



Advanced Virtual CEPC

Windows Embedded Compact 7 Technical Article

Writers: Mark McLemore, Ralph Brand

Technical Reviewer: Ravi Gandikota

Published: June 2011

Applies To: Windows Embedded Compact 7

Abstract

When developing applications for Windows Embedded Compact you can use a virtual CEPC to emulate a software development board (SDB). A vCEPC has many advantages over using hardware for testing applications:

- Eliminates hardware expense and reduces configuration overhead.
- Easier and quicker configuration changes.
- You can keep multiple configurations on hand.

This article explains advanced vCEPC tasks:

- Modifying configuration settings.
- Using a vCEPC without a network.
- Booting from a virtual floppy disk.
- Providing an SDK and stand-alone vCEPC for application developers.

We recommend reading [Getting Started with Virtual CEPC](#)

(<http://go.microsoft.com/fwlink/?LinkID=190470>) before reading this article.

Contents

Introduction	3
Modifying Configuration Settings	3
Showing the Current Settings	5
Changing the Boot Source	6
Changing the KITL Device.....	6
Changing Network Settings.....	7
Changing Display Settings	8
Changing the Debug Port.....	9
Using Virtual CEPC Without a Network.....	10
Booting from Virtual Floppy Disk.....	19
Creating a Stand-Alone Virtual CEPC.....	20
Creating a Stand-Alone OS Design	21
Creating a Stand-Alone Virtual Hard Disk.....	24
Creating a Stand-Alone VHD with DiskPrep	24
Creating a Stand-Alone VHD Manually.....	25
Preparing Your vCEPC for Stand-Alone Use	30
Copying Files to Your VHD	30
Linking Your VHD to Your vCEPC.....	33
Creating an SDK for Virtual CEPC.....	36
Generating the SDK Package	37
Verifying Your SDK	38
Conclusion	46
Additional Resources	46

Introduction

Virtual CEPC uses Virtual PC to emulate a Windows Embedded Compact x86-based hardware platform in a virtual machine environment on your development computer. As described in the introductory article, [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkID=190470) (<http://go.microsoft.com/fwlink/?LinkID=190470>), Virtual CEPC normally boots from a virtual hard disk (VHD) image that contains a boot loader, and this boot loader uses a network connection to download your OS image from Platform Builder. When you use this boot loader, you can enter a configuration menu that lets you change display settings, select a different debug port, change network settings, or select a different boot source. You can save these settings so that your vCEPC uses them again on subsequent reboots. The VHD image provided with Windows Embedded Compact does not include an OS image, so you have to download your OS image from Platform Builder or create a stand-alone VHD if you use this boot loader.

You can also use Virtual CEPC without a network. The default configuration assumes that your host computer is attached to a network with a Dynamic Host Configuration Protocol (DHCP) server and that you are using Platform Builder for your development work. However, you can configure Virtual CEPC to work without a network connection and without a DHCP server.

Virtual CEPC provides many different boot options. You can boot from a VHD with a boot loader and download your OS image from Platform Builder, or you can create a stand-alone vCEPC by including a boot loader and your OS image in a VHD. If you don't want a boot loader in your VHD, you can create a virtual floppy disk with a boot loader instead. When you create a stand-alone vCEPC, you can also generate an SDK package for your OS design and use this SDK with your stand-alone vCEPC as an application development environment that doesn't require the presence of Platform Builder or Windows Embedded Compact.

This article explains all of these options and provides you with detailed steps for customizing your vCEPC so that it best serves your needs.

Modifying Configuration Settings

When you start your vCEPC by using the VHD provided with Windows Embedded Compact 7, the boot loader displays messages about the boot loader version and the amount of memory available, which is then followed by a series of prompts for entering the configuration menu, as shown here:

```
Hit space to enter configuration menu 5...
Hit space to enter configuration menu 4...
Hit space to enter configuration menu 3...
Hit space to enter configuration menu 2...
Hit space to enter configuration menu 1...
```

If no boot loader messages or configuration menu prompts are displayed on the Virtual PC console, your VHD image may have been configured without a boot loader. To remedy this problem, you can configure your vCEPC with the VHD image included with Windows Embedded Compact as described in [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkID=190470) (<http://go.microsoft.com/fwlink/?LinkID=190470>). If you are using Virtual PC 2007, see “Booting from Virtual Floppy Disk” later in this article for an alternate booting method.

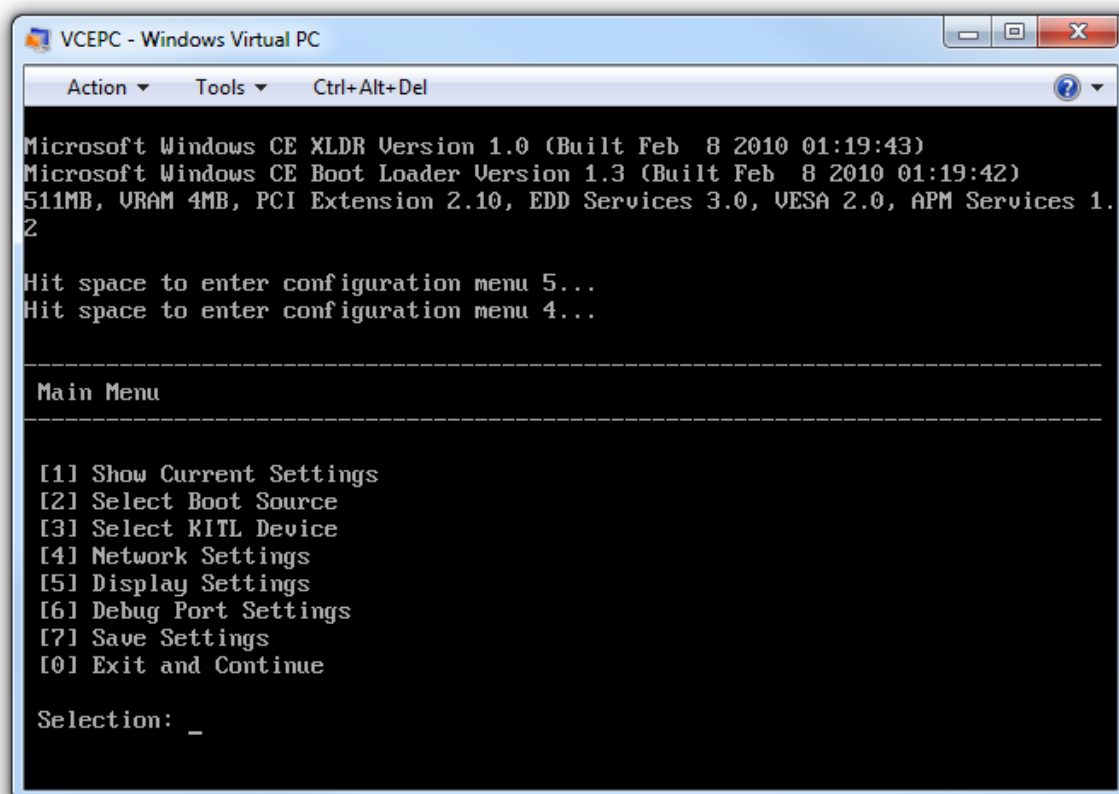
► **To enter the configuration menu**

1. When you see the configuration menu prompts appear on the Virtual PC console, press the spacebar to open the configuration menu

 **Note**

You have approximately five seconds to enter the configuration menu before the boot loader continues to the next phase of the boot process.

Figure 1 – Configuration Main Menu



2. The boot loader displays the Main Menu of configuration options:

Main Menu

```
-----
[1] Show Current Settings
[2] Select Boot Source
[3] Select KITL Device
[4] Network Settings
[5] Display Settings
[6] Debug Port Settings
[7] Save Settings
[0] Exit and Continue
```

Depending on which setting you want to change, type the number in the **Main Menu** that is associated with the submenu for the setting you want to modify. For example, you type **5** to enter the **Display Settings** submenu if you want to change the display resolution

3. To save changes that you made in one or more submenus, type **7** in the **Main Menu**.
These changes are saved across vCEPC reboots. For example, when you change the display resolution by using the **Display Settings** submenu and then type **7** to save the settings, your vCEPC will continue to use your new display resolution setting whenever you reboot it.
4. Type **0** to exit the **Main Menu** and continue the booting process.

The next sections describe the **Main Menu** options in more detail.

Showing the Current Settings

The Show Current Settings submenu displays the boot source device, the Kernel Independent Transport Layer (KITL) device, the KITL configuration, the display resolution, the name of the debug port, and whether DHCP is enabled.

To show the current settings

- Type **1** in the **Main Menu** to view the current settings.

Here is an example of the **Show Current Settings** display:

```
Main:
Boot source:    DEC21140 at PCI bus 0 dev 10 fnc 0
KITL device:    DEC21140 at PCI bus 0 dev 10 fnc 0
KITL config:    enabled, interrupt mode, VMINI enabled
DHCP:           enabled
Display:        640 x 480 x 32 / Full Screen
Debug Port:     COM1 38400 baud
```

After displaying the current settings, the boot loader returns you to the **Main Menu**.

Changing the Boot Source

You use the **Select Boot Source** submenu to configure the boot loader for loading the OS from over the network or from the boot drive. By default, the boot loader attempts to contact Platform Builder to load your OS image over a virtual network device. For more about this process, see the introductory article, [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkID=190470) (<http://go.microsoft.com/fwlink/?LinkID=190470>).

If you are using a custom VHD that contains an OS image, you can change the boot source to your VHD instead of the network.

▶ To change the boot source

1. Type **2** in the **Main Menu** to enter the **Select Boot Source** submenu.
2. In the **Select Boot Source** submenu, type the number associated with the boot source that you want to use. For example, type **1** to select **Boot Drive**. Choosing **1** selects the boot drive as the boot source and exits the **Boot Drive** submenu.
3. Type **1** in the **Main Menu** to show the current settings. Verify that the boot source is what you chose in step 2.
4. If you want to continue to use this boot source on each subsequent vCEPC reboot, type **7** in the **Main Menu** to save the settings.
5. In the **Main Menu**, you can type **0** to exit and continue the boot loading process, or you can select another submenu to change another setting.

Changing the KITL Device

KITL is the communication link between your host computer and the vCEPC that you use for debugging your OS design. Because Virtual CEPC defaults to only one virtual network device (which is the one that KITL uses by default), you typically don't change this setting. However, if you have a custom vCEPC configuration with multiple virtual network devices, you can use the **Select KITL Device** submenu to select which network device to use for KITL.

▶ To change the KITL device

1. Type **3** in the **Main Menu** to enter the **Select KITL Device** submenu.
2. In the **Select KITL Device** submenu, type the number associated with the desired KITL device.
3. Type **1** in the **Main Menu** to show the current settings. Verify that the KITL device is what you chose in step 2.
4. If you want to continue to use this KITL device on each subsequent vCEPC reboot, type **7** in the **Main Menu** to save the settings.
5. In the **Main Menu**, you can type **0** to exit and continue the boot loading process, or you can select another submenu to change another setting.

For more about KITL, see [Kernel Independent Transport Layer](http://go.microsoft.com/fwlink/?LinkID=166260) (<http://go.microsoft.com/fwlink/?LinkID=166260>) in the Windows Embedded Compact documentation.

Changing Network Settings

You use the **Network Settings** submenu to assign a static IP address to your vCEPC, enable or disable DHCP, enable or disable VMINI (the VMini Virtual Adapter), or configure KITL.

By default, the boot loader attempts to contact a DHCP server on the local network to acquire an IP address for your vCEPC to use. However, you can disable DHCP and configure your vCEPC to use a static IP address instead.

To disable DHCP and configure a static IP address

1. Type **4** in the **Main Menu** to enter the **Network Settings** submenu.
2. In the **Network Settings** submenu, type **1** to view the current network settings. For example:

```
Network:
  KITL state:      enabled
  KITL mode:       interrupt
  DHCP:           enabled
  IP address:      0.0.0.0
  VMINI:          enabled
```

3. If the current network settings display shows that DHCP is enabled, type **6** in the **Network Settings** submenu to disable DHCP.
4. When you see the prompt **Disable DHCP (actually enabled) [y/-]**, type **y**.
5. In the **Network Settings** submenu, type **5** to set the IP address.
6. When you see the prompt Enter Device IP address, enter the static IP address that you have reserved for your vCEPC. For example,

```
Enter Device IP address (actual 0.0.0.0): 192.168.10.42
```

Note

The IP address you enter must be a valid IP address for your network and this address cannot be assigned to another host on your network. If you do not know what IP address to use, contact your network administrator.

7. Type **1** in the **Network Settings** submenu to show the current settings. Verify that DHCP is disabled and that your IP address is configured correctly. For example:

```
Network:
  KITL state:      enabled
  KITL mode:       interrupt
  DHCP:           disabled
  IP address:      192.168.10.42
```

VMINI: enabled

8. Type **0** in the **Network Settings** submenu to return back to the **Main Menu**.
9. If you want to continue to use these network settings on each subsequent vCEPC reboot, type **7** in the **Main Menu** to save the settings.
10. In the **Main Menu**, you can type **0** to exit and continue the boot loading process, or you can select another submenu to change another setting.

Changing Display Settings

You use the **Display Settings** submenu to change the display resolution (width and height in pixels) and bit depth (bits per pixel) of your vCEPC screen. The default display resolution, 640 x 480, can be increased to a larger resolution depending on which bit depth you choose. The following table shows the display resolution and bit depth combinations that are supported by Virtual CEPC.

Display Resolution and Bits Per Pixel (BPP) Supported by Virtual CEPC

Display Resolution	4 BPP	8 BPP	16 BPP	32 BPP
640 x 400		x		
640 x 480		x		x
800 x 512		x	x	x
896 x 600	x	x	x	x
800 x 600		x	x	x
1024 x 768	x	x	x	x
1152 x 768		x	x	x
1280 x 1024	x	x	x	
1600 x 1024		x		
1600 x 1200		x	x	

To change the display settings

1. Type **5** in the **Main Menu** to enter the **Display Settings** submenu.
2. Type **2** in the **Display Settings** submenu to change the display resolution.
3. Select the desired bit depth (4, 8, 16, or 32 bits per pixel) by typing the corresponding letter as shown. For example:

```
-----
Select Desired Bits Per Pixel
-----
```

```
[a]    8
[b]    4
[c]   32
[d]   16
```

In this example, type **c** to select 32 bits per pixel.

4. Select the desired display resolution by typing the corresponding letter as shown. For example:

```
-----
Select Display Resolution
-----
```

```
[a]   640 x 480 x 32
[b]   800 x 600 x 32
[c]  1024 x 768 x 32
[d]   800 x 512 x 32
[e]   896 x 600 x 32
[f]  1152 x 768 x 32
```

In this example, you type **b** to select the display resolution 800 x 600 (800 pixels wide by 600 pixels high) with a bit depth of 32 bits per pixel.

5. Type **1** in the **Display Settings** submenu to show the current settings. Verify that the display resolution and bit depth are what you selected in steps 3 and 4.
6. Type **0** to exit the **Display Settings** submenu.
7. If you want to continue to use these display settings on each subsequent vCEPC reboot, type **7** in the **Main Menu** to save the settings.
8. In the **Main Menu**, you can type **0** to exit and continue the boot loading process, or you can select another submenu to change another setting.

Changing the Debug Port

You use the **Debug Port Settings** submenu to enable or disable the debug port, select a different COM port for debug, or set the baud rate. By default, Virtual CEPC enables COM1 for the debug port; however, you can select a different port from this submenu.

To change the debug port

1. Type **6** in the Main Menu to enter the **Debug Port Settings** submenu.
2. Type **2** in the **Debug Port Settings** submenu to select the COM port. A list of available COM ports to use for debugging appears. For example:

```
-----
Select Com Port
-----
```

```
[1] COM1
[2] COM2
[3] COM3
[4] COM4
```

```
Selection (actual COM1):
```

3. Enter the number corresponding to the COM port that you want to use. In the above example, type **3** to select COM3. Selecting a COM port returns you to the **Debug Port Settings** submenu.
4. Type **0** in the **Debug Port Settings** menu to return to the **Main Menu**.
5. Type **1** in the **Main Menu** to show the current settings. Verify that the debug port is what you chose in step 3. For example, if you chose COM3, your current settings display should show COM3 as the debug port:

```
Main:
```

```
Boot source:    DEC21140 at PCI bus 0 dev 10 fnc 0
KITL device:    DEC21140 at PCI bus 0 dev 10 fnc 0
KITL config:    enabled, interrupt mode, VMINI enabled
DHCP:           enabled
Display:        640 x 480 x 32 / Full Screen
Debug Port:     COM3 38400 baud
```

6. If you want to continue to use this debug port on each subsequent vCEPC boot, type **7** in the **Main Menu** to save the settings.
7. In the **Main Menu**, you can type **0** to exit and continue the boot loading process, or you can select another submenu to change another setting.

Using Virtual CEPC Without a Network

If your host computer is not attached to a network, or if your host computer is attached to a network without a DHCP server, your vCEPC will be unable to connect to Platform Builder to download your OS image. However, you can use the network loopback adapter on your host computer to connect Virtual CEPC to Platform Builder.

In brief, the steps to configure your Virtual CEPC to work without a network or without a DHCP server are:

1. Enable the loopback adapter on your host computer.

2. Derive an IP address for your vCEPC that is compatible with your loopback adapter.
3. Configure your vCEPC to use this derived IP address.

The following procedures describe this process in detail. Because the steps for configuring the loopback adapter are different for Windows 7 than for Windows Vista and Windows XP, choose the appropriate set of instructions for your host computer.

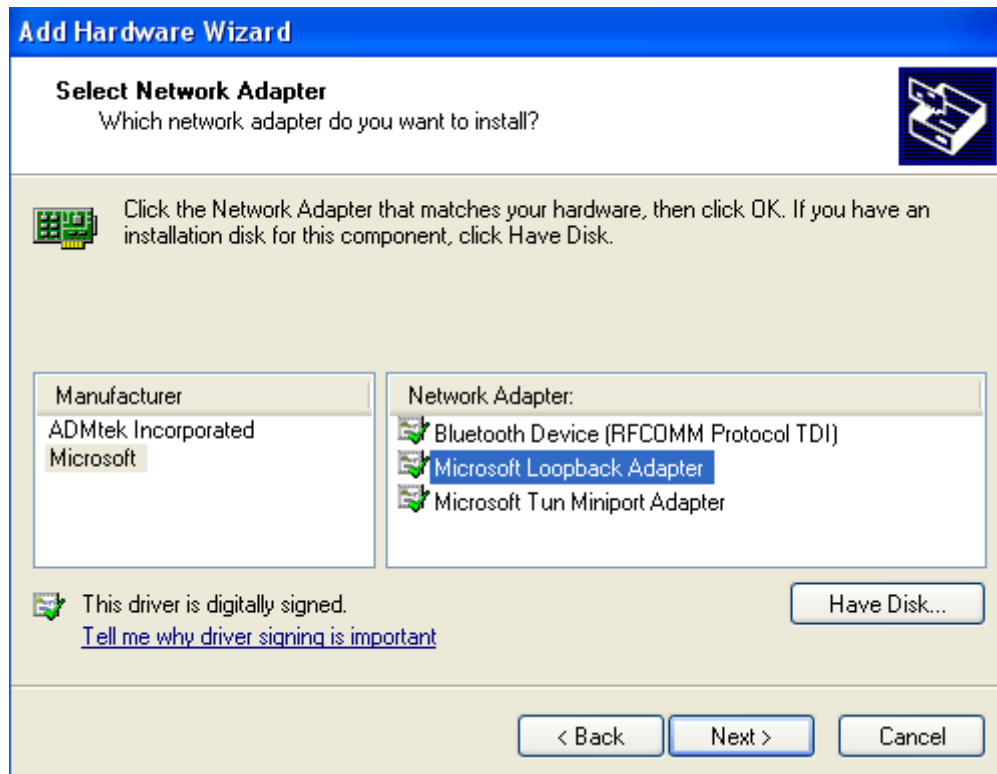
Windows Vista and Windows XP

Enable the loopback adapter on your host computer by following the instructions below. If you have an existing network adapter on your host computer, the loopback adapter appears as “Local Area Network Connection 2” when you open **Network Connections** in Control Panel.

To enable the loopback adapter

1. On your host computer, open **Control Panel**, click **Switch to Classic View**, and double-click **Add Hardware**.
2. When the **Add Hardware Wizard** displays the **Welcome to the Add Hardware Wizard** page, click **Next**.
3. When you see the page with the message **Is the hardware connected?**, select the option **Yes, I have already connected the hardware**. Click **Next**.
4. When you see the page with the message **The following hardware is already installed on your computer**, scroll down the **Installed Hardware** list until you find **Add a new hardware device**. Click **Add a new hardware device**, and then click **Next**.
5. When you see the page with the message **The wizard can help you install other hardware**, click **Install the hardware that I manually select from a list (Advanced)**, and then click **Next**.
6. When you see the page with the message **From the list below, select the type of hardware you are installing**, click **Network adapters**, and then click **Next**.
7. When you see the **Select Network Adapter** page, click **Microsoft** in the **Manufacturer** column, click **Microsoft Loopback Adapter** in the **Network Adapter** column (as shown in the following figure), and then click **Next**.

Figure 2 - Selecting the Loopback Adapter in Windows XP



8. When you see the **The wizard is ready to install your hardware** page, click **Next**.
9. In the **Completing the Add Hardware Wizard** page, click **Finish**.

At this point, you have added a network loopback adapter that you will use to provide connectivity between Platform Builder and your vCEPC. In the next set of steps, you determine an IP address to use for your vCEPC so that it can connect to this loopback adapter.

When you enable the loopback adapter, your host computer uses *address autoconfiguration* to automatically assign a *link-local* IP address to the loopback adapter. This IP address, which begins with 169.254, represents a communications link that resides entirely on your host computer. Platform Builder uses this link to communicate with the vCEPC. You must assign an IP address to your vCEPC so that it communicates on the same link-local network as the loopback adapter. In the next set of steps, you derive an IP address for your vCEPC that meets this criteria. For more information about address autoconfiguration and link-local IP addresses, see [Dynamic Configuration of IPv4 Link-Local Addresses](http://go.microsoft.com/fwlink/?LinkId=215916) (<http://go.microsoft.com/fwlink/?LinkId=215916>).

► To derive an IP address for your vCEPC

1. On your host computer, click **Start**, and then click **Run**.
2. In the **Open** box, type **cmd** and click **OK**.
3. In the Command Prompt window, type the following command and then press Enter:

```
ipconfig
```

4. In the resulting output, look for the local area connection listing that has an **Autoconfiguration IP Address** setting. For example:

```
Ethernet adapter Local Area Connection 2:
Connection-specific DNS Suffix . . :
Autoconfiguration IP Address. . . : 169.254.23.189
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
```

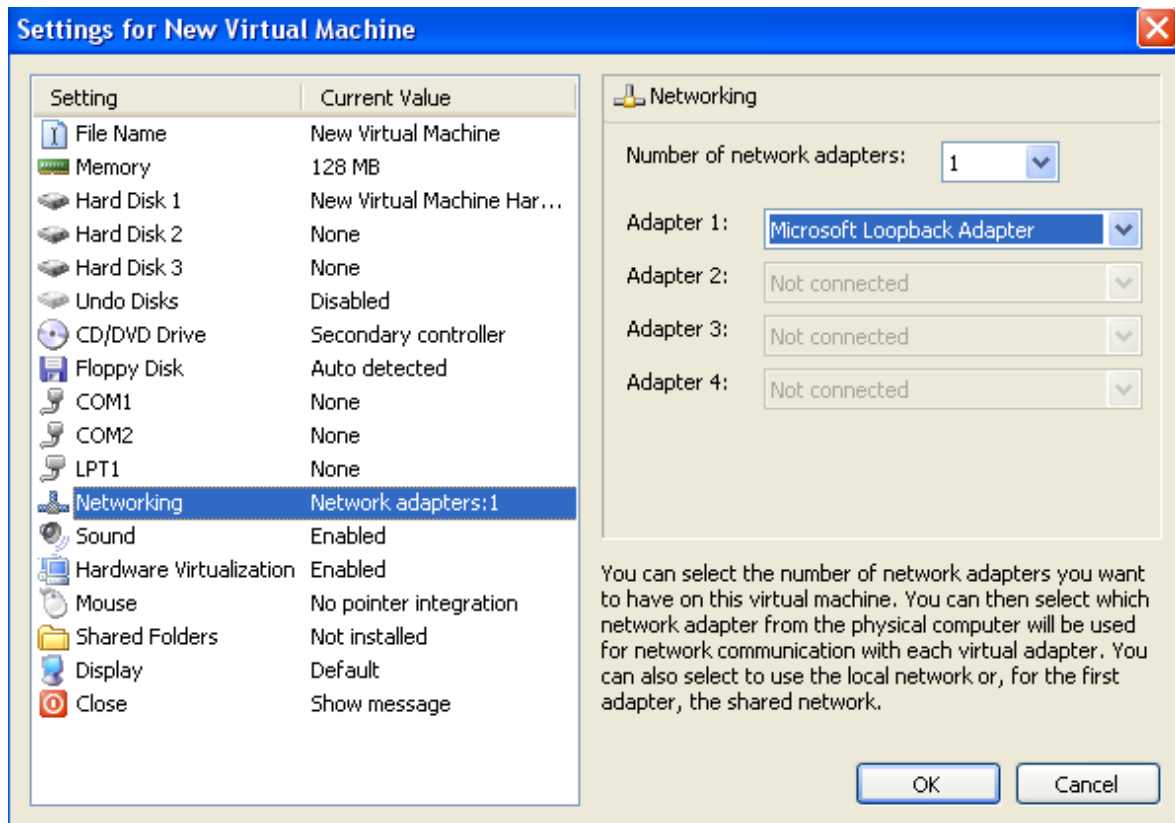
This “Ethernet adapter” is your loopback adapter, and its IP address is the address shown to the right of **Autoconfiguration IP Address**. In this example, the loopback adapter’s IP address is 169.254.23.189.

5. Derive a new IP address by taking the loopback adapter’s IP address and changing the last octet to a different number (*octets* are the numbers in the IP address that are separated by periods). For the last octet, choose a new number that is greater than 1 and less than 255. For example, if your loopback adapter’s address is 169.254.23.189, the last octet is 189. You can change this octet to 188, which gives you a derived IP address of 169.254.23.188.
6. Assign this new derived IP address to your vCEPC as described in the next set of steps.

▶ **To configure your vCEPC with the derived IP address**

1. If you have not yet created a Virtual CEPC virtual machine in Virtual PC 2007, create a virtual machine using the instructions in the introductory article [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkID=190470) (<http://go.microsoft.com/fwlink/?LinkID=190470>).
2. Start Virtual PC 2007. The **Virtual PC Console** appears.
3. On the **Virtual PC Console**, click **Settings**. The **Settings for New Virtual Machine** dialog box appears.
4. In the **Setting** column, click **Networking**.
5. Under **Networking** on the right side of the dialog box, change the **Number of network adapters** setting to 1 (if it is not already set to 1) as shown in the following figure.

Figure 3 – Virtual PC 2007 Settings



6. In the **Adapter 1** list, select **Microsoft Loopback Adapter** and click **OK**.
7. On the **Virtual PC Console**, click **Start** to start your vCEPC.
8. When your vCEPC boot loader prompts you to enter the configuration menu, press the spacebar to enter the configuration menu.
9. Type **4** in the **Main Menu** to select the **Network Settings** submenu.
10. In the **Network Settings** submenu, type **4** to disable DHCP.
11. Type **5** to select the **Set IP address** option.
12. When you see the prompt **Enter Device IP address**, enter the IP address that you derived previously. For example,

```
Enter Device IP address (actual 0.0.0.0): 169.254.23.188
```

13. Type **1** in the **Network Settings** submenu to show the current settings. Verify that DHCP is disabled and that your IP address is configured correctly. For example:

```
Network:
KITL state:      enabled
KITL mode:       interrupt
DHCP:            disabled
IP address:      169.254.23.188
```

```
VMINI:          enabled
```

14. Type **0** in the **Network Settings** submenu to return to the **Main Menu**.
15. If you want to continue to use these network settings on each subsequent vCEPC reboot, type **7** in the **Main Menu** to save the settings.

At this point, your vCEPC can connect to Platform Builder through the loopback adapter on your host computer. For more information about connecting your vCEPC to Platform Builder, see the introductory article [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkID=190470) (<http://go.microsoft.com/fwlink/?LinkID=190470>).

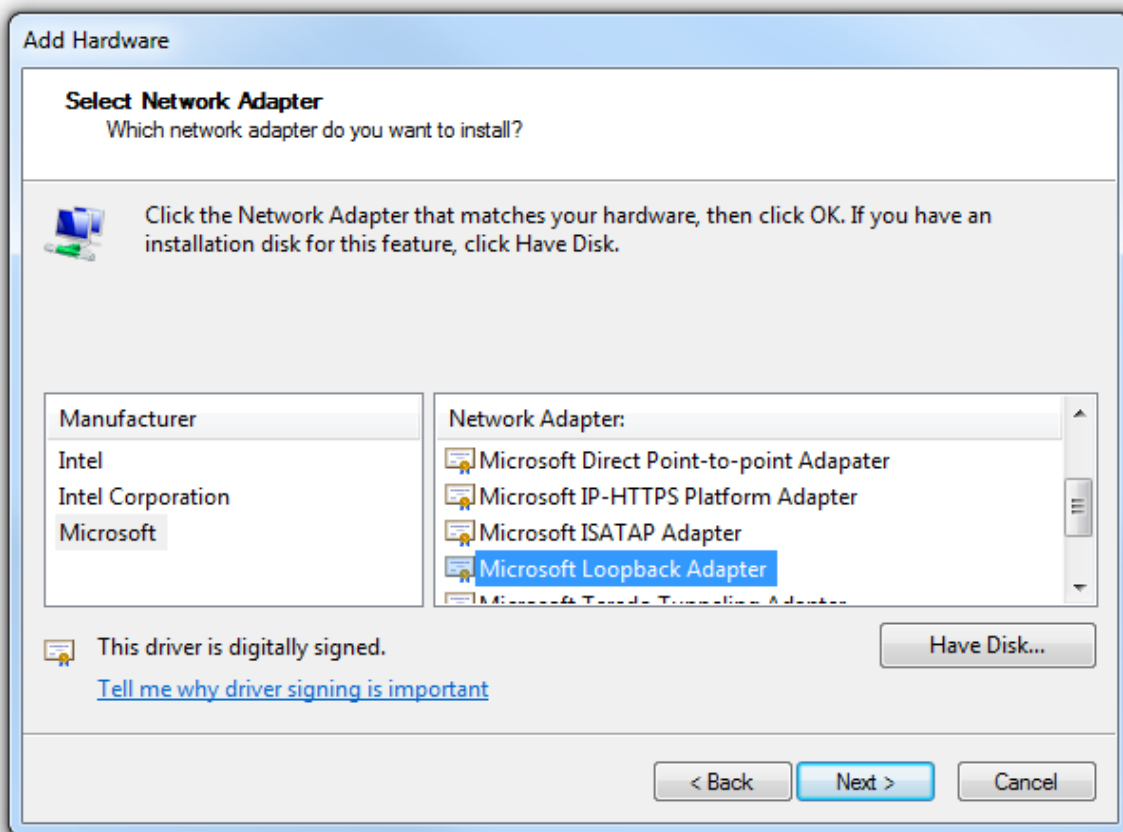
Windows 7

Enable the loopback adapter on your host computer by following these instructions. If you have an existing network adapter on your host computer, the loopback adapter appears as “Local Area Network Connection 2” in the Network and Sharing Center.

To enable the loopback adapter

1. On your host computer, click **Start**, and then click **Run**.
2. In the **Open** box, type **hdwwiz** in the text box and press **Enter**.
3. When the **Welcome to the Add Hardware Wizard** page appears, click **Next**.
4. In the **Add Hardware** page, click **Install the hardware that I manually select from a list (Advanced)**, and then click **Next**.
5. When the **Add Hardware** page presents a list of common hardware types, click **Network adapters**, and then click **Next**.
6. When the **Select Network Adapter** page appears, wait until the **Manufacturer** column displays a list of names (this may take a few seconds). Do the following:
 - a. Under **Manufacturer**, click **Microsoft**.
 - b. Under **Network Adapter**, click **Microsoft Loopback Adapter** as shown in the following figure.
 - c. Click **Next**.

Figure 4 – Selecting the Loopback Adapter in Windows 7



7. When the **The wizard is ready to install your hardware** page appears, click **Next**.
8. In the **Completing the Add Hardware Wizard** page, click **Finish**.

At this point, you have added a network loopback adapter that you will use to provide connectivity between Platform Builder and your vCEPC. In the next set of steps, you determine an IP address to use for your vCEPC so that it can connect to this loopback adapter.

When you enable the loopback adapter, your host computer uses *address autoconfiguration* to automatically assign a *link-local* IP address to the loopback adapter. This IP address, which begins with 169.254, represents a communications link that resides entirely on your host computer. Platform Builder uses this link to communicate with the vCEPC. You must assign an IP address to your vCEPC so that it communicates on the same link-local network as the loopback adapter. In the next set of steps, you will derive an IP address for your vCEPC that meets this criteria. For more about address autoconfiguration and link-local IP addresses, see [Dynamic Configuration of IPv4 Link-Local Addresses](http://go.microsoft.com/fwlink/?LinkId=215916) (<http://go.microsoft.com/fwlink/?LinkId=215916>).

► To derive an IP address for your vCEPC

1. On your host computer, click **Start**, and then click **Run**.

2. In the **Open** box, type **cmd** and press **Enter**.
3. In the **Command Prompt** window, type the following command and then press **Enter**:

```
ipconfig
```
4. In the resulting output, look for the local area connection listing that has an **Autoconfiguration IPv4 Address** setting. For example,

```
Ethernet adapter Local Area Connection 2:
    Connection-specific DNS Suffix . . : 
    Link-local IPv6 Address . . . . . : fe80::2500:9178:934e:17bd%20
    Autoconfiguration IPv4 Address. . : 169.254.23.189
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . :
```

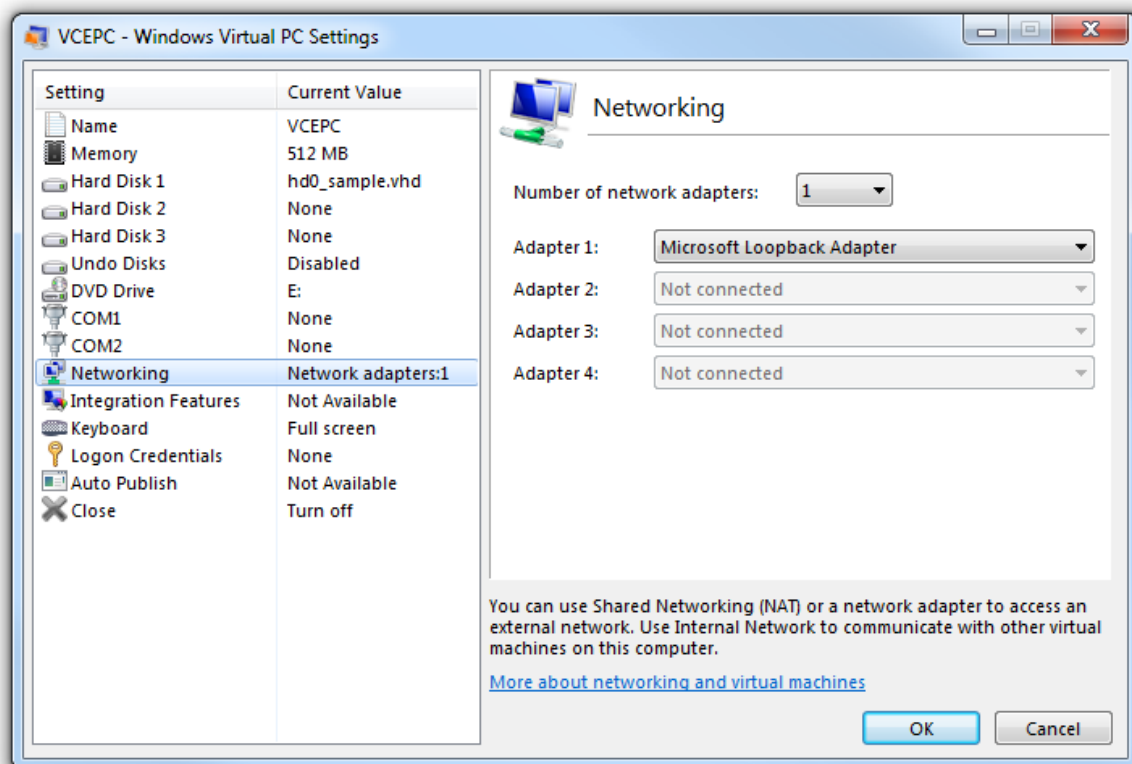
This “Ethernet adapter” is your loopback adapter, and its IP address is the address shown to the right of **Autoconfiguration IPv4 Address**. In this example, the loopback adapter’s IP address is 169.254.23.189.

5. Derive a new IP address by taking the loopback adapter’s IP address and changing the last octet to a different number (octets are the numbers in the IP address that are separated by periods). For the last octet, choose a new number that is greater than 1 and less than 255. For example, if your loopback adapter’s address is 169.254.23.189, the last octet is 189. You can change this octet to 188, which gives you a derived IP address of 169.254.23.188.
6. Assign this new derived IP address to your vCEPC as described in the next set of steps.

▶ To configure your vCEPC with the derived IP address

1. If you have not yet created a Virtual CEPC virtual machine in Windows Virtual PC, create a virtual machine by using the instructions in the introductory article [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkID=190470) (<http://go.microsoft.com/fwlink/?LinkID=190470>).
2. In your virtual machines folder, right-click your .vmcx or .vmc file (for example, VCEPC.vmcx) and click **Settings**.
3. In the **Windows Virtual PC Settings** dialog box, do the following:
 - a. Under **Setting**, click **Networking**.
 - b. Under **Networking** on the right side of the dialog box, change the **Number of network adapters** to 1 (if it is not already set to 1).
 - c. In the **Adapter 1** list, click **Microsoft Loopback Adapter** (as shown in the following figure), and then click **OK**.

Figure 5 – Windows Virtual PC Settings



4. In your virtual machines folder, double-click your .vmcx or .vmc file (for example, VCEPC.vmcx) to start your vCEPC.
5. When your vCEPC boot loader prompts you to enter the configuration menu, press the spacebar to enter the configuration menu.
6. In the **Main Menu**, type **4** to select the **Network Settings** submenu.
7. In the **Network Settings** submenu, type **4** to disable DHCP.
8. Type **5** to select the **Set IP address** option.
9. When you see the prompt **Enter Device IP address**, enter the IP address that you derived previously. For example,


```
Enter Device IP address (actual 0.0.0.0): 169.254.23.188
```
10. Type **1** in the Network Settings submenu to show the current settings. Verify that DHCP is disabled and that your IP address is configured correctly. For example:

```
Network:
KITL state:      enabled
KITL mode:       interrupt
DHCP:            disabled
IP address:      169.254.23.188
```

```
VMINI:          enabled
```

11. Type **0** in the **Network Settings** submenu to return to the **Main Menu**.
12. If you want to continue to use these network settings on each subsequent vCEPC reboot, type **7** in the **Main Menu** to save the settings.

At this point, your vCEPC can connect to Platform Builder through the loopback adapter on your host computer. For more information about connecting your vCEPC to Platform Builder, see the introductory article [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkID=190470) (<http://go.microsoft.com/fwlink/?LinkID=190470>).

Booting from Virtual Floppy Disk

If you are using a custom VHD image that does not contain a boot loader, you can instead boot the vCEPC from an image on a virtual floppy disk. Because the steps for booting from virtual floppy disk are different for Windows Virtual PC (for Windows 7) and Virtual PC 2007 (for Windows Vista and Windows XP), choose the appropriate set of instructions for your host computer.

Windows Vista and Windows XP

When you are running Virtual PC 2007, you can boot from a virtual floppy disk by using the UI.

To boot Virtual PC 2007 from virtual floppy disk

1. In the **Virtual PC Console**, click **Start** to boot the vCEPC.
2. When the vCEPC boot screen (in the **Virtual PC Console**) appears, click the **Floppy** menu, and then click **Capture Floppy Disk Image**.
3. When the **Select Floppy Image to Capture** dialog box appears and prompts for the file name of the virtual floppy disk image, use the provided sample virtual floppy disk image vpc_bootce.vfd, located at

C:\WINCE700\platform\VirtualPC\VM\vpc_bootce.vfd

This sample floppy disk image contains a boot loader that you can use to boot your vCEPC.

4. After typing the virtual floppy disk location, click **Open**.

The virtual floppy disk image is now attached to the vCEPC for use on the next boot. To reboot the vCEPC by using the virtual floppy disk image, click the **Action** menu, and then click **Ctrl+Alt+Del**.

When the vCEPC boots from the floppy disk image, it displays messages about the boot loader version and the amount of memory available, followed by a series of prompts for entering the configuration menu as previously described. You can enter the boot loader menu and select boot settings as you would when you boot from a VHD image.

Windows 7

Windows Virtual PC does not provide UI support for virtual floppy disks, but you can boot from a virtual floppy disk by using scripting.

▶ To boot Windows Virtual PC from virtual floppy disk

1. Download Floppy Drive Scripts.zip from Using Floppy Disks with Windows Virtual PC on [MSDN Blogs](http://go.microsoft.com/fwlink/?LinkId=199474&clcid=0x409) (<http://go.microsoft.com/fwlink/?LinkId=199474&clcid=0x409>).
2. Extract FloppyDrive.ps1 and FloppyDrive.vbs from the .zip file and save them in the same folder as your virtual machine. These two scripts do the same thing; you can use either of them to connect your vCEPC to a virtual floppy disk.
3. Open a Command Prompt window and navigate to the folder where you saved the files.
4. To use the Microsoft Visual Basic Scripting Edition (VBScript) file, type the following at the command prompt, and then press Enter:

```
cscript FloppyDrive.vbs /vm:"<vCEPC_Name>"/action:vfd /floppy: "<VFD
location>"
```

For the virtual floppy disk location, use the provided sample virtual floppy disk image, vpc_bootce.vfd, located at

```
C:\WINCE700\platform\VirtualPC\VM\vpc_bootce.vfd
```

This sample floppy disk image contains a boot loader that you can use to boot your vCEPC.

For example, if your vCEPC is named "VCEPC," the command would be

```
cscript FloppyDrive.vbs /vm:"VCEPC" /action:vfd
/floppy:"C:\WINCE700\platform\VirtualPC\VM\vpc_bootce.vfd"
```

When the script successfully runs, you will see a message similar to the following:

Attaching C:\WINCE700\platform\VirtualPC\vm\vpc_bootcd.vfd to the floppy drive.

The virtual floppy disk image is now attached to the vCEPC for use on the next boot.

5. Start your vCEPC.

The vCEPC will boot from the virtual floppy disk and display messages about the boot loader version and the amount of memory available, followed by a series of prompts for entering the configuration menu as previously described. You can enter the boot loader menu and select boot settings as you would when you boot from a VHD image.

Creating a Stand-Alone Virtual CEPC

The default configuration for Virtual CEPC is to boot from the provided VHD sample and download your OS image from Platform Builder. However, because most application developers are not concerned with OS design or board support package (BSP) development, they typically do not use Platform

Builder. As an application developer, you can use a customized, stand-alone vCEPC that you package with the following components:

- A pre-built OS design image that includes the necessary drivers and any additional catalog items that you need to run in stand-alone mode. This OS design image must have Kernel Independent Transport Layer (KITL) disabled.
- A software development kit (SDK) for the OS design that will be used on the vCEPC.
- A custom VHD that contains a boot loader, the OS design image, communications files, and any special files that you need to develop your application.

You create a stand-alone Virtual CEPC for application development by using the steps described in the following sections. You need a computer that is running Windows 7 to run the tools for creating a customized stand-alone vCEPC VHD, but you can deploy this VHD to application development computers running Windows XP, Windows Vista, or Windows 7.

Creating a Stand-Alone OS Design

You begin the process of creating a stand-alone vCEPC by creating an OS design that runs in stand-alone mode. Your OS design image must have a network driver enabled for communication with the outside world and it must have storage and file system drivers enabled so that it can access files in the VHD. In addition, you may need to enable other catalog items in your OS design, depending on your application development needs. Finally, KITL transport must be disabled in your OS image because your OS will not communicate with Platform Builder. This section explains how to configure these settings in your OS design.

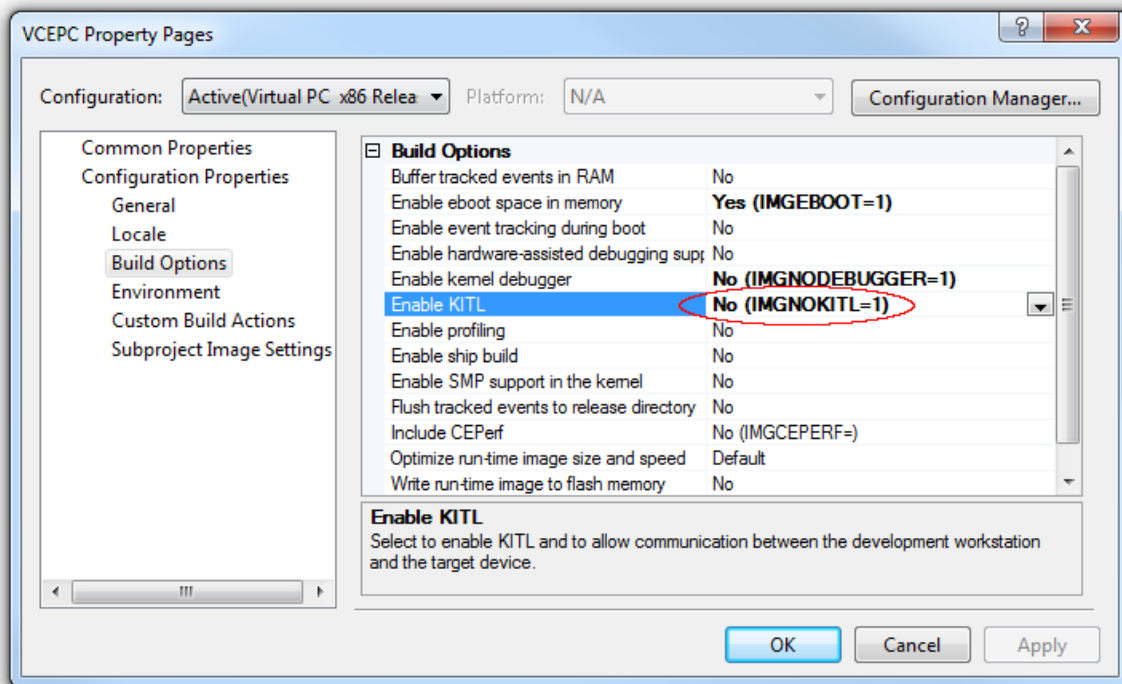
Disabling KITL

If your OS design image has KITL enabled (it is enabled by default), use the following steps to disable it.

To disable KITL in your OS design

1. Open your OS design project in Visual Studio.
2. On the **Project** menu, click **<Your Project Name> Properties**. For example, if your project name is VCEPC, click **VCEPC Properties**.
3. In the **Property Pages** dialog box, under **Build Options**, select **Enable KITL** and change its setting to **No (IMGNOKITL=1)** as shown in the following figure.

Figure 6 - Disabling KITL



4. Click **OK** to save this setting.

Enabling the Network Driver

For your stand-alone vCEPC to communicate with Visual Studio, you must enable the device driver for the virtual DEC 21x40 Ethernet device that is simulated by Virtual CEPC. Your vCEPC uses this virtual network device to communicate with the physical network that is attached to your host computer (or your host computer's loopback adapter if you are running your vCEPC "off network").

▶ To enable the networking driver

1. In Visual Studio, open your OS design project.
2. In the **Catalog Items View** pane, expand **BSP, Virtual PC : x86, Device Drivers**, and then **Networking**.
3. Select **DEC 21x40 Ethernet**.
4. On the **File** menu, click **Save All** to save your changes.

These steps enable the BSP_NIC_DC21X4 network driver in your OS design.

Enabling the File System

Your stand-alone vCEPC must be able to mount the FAT32 file system partition in your VHD so it can access files that your vCEPC needs for communicating with Visual Studio. To make this possible, your OS image must have the following catalog items enabled:

- Storage Block Driver
- Storage Manager
- Partition Driver
- FAT File System

Use the following steps to enable these catalog items.

To enable the Storage Block Driver

1. In Visual Studio, open your OS design project.
2. In the **Catalog Items View** pane, expand **Core OS, Windows Embedded Compact, Device Drivers**, and then **Storage Devices**.
3. Select **ATAPI PCI Support**.
4. Expand **ATAPI PCI Support** and select **Basic ATAPI PCI CD/DVD-ROM Support**.
5. Expand **Basic ATAPI PCI CD/DVD-ROM Support** and select **ATAPI PCI/IDE Storage Block Device Driver**.
6. On the **File** menu, click **Save All** to save your changes.

These steps add the Sysgen variable SYSGEN_ATAPI to your OS design, together with the Sysgen variables that SYSGEN_ATAPI depends upon (such as SYSGEN_ATAPI_PCIO and SYSGEN_ATAPI_PCIO_CD).

Note that this storage block driver does not access physical ATAPI or PCI devices in your host computer. Instead, it communicates with a simulated ATAPI device that Virtual CEPC uses for accessing your VHD.

To enable the Storage Manager, Partition Driver, and FAT File System

1. In Visual Studio, open your OS design project.
2. In the **Catalog Items View** pane, expand **Core OS, Windows Embedded Compact**, and then **File Systems and Data Store**.
3. Select **Storage Manager**.
4. Expand **Storage Manager** and select **FAT File System** and **Partition Driver**.
5. On the Visual Studio **File** menu, click **Save All** to save your changes.

These steps enable the Sysgen variables SYSGEN_STOREMGR, SYSGEN_FATFS, and SYSGEN_MSPART in your OS design.

Enabling Silverlight for Windows Embedded

If you are creating applications that use Silverlight for Windows Embedded, you must enable the Silverlight for Windows Embedded catalog item in your OS design.

To enable Silverlight for Windows Embedded

1. In Visual Studio, open your OS design project.
2. In the **Catalog Items View** pane, expand **Core OS, Windows Embedded Compact, Shell and User Interface**, and then **Silverlight for Windows Embedded**.
3. Select **Silverlight for Windows Embedded**.
4. On the **File** menu, click **Save All** to save your changes.

These steps enable the Sysgen variable SYSGEN_XAML_RUNTIME in your OS design.

Rebuilding Your OS Design

After you have disabled KITL and enabled the necessary catalog items, you are ready to rebuild your OS design. On the **Build** menu, click **Rebuild Solution**. When the build completes, your OS design image (NK.bin) will have KITL disabled and the necessary catalog items enabled so that it can boot in a stand-alone Virtual CEPC.

Creating a Stand-Alone Virtual Hard Disk

The sample VHD provided with Windows Embedded Compact contains only a boot loader, so you must create a custom VHD that has a boot loader and your stand-alone OS design image so that your vCEPC can boot without having to download an OS image from Platform Builder.

After you have created a stand-alone OS design image, use the following procedures on a computer running Windows 7 to create a custom VHD for use with your vCEPC. If you are using the boot loader that is included with Windows Embedded Compact 7, you can use the DiskPrep tool to create your VHD. If you want to use your own custom boot loader, then you must create your VHD image manually.

Creating a Stand-Alone VHD with DiskPrep

The easiest way to create a stand-alone VHD image is to use the DiskPrep tool to create a VHD that contains the Windows Embedded Compact 7 boot loader and your OS image. You can download DiskPrep from [Windows Embedded CE DiskPrep PowerToy](http://go.microsoft.com/fwlink/?LinkId=199594) (<http://go.microsoft.com/fwlink/?LinkId=199594>) on MSDN Code Gallery. You can use DiskPrep to create a new VHD or use an existing VHD, but you cannot use DiskPrep on a VHD that is being used by a running virtual machine. As a safety precaution, DiskPrep will not present your boot, system, or swap volumes as possible target disks. Although DiskPrep runs on Windows 7 and Windows Vista, it requires Windows 7 to create a VHD.

► **To create a stand-alone VHD on Windows 7**

1. Download the [DiskPrep](http://go.microsoft.com/fwlink/?LinkId=199594) (<http://go.microsoft.com/fwlink/?LinkId=199594>) tool from MSDN Code Gallery.
2. Double-click DiskPrep.exe.
3. When you see the initial **DiskPrep** window, in the **Disk Selection** list, select **<VHD I create or specify when I click OK>**.
4. In the **Use File System Format** list, select **FAT32**.
5. Select the **BOOT.INI** options that are appropriate for your device. You can edit the **Boot.ini** file on the disk after it is created to change these settings.
6. Click **Load specific image file copied from** and click **Browse**.
7. Navigate to the **KITL-disabled OS image (NK.bin)** that you created earlier and click **Open**.
8. In the **Create or Select Virtual Disk** dialog box, click **Create a new Virtual Hard Disk (VHD)**.
9. Type the name of the new VHD file. For example, if your vCEPC OS design folder resides at **C:\WINCE700\OSDesigns\VCEPC**, you could use the following path for your VHD:
 C:\WINCE700\OSDesigns\VCEPC\custom.vhd
10. In the **Size in Megabytes** box, enter a size of at least 255 MB for the VHD. You may need a larger size depending on the number of additional files that you intend to copy to your VHD.
11. Click **Static Size** and then click **OK**. Depending on the speed of your computer, DiskPrep may take a few seconds to begin the VHD creation process.
12. When the **Confirm Dangerous Action** dialog box appears, click **OK**.
13. When DiskPrep displays the message, **Disk Preparation Completed ok**, click **OK**.

After you have completed these steps, your VHD can be used to boot a Virtual CEPC virtual machine without Platform Builder.

Creating a Stand-Alone VHD Manually

The most general way to create a stand-alone VHD image is to use the Windows 7 Disk Management tool to create your VHD and use the cesys tool to embed a boot loader in your VHD. Use this method if you don't want to use DiskPrep or if you need to create a VHD with your own boot loader.

► **To manually create a stand-alone VHD on Windows 7**

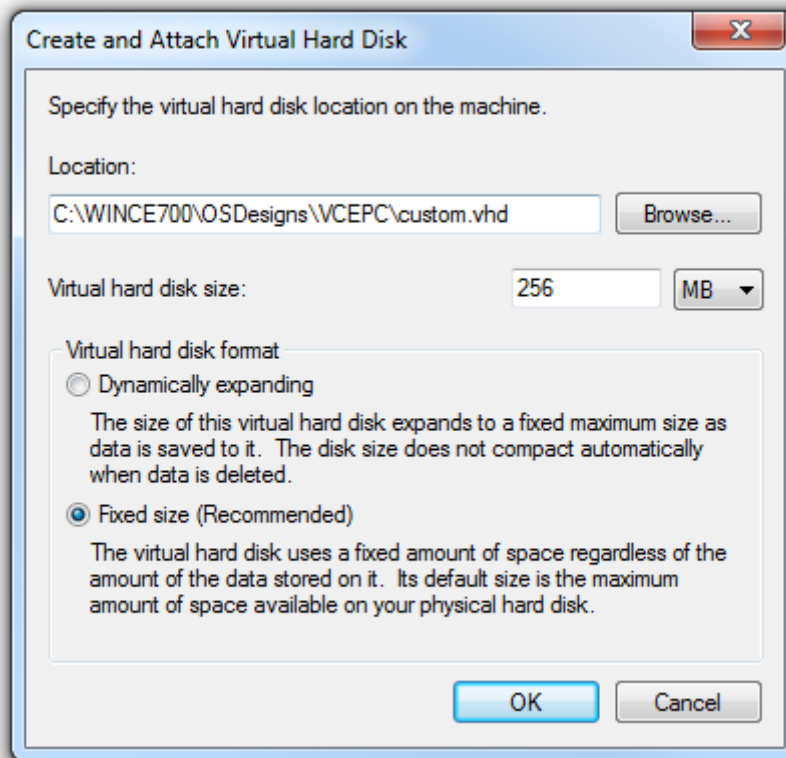
1. On your host computer, click **Start**, and then click **Run**.
2. In the **Open** box, type **diskmgmt.msc** and press **Enter**. This starts the **Disk Management** tool.
3. On the **Action** menu, click **Create VHD**.
4. In the **Create and Attach Virtual Hard Disk** dialog box, type the location on your host computer where you want the VHD file to be stored. For example, if your vCEPC OS design folder resides at **C:\WINCE700\OSDesigns\VCEPC**, you could use the following path for your

VHD:

C:\WINCE700\OSDesigns\VCEPC\custom.vhd

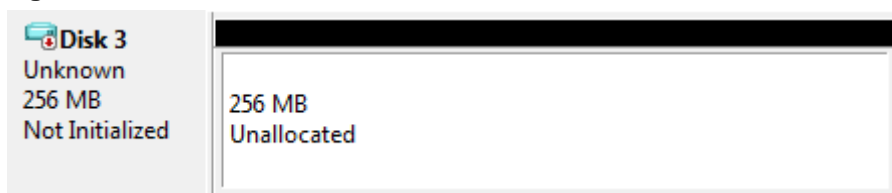
5. In the **Virtual hard disk size** box, enter a size of at least 255 MB for the VHD. You may need a larger size depending on the number of additional files that you intend to copy to your VHD.
6. Under **Virtual hard disk format**, select **Fixed size (Recommended)** as shown here, and then click **OK**.

Figure 7 - Create and Attach Virtual Hard Disk



The new VHD appears in the list of drives, marked as Not Initialized and containing an empty partition marked as **Unallocated** as shown in the following figure.

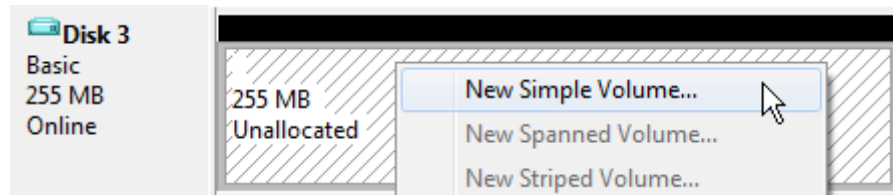
Figure 8 – New Virtual Hard Disk Partition



► **To create the volume on the VHD**

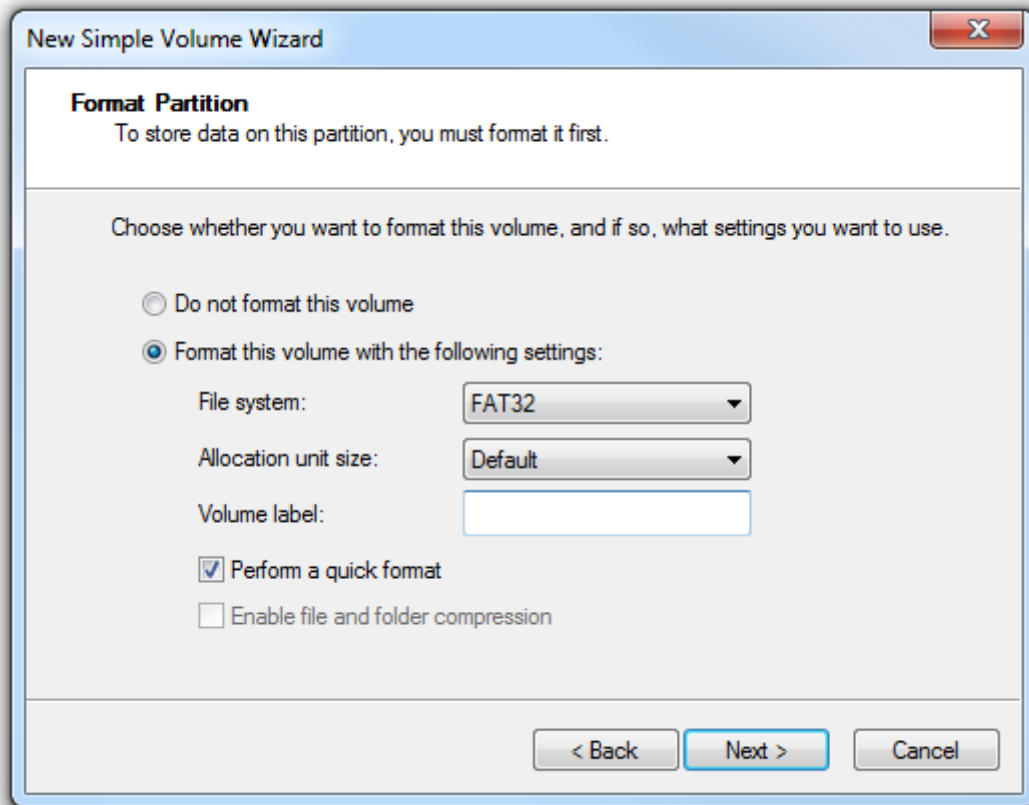
1. In the **Disk Management** tool, right-click the new VHD and select **Initialize Disk**.
2. When the **Initialize Disk** dialog box appears, verify that the selected disk is the one that you created above.
3. Click **MBR (Master Boot Record)** for the partition style and click **OK**.
4. Right-click the unallocated disk partition and select **New Simple Volume** as shown here.

Figure 9 - New Simple Volume



5. In the **New Simple Volume Wizard**, click **Next** until you reach the **Format Partition** page. Do the following:
 - a. In the **File System** list, select **FAT32**.
 - b. Clear the **Volume label** box to make the volume label blank. The **New Simple Volume Wizard** should appear as shown here.

Figure 10 - New Simple Volume Wizard



◆ Important

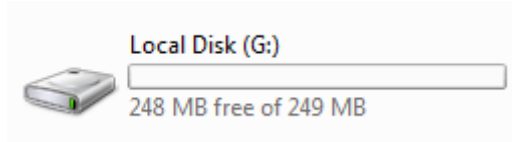
If **Volume label** is not blank, you will be unable to embed a boot loader in this VHD in the next series of steps. You can set the volume label after you have embedded the boot loader.

6. Click **Next**, and then click **Finish**.

▶ To activate the VHD

1. In the **Disk Management** tool, right-click the partition you created earlier and select **Mark Partition as Active**.
2. To exit the **Disk Management** tool, click the **File** menu and click **Exit**.
3. Click **Start** and click **Computer** to view the drives on your host computer to verify that your VHD has been activated. After you activate the VHD, you should see the new VHD with a new drive letter as shown here.

Figure 11 - New VHD Drive



You may see a different drive letter depending on the number of other drives in your host computer. Remember the drive letter assigned to your VHD so that you can use it in the following steps.

▶ To add the boot loader to the VHD

1. Click Start, click All Programs, and then click Accessories. Right-click Command Prompt and select Run as administrator.
2. Add `%_WINCEROOT%\platform\VirtualPC\src\boot\i386` to your PATH environment variable. For example if your Windows Embedded Compact installation is located at the default path `C:\WINCE700`, you can enter the following command in the command prompt window:

```
PATH = %PATH%;C:\WINCE700\platform\VirtualPC\src\boot\tools\bin\i386
```

3. Go to the release folder for your Virtual CEPC. If you named your Virtual CEPC OS design "VCEPC" as shown in previous examples, you can enter the following command in the Command Prompt window:

```
chdir /D C:\WINCE700\OSDesigns\VCEPC\VCEPC\RelDir\VirtualPC_x86_Release
```

4. Use `cesys` to embed the boot loader in your VHD, passing it the drive letter for the VHD that you created above. For example, if your VHD has the drive letter G:, enter the following command:

```
cesys G: wceldrC -f
```

When `cesys` successfully installs the boot loader, you will see a message like the following:

```
Boot Loader transferred on drive G:
```

5. If you see the error message, **Cannot obtain exclusive access to drive ...**, then close all other windows and applications and retry the `cesys` command again. Typically, this error occurs if the Disk Management tool is still running or if a Windows Explorer window remains open. If you still see this error after closing all windows and applications, it may be caused by your antivirus application.

After you have completed these steps, your VHD will have a boot loader that your vCEPC can use to boot from the VHD. This boot loader, when it starts, will check for the existence of the file `NK.bin` (your OS image) in the VHD partition, copy it to vCEPC memory, and transfer control to this OS image to start Windows Embedded Compact.

📌 Note

You must embed the boot loader in your VHD before you copy any files to the VHD. If you attempt to embed the boot loader after copying files to the VHD, you will see a message like

the following when you attempt to boot your vCEPC:

```
Microsoft Windows CE XLDR Version 1.0 (Built Apr 20 2011 15:44:40)
BootLoader Record Checksum Verification Failed!
*** Spin For Ever ***
```

You can remedy this error by using the Disk Management tool to reformat the VHD disk partition as described earlier and then re-embedding the boot loader by using `cesys`.

In the next series of steps, you will add your NK.bin (OS image) to your VHD.

To add an OS image to the VHD

1. Locate the OS image file for your OS design (NK.bin). For example, if your OS design resides in C:\WINCE700\OSDesigns\VCEPC, the path to the NK.bin file is:

```
C:\WINCE700\OSDesigns\VCEPC\VCEPC\RelDir\VirtualPC_x86_Release\NK.bin
```

2. Copy NK.bin to your VHD. For example, if your NK.bin resides at the path in the previous example and your VHD has drive letter G:, you can use the following commands:

```
chdir /D C:\WINCE700\OSDesigns\VCEPC\VCEPC\RelDir\VirtualPC_x86_Release
copy NK.bin G:
```

Note

By default, the OS image name used by the boot loader is “NK.bin” – if the file name for your OS image is not NK.bin, you must rename it to NK.bin before copying it to your VHD drive.

After you have completed these steps, you can use your VHD to boot a Virtual CEPC virtual machine without Platform Builder. In the next section, you will prepare your vCEPC to use this VHD.

Preparing Your vCEPC for Stand-Alone Use

After you create a stand-alone VHD (either manually or by using DiskPrep) you can prepare your vCEPC for stand-alone use by installing communication files used by Visual Studio and any other files that you need for application development. When your VHD is ready, you can connect it to your vCEPC so that your vCEPC can boot and load your OS image from the VHD.

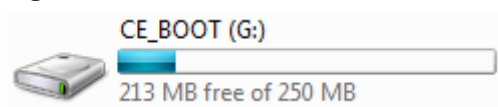
Copying Files to Your VHD

To copy files to your VHD, you attach your VHD to your host computer so that you can access it like a physical drive. After you copy files to your VHD, you detach it from your host computer so you can attach it to your vCEPC.

► **To attach your VHD to your host computer**

1. On your host computer, click **Start** and then click **Run**.
2. In the **Open** box, type **diskmgmt.msc** and press Enter. This command starts the **Disk Management** tool.
3. On the **Action** menu, click **Attach VHD**.
4. In the **Attach Virtual Hard Disk** dialog box, click **Browse** and navigate to the VHD image file that you created earlier. For example:
C:\WINCE700\OSDesigns\VCEPC\custom.vhd
5. Click **OK**.
6. Click **Start** and click **Computer** to view the drives on your host computer. You should see your VHD attached to your host computer with a drive letter as shown here.

Figure 12 - Attached VHD



You may see a different drive letter depending on the number of other drives in your host computer. Remember the drive letter assigned to your VHD so that you can use it in the following steps. If you used DiskPrep to create your VHD, it should have the volume name "CE_BOOT" as shown in the previous figure (DiskPrep sets this volume name after embedding the boot loader).

► **To add communications files to the VHD**

1. Click **Start**, click **All Programs**, and then click **Accessories**. Right-click **Command Prompt** and click **Run as administrator**.
2. In the **Command Prompt** window, enter the following command to go to the folder that contains the Visual Studio connectivity files:

```
chdir /D "%ProgramFiles%\Common Files\
Microsoft Shared\CoreCon\1.0\Target\wce400\x86"
```

3. Copy the following files to your VHD:

- Clientsshutdown.exe
- ConmanClient2.exe
- CMaccept.exe
- eDbgTL.dll
- TcpConnectionA.dll

For example, if your VHD is using drive letter G:, enter the following commands:

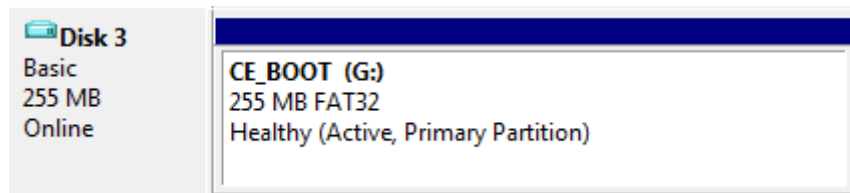
```
copy Clientshutdown.exe G:
copy ConmanClient2.exe G:
copy CMaccept.exe G:
copy eDbgTL.dll G:
copy TcpConnectionA.dll G:
```

At this point, you can copy any additional files to your VHD that you need for development. When your VHD is ready, detach it from your host computer as described in the next set of steps.

► **To detach the VHD from your host computer**

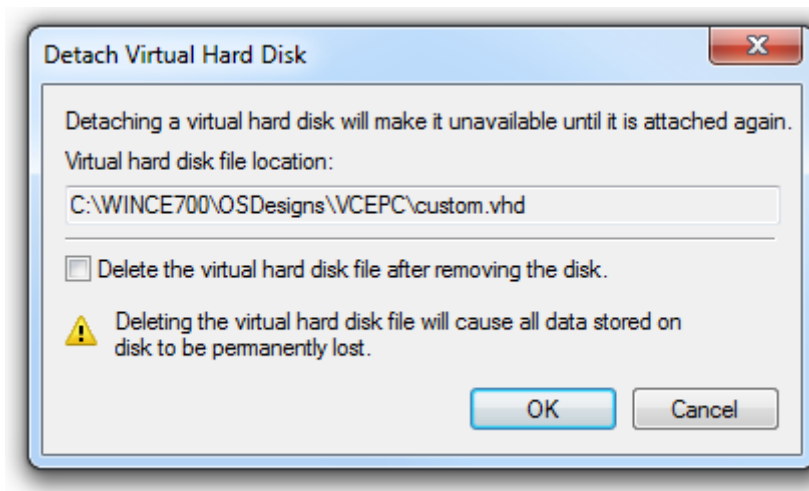
1. Locate the list of disks in the lower-left pane of the **Disk Management** tool.
2. Find your VHD in the list of disks. It should have a FAT32 partition and the drive letter that you observed earlier, as shown in the following figure. If you used DiskPrep to create your disk, it will have the volume label "CE_BOOT."

Figure 13 - VHD in the Disk Management Tool



3. Right-click this disk and select **Detach VHD**.
4. When the **Detach Virtual Hard Disk** dialog box appears, clear the **Delete the virtual hard disk file after removing the disk** check box if it is selected, as shown in the following figure.

Figure 14 - Detach Virtual Hard Disk



5. Click **OK**.
6. To exit the **Disk Management** tool, click the **File** menu and click **Exit**.

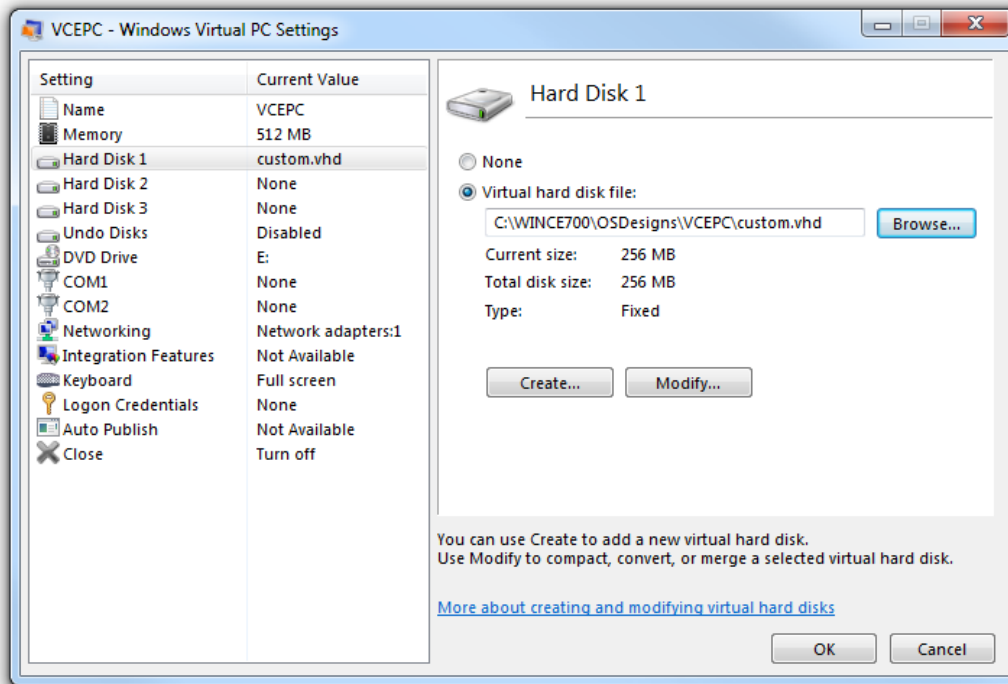
Linking Your VHD to Your vCEPC

Before your vCEPC can boot from your VHD, you must link the VHD image file to your vCEPC. You link your VHD to your vCEPC by changing the Virtual PC hard disk settings and by using the boot loader configuration menu to change the boot source.

To link your VHD to the vCEPC

1. If you have not yet created a Virtual CEPC virtual machine in Windows Virtual PC, create a virtual machine by using the instructions in the introductory article [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkID=190470) (<http://go.microsoft.com/fwlink/?LinkID=190470>).
2. If your host computer is running Windows XP or Windows Vista, start Virtual PC 2007 and click **Settings** on the **Virtual PC Console**. If your host computer is running Window 7, right-click your .vmcx or .vmc file (for example, VCEPC.vmcx) in your virtual machines folder and select **Settings**.
3. In the **Windows Virtual PC Settings** dialog box, do the following:
 - a. Under **Setting** on the left side of the dialog box, select **Hard Disk 1** and verify that **Virtual hard disk file** is selected on the right of the dialog box as shown in the following figure.
 - b. Click **Browse** and navigate to the custom VHD that you created earlier.
 - c. Click **OK** to link this VHD to your vCEPC.

Figure 15 - Selecting the Custom Virtual Hard Disk



► To change the boot source

1. If your host computer is running Windows XP or Windows Vista, start Virtual PC 2007 and click **Settings** on the **Virtual PC Console**. If your host computer is running Window 7, right-click your .vmcx file (for example, VCEPC.vmcx) in your virtual machines folder and select Settings.
2. When you see the configuration menu prompts appear on the **Virtual PC console**, press the spacebar immediately to open the configuration menu. The **Main Menu** appears with a set of configuration options:

```
-----
Main Menu
-----
```

```
[1] Show Current Settings
[2] Select Boot Source
[3] Select KITL Device
[4] Network Settings
[5] Display Settings
[6] Debug Port Settings
[7] Save Settings
[0] Exit and Continue
```

 **Note**

If your VHD is not prepared correctly, the boot loader will not start and you will not see this menu. Instead, Virtual PC will display the following message in the console:

```
Reboot and Select proper Boot device  
or Insert Boot Media in selected Boot device
```

If you see this message, try creating your custom VHD again using the steps described previously. Typically, this error message occurs when the partition is not marked as active in the VHD.

3. Type **2** in the **Main Menu** to enter the **Select Boot Source** submenu.
4. In the **Select Boot Source** submenu, type **1** to select **Boot Drive**. This option selects the boot drive as the boot source and exits the **Boot Drive** submenu.
5. Type **1** in the **Main Menu** to show the current settings. Verify that the boot source is set to **Boot Drive**.
6. Type **7** in the **Main Menu** to save the settings.

In the Main Menu, you can type **0** to exit and continue the boot loading process. Or, if your stand-alone vCEPC must also work without a network, you can continue to use the configuration menu to prepare your vCEPC for “off-network” use as described previously in the section “Using Virtual CEPC Without a Network”.

When your vCEPC starts, it loads your OS image and displays the Windows Embedded Compact desktop as shown in the following figure.

Figure 16 - Stand-Alone vCEPC

Creating an SDK for Virtual CEPC

OEMs typically use Platform Builder to create a software development kit (SDK) for a particular Windows Embedded Compact powered device or product platform. When you provide an SDK for your target device, application developers do not need to install Windows Embedded Compact or Platform Builder to create applications for your target device. After you create an installable SDK package for your Virtual CEPC OS design, application developers need only the following items to develop and test applications on your Windows Embedded Compact powered Virtual CEPC device:

- Visual Studio
- Virtual PC
- Your SDK
- A stand-alone VHD containing your OS design image

The following sections explain how to create an SDK for Virtual CEPC. Before creating this SDK, you must first create a Virtual CEPC OS design and a stand-alone Virtual CEPC VHD that boots from this OS design image. For more about creating a Virtual CEPC OS design, see the introductory article [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkID=190470) (<http://go.microsoft.com/fwlink/?LinkID=190470>). For more about modifying this OS design for stand-alone use, see the section “Creating a Stand-Alone OS Design.” For more about creating a Virtual CEPC VHD, see the earlier section, “Creating a Stand-Alone Virtual Hard Disk.”

Generating the SDK Package

Platform Builder can automatically generate an SDK installer package based on the OS design that you create for Virtual CEPC.

► To generate an SDK package for your OS design

1. Open your Virtual CEPC OS design project in Platform Builder.
2. In the **Solution Explorer** pane, right-click **SDKs** and click **Add New SDK**.
3. In the **SDK1 Property Pages** dialog box, enter the name of your SDK in the **SDK Name** box. For example, if the name of your product is “MyDevice”, you could use “MyDevice Software Development Kit.”
4. In the **Product Name** box, enter the name of the target device or product platform that this SDK is intended for.
5. In the **Product Version** box, enter the version number of your product that this SDK is intended for.
6. In the **Company Name** box, enter the name of your company.
7. In the **Company Website** box, enter the URL for your company’s website.
8. Click **OK**.
9. In **Solution Explorer**, when you see the name of your SDK appear under **SDKs**, right-click your SDK and click **Build**.

The build process can take several minutes or more, depending on the speed of your computer. When the build completes, a message similar to the following appears in the output pane:

```
C:\WINCE700\build.log(1096) : Message:
Exported SDK to: C:\WINCE700\OSDesigns\VCEPC\VCEPC\SDKs\SDK1\MSI\SDK1.msi
C:\WINCE700\OSDesigns\VCEPC\VCEPC\SDKs\SDK1\SDK1.sdkcfg - 0 error(s), 0 warning(s)
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
```

The SDK package can be found in the path as shown in the output pane. For example, if “VCEPC” is the name of your OS design, you can find the SDK package at the location:

C:\WINCE700\OSDesigns\VCEPC\VCEPC\SDKs\SDK1\MSI\SDK1.msi

In the next section, you use this SDK to create a simple “Hello World” application for Virtual CEPC.

Verifying Your SDK

After you create your SDK package, you can install it and create a simple Virtual CEPC application to verify that it works.

▶ To install your SDK package

1. Exit Visual Studio.
2. Navigate to the location of the SDK MSI file created earlier and double-click it to install the SDK.
3. Follow the instructions displayed by the installation wizard and verify that the information you used to create the SDK is displayed properly.

▶ To create a project that uses your SDK package

1. Start Visual Studio.
2. On the **File** menu, click **New**, and then click **Project**.
3. When Visual Studio displays the **New Project** dialog box, under **Project types**, expand **Visual C++** if it is not already expanded, and select **Smart Device**.
4. Under **Templates**, select **Win32 Smart Device Project**.
5. In the **Name** box, type a name for your project (for example, "MyApplication").
6. In the **Location** box, you can accept the default location C:\Users\<your login name>\Documents\Visual Studio 2008\Projects or you can browse to a different location.
7. In the **Solution Name** box, type the solution name that you want to use for your application (or accept the default name that you typed in step 5), and then click **OK**.
8. When you see **Welcome to the Win32 Smart Device Project Wizard**, click **Next**.
9. When the **Win32 Smart Device Project Wizard** displays the **Platforms** page, click the << button to clear the **Selected SDKs** column.
10. Under **Installed SDKs**, select the name of your SDK, click the > button to move it to the **Selected SDKs** column and click **Next**.
11. On the **Project Settings** page, under **Application type**, select **Windows application**, and click **Finish**.

When the **Win32 Smart Device Project Wizard** finishes, Visual Studio opens your project and displays the source file that contains the **WinMain** function. For example, if the name of your project is "MyApplication," you will find **WinMain** in MyApplication.cpp.

In the next set of steps, you replace this functionality with a simple "Hello World" example application and use it to verify your SDK.

▶ To create the sample application

1. In Visual Studio, open the project that you created in the previous set of steps and edit the file

that contains **WinMain** (for example, MyApplication.cpp).

2. Replace the contents of this file with the following code sample:

```
#include "stdafx.h"

// The main window class name.

TCHAR szWindowClass[] = TEXT("HelloWorldApp");

// The string that appears in the application's title bar.

TCHAR szTitle[] = TEXT("Hello World");

// Forward declaration of callback function.

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

int WINAPI WinMain(HINSTANCE hInstance,
                  HINSTANCE hPrevInstance,
                  LPTSTR lpCmdLine,
                  int nCmdShow)
{
    WNDCLASS wc;
    wc.style = CS_HREDRAW | CS_VREDRAW;
    wc.lpfnWndProc = (WNDPROC) WndProc;
    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;
    wc.hInstance = hInstance;
    wc.hIcon = 0;
    wc.hCursor = 0;
    wc.hbrBackground = (HBRUSH) GetStockObject(WHITE_BRUSH);
    wc.lpszMenuName = 0;
    wc.lpszClassName = szWindowClass;
```

```

// Register the application

RegisterClass(&wc);

// Create the window for the application: specify the name
// of the application, the title bar text, flags to make
// the window visible and turn on decorations. The window
// is 300x200 in size and there is no parent window or menu.

HWND hWnd = CreateWindow (
    szWindowClass, szTitle, WS_VISIBLE | WS_CAPTION,
    150, 100, 300, 200, NULL, NULL, hInstance, NULL);

// Display the window.

ShowWindow (hWnd, nCmdShow);

// Send a WM_PAINT message to the window.

UpdateWindow (hWnd);

// Message loop that listens for OS messages. When
// the application receives a message, this loop dispatches
// the message to the WndProc function.

MSG msg;
while (GetMessage(&msg, NULL, 0, 0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
return msg.wParam;
}

```

```

// Function that processes messages for the main (and only) window:

LRESULT CALLBACK WndProc(HWND hWnd, UINT message,
                        WPARAM wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;
    HDC hdc;
    TCHAR szMessage[] = L"Hello World!";

    switch (message)
    {
        case WM_PAINT:

            // Display the message:

            hdc = BeginPaint(hWnd, &ps);
            RECT rt;
            GetClientRect(hWnd, &rt);
            DrawText(hdc, szMessage,
                    _tcslen(szMessage), &rt, DT_CENTER);
            EndPaint(hWnd, &ps);
            break;

        case WM_DESTROY:
            PostQuitMessage(0);
            break;

        default:
            return DefWindowProc(hWnd, message, wParam, lParam);
    }
    return 0;
}

```

3. On the **File** menu, click **Save All** to save your changes.
4. On the **Build** menu, click **Build Solution**. When the build completes successfully, a message similar to the following appears in the output pane:

```
1>MyApplication - 0 error(s), 0 warning(s)
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

In the next set of steps, you start this application from Visual Studio and run it on your stand-alone vCEPC. First, you configure Visual Studio with the IP address of your vCEPC, and then you connect your vCEPC to Visual Studio and start your application from within Visual Studio.

 **To configure Visual Studio with the IP address of your vCEPC**

1. Start your vCEPC if it is not already running.
2. On your vCEPC desktop, click **Start**, click **Programs**, and then click **Command Prompt**.
3. In the **Command Prompt** window, type the following command to display the IP address of your vCEPC:

```
ipconfig
```

4. When this command finishes, a message similar to the following appears in the Command Prompt window:

```
Windows IP configuration

Ethernet adapter [PCI\DC21X41]:

    IPv4 Address . . . . : 10.125.149.75
    Subnet Mask . . . . : 255.255.254.0
    IPv6 Address . . . . : fe80:6468:9205:4eeb:a79b%4
    Default Gateway . . . : 10.125.149.1
```

5. Find the IP address for your vCEPC. The IP address can be found to the right of **IPv4 Address**. In the above example, the IP address is 10.125.149.75. You will enter this IP address into Visual Studio in step 11 below.
6. In Visual Studio, on the **Tools** menu, click **Options**.
7. In the **Options** dialog box, expand **Device Tools** and click **Devices**.
8. Under **Devices**, select the device name that is associated with your SDK (for example, "MyDevice Software Development Kit x86 Device").
9. Click **Properties** to open the **Device Properties** dialog box.
10. In the **Device Properties** dialog box, click the **Configure** button to the right of **TCP Connect Transport**.
11. In the **Configure TCP/IP Transport** dialog box, in the **Use specific IP address** box, enter the IP address for your vCEPC. Click **OK**.
12. In the **Device Properties** dialog box, click **OK**.

13. In the **Options** dialog box, click **OK**.

At this point, Visual Studio has the IP address of your vCEPC so that Visual Studio can connect to your vCEPC to start and debug applications. If you reboot your vCEPC, you don't have to repeat these steps.

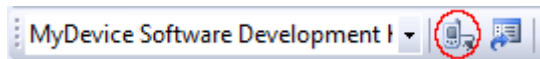
▶ **To run the sample application on your vCEPC**

1. In the **Command Prompt** window of your vCEPC, enter the following commands:


```
cd hard disk
ConmanClient2.exe
```
2. On the Virtual CEPC desktop, click **Start**, click **Programs**, and then click **Command Prompt** to open a second **Command Prompt** window.
3. In the second **Command Prompt** window of your vCEPC, enter the following commands:

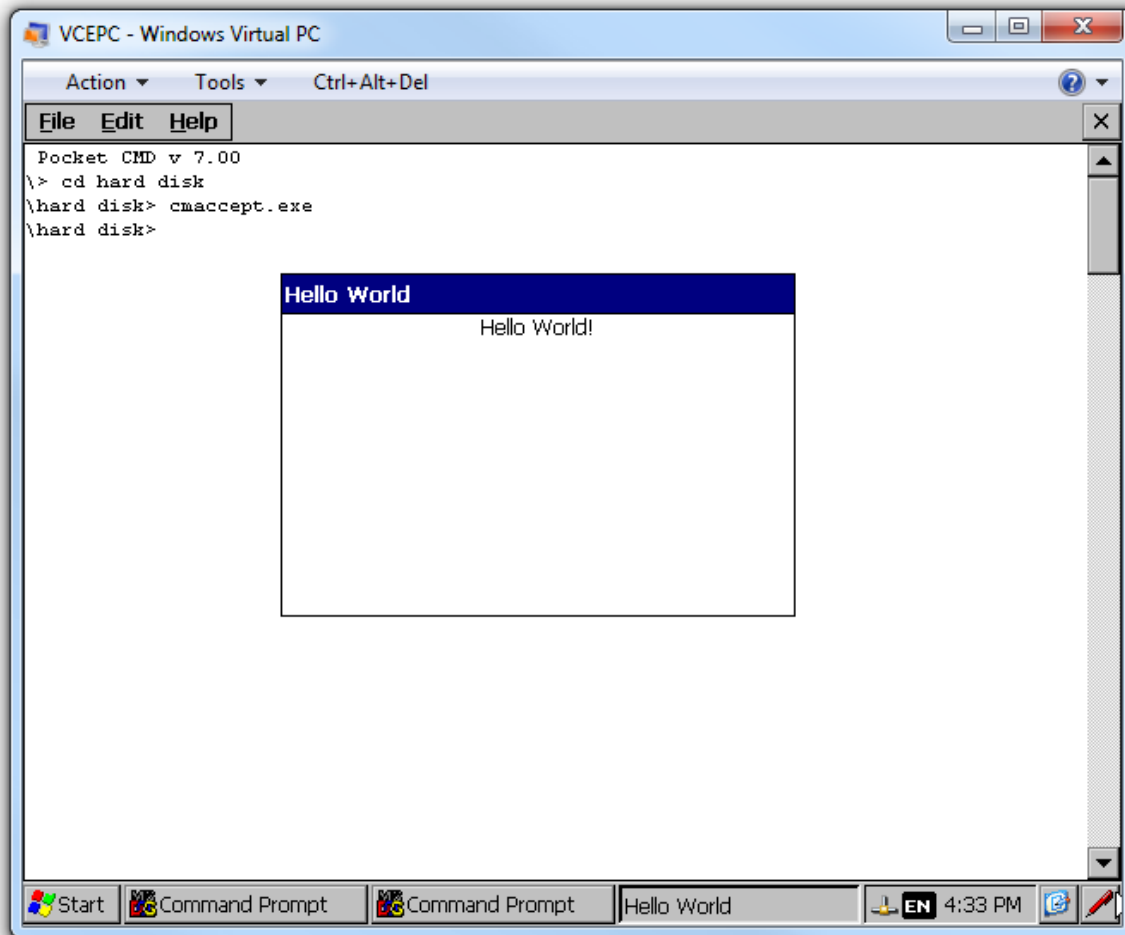

```
cd hard disk
CMAccept.exe
```
4. In Visual Studio, click the "Connect to Device" icon as shown here:

Figure 17 - Connect to Device Icon



5. When Visual Studio displays a window with the message "Connection succeeded," click the Close button.
6. In Visual Studio, on the Debug menu, click Start Debugging. The sample application appears on the desktop of your vCEPC as shown in the following figure.

Figure 18 - Running the Sample Application



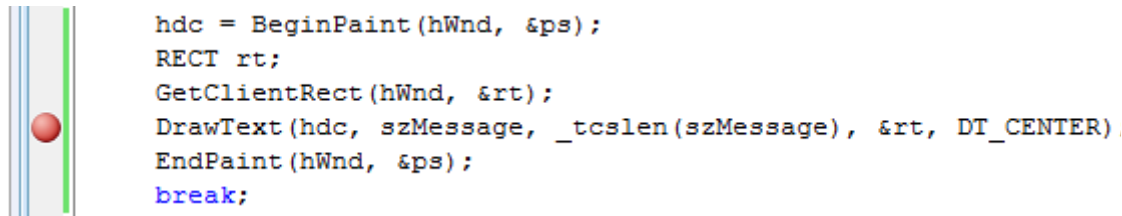
To stop the sample application, right-click **Hello World** on the vCEPC taskbar and click **Close**.

7. If you want to restart your vCEPC, you only need to run ConmanClient2.exe and CMAccept.exe again as described above before starting and debugging your application.

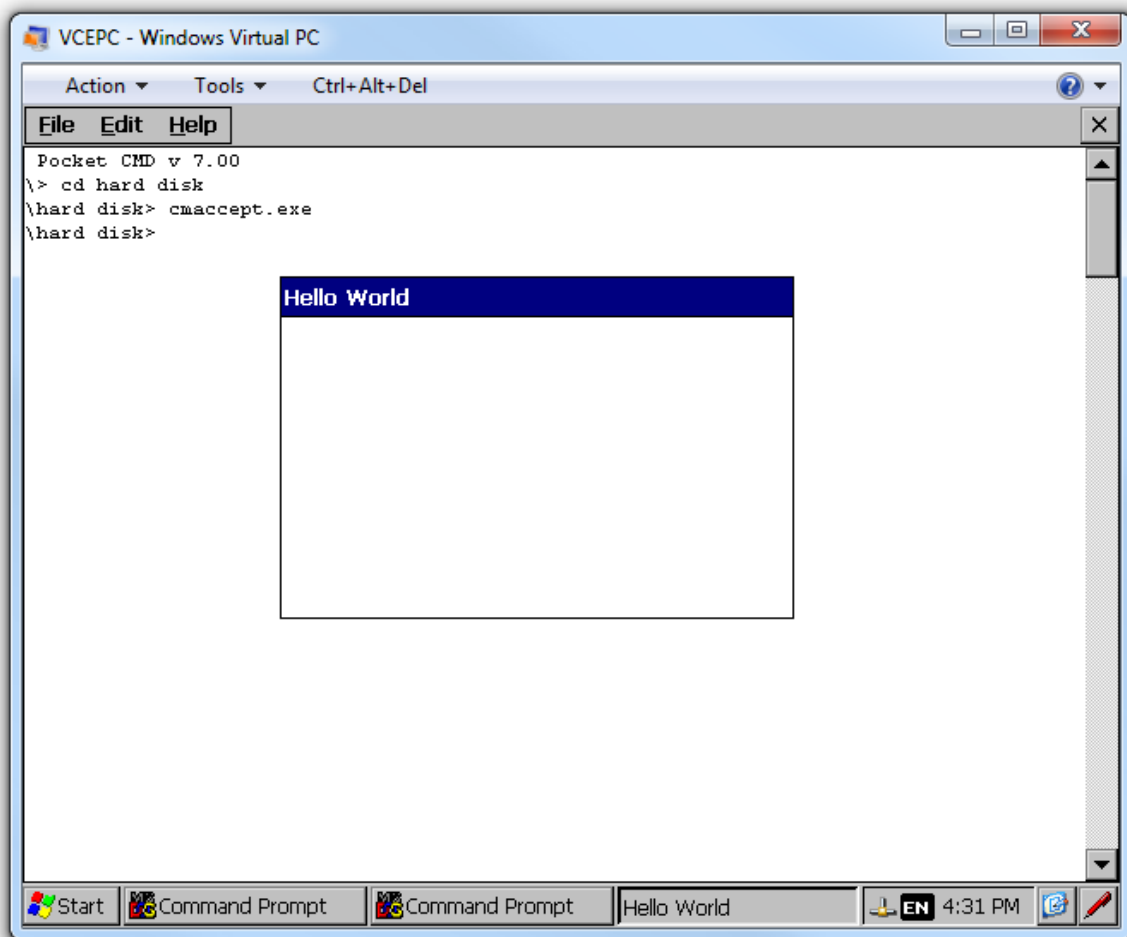
After you verify that the sample application runs successfully on your vCEPC, you can verify that you have debug capability by setting breakpoints in this sample application before running it again.

▶ To verify debug capability

1. In Visual Studio, open the file in the sample application that contains **WinMain** (for example, MyApplication.cpp).
2. Find the line of code that calls the function **DrawText**.
3. Click in the gray area on the left-hand side of the call to **DrawText** to set a break point at this location. The break point is represented as a reddish circle as shown here.

Figure 19 - Setting a Break Point

4. Press F5 (or click **Debug** and click **Start Debugging**) to start the application in the debugger. The sample application should break before it prints the message to the application window, as shown in the following figure.

Figure 20 - Application at Break Point

5. Press F5 to continue (or click **Debug** and click **Continue**). The sample application should

continue past the break point and call the **DrawText** function, which displays the “Hello World!” message in the application window (as shown in Figure 18).

These verification steps demonstrate how you can use Virtual CEPC to develop your Windows Embedded Compact applications in Visual Studio—like any other Visual Studio application—without using Platform Builder and without having to install Windows Embedded Compact on the computer that you use for application development.

Conclusion

When you use Windows Embedded Compact 7, you have the option to run Windows Embedded Compact in an emulated software development environment called a virtual CEPC (vCEPC). You can use the advanced features of Virtual CEPC to customize how you use it so it accommodates your development needs and preferences. You can change the display resolution, boot from a network device or from a boot drive, and select a different COM port for debug use. You can run “off network” if your host computer does not have a network connection or DHCP server access.

Virtual CEPC provides many different boot options. You can boot from the sample VHD and download your OS design from Platform Builder, you can create a stand-alone VHD that contains your OS design so that you don’t have to download it from Platform Builder, or you can boot from virtual floppy disk so that you don’t have to install a boot loader in your VHD.

You can customize your OS design and your VHD to create a stand-alone Windows Embedded Compact application development platform. You can easily generate an SDK package for your Virtual CEPC OS design, so you can use Virtual CEPC for application development on computers that do not have Platform Builder and Windows Embedded Compact installed. When you install this SDK on your host computer and use it with a stand-alone virtual CEPC, you can develop and debug Windows Embedded Compact powered applications in Visual Studio as if you were developing a native application on your host computer.

In short, Virtual CEPC provides you with an inexpensive and easy-to-use platform for developing Windows Embedded Compact applications by eliminating the expense and configuration overhead of physical hardware. In addition to the reduced expense of creating and testing different emulated hardware and software configurations, you can modify Virtual CEPC to suit your development needs and preferences.

Additional Resources

To learn more about Virtual PC, see the following links:

- [Windows Virtual PC website](http://go.microsoft.com/fwlink/?LinkId=78095) (http://go.microsoft.com/fwlink/?LinkId=78095)
- [Windows Virtual PC download page](http://go.microsoft.com/fwlink/?LinkId=186188) (http://go.microsoft.com/fwlink/?LinkId=186188)
- [Virtual PC 2007 download page](http://go.microsoft.com/fwlink/?LinkId=186183) (http://go.microsoft.com/fwlink/?LinkId=186183)

To learn more about Windows Embedded, see the following link:

- [Windows Embedded website](http://go.microsoft.com/fwlink/?LinkId=183524) (http://go.microsoft.com/fwlink/?LinkId=183524)

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2011 Microsoft. All rights reserved.