Getting Started with Virtual CEPC

Windows Embedded Compact 7 Technical Article

Writer: Mark McLemore

Technical Reviewers: Shajib Sadhukha, Francisco Blanquicet

Published: May 2011

Applies To: Windows Embedded Compact 7

Abstract

OEMs often use a software development board (SDB) to evaluate Windows Embedded Compact or to develop new Windows Embedded Compact-based applications. However, to use an SDB for your development platform, you must invest in additional hardware, install and configure development tools specific to your SDB, and connect the hardware to your host computer for testing and debugging your application.

When you use Windows Embedded Compact 7, you have the alternative of running Windows Embedded Compact in an emulated software development board environment called a *virtual CEPC* (vCEPC). A vCEPC eliminates the hardware expense and configuration overhead of physical hardware so that you can begin developing with Windows Embedded Compact quickly and inexpensively. In addition, you can use a vCEPC to create, save, and test different emulated x86 hardware configurations without having to swap and configure hard disk drives, memory, display controllers, network adapters, and sound cards.

This article shows you how to set up a vCEPC and use it to develop and run a simple Windows Embedded Compact application. A companion article, Advanced Virtual CEPC (http://go.microsoft.com/fwlink/?LinkID=206041), explains the advanced features of Virtual CEPC.

Introduction

When you use a virtual CEPC (vCEPC) for Windows Embedded Compact-based development, you use Windows Virtual PC to emulate a Windows Embedded Compact x86-based hardware platform in a virtual machine environment on your development computer. A vCEPC uses Virtual PC to emulate a standard x86 hardware board (CEPC) on any host computer running Windows 7, Windows Vista, or Windows XP.

In brief, this is what you do to configure and use a vCEPC for application development:

1. Create a Windows Embedded Compact OS design that supports a vCEPC.
2. Configure Windows Virtual PC to run the vCEPC virtual machine.
3. Boot your Windows Embedded Compact OS image in Virtual PC.
4. Create a Platform Builder subproject for your application.
5. Use Platform Builder to start and run your application on your vCEPC.

This article explains this process in detail. When you complete this process, you will be able to load and boot your OS image in a vCEPC virtual machine on your development host computer and run your application in this virtual machine. You can interact with the vCEPC in the same way that you would interact with a physical development board, using the vCEPC to develop, test, and debug Windows Embedded Compact applications.

Prerequisites

Virtual CEPC requires the following minimum software configurations.

Windows 7

- Windows 7 Enterprise, Windows 7 Ultimate, or Windows 7 Professional
- Windows Virtual PC
- Microsoft Visual Studio 2008
- Windows Embedded Compact 7

Windows Vista, Windows XP

- Windows Vista Enterprise, Windows Vista Ultimate, Windows Vista Business, or Windows XP Professional
- Microsoft Virtual PC 2007
- Visual Studio 2008
- Windows Embedded Compact 7

You can download Windows Virtual PC from the Windows Virtual PC download page (http://go.microsoft.com/fwlink/?LinkId=186188). Virtual PC 2007 is available from the Virtual PC 2007 download page (http://go.microsoft.com/fwlink/?LinkId=186183).

When you install Windows Embedded Compact 7 and choose **Installs all required, optional, and 3rd party options** in the **Install Options** dialog box, Virtual CEPC is included automatically. If you choose **Customize your installation** in the **Install Options** dialog box, select the following options in the **Customize Installation** dialog box to enable Virtual CEPC support:

- Platform Builder
- x86 Architecture

For more information about Windows Embedded development and Platform Builder, see the article Integrated Development Environment Build Process (http://go.microsoft.com/fwlink/?LinkId=178104).

Create a Virtual CEPC OS Design

To run Windows Embedded Compact on a vCEPC, you must first create an OS design that supports a vCEPC. To create a vCEPC-capable OS design, you perform the following tasks:

1. Create a Platform Builder project as the basis for your OS design.
2. Configure your OS design for a vCEPC.
3. Customize your OS design by adding or removing catalog items.
4. Select an appropriate Virtual PC keyboard driver.
5. Build your OS design.

The following subsections describe the steps for each of these tasks. When you complete these tasks, you will have a boot image for your OS design that you can load and run in Virtual PC.

Create a Project for Your OS Design

If you do not have an existing Platform Builder project for your OS design, use the following steps to create a new project for your OS design.

To create a project for your OS design

1. In Visual Studio 2008, on the **File** menu, click **New**, and then click **Project**.
2. When Visual Studio displays the **New Project** dialog box, under **Project types**, select **Platform Builder**.
3. In the **Name** box, type a name for your project (for example, “VCEPC”).
4. In the **Location** box, you can accept the default location C:\WINCE700\OSDesigns, or you can browse to a different location. Platform Builder will create a new folder for your vCEPC OS design at the location you choose.
5. In the **Solution Name** box, type the solution name that you want to use or accept the default name that you typed in step 3. Click **OK**.

Configure Your OS Design

After you create a project for your OS design, you configure your OS design through the OS Design Wizard. You start with an OS design template and then add or remove catalog items to build the functionality that you want into your OS design.

To configure your vCEPC OS design

1. After you type a name and location for your project as directed in the previous list of steps, Platform Builder starts the OS Design Wizard. When the **Welcome to the OS Design Wizard** page appears, click **Next**.

2. On the **Board Support Packages** page, under **Select one or more BSPs**, choose **Virtual PC: x86**, and then click **Next**.
3. On the **Design Templates** page, you customize your OS design by selecting a design template that best approximates the characteristics of the physical device you want to emulate. The **Enterprise Handheld** template is a good starting point to use with your vCEPC. To create an OS design based on the **Enterprise Handheld** template, expand **Enterprise Device**, select **Handheld**, and then click **Next**.
4. When the OS Design Wizard presents the **Applications and Media** page, select any additional applications or services that you want to include in your OS design, and then click **Next**.
5. On the **Networking and Communications** page, expand **Local Area Network (LAN)** and select **Wired Local Area Network**. Click **Next**.
6. When the OS Design Wizard presents the **OS Design Project Wizard Complete** page, clear the **Build the following configuration after I click Finish** check box (if it is selected) so that you can build the OS design in a later step. Click **Finish**.
7. Platform Builder may present you with a notification window that explains security warnings that are associated with certain catalog items that are included in your OS design. Click **Acknowledge** to indicate that you have read and fully understand these warnings.

Add Catalog Items to Your OS Design

If there are any additional catalog items that you would like to include in your OS design, you must add them before you build your OS design.

To add catalog items to your OS design

1. You add new catalog items by using the **Catalog Items View**. If the **Catalog Items View** is not visible, follow these steps to make it visible:
   a. In Visual Studio, click the **View** menu.
   b. Click **Other Windows** and then click **Catalog Items View**. The **Catalog Items View** pane appears in Visual Studio.
2. In the **Catalog Items View** pane, select one or more catalog items to include in your OS design. For example, to add the catalog item **Silverlight for Windows Embedded**, follow these steps:
   a. Expand **Core OS**, expand **Windows Embedded Compact**, expand **Shell and User Interface**, and then **Silverlight for Windows Embedded**.
   b. Select the **Silverlight for Windows Embedded** catalog item.
   c. On the **File** menu, click **Save All** to save your catalog item changes.

Select a Virtual CEPC Keyboard Driver

Depending on which version of Windows you are running on your host computer, you must select a specific keyboard driver to include in your OS design for support of the Virtual PC installation that you installed for that Windows version.

To select a keyboard driver

1. In the **Catalog Items View** pane, expand **BSP**, **Virtual PC : x86**, **Device Drivers**, and then **Keyboard**.
2. If your host computer is running Windows 7, select the check box for **VCEPC Keyboard driver:WIN7VPC**. If your host computer is running Windows Vista or Windows XP, select the check box for **VCEPC Keyboard driver:VPC2007**.

After you have chosen the appropriate keyboard driver, click the **File** menu, and click **Save All** to save your changes.

Build Your OS Design

At this point, the OS Design Wizard has created your OS design but it has not yet built an OS design boot image that you can load and boot in Virtual PC.

To build your OS design

1. In Visual Studio, click the **Build** menu.
2. Click **Build Solution** to build your OS design.

When you build your OS design, Platform Builder compiles and links the modules that go into your OS design boot image. The Visual Studio **Output** window displays the progress of the build. The build process may take several minutes or more to complete, depending on the speed of your computer.

Platform Builder creates your OS design in the folder C:\WINCE700\OSDesigns, using the name of your OS design as the folder name. For example, if you choose the name “VCEPC” for your OS design, Platform Builder creates an OS design folder in the following location:

C:\WINCE700\OSDesigns\VCEPC

When the build completes successfully, Platform Builder creates your OS boot image in your OS design folder. For a release build of “VCEPC”, for example, your boot image can be found at the following location:

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

The next section explains how to configure Virtual PC to load and run this vCEPC OS design boot image.

Create a Virtual Machine

After you have built an OS image for vCEPC, you create a virtual machine with a virtual hard disk (VHD) image. In the following steps, you use a virtual hard disk image that is included with Windows Embedded Compact 7. This VHD includes a boot loader – you interact with this boot loader in the Virtual PC console to download and boot your OS image to Virtual PC.

If you want to create a custom VHD with your own boot loader or you want to boot your OS image from your VHD (rather than downloading your OS image from Platform Builder), see “Creating a Stand-Alone Virtual Hard Disk” in the companion article Advanced Virtual CEPC (http://go.microsoft.com/fwlink/?LinkID=206041). If you want to use a custom VHD image that does not contain a boot loader, you can instead boot your vCEPC from a virtual floppy disk (VFD) image. For more about virtual floppy disk booting, see "Booting from Virtual Floppy Disk" in the companion article Advanced Virtual CEPC.

Because the steps for configuring the VHD and virtual machine 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 version of Virtual PC.

Virtual PC 2007

If your computer is running Windows Vista or Windows XP, you use Virtual PC 2007 to run your vCEPC. To configure Virtual PC 2007 to run a vCEPC, you use the New Virtual Machine Wizard to create a new virtual machine, and then configure this virtual machine to use the VHD image included with Windows Embedded Compact.

To configure Virtual PC 2007 to run a vCEPC

1. If you have not already done so, download and install Virtual PC 2007 (http://go.microsoft.com/fwlink/?LinkId=186183).
2. Start Virtual PC 2007. The **Virtual PC Console** and the **New Virtual Machine Wizard** appear.
3. If the **New Virtual Machine Wizard** does not appear, click the **File** menu of the **Virtual PC Console**, and then select **New Virtual Machine Wizard**.
4. When the **Welcome to the New Virtual Machine Wizard** page appears, click **Next**.
5. On the **Options** page, select **Create a virtual machine**, and then click **Next**.
6. On the **Virtual Machine Name and Location** page, in the **Name and Location** box, type a name for your virtual machine file. If you restrict this name to sixteen characters or fewer, you can reuse this name later when you name your Platform Builder target device. In the following examples, we use "VCEPC" for the virtual machine name. The virtual machine file is automatically saved to the **My Virtual Machines** folder. To save it to a different location, use the **Browse** button. Click **Next**.
7. On the **Operating System** page, click the **Operating System** menu and click **Other**. Click **Next**.
8. On the **Memory** page, you can adjust the amount of RAM that the virtual machine will use or accept the default size of 128 MB. To accept the default size, click **Using the recommended RAM**. Click **Next**.

Before you can boot an OS on your virtual machine, you must add a new or existing VHD image. A VHD image is a .vhd file that is stored on your physical hard disk. The VHD typically contains a boot loader and may contain an OS, applications, and data files used by a vCEPC.

To create a vCEPC virtual machine with a virtual hard disk

1. After you click **Next** in step 8 above, the **New Virtual Machine Wizard** displays the **Virtual Hard Disk Options** page. Under **Do you want to use**, select **An existing virtual hard disk**. Click **Next**.

2. On the **Virtual Hard Disk Location** page, the wizard asks you for the location of the existing VHD image. Under **Virtual hard disk location**, click **Browse** and navigate to the following location.

   C:\WINCE700\Platform\VirtualPC\VM\hd0_sample.vhd

3. Click **Next**.
4. After the **Completing the New Virtual Machine Wizard** page provides you with a summary of the attributes for your virtual machine, click **Finish** to close the New Virtual Machine Wizard and create this virtual machine.

When the wizard creates your virtual machine, the **Virtual PC Console** dialog box displays the name of your virtual machine followed by “Not running” as shown in this figure.

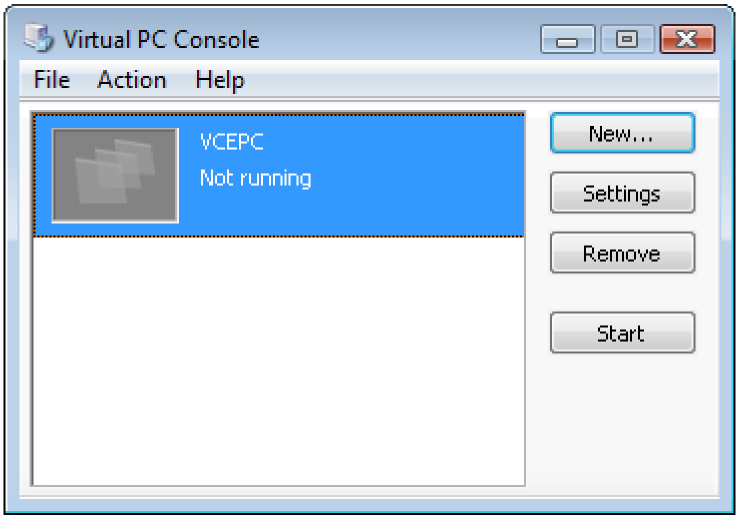


Figure 1 - Virtual PC 2007 Console

For now, leave this console window open. A later section explains how to start and connect this virtual machine to Platform Builder for the purpose of downloading your OS design boot image into Virtual PC 2007.

Windows Virtual PC

If your computer is running Windows 7, you use Windows Virtual PC to run your vCEPC. To configure Windows Virtual PC to run a vCEPC, you create a new virtual machine and configure this virtual machine to use the vCEPC hard disk image included with Windows Embedded Compact.

To configure Windows Virtual PC to run a vCEPC

1. If you have not already done so, download and install Windows Virtual PC (http://go.microsoft.com/fwlink/?LinkId=186188).
2. Start Windows Virtual PC. When Windows Virtual PC starts, it opens and displays your **Virtual Machines** folder.

3. On the menu bar of the **Virtual Machines** folder window, click **Create virtual machine**.
4. In the **Create a virtual machine** wizard, on the **Specify a name and location for this virtual machine** page, do the following:

   a. Type a name for your virtual machine in the **Name** box. If you restrict this name to sixteen characters or fewer, you can reuse this name later when you name your Platform Builder target device. In the following examples, we use “VCEPC” for the virtual machine name.

   b. In the **Location** box, you can accept the default location for your virtual machine file or click **Browse** to navigate to a different location. The default location is the **Virtual Machines** folder, which is located at

      C:\Users\<*user name*>\AppData\Local\Microsoft\Windows Virtual PC\Virtual Machines

   c. Click **Next**.

5. On the **Specify memory and networking options** page, you can adjust the amount of RAM that the virtual machine will use or you can accept the default size of 512 MB. To change the amount of memory, type the number of megabytes in the **RAM** box. Leave the **Use computer network connections checkbox** selected and click **Next**.

   Note Before you can boot an OS on your virtual machine, you must add a new or existing VHD image. A VHD image is a .vhd file that is stored on your physical hard disk. The VHD typically contains a boot loader and may also contain an OS, applications, and data files used by your vCEPC.

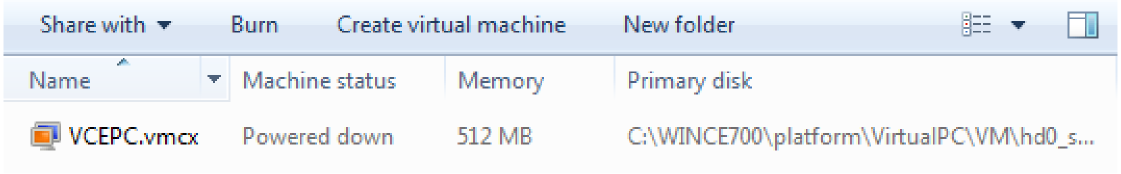
6. On the **Add a virtual hard disk** page, do the following:

   a. Select **Use an existing virtual hard disk**.

   b. In the **Location** box, type the following path. Alternately, you can click **Browse**, navigate to this location, and then click **Open**.

      C:\WINCE700\Platform\VirtualPC\VM\hd0_sample.vhd

   c. Click **Create**. If you chose the name “VCEPC” for your virtual machine, the file VCEPC.vmcx appears in the **Virtual Machines** folder.



Name	Machine status	Memory	Primary disk
VCEPC.vmcx	Powered down	512 MB	C:\WINCE700\platform\VirtualPC\VM\hd0_s...

Figure 2 - Virtual Machines Folder

A later section explains how to start and connect this virtual machine to Platform Builder for the purpose of downloading your OS design boot image into Virtual PC.

The Virtual CEPC Boot Loader

The virtual hard disk image included with Windows Embedded Compact includes a boot loader that you can use to download your OS image into your virtual machine. This boot loader runs immediately after you start your virtual machine, and you interact with this boot loader through the Virtual PC console. Before you begin the process of downloading your OS image to the boot loader, you should become familiar with how the boot loader works as described in this section.

After you start your vCEPC, the boot loader begins execution and progresses through three stages:

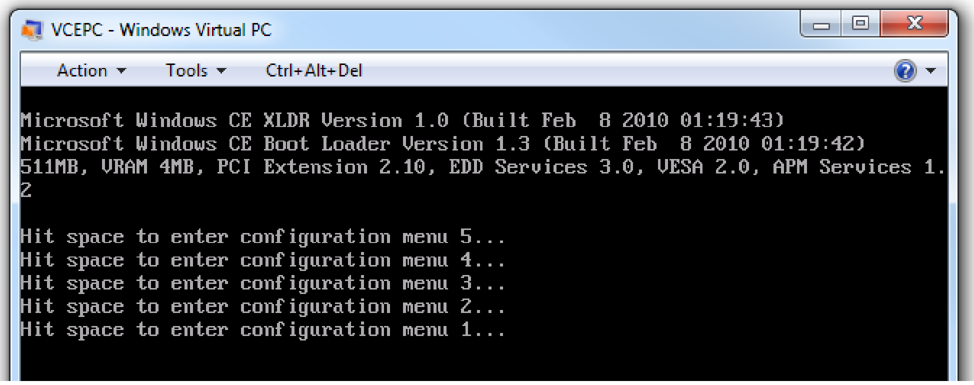
1. Configuration Menu stage
2. Network Address stage
3. BOOTME stage

Configuration Menu Stage

During this stage, the boot loader provides you with an opportunity to enter the configuration menu so you can change vCEPC settings. The boot loader displays the following lines (spaced by one-second intervals) before it progresses to the Network Address stage:

```
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 you press the space bar before the boot loader displays the final prompt, which is "`Hit space to enter configuration menu 1...`", the boot loader enters the configuration menu.



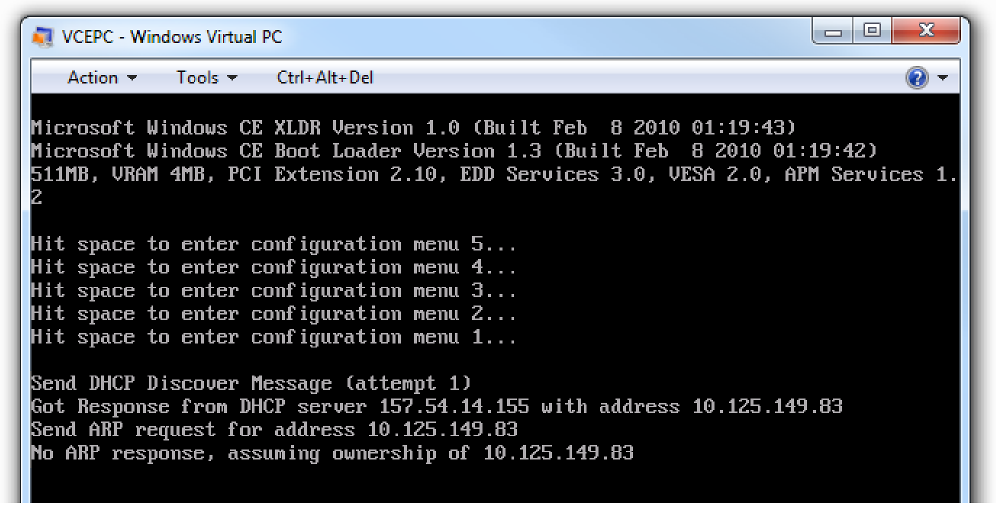
```
Microsoft Windows CE XLDR Version 1.0 (Built Feb  8 2010 01:19:43)
Microsoft Windows CE Boot Loader Version 1.3 (Built Feb  8 2010 01:19:42)
511MB, VRAM 4MB, PCI Extension 2.10, EDD Services 3.0, VESA 2.0, APM Services 1.
2

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...
```

Figure 3 - Configuration Menu Stage

You can use the configuration menu to select a different boot source, change network settings, change display settings, or modify debug port settings. For more information about the configuration menu, see "Modifying Configuration Settings" in the companion article Advanced Virtual CEPC (http://go.microsoft.com/fwlink/?LinkID=206041).

Network Address Stage

After it completes the configuration menu stage, the boot loader enters the network connection stage. During the network connection stage, the boot loader attempts to contact a Dynamic Host Configuration Protocol (DHCP) server on your local network to request an IP address to use for your vCEPC session (see Figure 4).



```
Microsoft Windows CE XLDR Version 1.0 (Built Feb  8 2010 01:19:43)
Microsoft Windows CE Boot Loader Version 1.3 (Built Feb  8 2010 01:19:42)
511MB, VRAM 4MB, PCI Extension 2.10, EDD Services 3.0, VESA 2.0, APM Services 1.
2

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...

Send DHCP Discover Message (attempt 1)
Got Response from DHCP server 157.54.14.155 with address 10.125.149.83
Send ARP request for address 10.125.149.83
No ARP response, assuming ownership of 10.125.149.83
```

Figure 4 - Network Address Stage

If the boot loader displays an error message after the last configuration menu prompt, (instead of “Send DHCP Discover Message …”) then your boot loader could be configured incorrectly. See “Troubleshooting” later in this article for a description of this problem and a way to solve it.

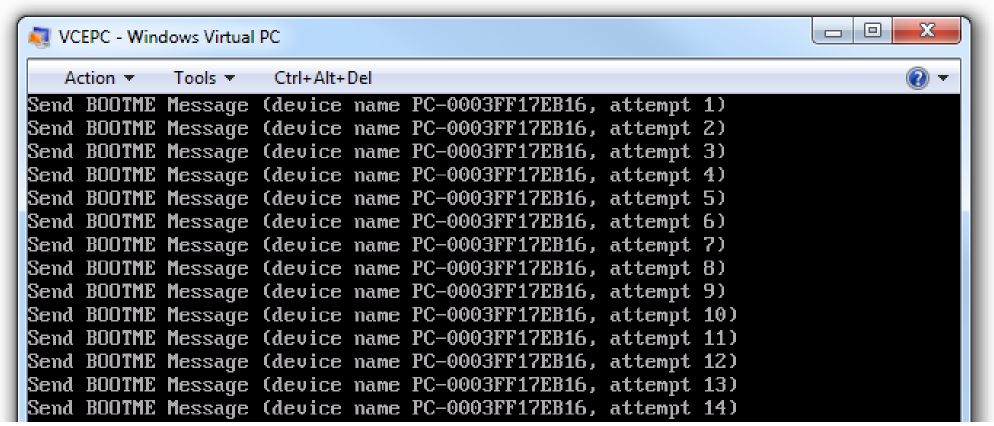
If your computer is not connected to a network with a DHCP server, the boot loader will be unable to acquire an IP address for communication with Platform Builder (see “DHCP Time-Out” later in this article). In this case, you can configure your vCEPC to connect to Platform Builder and boot without a network. See “Using Virtual CEPC Without a Network” in the companion article Advanced Virtual CEPC).

BOOTME Stage

After the boot loader acquires an IP address from a DHCP server, it transmits “BOOTME” messages on the network (see Figure 5). The boot loader broadcasts these messages in an attempt to connect to your Platform Builder session for downloading your OS image. The boot loader displays a series of status messages on the console as it sends BOOTME messages on the network. For example:

```
Send BOOTME Message (device name PC-0003FF17EB16, attempt 1)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 2)
```

Each line reveals the unique *device name* of your vCEPC on the network. Look for “device name” in the console output to find the device name of your vCEPC. In the preceding example console output, the device name is “PC-0003FF17EB16.” When you attach Platform Builder to your vCEPC, you use this device name to tell Platform Builder how to connect to your vCEPC session (see the section “Download Your Boot Image to Virtual PC” later in this article for more on this process).



```
Send BOOTME Message (device name PC-0003FF17EB16, attempt 1)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 2)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 3)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 4)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 5)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 6)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 7)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 8)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 9)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 10)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 11)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 12)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 13)
Send BOOTME Message (device name PC-0003FF17EB16, attempt 14)
```

Figure 5 - BOOTME Stage

Download and Boot Your OS Image

After you have created an OS design and a virtual machine, you are ready to prepare Platform Builder for OS download, start the boot loader, and download your boot image into Virtual PC.

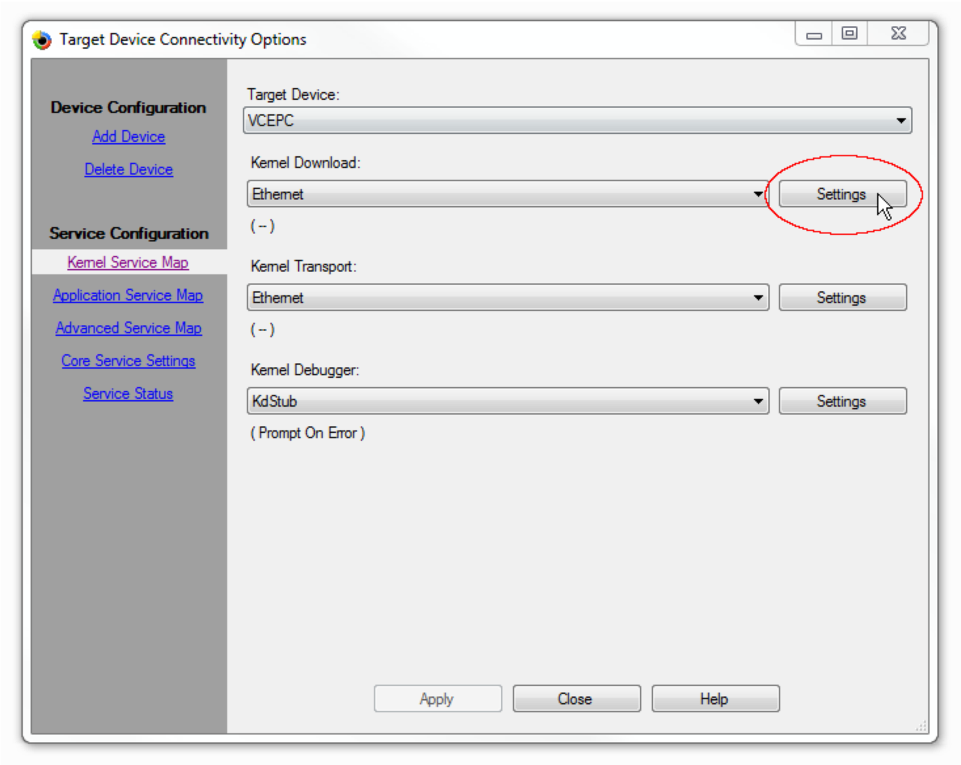
Prepare Platform Builder to Download Your Boot Image to Virtual PC

Platform Builder downloads your OS design image to a vCEPC boot loader running in Virtual PC. After the boot loader transfers the OS image from Platform Builder to Virtual PC, it boots this image in Virtual PC.

To prepare Platform Builder for download to Virtual PC

1. In Visual Studio, click the **Target** menu, and then click **Connectivity Options**.
2. In the **Target Device Connectivity Options** dialog box, click **Add Device** and type a name for your vCEPC in the **New target device name** box. You can use the name that you chose earlier for your virtual machine. For example, you could use “VCEPC” as the name of your vCEPC. For **Associated OS Design/SDK**, select **Windows CE**, and then click **Add**.
3. Select the **Kernel Service Map** service configuration if it is not already selected. In the **Target Device** list, select the name you chose for your device (for example, “VCEPC”).
4. In the **Kernel Download** list, select the **Ethernet** option.
5. Click the **Settings** button under and to the right of **Kernel Download** (as shown in Figure 6) to open the **Ethernet Download Settings** dialog box.

> **Note** If you see a Windows Security Alert dialog box with the message, “Windows Firewall has blocked some features of this program”, you can click **Allow Access** to allow the Platform Builder CE Service Host to communicate with domain and private networks.



Target Device: VCEPC

Kernel Download: Ethernet

(--)

Kernel Transport: Ethernet

(--)

Kernel Debugger: KdStub

(Prompt On Error)

Figure 6 - Target Device Connectivity Options

The **Ethernet Download Settings** box initially appears as shown in the following figure. While the **Ethernet Download Settings** box is displayed, Platform Builder listens for BOOTME messages on the local network. As described earlier, these messages are connection requests that the vCEPC boot loader broadcasts on the network in an attempt to connect to Platform Builder.

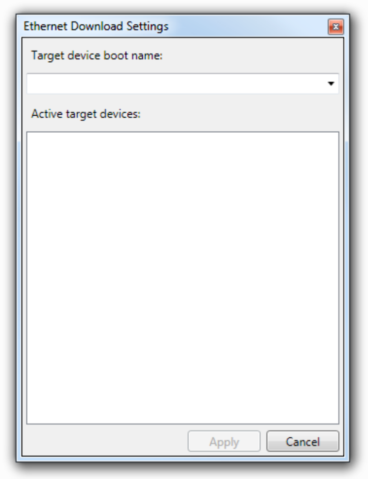
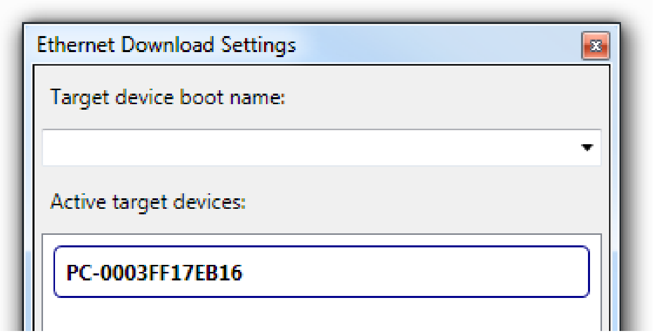


Figure 7 - Ethernet Download Settings

When Platform Builder receives a BOOTME message from a Virtual CEPC boot loader, it prints the device name of that vCEPC in the **Ethernet Download Settings** box (the device name is also called the "Target device boot name") as shown in the next figure. The device name uniquely identifies a Platform Builder target device on the network. If there are other Virtual CEPCs (or CEPCs) on your network, you may see multiple device names appear in the **Ethernet Download Settings** box.



Ethernet Download Settings

Target device boot name:

Active target devices:

PC-0003FF17EB16

Figure 8 - Target Device

You will connect your vCEPC to Platform Builder by clicking the device name for your vCEPC after it appears in this box. Leave the **Target Device Connectivity Options** and **Ethernet Download Settings** dialog boxes open as you proceed through the remaining steps.

Start the Virtual CEPC Boot Loader

After you have prepared Platform Builder for boot image download, you start your vCEPC and use its boot loader to download your OS image from Platform Builder.

> **Important** Before you start your vCEPC, keep in mind that you have only a few seconds to enter the boot loader configuration menu if you want to change vCEPC settings, and you have only a minute or so to connect Platform Builder to the vCEPC boot loader before the boot loader "times out" and stops sending connection requests on the network.

If you are unable to connect Platform Builder to your vCEPC in time, you can simply restart your vCEPC and try again. For more on restarting your vCEPC, see the section "Restarting the Boot Loader" later in this article.

Note that this manual coordination between Platform Builder and your vCEPC is required only when you are establishing a connection between Platform Builder and your vCEPC for the first time. After Platform Builder makes an initial connection to your vCEPC and knows its boot name, it can automatically synchronize with the vCEPC boot loader and re-download your boot image to the vCEPC whenever the vCEPC restarts.

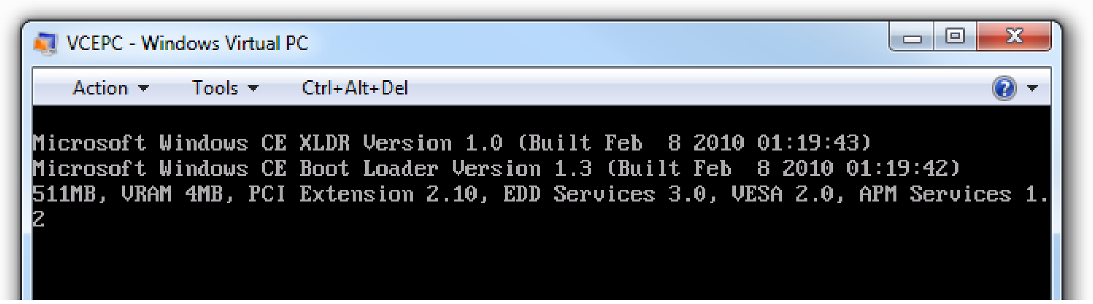
Step by step instructions follow.

To start Virtual CEPC in Virtual PC 2007

1. Click the **Start** button in the Virtual PC 2007 console.
2. Verify that Virtual PC 2007 starts and displays boot loader messages in the Virtual PC 2007 console. See Figure 9 for an example of boot loader output.

To start Virtual CEPC in Windows Virtual PC

1. Double-click the virtual machine configuration file (for example, VCEPC.vmcx) or virtual machine file (for example, VCEPC.vmc) for your vCEPC.
2. Verify that Virtual PC starts and displays the boot loader in the Windows Virtual PC console as shown here.



```
Microsoft Windows CE XLDR Version 1.0 (Built Feb  8 2010 01:19:43)
Microsoft Windows CE Boot Loader Version 1.3 (Built Feb  8 2010 01:19:42)
511MB, VRAM 4MB, PCI Extension 2.10, EDD Services 3.0, VESA 2.0, APM Services 1.
2
```

Figure 9 - Virtual CEPC Boot Loader

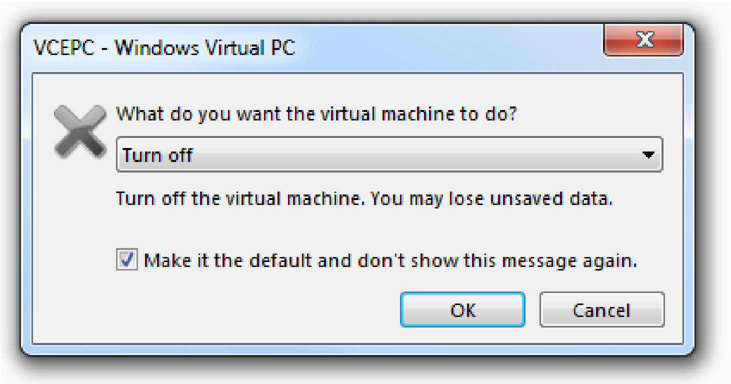
After you start Virtual CEPC, the boot loader progresses through each of its boot stages as described in the section “The Virtual CEPC Boot Loader.” If the boot loader halts with an error message, see the “Troubleshooting” section later in this article for information about how to resolve the error.

Restarting the Boot Loader

If you are unable to connect Platform Builder to the vCEPC and download your boot image before the boot loader displays the last ‘BOOTME’ message, you can simply restart your vCEPC and try again. This causes the boot loader to start over.

To restart your vCEPC

1. Click **Action** on the Windows Virtual PC console and click **Close** (In Virtual PC 2007, click **Reset**).
2. If Windows Virtual PC presents a dialog box asking you the question, **What do you want the virtual machine to do?** (see Figure 10), you can choose the option **Turn off**. Because your vCEPC is booting an existing virtual hard disk image without making any changes to it, you can turn off your vCEPC without losing any unsaved data.
3. If you want Virtual PC to automatically turn off your vCEPC on subsequent closes, you can select **Make it the default and don’t show this message again** (see Figure 10) so that Virtual PC will automatically turn off your vCEPC when you close it again.



VCEPC - Windows Virtual PC

What do you want the virtual machine to do?

Turn off

Turn off the virtual machine. You may lose unsaved data.

☑ Make it the default and don't show this message again.

Figure 10 - Closing the Virtual PC Console

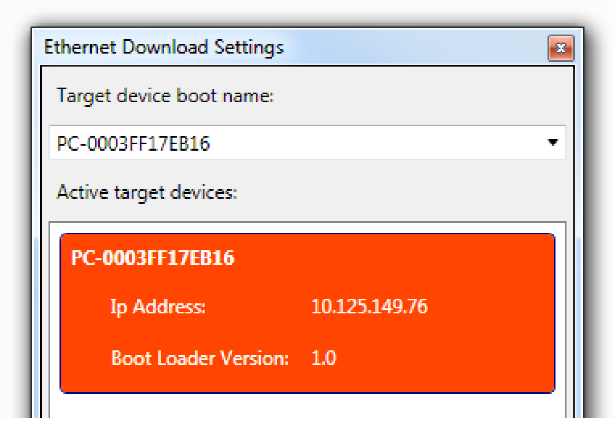
4. Click OK.
5. Double-click your virtual machine file (for example, VCEPC.vmcx) to restart the boot loader in a new Windows Virtual PC console.

Connect Platform Builder to Virtual PC

You use the Ethernet Download Settings dialog box that you left open earlier to connect Platform Builder to the Virtual PC session that is running your vCEPC.

To connect Platform Builder to the running Virtual PC session

1. Under Active target devices, look for the name that corresponds to the “device name” displayed by the boot loader in the “BOOTME” messages on the Virtual PC console.
2. When you see the device name for your vCEPC appear under Active target devices, click that device name. For example, if the console “BOOTME” messages display your device name as “PC-0003FF17EB16”, click the device name PC-0003FF17EB16 under Active target devices.



Ethernet Download Settings

Target device boot name:

PC-0003FF17EB16

Active target devices:

PC-0003FF17EB16

Ip Address: 10.125.149.76

Boot Loader Version: 1.0

Figure 11 - Connected Target Device

The selected device name becomes highlighted with a red background in the Ethernet Download Settings window and the boot name appears under Target device boot name as shown.

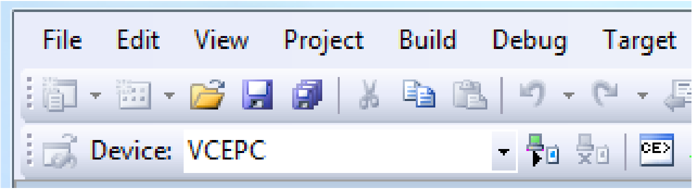
3. Click **Apply**. You have now set up a communications link between Platform Builder and your vCEPC.
4. Click **Close** in the **Target Device Connectivity** dialog box.

Download Your Boot Image to Virtual PC

To initiate download of your boot image to Virtual PC, you must first attach Platform Builder to your running vCEPC as the target device.

To download your boot image to Virtual PC

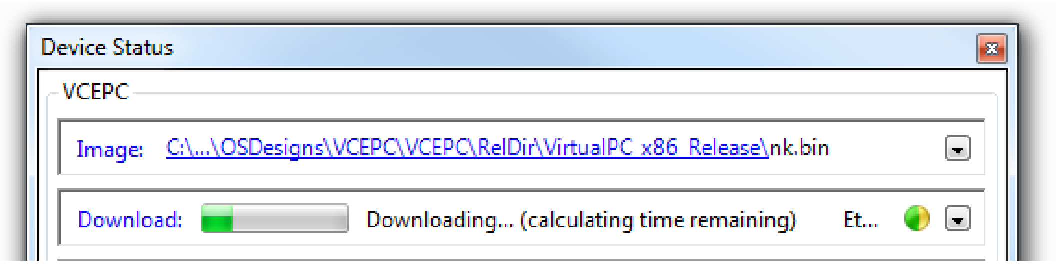
1. In Visual Studio, select your vCEPC in the **Device** list. For example, if you named your vCEPC “VCEPC”, you will see **VCEPC** in this list. Note that this name is not the same as the “device name” that is manufactured and displayed by the boot loader.



Device: VCEPC

Figure 12 - Device List

2. In Visual Studio, click **Target** and click **Attach Device** to initiate the boot image download to the Virtual PC.
3. Verify that the download has started. When Platform Builder downloads your boot image to the boot loader running in Virtual PC, it reports download progress in a **Device Status** window as shown this figure.


Device Status

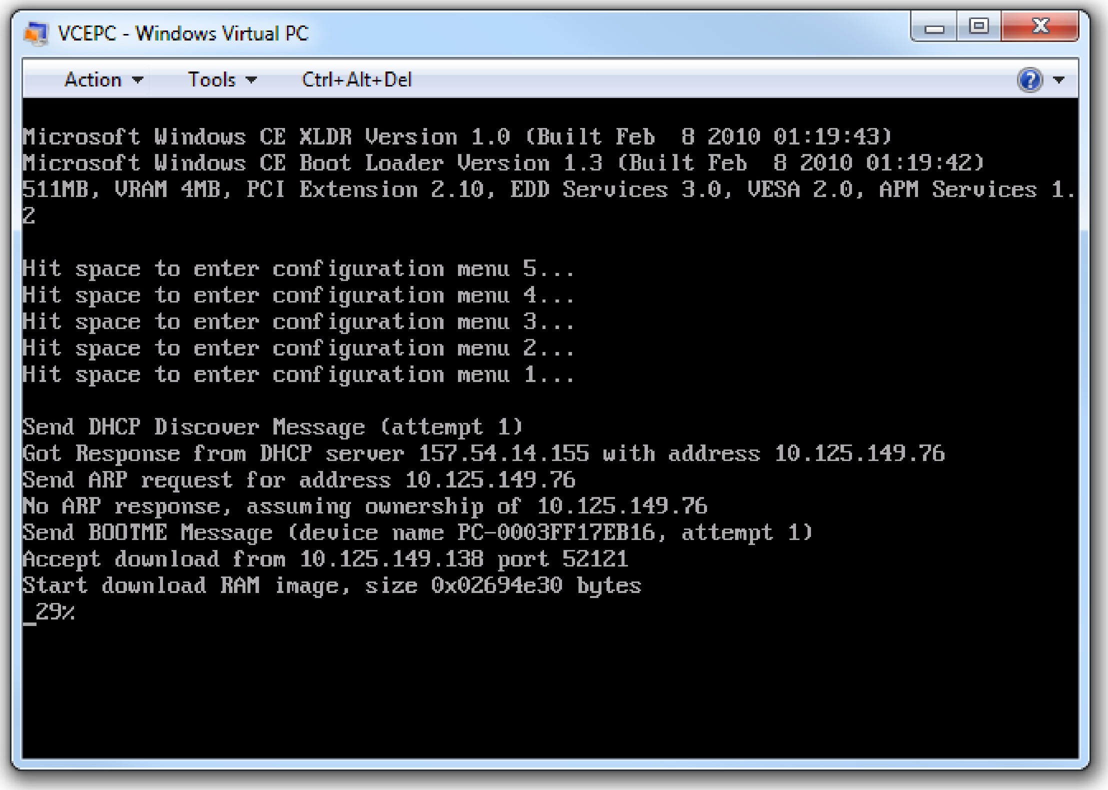
VCEPC

Image: C:\...\OSDesigns\VCEPC\VCEPC\RelDir\VirtualPC_x86_Release\nk.bin

Download: Downloading... (calculating time remaining) Et...

Figure 13 - Platform Builder Download Status

The boot loader in the Virtual PC console also provides download status. For example, you will see a percentage complete indication on the Virtual PC console as shown in the following figure.



```
Microsoft Windows CE XLDR Version 1.0 (Built Feb  8 2010 01:19:43)
Microsoft Windows CE Boot Loader Version 1.3 (Built Feb  8 2010 01:19:42)
511MB, VRAM 4MB, PCI Extension 2.10, EDD Services 3.0, VESA 2.0, APM Services 1.
2

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...

Send DHCP Discover Message (attempt 1)
Got Response from DHCP server 157.54.14.155 with address 10.125.149.76
Send ARP request for address 10.125.149.76
No ARP response, assuming ownership of 10.125.149.76
Send BOOTME Message (device name PC-0003FF17EB16, attempt 1)
Accept download from 10.125.149.138 port 52121
Start download RAM image, size 0x02694e30 bytes
_29%
```

Figure 14 - Virtual PC Console Download Status

When the OS image download finishes, the boot loader boots your image in the Virtual PC window, displaying the Windows Embedded Compact desktop in the Virtual PC console as shown in Figure 15. Now you can interact with Windows Embedded Compact 7 in a virtual machine running on your development computer.

Input Focus and Virtual CEPC

When you click inside the Virtual PC window, Virtual PC transfers your keyboard and mouse focus to the virtual machine running Virtual CEPC. When you move the mouse pointer outside of the vCEPC desktop, Virtual PC automatically transfers keyboard and mouse focus back to your host computer’s desktop. In some cases, you may have to use a key combination to release keyboard and mouse focus back to your host computer. To restore keyboard and mouse focus back to your host computer, use the release key combination that applies to your version of Virtual PC as shown in Table 1.

Table 1 – Key Sequence to Restore Keyboard and Mouse Focus

Virtual PC	Restore Keyboard and Mouse Focus
Virtual PC 2007	Right Alt + Keypad Enter
Windows Virtual PC	Ctrl + Alt + Left Arrow

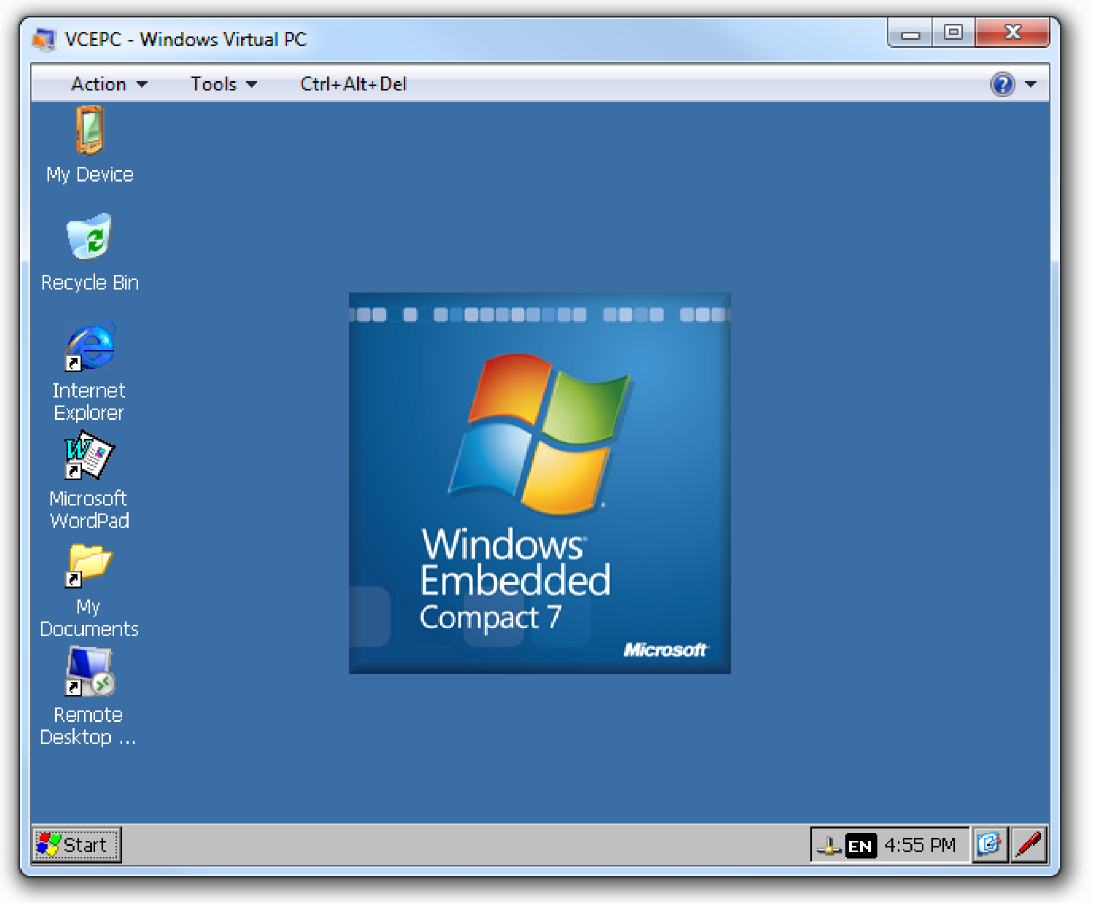


Figure 15 - Running Virtual CEPC

Shutting Down Virtual CEPC

When you are done with your vCEPC, you can shut it down and exit the Virtual PC application.

To shut down your vCEPC

1. Click **Action** on the Windows Virtual PC console and click **Close** (In Virtual PC 2007, click **Close**).
2. If Windows Virtual PC presents a dialog box asking you the question, **What do you want the virtual machine to do?** (see Figure 10), you can choose the option **Turn off**. Because your vCEPC is booting an existing virtual hard disk image without making any changes to it, you can turn off your vCEPC without losing any unsaved data.
3. If you want Virtual PC to automatically turn off your vCEPC on subsequent closes, you can click the checkbox next to **Make it the default and don't show this message again** so that Virtual PC will automatically turn off your vCEPC when you close it again.
4. Click **OK**.

Synchronizing Platform Builder and Virtual CEPC

Virtual CEPC includes an automatic download capability that makes it easier for you to synchronize your vCEPC with Platform Builder whenever your vCEPC restarts. When you create the first connection between Platform Builder and your vCEPC (as described previously), Platform Builder remembers the target device boot name of your vCEPC. Whenever you restart your vCEPC, the vCEPC's boot loader broadcasts BOOTME messages with this device name on the network. Because Platform Builder already knows about this device name, it can automatically initiate the download of the OS image to your vCEPC when it receives these BOOTME messages.

This capability simplifies coordination between Platform Builder and your vCEPC on subsequent restarts. For example, you can click the Visual Studio **Target** menu and click **Attach Device** before you start your vCEPC: once Platform Builder sees the first BOOTME message from your vCEPC's boot loader, it automatically downloads the OS image to your vCEPC. Similarly, if you restart your vCEPC while it is attached to Platform Builder, Platform Builder remembers the attached state and automatically re-downloads your OS image to your vCEPC when it receives a BOOTME message. To prevent Platform Builder from re-downloading the OS image to your vCEPC on vCEPC restart, detach the vCEPC from Platform Builder by clicking the Visual Studio **Target** menu and clicking **Detach Device.**

Develop and Run Your Application

After you have booted your vCEPC, you can create your application and run it on your vCEPC. In this section, you will learn how to create a simple Windows Embedded Compact application that displays the target device boot name that your vCEPC uses to connect to Platform Builder.

Create a Platform Builder Subproject

The easiest way to get your Windows Embedded Compact application started is to develop it as a Platform Builder subproject. Use the following steps to create an initial Platform Builder subproject for your application.

To create a subproject in your OS design

1. Open your OS design project in Platform Builder, such as the VCEPC project that you created earlier.
2. On the **Project** menu, click **Add New Subproject**.
3. When the Subproject Wizard displays a list of available templates, choose the **WCE Application** template.
4. In the **Subproject name** box, enter the name of your application. For example, you can use the name “MyBootName” for this example application. In the **Location** box, you can enter a new location for your subproject or use the path that is provided. Click **Next**.
5. On the **Auto-generated subproject files** page, select **A simple application** and click **Finish**.

After you create this new subproject in your OS design, you can compile and link the generated application template to run on your vCEPC.

To build your subproject

1. In the **Solution Explorer** pane, expand **Subprojects** to view all subprojects.
2. Locate and right-click the subproject that you created for your application, then click **Build**.
3. Verify that there are no build errors in the output display window. For example, if you named your subproject “MyBootName,” the output window should display the following messages:

```
C:\WINCE700\OSDesigns\VCEPC\VCEPC\MyBootName\sources - 0 error(s), 0 warning(s)
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

After you successfully create and build your initial subproject, you can add functionality to create your application.

Add Your Application Code

When you use the Subproject Wizard to create a simple application template, it creates a .cpp source file with an empty **WinMain** function – you use this file as the starting point for developing your application. If you choose “MyBootName” for your project name, the Subproject Wizard creates the source file MyBootName.cpp with a few lines of template code:

```
#include "stdafx.h"

int WINAPI WinMain(HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPTSTR    lpCmdLine,
                   int       nCmdShow)
{
    // TODO: Place code here.
    return 0;
}
```

In the following steps, you will replace this code template with sample application code that retrieves the boot name of your vCEPC and displays this boot name in a small window on your vCEPC desktop.

To create the MyBootName sample application

1. In the **Solution Explorer** pane, expand the **MyBootName** subproject that you created earlier.
2. Locate the **Source files** folder and expand it to reveal the source file MyBootName.cpp as shown here:

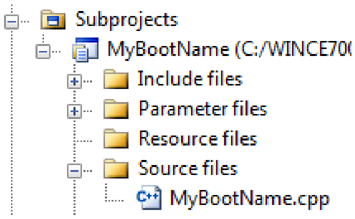


Figure 16 - Subproject Tree

3. Double-click **MyBootName.cpp** to open it, then replace the contents of this file with the following code sample:

```
#include "stdafx.h"

// The main window class name.

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

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

TCHAR szTitle[] = TEXT("Boot Name");

// 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 140x60 in size and there is no parent window or menu.

   HWND hWnd = CreateWindow (
         szWindowClass, szTitle, WS_VISIBLE | WS_CAPTION,
         400, 150, 140, 60, 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 szBootName[128] = L"";

   switch (message)
   {
        case WM_PAINT:

            // Get the boot name and display it:

            SystemParametersInfo (SPI_GETBOOTMENAME,
                    sizeof(szBootName), szBootName, 0);
            hdc = BeginPaint(hWnd, &ps);
            RECT rt;
            GetClientRect(hWnd, &rt);
            DrawText(hdc, szBootName,
                    _tcslen(szBootName), &rt, DT_CENTER);
            EndPaint(hWnd, &ps);
            break;

        case WM_DESTROY:
            PostQuitMessage(0);
            break;

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

4. Click the **File** menu and click **Save All** to save your changes.

Build and Run Your Application

To build MyBootName

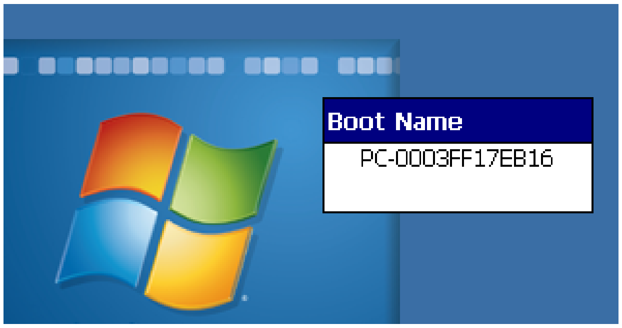
1. Locate and right-click the **MyBootName** subproject that you created earlier, then click **Build**.
2. Verify that there are no build errors in the output display window. For example, the output window should display the following messages:

```
C:\WINCE700\OSDesigns\VCEPC\VCEPC\MyBootName\sources - 0 error(s), 0 warning(s)
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

After you successfully create and build MyBootName.exe, you can start it and see it run on your vCEPC desktop.

To run MyBootName.exe on your vCEPC

1. If your vCEPC is not already booted and connected to Platform Builder, follow the steps in the section, “Download and Boot Your OS Image” to start your vCEPC and prepare it for running your application.
2. Click the **Target** menu in Platform Builder and select **Run Programs**, then select **MyBootName.exe**.
3. Verify that your application starts and runs successfully. When MyBootName runs successfully, it prints the boot name of your vCEPC in a small window as shown here:



Boot Name

PC-0003FF17EB16

Figure 17 - Running MyBootName.exe

To stop MyBootName, right-click **Boot Name** on the vCEPC task bar and select **Close**.

Using the Shared Release Folder

When you double-click **My Device** on your vCEPC desktop, you see several folders, including the Release folder, as shown in the following figure.

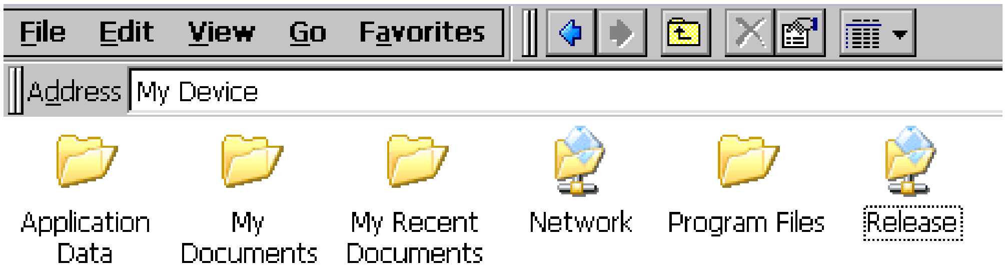


Figure 18 - Virtual CEPC Release Folder

As shown here, the Release folder is a network share – Platform Builder exports this share from your host computer. If you are using "VCEPC" for the name of your OS design, Platform Builder exports the following path on your host computer to create the Release network share:

C:\WINCE700\OSDesigns\VCEPC\VCEPC\RelDir\VirtualPC_x86_Release

On your vCEPC, this shared folder is available at the path:

\Release

Note that this Windows Embedded Compact path is not prefixed with a drive letter.

You can use this shared folder to transfer files to and from your vCEPC. For example, you can copy media files into this folder for testing media applications on your vCEPC, or you may have output files generated by your application that you want to transfer back to your host computer for examination.

In the previous example, you started your application from the **Target** menu in Platform Builder. You can also run your application from the vCEPC command prompt, which includes \Release in its PATH environment variable.

To run MyBootName from the command prompt on your vCEPC

1. Click **Start** on your vCEPC desktop, click **Programs**, and then **Command Prompt**.
2. Type **MyBootName** at the command prompt and press Enter.

This command starts MyBootName.exe from the \Release path on your vCEPC. When you build your application in Platform Builder, the build process copies the final executable to the VirtualPC_x86_Release folder on your host computer; therefore, this executable is also available in the \Release path on your vCEPC. When you start MyBootName.exe on your vCEPC, you are actually starting MyBootName.exe from the VirtualPC_x86_Release folder on your host computer.

Troubleshooting

The following sections provide solutions to common problems that you may experience while trying to get your vCEPC to boot and connect to Platform Builder.

No Boot Loader Found

When Virtual PC starts, it displays the following error message if it is unable to find a boot device with a valid boot loader:

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

This error typically occurs when you do not configure Virtual PC with either a virtual hard disk image or a virtual floppy disk image that contains a boot loader.

To remedy this problem, configure your virtual machine with the VHD image included with Windows Embedded Compact as described in the section “Create a Virtual Machine,” or configure your virtual machine with a virtual floppy disk image as described in the companion article Advanced Virtual CEPC (http://go.microsoft.com/fwlink/?LinkID=206041).

No Bootable OS Found

When the boot loader starts, it displays the following error message if it is unable to find an OS image to boot:

```
ERROR: No bootable OS found!

*** Setting to lower power state ***

*** HALT ***
```

This error typically occurs when you use the VHD image provided with Windows Embedded Compact (which does not contain an OS image) and your boot source is configured as a boot drive rather than a network device.

To remedy this problem, use the following steps to configure your Virtual CEPC boot loader to boot from the network device so that it can load your OS design image from Platform Builder.

To configure the boot loader to use the network as the boot source

1. Reboot your Virtual CEPC.
2. When you see the configuration menu prompts appear on the Virtual PC console, press the space bar to open the configuration menu.
3. The boot loader presents a **Main Menu** of configuration options:

   ```
   [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
   ```

   Type **2** to select a boot source.
4. The **Select Boot Device** menu lists two devices to boot from, a boot drive or a network device:

   ```
   [1] Boot Drive
   [2] DEC21140 at PCI bus 0 dev 10 fnc 0
   ```

Type **2** to select the network device shown. The network device, for example, DEC21140, is not an actual physical device on your computer. It is a simulated network adapter that the vCEPC uses for sharing the physical network adapter of your host computer. After you select the network device, the boot loader confirms your selection. For example:

```
Boot Device is set to DEC21140 at PCI bus 0 dev 10 fnc 0
```

5. On the **Main Menu**, type **1** to display the boot loader settings. The settings display includes information about the boot source, the display, kernel independent transport layer (KITL), dynamic host configuration protocol (DHCP), and the debug port. For example:

   ```
   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
   ```

   Verify that the boot source is set to the network device rather than the boot drive. In addition, verify that VMINI is enabled and that the network device is configured for KITL as shown in this example. These settings are enabled by default so that the vCEPC can access the network and so that you can use debugging tools and access the vCEPC from your development computer.

6. Type **7** to save the settings so that you don't have to change the boot source to the virtual network device on every subsequent boot of the vCEPC.
7. Type **0** to exit and continue; this action begins the boot process.

If you do not want to boot your OS image over the network device from Platform Builder, you can create a stand-alone VHD image that contains your OS design image and a boot loader. For more about creating a stand-alone VHD image, see the companion article Advanced Virtual CEPC (http://go.microsoft.com/fwlink/?LinkID=206041).

DHCP Time-Out

The Virtual CEPC boot loader sends 26 DHCP discover messages before it aborts the DHCP server discovery process. When this time-out occurs, you will see an error message in the Virtual PC console similar to the following:

```
Send DHCP Discover Message (attempt 26)
ERROR: BootTransportPb: Failed get IP address from DHCP server!
ERROR: BootDownloadBin!ImageType: BootTransportRead Failed!
```

This error means that your host computer is not attached to a network or it is attached to a network without a DHCP server. To use Virtual CEPC on a host computer without a network connection or a DHCP server, see "Using Virtual CEPC Without a Network" in the companion article Advanced Virtual CEPC (http://go.microsoft.com/fwlink/?LinkID=206041).

Boot Time-Out

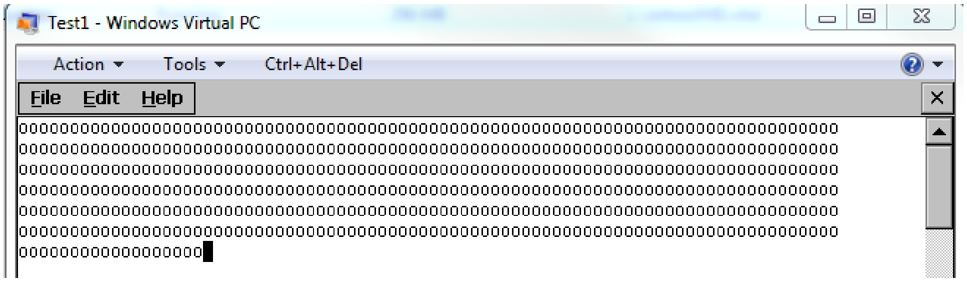
The vCEPC boot loader sends sixty "BOOTME" messages before it aborts the boot process. When this time-out occurs, you will see an error message in the Virtual PC console similar to the following:

```
Send BOOTME Message (device name PC-0003FFC67ACA, attempt 60)
ERROR: BootDownloadBin!ImageType: BootTransportRead Failed!
```

If you are unable to initiate the boot image download from Platform Builder before the boot loader times out and prints this message, you can restart the vCEPC to retry the boot loading process. See "Restarting the Boot Loader" earlier in this article for information about restarting the vCEPC.

Corrupted Command Prompt Input

After you successfully boot your vCEPC and open a Command Prompt window on the Virtual CEPC desktop, you can type commands in the same way that you would in a Command Prompt window on your host computer. If you see corrupted input appear on the Virtual CEPC command console (or, when opening the Command Prompt window, you see lines of zeroes as shown in the following figure), then you do not have the correct keyboard driver installed in your OS design.



```
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000
```

Figure 19 - Wrong Keyboard Driver

To remedy this problem, verify that you have selected the correct keyboard driver as described in the section "Select a Virtual CEPC Keyboard Driver" and rebuild your OS design.

Conclusion

You can configure and boot a virtual CEPC (vCEPC) for use as a Windows Embedded Compact 7 development platform. To do so, you create a Platform Builder project and OS design for the vCEPC, select catalog items and a keyboard driver for your vCEPC OS design, and build an image that runs on the vCEPC. You also configure Virtual PC with a virtual machine and a VHD image for Windows Embedded Compact 7. You connect Platform Builder to your running Virtual PC session so that you can download your OS image to Virtual PC and boot it in a virtual machine. With your vCEPC booted and connected to Platform Builder, you can develop and build your application in a Platform Builder subproject and run it on your vCEPC. You can also copy files to and from your vCEPC through a folder that is shared between your vCEPC and your host computer.

With Virtual CEPC, you can develop, test, and debug Windows Embedded Compact applications entirely on your development computer without having to purchase, install, and configure special hardware.

Additional Resources

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

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

Copyright