grunt work on the device for you. As usual it's provided as-is and without warranty, but if you have a problem let us know.

As of this writing CeDbgSetup.exe supports only ARMv4I and x86 devices. Sorry to you MIPS and SHx developers, but I don't have a device with one of those processors, so I can't build or test the tool for those architectures. If you have an SDK for one of them, let us know and we may be able to update the tool.

- As of this writing we've only verified that this works with Studio running on Windows XP.
- You probably should shut off or put an exception in firewalls and other security stuff that might interfere with the transport
- We've tested with the PC and the device running 802.11 connections, so yes, this work wireless on either or both sides
- You don't need ActiveSync. Need we say more?

## Configure the device

1. First make sure the device has a valid IP address. Record the IP for future use.  
Using a fixed IP, especially on networks where the DHCP lease is short and addresses change frequently. If the device IP address changes, this entire process must be done again.
2. Deploy and Run **CeDbgSetup.exe** on the device. This application is provided for free from OpenNETCF Consulting. If your device has a mechanism to persist the file and auto-launch it on boot, then we recommend that you do so to make things simple.  
When successfully run, shortcuts will be added to the device desktop. In normal operation you should not actually need to use these shortcuts.

## Configure the PC

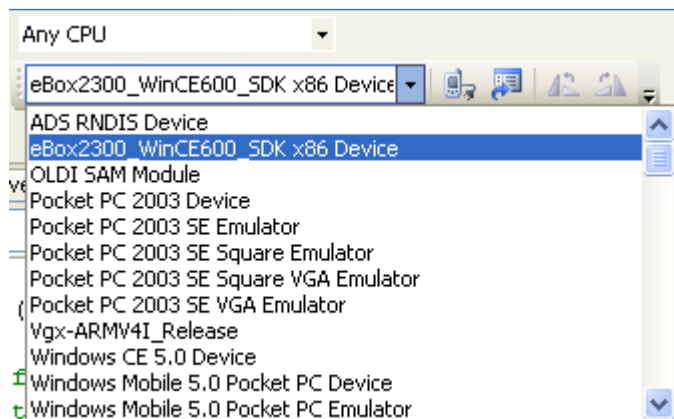
1. Verify you can ping the device to ensure that there is a solid communication channel.

## Configure Visual Studio

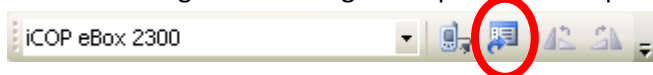
1. Enable the Device toolbar by right-clicking on Studio's toolbar and checking 'Device'

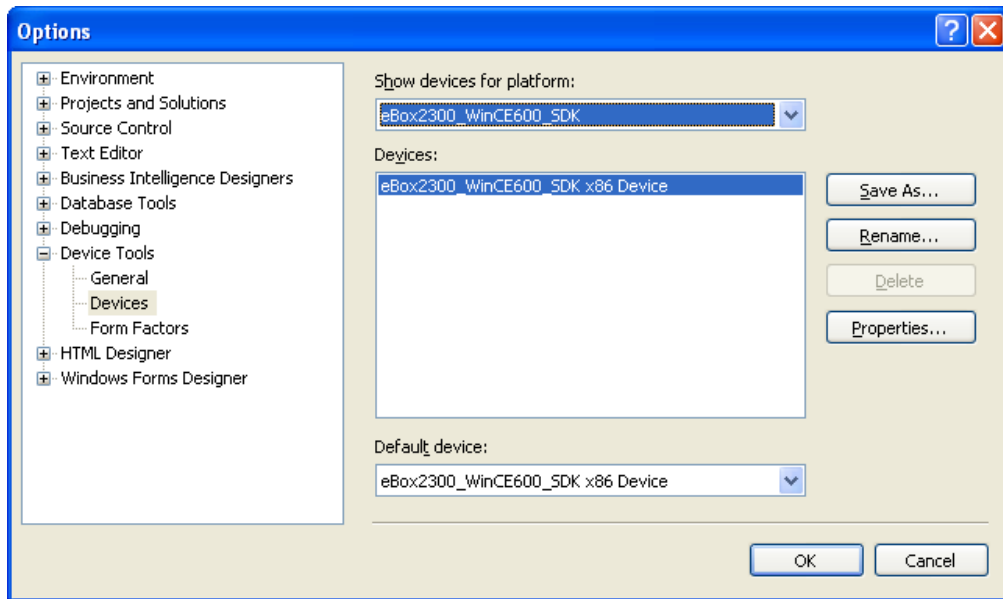


2. Select your target device. In this example we'll choose the iCOP eBox 2300, which is running CE 6.0. You'll see that I've got several other devices to choose from.

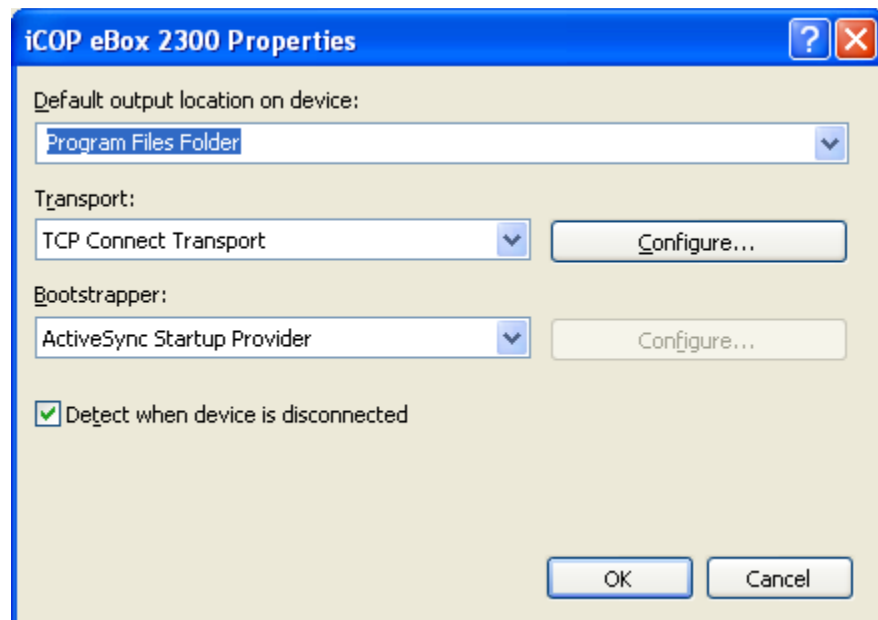


3. Open the Devices option window by either clicking the 'Device Options' button on the Device toolbar or using the following menu path Tools->Options->Device Tools->Options.

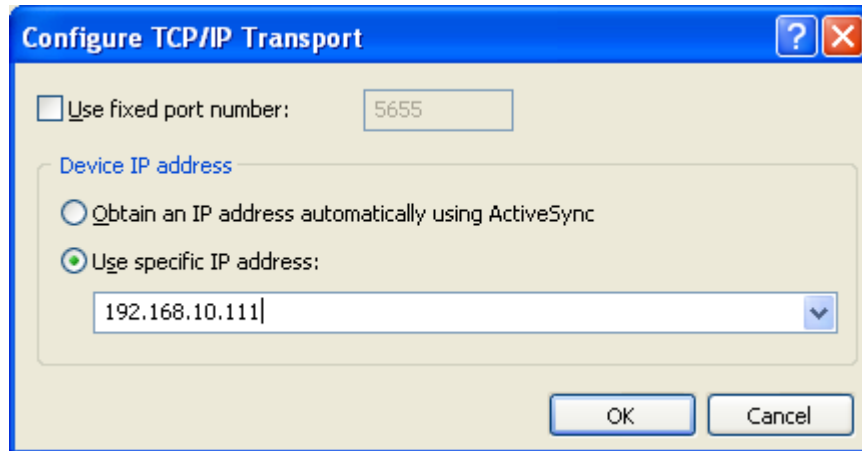




4. Select your platform from the “Show devices for Platform” dropdown.
5. In the “Devices” list, select your target device. At this point it may be useful to rename the device. You can also “clone” any device by clicking “Save As” which can be helpful if you want to have different transports or IP addresses for the same type of target device.
6. Select your device in the “Devices” list and click “Properties”



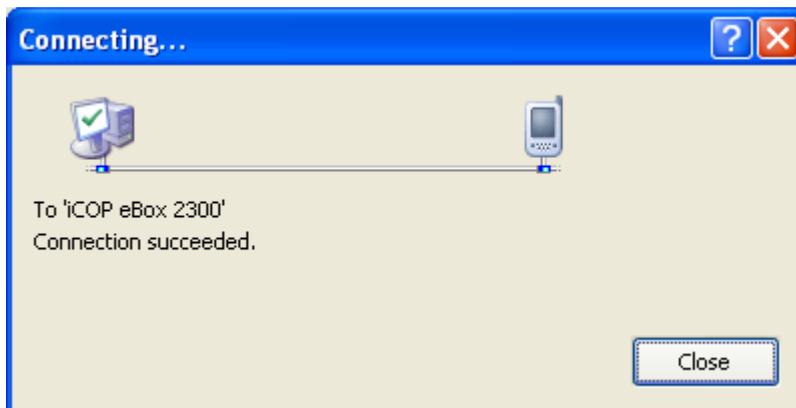
7. Click on the “Configure” button



8. Select the “Use specific IP address” radio button and enter device’s IP address you recorded way back at the start of this process.
9. Click the “OK” button on all 3 dialogs to bring you back to Studio.
10. Click the “Connect to Device” button on the Device toolbar



You should see the following:



At this point Visual Studio can communicate with the device and you’re ready to develop.