



Windows Embedded Standard 2011

CTP2 Lab Manual

Table of Contents

1	Preparing to Use Windows Embedded Standard 2011	5
1.1	Introduction	5
1.2	Related Windows 7 Documentation	5
1.3	Minimum Device Hardware Requirements	5
1.4	Release Notes	5
1.5	Install Windows Embedded Standard 2011 Toolkit	6
1.5.1	Open Windows Embedded Standard 2011 Toolkit Help	6
1.6	Create a Utility Drive	7
1.6.1	Make Your Utility Disk Bootable	7
1.6.2	Add Image Builder Wizard to Your Utility Disk	7
1.7	Create a Windows Preinstallation Environment Utility Drive (Optional)	8
1.7.1	Create a Windows PE Utility Disk with Image Configuration Editor	8
1.7.2	Create a Custom Windows PE Image	8
2	Build Windows Embedded Standard 2011 Images	9
2.1	Build a Windows Embedded Standard 2011 Image with Image Builder Wizard	9
2.1.1	Build a Thin Client Image with Embedded Enabling Features	9
2.1.2	Build an Image with Embedded Core Only	11
2.2	Build a Windows Embedded Standard 2011 Image Using Image Configuration Editor	11
2.2.1	Build a Thin Client Image with Custom Settings	11
2.2.2	Add Update Packages to a Distribution Share (Optional)	14
2.2.3	Add Third-Party Software (Optional)	15
2.2.4	Run a Custom Script (Optional)	16
2.2.5	Add Third-Party Drivers (Optional)	17
2.2.6	Make Your Install Fully Unattended (Optional)	17
2.2.7	Create Image Builder Wizard Disk from Answer File (optional)	24
3	Deploy an Image	25
3.1	Prepare and Capture an Image for Deployment	25
3.1.1	Sysprep the Image	25
3.1.2	Capture the Image Into a Windows Image (.wim) File	25
3.2	Deploy an Image Using ImageX	26
3.2.1	Prepare your Destination Device	26
3.2.2	Apply an Image to Your Device's Hard Drive	27

3.2.3	Shut Down the Destination Device	27
3.3	Deploy an Image Using Image Builder Wizard.....	27
3.3.1	Start Image Builder Wizard with your Image	28
3.3.2	Customize a Captured Image.....	29
4	Service an Image	31
4.1	Service an Image with Image Configuration Editor and DISM.....	31
4.1.1	Create a Configuration Set using Image Configuration Editor.....	31
4.1.2	Install using your Configuration Set	32
4.1.3	Test your image	34
4.2	Service an Image with DISM	34
4.2.1	Mount your image	34
4.2.2	Add sample packages.....	34
4.2.3	Adding a Package Using DISM	35
4.2.4	Unmount and Commit Changes.....	35
4.2.5	Redeploy and Verify	35
4.2.6	Add a Language Pack (optional).....	35
4.3	Install Updates with WUSA.....	36
4.4	Service an Image with Package Scanner	36
4.4.1	Package Enumeration.....	37
4.4.2	Find Applicable Updates.....	37
4.4.3	Scavenging.....	37
5	Additional Windows Embedded Standard 2011 Labs.....	39
5.1	Suppress OOBE	39
5.1.1	Create an Answer File	39
5.1.2	Using your answer file with Image Builder Wizard	39
5.2	Using a Custom Shell and Custom Branding.....	40
5.2.1	Hide boot screens	40
5.2.2	Use Shell Launcher	40
5.2.3	Remove Windows branding	41
5.2.4	Replace the startup screen background image	41
5.2.5	Add Message Blockers.....	42
5.3	Build a USB bootable Windows Embedded 2011 image.....	44
5.3.1	Install from Image Builder Wizard directly to UFD :	44
5.3.2	Install from Image Builder Wizard to Hard Drive, ImageX to UFD:.....	44
5.4	Create Custom Templates for IBW	46

5.4.1	Creating Templates	46
5.4.2	Using Custom Templates in IBW	46

1 Preparing to Use Windows Embedded Standard 2011

1.1 Introduction

Welcome to Windows Embedded Standard 2011. This lab manual is a guide to help you use and evaluate Windows Embedded Standard 2011. In addition to preparation steps, such as toolkit installation, this manual includes labs for key scenarios such as building an image, deploying an image and servicing an image.

The labs are most easily followed in the order presented. You can also choose labs individually based on your interests and previous experience with Windows Embedded, but please note that several lab scenarios assume you have the output of an earlier lab scenario.

1.2 Related Windows 7 Documentation

Windows Embedded Standard 2011 is based on Windows 7; therefore much of the Windows 7 documentation can be used as reference material.

- Windows 7 Automated Installation Kit (Windows AIK) online documentation <http://www.microsoft.com/downloads/details.aspx?FamilyID=f1bae135-4190-4d7c-b193-19123141edaa&displaylang=en>
- Windows 7 Technical Library on Microsoft TechNet <http://technet.microsoft.com/en-us/library/dd349342.aspx>
- Windows Developer Center for Windows 7 <http://msdn.microsoft.com/en-us/windows/dd433113.aspx>

1.3 Minimum Device Hardware Requirements

You must have the following minimum hardware to be able to build a Windows Embedded Standard 2011 image on your device:

- 1 GHz x86 or amd64 processor
- 1 GB of flash or hard drive space (4 GB recommended)
- 512 MB of RAM (1 GB recommended for amd64 devices)
- 900 MHz CPU or equivalent
- At least one of the following bootable media types:
 - Bootable DVD-ROM drive
 - Bootable USB 2.0 port and a USB Flash Drive (UFD) with 4 GB free space, or access to a local network
- BIOS supporting Windows Preinstallation Environment (Windows PE) 3.0
- Minimum hardware requirements, particularly RAM requirements, may be greater depending on the size and type of feature packages selected.

1.4 Release Notes

Although every attempt has been made to provide workarounds and additional usage notes for scenarios that are affected by known issues in the pre-release versions of Windows Embedded Standard 2011, we strongly recommend that you refer to the release notes provided with this release before beginning any of the labs described in this manual.

1.5 Install Windows Embedded Standard 2011 Toolkit

You can install Windows Embedded Standard 2011 Toolkit on your development computer from a DVD or from setup files downloaded from Microsoft Connect.

1. Run Setup.exe
 - If you have the Windows Embedded Standard 2011 Toolkit DVD, installation should begin when you insert the DVD into the DVD drive. If it doesn't start automatically, the Setup.exe file can be found at:

`<DVD Drive>:\WindowsEmbeddedStudio.msi.`
2. On the Setup Type page, do one of the following:
 - To install the tools and distribution share to the default location, select Complete. The default locations are:
 - 32-bit operating system: *[System Drive]:\Program Files\Windows Embedded Standard 2011*
 - 64-bit operating system: *[System Drive]:\Program Files(x86)\Windows Embedded Standard 2011*
 - To install the tools and distribution share to a different location, select Custom.
3. Follow the instructions in the installation wizard to complete the installation process.

1.5.1 Open Windows Embedded Standard 2011 Toolkit Help

The Windows Embedded Standard 2011 Toolkit Help contains more detailed information on many of the topics and steps contained in this manual.

To access the Windows Embedded Standard 2011 Toolkit Help:

1. Start Image Configuration Editor on your development computer
 - From the Start menu, click Programs, click Windows Embedded Standard 2011, and then click Image Configuration Editor.
2. On the toolbar, click the Help icon to launch Help.

1.6 Create a Utility Drive

If your device is able to boot from a DVD, you can use the Windows Embedded Standard 2011 DVD appropriate to your device's architecture to install the OS image directly to your device.

Otherwise you can create a bootable USB drive by using the DiskPart tool and then loading the USB Drive with Image Builder Wizard (IBW) or WindowsPreinstallation Environment (WinPE). Diskpart supports the partitioning and formatting of a USB Flash Device (UFD) as a bootable device. A USB drive with a minimum of 4 GB is recommended for a utility drive loaded with Image Builder Wizard.

1.6.1 Make Your Utility Disk Bootable

1. Attach your USB drive to your development computer.
2. From a Windows Vista, Windows 7 or Windows Preinstallation (Windows PE) 3.0 environment, run the DiskPart tool by typing the following at a command prompt:

```
diskpart
```

Note:

The version of the DiskPart tool provided by Windows Vista, Windows 7 and Windows PE 2.0, 2.1 and 3.0 supports the partitioning and formatting of a UFD as a bootable device. Previous versions of the DiskPart tool, including the version provided by Windows XP, do not fully support partitioning and formatting a UFD to be bootable and should not be used.

3. Use the DiskPart tool to determine the disk number and device size to be used for the next step by typing the following at the DiskPart prompt:

```
list disk
```

4. Use the DiskPart tool to partition and format the drive and make it bootable. At the DiskPart prompt, type the following, replacing <disk_number> with the disk number of the USB drive:

```
select disk <disk_number>
clean
create part pri
select part 1
active
format fs=ntfs quick
assign
exit
```

Your USB drive is now bootable.

1.6.2 Add Image Builder Wizard to Your Utility Disk

The Image Builder Wizard disks provide the ability to quickly generate a new IBW disk in the event IBW has been serviced or new packages have been added to the distribution share.

To generate an Image Builder Wizard Disk:

1. Start Image Configuration Editor on your development computer
 - From the Start menu, select **Programs**, select **Windows Embedded Standard 2011**, then select **Image Configuration Editor**.
2. On the **Tools** menu, select **Media Creation**, then select **Create IBW Disk**.
3. Enter the desired distribution share to be copied.
4. Select your USB drive as the target folder.
5. Select the desired disk architecture.
6. Click **OK**.

1.7 Create a Windows Preinstallation Environment Utility Drive (Optional)

In some instances, you may want to create a utility disk that includes the Windows Preinstallation Environment (Windows PE). Windows PE is a lightweight version of Windows used mainly for deployment. You may prefer to use Windows PE instead of Image Builder Wizard if your USB drive is smaller than 2 GB, if you have space limitations on your device or if your device requires custom drivers that you will install on a custom version of Windows PE.

1.7.1 Create a Windows PE Utility Disk with Image Configuration Editor

The Windows PE Utility Disk feature of the Media Creation tool in Image Configuration Editor provides the ability to quickly generate a Windows PE disk for gathering information about your target device or for deploying an image using Image Builder Wizard. The generated disk includes TAP.exe, ImageX and Package Scanner.

To generate a Windows PE utility disk:

1. Create a bootable utility disk (section 1.6.1 above).
2. Start Image Configuration Editor on your development computer
 - From the Start menu, select **Programs**, select **Windows Embedded Standard 2011**, then select **Image Configuration Editor**.
3. On the **Tools** menu, select **Media Creation**, then select **Create PE Image**.
4. Enter a target folder on your development computer to which the binaries will be copied.
5. Select the desired disk architecture.
6. Click **OK**.
7. Copy the contents of the ISO sub-folder from the target folder on your development computer to the root directory of your bootable utility disk.

1.7.2 Create a Custom Windows PE Image

See the topic “Create a Custom Windows PE Image” in the Windows Embedded Standard 2011 Toolkit Help for more information about creating a custom Windows PE Image and gathering other files to add to the bootable USB drive you created in section 1.6.1.

2 Build Windows Embedded Standard 2011 Images

2.1 Build a Windows Embedded Standard 2011 Image with Image Builder Wizard

Image Builder Wizard (IBW) is a tool you can use to create, configure and install Windows Embedded Standard 2011 on your device. The wizard runs directly on your device and guides you through a set of configuration choices. After you make your selections, the wizard creates and installs Windows Embedded Standard 2011 onto the device. You can further customize the installation or you can capture it to an image file and deploy it to other devices.

In this lab, you will use IBW on your device to create different Windows Embedded Standard 2011 images. You can follow the steps provided in this lab to create the following variations:

- An image that can be used for a thin client device.
- A “minboot” image that contains only the base packages (referred to as Embedded Core).

2.1.1 Build a Thin Client Image with Embedded Enabling Features

To build a thin client image using IBW:

1. Start Image Builder Wizard on your device by either:
 - Start the wizard from your Windows Embedded Standard 2011 DVD, your ISO image, or the USB drive prepared in section 1.6
 - Boot your device into Windows PE and run setup.exe from another location (USB drive, network share, etc.).

Note:

Image Builder Wizard will setup and install Windows Embedded Standard 2011. It's important to make sure you are running the wizard on your device itself and not on your development computer.

2. On the first page, select Build an Image
This option starts IBW and allows you to choose packages, drivers and languages to include in your image. You can optionally start from a template.
3. Accept the End User License Agreement (EULA).
4. The Choose the way you want to build your image page allows you to start from a template or from a blank configuration. In this exercise, select the Thin Client template and click Next.
5. Select a language, time and currency format and keyboard or input method to install on the final image. These selections are for the primary language of your image. You can add additional languages later. Click Next to continue.
6. The Summary of Drivers and Features page shows a summary of the drivers to be installed, detected devices and feature packages in your template. Check the Modify Drivers and Modify Features checkboxes and click Next.

7. On the Find and Select Drivers page, choose one of the following options and click Next to continue:
 - The Automatically detect devices option detects the drivers on your device and attempts to find drivers for them.
 - The Choose a PMQ option makes it possible for you to import a device list from a previously generated PMQ file. As with Windows Embedded XP, TAP.exe is used to generate this PMQ file.
 - The Do not select additional drivers option includes only the drivers in Embedded Core. Your image will still be bootable but drivers for non-boot-critical devices may not be installed.
8. The Confirm drivers to be installed page shows drivers that will be installed as well as the devices that were detected on your computer for which we do not have drivers. Click Browse if you want to add custom drivers. Click Next to continue.
9. On the Please select Feature Packages to include in your image page you can add additional features to your configuration. Because you started from the Thin Client template, several packages have been preselected for you. Choose any additional packages you wish to include.
10. Determine which type of write filters you want to use for your thin client and select the appropriate feature packages.

To use File Based Write Filter (FBWF) and Registry Filter, select:

 - Embedded Enabling Features\File Based Write Filter (FBWF)
 - Embedded Enabling Features\Registry Filter

OR

To use Enhanced Write Filter (EWF), Hibernate Once Resume Many (HORM) and Registry Filter, select:

 - Embedded Enabling Features\Enhanced Write Filter with HORM
 - Embedded Enabling Features\Registry Filter
 - Boot Environments\Enhanced Write Filter Boot Environment

Note:

Although EWF can be used without HORM, using HORM requires EWF.

Adding EWF in IBW (without an answer file) will configure all existing volumes to be protected in RAM-REG mode; however EWF will be disabled for all of them.
11. Click the Resolve Dependencies button. If a pop-up window asks you to choose between multiple packages, make the following selections:
 - a. If you are installing HORM, select Embedded Windows 7 Boot Environment, otherwise select the Windows 7 Boot Environment.
 - Select Windows Embedded Standard Startup Screens.
 - Select Windows Explorer.
 - Select Standard Windows USB Stack.

- b. Resolve all other dependencies, then click Done. Click Next to continue.
12. The Summary of Drivers and Features page gives you a final overview of packages and drivers to be installed. If you are satisfied with your selections, click Next.
13. Select the disk or partition where you would like to install the image, then click Next to begin installation and set up of the customized Windows Embedded Standard 2011 image on your device.
14. To enable Embedded Enabling Features after installation is complete, open a command prompt on your device and run the following commands:
 - For FBWF and Registry Filter:

```
fbwfmgr /enable  
fbwfmrg /addvolume c:
```
 - For EWF, HORM and Registry Filter:

```
ewfmgr c: -enable  
ewfmgr c: -activatehorm
```
15. Once HORM has been activated, it should be tested:
 - a. Reboot so that EWF is enabled.
 - b. Start Internet Explorer
 - c. Enable hibernation by typing the following at a command prompt:

```
powercfg -h on
```
 - d. Reboot and device should resume from hibernation.
 - e. Make changes and reboot your device again
 - f. Verify your device state returns to that of step d above and that the changes made in step e were not retained.

2.1.2 Build an Image with Embedded Core Only

A Windows Embedded Standard 2011 image that contains just the Embedded Core package is described as a “minboot” image.

To build an Embedded Core (eCore) image using Image Builder Wizard, follow the steps in section 2.1.1 Build a Thin Client Image with Embedded Enabling Features with the following changes:

1. At step 4, choose the Minimum Configuration template
2. Complete steps 10 and 15 only if you want to enable write filters.

2.2 Build a Windows Embedded Standard 2011 Image Using Image Configuration Editor

2.2.1 Build a Thin Client Image with Custom Settings

1. Start Image Configuration Editor on your development computer.
 - From the Start menu, select Programs, select Windows Embedded Standard 2011, then select Image Configuration Editor.

2. On the File menu, select Distribution Share, then navigate to the desired distribution share. The default distribution share locations are:
 - On a development computer running a 32-bit operating system:
 - x86 distribution share: *C:\Program Files\Windows Embedded Standard 2011\DS*
 - amd64 distribution share: *C:\Program Files\Windows Embedded Standard 2011\DS64*
 - On a development computer running a 64-bit operating system:
 - x86 distribution share: *C:\Program Files(x86)\Windows Embedded Standard 2011\DS*
 - amd64 distribution share: *C:\Program Files(x86)\Windows Embedded Standard 2011\DS64*
3. On the File menu, click New Answer File. By default, the Embedded Edition package is added to this new answer file. This is the Embedded Core package.
4. Add Driver Packages
 - a. On the File menu, click Import then select Import PMQ to add device drivers using a PMQ file. In the Messages pane, in the Import PMQ tab, you can review the results of mapping the devices in a PMQ file to driver packages.
 - Successfully mapped devices will be listed. To view the package in the answer file that the device was mapped to, double-click the device name.
 - The warning icons denote devices that were not mapped to driver packages. If you need support for these devices, see section 2.2.3

Note:

For more information on how to generate a PMQ file, refer to the Windows Embedded Standard 2011 Help topic “How to Generate a .PMQFile Using Target Analyzer”.

- b. In the Distribution Share pane, under Packages/Driver, you can add additional driver packages. Double-click any driver package (leaf node in the tree) to add it to your answer file.
5. Add Feature Packages
 - a. In the Distribution Share pane, expand the Packages/FeaturePack node.
 - b. Expand the Browsers/Internet Explorer 8 node, right-click on Internet Explorer 8 Browser and select Add to Answer File.
 - c. Expand the Graphics and Multimedia node, right-click on Windows Media Player 12 and select Add to Answer File.
 - d. Expand the Remote Connections node, right-click on Remote Desktop Connection and select Add to Answer File.
6. Determine which type of write filters you want to use for your thin client. In the Distribution Share pane, under Packages/FeaturesPack, right-click on the appropriate feature packages and select Add to Answer File.

To use File Based Write Filter (FBWF) and Registry Filter, select:

- Embedded Enabling Features\File Based Write Filter (FBWF)
- Embedded Enabling Features\Registry Filter

OR

To use Enhanced Write Filter (EWF), Hibernate Once Resume Many (HORM) and Registry Filter, select:

- Embedded Enabling Features\Enhanced Write Filter with HORM
- Embedded Enabling Features\Registry Filter
- Boot Environments\Enhanced Write Filter Boot Environment

Note:

Although EWF can be used without HORM, using HORM requires EWF.

Adding EWF in IBW (without an answer file) will configure all existing volumes to be protected in RAM-REG mode; however EWF will be disabled for all of them.

7. To add language packs, in the Distribution Share pane, expand the Packages/LanguagePack/en-US node. Right-click the English (US) Language Pack package and select Add to Answer File. Add additional language packs the same way.
8. To change Internet Explorer 8 settings:
 - a. In the Answer File pane, click on the Internet Explorer 8 Browser package.
 - b. In the Settings pane, change Filter View to “4 Specialize”.
 - c. In the Settings pane, click on the Value column to update the value for each of the following settings:
 - d. Set Home_Page to www.microsoft.com/embedded
 - e. Set IEWelcomeMsg to false
 - f. Save your answer file.

You have now changed the way Internet Explorer behaves by using Image Configuration Editor's settings UI. You can change additional settings in your answer file the same way.

9. To resolve dependencies, from the Validate menu, select Add Required Packages.
 - a. If there are any errors listed in the Validation tab of the Messages pane that state “Dependencies of the source package are not satisfied,” double-click the error message and use the Resolve Dependencies dialogue box to satisfy all required package dependencies.
 - If you are asked to choose between two USB stacks, choose “Bootable Windows USB Stack” only if you are using the USB Boot Embedded Enabling Feature and enabling a USB bootable image (see section 5.3). Otherwise choose “Standard Windows USB Stack.”
 - If you are asked to choose between two boot environment packages, choose “Enhanced Write Filter Boot Environment” if you are installing HORM with EWF. Otherwise choose “Windows Boot Environment”.
 - b. Warnings that state “Optional Dependencies exist for the source package” are acceptable and may be ignored.

- c. Continue until all required dependencies are resolved.

Note:

For more information on the three validation options, refer to Windows Embedded Standard 2011 Help's topic "Validating an Answer File in Image Configuration Editor".

10. Use the answer file you just created to install the thin client image on your device:
 - a. Boot your device into Image Builder Wizard using the DVD or utility disk you created in section 1.6.
 - b. On the Build an Image / Deploy an Answer File or WIM page, select Deploy an Answer File or WIM.
 - c. In the browse to file dialog box, browse to your answer file and click Open.

Note:

You can use this answer file to create a configuration set. A configuration set is a folder that contains the packages, applications, images and drivers used in an answer file. Image Builder Wizard uses the configuration set and the answer file to build or modify an installation. For more information, see the Help documentation that was installed with Windows Embedded Standard 2011.

11. To enable Embedded Enabling Features after installation is complete, open a command prompt on your device and run the following commands:

- For FBWF and Registry Filter:

```
fbwfmgr /enable
fbwfmrg /addvolume c:
```

- For EWF, HORM and Registry Filter:

```
ewfmgr c: -enable
ewfmgr c: -activatehorm
```

12. Once HORM has been activated, it should be tested:
 - a. Reboot so that EWF is enabled.
 - b. Start Internet Explorer
 - c. Enable hibernation by typing the following at a command prompt:


```
powercfg -h on
```
 - d. Reboot and device should resume from hibernation.
 - e. Make changes and reboot your device again
 - f. Verify that your device state returns to that of step d above and that changes made in step e were not retained.

2.2.2 Add Update Packages to a Distribution Share (Optional)

You can use the ImportPackage feature of Image Configuration Editor to import update packages into an existing Distribution Share. Both cabinet (CAB) files and Microsoft Update Standalone Package (MSU) files can be imported.

You can use the sample packages that were included with the pre-release build of Windows Embedded Standard 2011 Toolkit, found under the \Updates folder of the Image Builder Wizard DVD appropriate to your architecture. For more information about which updates are available, see section 4.2.2.

2.2.2.1 *To use the Import Package feature in Image Configuration Editor:*

1. Open Image Configuration Editor on your development computer.
 - On the Start menu, select Programs, select Windows Embedded Standard 2011, then select Image Configuration Editor.
2. On the File menu, click on Import, then click on Import Package(s).
3. In the Select Packages to Import dialog box, navigate to the file you want to import and click Open.

2.2.2.2 *To Import Packages from a command prompt:*

1. Open a command prompt window and navigate to: C:\Program Files\Windows Embedded Standard 2011\Tools\Image Configuration Editor
2. At the command prompt, type the following command to add the update package to the 32-bit distribution share:

```
run importpackage <updatepackage> C:\Program  
Files\Windows Embedded Standard 2011\DS
```

Note:

If your development computer is running a 64-bit operating system, the paths listed above will instead be under the C:\Program Files (x86)\ directory.

2.2.3 Add Third-Party Software (Optional)

The recommended way to add custom files to an Windows Embedded Standard 2011 runtime is by using Image Configuration Editor and the \$OEM\$ Folder.

Files placed in the \$OEM\$ Folder will be automatically copied to the embedded device during installation. The folder and file hierarchy will be preserved on the embedded device as you define it in the \$OEM\$ Folder. Details of the format of an \$OEM\$ Folder can be found in the Windows Embedded Standard 2011 Help under “Understanding Distribution Shares and Configuration Sets.”

Image Configuration Editor supports the management of multiple \$OEM\$ Folders in a distribution share. The folder hierarchy underneath “..\DS\\$OEM\$ Folders” determines how folders will be displayed in the Image Configuration Editor UI.

2.2.3.1 *Create an \$OEM\$ Folder:*

1. Right-click the distribution share path in the Distribution pane, then select Explore Distribution Share.
2. Navigate to the \$OEM\$ Folders directory and add a new sub-folder called MyApplication.
3. Under MyApplication, add a new sub-folder called \$OEM\$.

4. Under the \$OEM\$ folder you just created, add additional subfolders and files as necessary. For example, to deliver the file “foo.exe” to the root of the install partition, the full path should be: “..\DS\\$OEM\$ Folders\MyApplication\\$OEM\$\\$1\foo.exe”

Note:

You can also refer to the Microsoft Technet article, [Designing a Distribution Share](#), for additional information on the structure and parameters available for use with the \$OEM\$ Folder.

2.2.3.2 Add an \$OEM\$ Folder to an Answer File:

1. Expand the \$OEM\$ Folders node in the Distribution Share pane.
2. Right-click the MyApplication folder and select Insert OEM Folders Path.

2.2.4 Run a Custom Script (Optional)

To run a custom script during the installation process, you use the Synchronous command in Image Configuration Editor. You can use the Synchronous command to start the installation of an application, start a program on first boot, or to perform more advanced operations.

For example, the following steps will guide you through the process of finding a package that contains a specific file - in this example it's regedit.exe - and then using a .reg file to modify your registry.

2.2.4.1 Use Image Configuration Editor to search for the package that contains the file regedit.exe:

1. Open Image Configuration Editor on your development computer
 - From the Start menu, select Programs, select Windows Embedded Standard 2011, then select Image Configuration Editor.
2. On the Edit menu, click Find.
3. In the Look In group, select Distribution Share. Clear any other options.
4. In the Options group, select Search file names in packages. Clear the other options.
5. In the Find what box, type “regedit.exe” and click Find.
6. When the search is complete, look for regedit.exe in the Value column. It should be listed with a Path of System Management Utilities/Files.
7. Double-click that row and the package that contains regedit.exe is now displayed and selected in the Distribution Share pane.
8. Right-click on this package and select Add to Answer File.

2.2.4.2 Modify your registry using the Synchronous command:

1. On the Insert menu, click Synchronous Command, then select Pass 7 oobeSystem.
2. In the Enter a Command Line text box, enter the following:

```
cmd /c REGEDIT /s %windir%\mykeys.reg
```


3. Click OK

Note:

Follow the instructions in section 2.2.3 to create an \$OEM\$ Folder and add the \$OEM\$ Folder to an answer file to add the .reg file to your answer file.

2.2.5 Add Third-Party Drivers (Optional)

You can add device drivers during several phases of installation. To add third-party drivers using Image Builder Wizard with an answer file, you must include a device-driver path in an answer file.

To select and add a device-driver path to an answer file:

1. Open Image Configuration Editor with a Distribution Share and an answer file.
2. From the Insert menu, select Driver Path, then choose the configuration pass in which you want to install the driver (Windows PE for boot-critical drivers).

Note:

For more information on configuration passes, refer to the Windows Embedded Standard 2011 Help topics “How Configuration Passes Work” and “Image Builder Configuration Passes”.

3. Navigate to the folder containing the inf-type drivers you wish to include and click Open.
4. The Answer File pane now shows that the selected driver path is added to the answer file.

Note:

You can also right-click the Out-of-Box Drivers folder or any subfolder in the Distribution Share pane and select Add to Answer File to add the path to the Answer File pane.

2.2.6 Make Your Install Fully Unattended (Optional)

A fully unattended install is an installation that requires no user input after Image Builder Wizard is started with an answer file. To achieve a fully unattended installation, Image Configuration Editor must be used to create an answer file that provides all the information necessary for both the Image Builder Wizard and out-of-box experience (OOBE) processes. The following two sections guide you through the process of providing this information in Image Configuration Editor.

Note:

It is helpful to create an answer file with only the necessary settings and save that answer file with the name Fully_Unattend.xml. Then every time you create a new Windows Embedded Standard 2011 configuration, start by loading the Fully_Unattended.xml and saving it as a different answer file (for example: Working.xml). By creating this Fully_Unattend.xml answer file, you can always start with it and then add the packages and customizations you want for a specific Standard 2011 runtime.

2.2.6.1 Suppress Image Builder Wizard Dialogs

You must include certain settings in your answer file to prevent the Image Builder Wizard user interface (UI) from being displayed. The default behavior of Image Builder Wizard is to display the UI if any of the required settings are incorrect or not available.

In some cases there are multiple ways to configure your unattended installation; which way you choose depends on your build and installation preferences. The following tables describe only the minimum required settings to perform an unattended installation on a device with a single partition. All the settings are found in sub-nodes of the Product/Operating System/Foundation Core/Windows Embedded Edition node packages.

To update the settings:

1. Click on the package that contains the setting in the Answer File pane.
2. Change Filter View By to the configuration pass shown for each setting in the following table.
3. Click on the Value column and enter the example value for each setting.

Note:

The settings described in this document refer to a setup that has only one system partition. Unattended setup and deployment can be achieved on systems with multiple volumes and partitions, but the installation partition/volume will need to be specified using the Setup/ImageInstall/OSImage/InstallTo SMI setting.

The settings listed in these tables do not include their architecture abbreviation suffix in order to simplify the presentation. The suffix will appear as either “_x86” or “_amd64”, depending on the Distribution Share selected for your device’s architecture.

Image Builder Wizard Language and Locale Settings

SETTING	VALUE	DESCRIPTION	CONFIGURATION PASS
Microsoft-Windows-International-Core-WinPE	InputLocale = “en-US”	Specifies the default language to use on the installed OS.	1 windows PE
	UILanguage = “en-US”	Specifies the default language to use during Standard 2011 setup. During installation, Standard 2011 setup displays installation progress in the selected language.	

Image Builder Wizard Dialog Boxes

SETTING	VALUE	DESCRIPTION	CONFIGURATION PASS
Microsoft-Windows-Setup	ShowTemplateSelection = false	Specifies whether to show the IBW feature template selection page.	1 windowsPE
Microsoft-Windows-Setup Target Analysis Mapping	RunTargetAnalysisMapping = True ShowImportDriverUI = False WillShowUI = False	Specifies whether to run TAP.exe and map the .pmq file to driver packages. Specifies whether to show the IBW Import Driver UI page. Specifies whether to show the IBW Target Analysis Mapping pages.	1 windowsPE
Microsoft-Windows-Setup UserData	AcceptEULA = True	Specifies whether to accept the Microsoft License Software Terms during IBW	1 windowsPE

Image Installation

You must specify where you intend to install Windows. You can either specify the exact Disk ID and Partition ID or you can install Windows to the first available partition.

SETTING	VALUE	DESCRIPTION	CONFIGURATION PASS
Microsoft-Windows-Setup	CreateSeparateSystemPartition = false	Specifies whether a separate system partition should be created	1 windowsPE
Microsoft-Windows-Setup ImageInstall	InstallToAvailablePartition = true	Installs Standard 2011 to the first available partition.	1 windowsPE

		NOTE: Cannot be used in conjunction with Microsoft-Windows-Setup ImageInstall OSImage InstallTo shown below.	
Microsoft-Windows-Setup ImageInstall OSImage InstallTo	DiskID = 0 PartitionID = 1	Specifies the disk and partition to install Standard 2011 on. NOTE: Cannot be used in conjunction with Microsoft-Windows-Setup ImageInstall shown above.	1 windowsPE

2.2.6.2 Suppress OOBE

The Windows out-of-box experience (OOBE) or “Windows Welcome” is part of the setup UI that appears at the final stage of the setup process, or on first boot after a captured image is deployed. This UI element is run online (booting from the target device) and is a stand-alone application that runs when an image is set up with Image Builder Wizard or deployed from a sysprepped .WIM file. In the OOBE UI, an end-user can configure certain personalization settings, such as naming the computer, creating a user account, setting date and time, time zone, and so on. Many of these settings may not be applicable in embedded scenarios, and, therefore, you may wish to suppress the OOBE UI during setup.

In this lab, we suppress the OOBE UI that appears at the end of setup. Suppression of the OOBE UI is accomplished by providing the necessary information to the setup/OOBE process by using a properly populated answer file.

Some key points that you should consider when suppressing the OOBE UI are user account creation, EULA acceptance, and the product key. Although all the settings used in this section are required to suppress the OOBE UI, these settings — in particular — are required for your system to be usable. If you omit any of these settings from your answer file, not only will the OOBE UI appear, but your system might possibly be unusable.

Note:

The settings listed in these tables do not include their architecture abbreviation suffix in order to simplify the presentation. The suffix will appear as either “_x86” or “_amd64”, depending on the Distribution Share selected for your device’s architecture.

OOBE Dialog Box Settings

SETTING	VALUE	DESCRIPTION	CONFIGURATION PASS
Microsoft-Windows-Setup UserData	AcceptEula = true	Specifies the acceptance of the EULA.	1 windows PE
Microsoft-Windows-Shell-Setup	ComputerName = "*" ProductKey = <product key>	Specifies the name of the computer to apply to the Standard 2011 installation. Using an asterisk will assign a random name. Specifies the product key to use for activation. Note that all zeros may be used – "00000-00000-00000-00000-00000".	4 Specialize
Microsoft-Windows-International-Core	InputLocale = "en-US" SystemLocale = "en-US" UILanguage = "en-US" UserLocale = "en-US"	Specifies the default input locale of the Standard 2011 installation. Specifies the default system locale of the Standard 2011 installation. Specifies the default UI language of the Standard 2011 installation. Specifies the default user locale of the Standard 2011 installation Example: "en-US"	7 oobeSystem

Microsoft- Windows-Shell- Setup OOBE	<p>HideEULAPage = true</p> <p>ProtectYourPC = 3</p> <p>HideWirelessSetup inOOBE = true</p> <p>NetworkLocation = "Work"</p> <p>TimeZone = "Pacific Standard Time"</p>	<p>Suppresses the display of the Microsoft Software License Terms page..</p> <p>Specifies the protection level of the Standard 2011 installation.</p> <p>Hides the Wireless Network selection page.</p> <p>Specifies the location of the device.</p> <p>Specifies the time zone of the device.</p>	7 oobeSystem
Microsoft- Windows-Shell- Setup UserAccounts	Right-click to insert a new user account.	Specifies the user accounts to create on the Standard 2011 installation. There must be at least one user account created in the answer file or it will not be possible to log into the system	7 oobeSystem
Microsoft- Windows-Shell- Setup UserAccounts LocalAccount	<p>Name = <Name></p> <p>DisplayName = <Display Name></p>	<p>Specifies a valid account name not already present in your image.</p> <p>Specifies a display name for the user account.</p>	7 oobeSystem

Microsoft- Windows-Shell- Setup UserAccounts LocalAccount Password	value = <password>	Specifies a password for the user account.	7 oobeSystem
Microsoft- Windows-Shell- Setup UserAccounts AdministratorPass word	value = <password>	Specifies a password for the administrator account.	7 oobeSystem
Microsoft- Windows-Core- Settings	HideEvaluationWar ning = 1	Suppresses the display of the evaluation warning	7 oobeSystem

AutoLogon Settings

Although not actually part of the OOBE suppression, the following settings enable AutoLogon which may be desired for some automation scenarios.

SETTING	VALUE	DESCRIPTION	CONFIGURATION PASS
Microsoft- Windows-Shell- Setup AutoLogon	Enabled = true Username = <autologon account name>	Enables Autologon. Specifies the account that will be automatically logged on.	7 oobeSystem
Microsoft- Windows-Shell- Setup AutoLogon Password	LogonCount = 9999 PlainText = false Value = <encrypted password string>	Specifies the number of times someone can autologon to the device. Specifies whether the Autologon password is hidden in the unattended answer file. Specifies the encrypted password for the autologon account.	7 oobeSystem

2.2.7 Create Image Builder Wizard Disk from Answer File (optional)

The Media Creation tool in Image Configuration Editor includes an IBW Disk from Answer File option that allows you to quickly deploy an image with a specified configuration. This option creates an IBW disk and a configuration set so that, when booting from this disk, IBW will automatically consume the provided answer file and install that configuration on the target device.

To generate an Image Builder Wizard Disk from an Answer File:

1. Create a bootable utility disk (see section 1.6.1 above).
2. In the Tools menu, select Media Creation, then select Create IBW Disk from Answerfile.
3. Enter the desired answer file to be copied, select the utility disk as the target folder and click OK.
4. Attach the disk to your device and reboot to run setup and install your image, making sure your device boots to this disk.

3 Deploy an Image

In Section 2 you developed a Windows Embedded Standard 2011 image using Image Builder Wizard and/or Image Configuration Editor and then installed it on your reference device. In this section you will learn how to prepare, capture and deploy that image to additional devices.

3.1 Prepare and Capture an Image for Deployment

3.1.1 Sysprep the Image

You can use the Sysprep tool with other deployment tools to install the Windows Embedded Standard 2011 operating system onto new hardware. Sysprep prepares a computer for disk imaging or delivery to a customer by configuring the device to create a new computer security identifier (SDI) when the computer is restarted. In addition, Sysprep cleans up user-specific and device-specific settings and data that must not be copied to a destination device. This process is called “generalization.”

To prepare a device for disk imaging:

1. Start with the device on which you already installed a Windows Embedded Standard 2011 image. (For more information about building an image, see Section 2.)
2. Open a command prompt and run the following command. This will remove all device and user-specific information from the image, and shut down the device.

```
sysprep /generalize /shutdown
```

3. [Optional] You can also use an answer file with Sysprep by adding the /unattend command.

```
sysprep /generalize /shutdown /unattend:<file_path>
```

Note:

Sysprep can be found on your device in the folder `\Windows\System32\Sysprep`. You may need to navigate to the root directory or to the Sysprep sub-folder to run Sysprep.

For additional sysprep command line options, refer to the Windows Embedded Standard 2011 Toolkit Help.

3.1.2 Capture the Image Into a Windows Image (.wim) File

Now that you have a generalized Standard 2011 image installed on your device, you can use ImageX to capture it into a .wim file so you can deploy it to other devices.

To create a .wim file:

1. On your device, do one of the following:
 - Boot into Image Builder Wizard from your utility disk or the Windows Embedded Standard 2011 DVD and open a command prompt by either:
 - On the Build an Image/Deploy an Answer File or WIM page, press Shift + F10.
 - On the Build an Image/Deploy an Answer File or WIM page, select Launch WinPE Command Prompt.
 - Boot into Windows PE from your utility disk.

2. If necessary, connect to your network share or the media share that you want to copy your captured image to. For example, the following command will connect you to a network share located at [\\mycomputer\images](#) with the user “mydomain\user” and a password of “password.”

```
net use * \\mycomputer\images /USER:mydomain\user password
```

3. Capture the image using ImageX. You can do this by entering the following command, where C: is the drive you want to capture and Z:\Mydevice.wim is the location where you want to save your Windows Image file.

```
imagex /capture C: Z:\mydevice.wim "Drive C" /verify
```

4. You can also compress the captured image to reduce the size of the image. For example:

```
imagex /capture /compress maximum C: Z:\mydevice.wim "Drive C" /verify
```

Note:

For more information about running ImageX, refer to the [Windows Embedded Standard 2011 Toolkit Help](#).

3.2 Deploy an Image Using ImageX

In this lab, you will deploy your Windows Embedded Standard 2011 image onto a device by using ImageX and Windows PE. To do so, you must first format the hard drive of the destination device with two partitions, apply the image to the second partition, then run BCDBoot to reconfigure the BCD store.

3.2.1 Prepare your Destination Device

To begin this deployment, you will need to boot your destination device and prepare the hard drive by using Windows PE and DiskPart. If you do not have a separate destination device, you can reuse your reference device as the destination device.

To prepare the hard drive:

1. On your device, do one of the following:
 - Boot into Image Builder Wizard from your utility disk or the Windows Embedded Standard 2011 DVD and open a command prompt by either:
 - On the Build an Image/Deploy an Answer File or WIM page, press Shift + F10.
 - On the Build an Image/Deploy an Answer File or WIM page, select Launch WinPE Command Prompt.
 - Boot into Windows PE from your utility disk.

Note:

You may need to override the boot order to boot from the utility disk or CD/DVD-ROM drive. During initial boot, select the appropriate function key to override the boot order.

2. Run the DiskPart tool by typing the following at the command prompt:

```
diskpart
```

3. Use the DiskPart tool to determine the disk number and device size to be used for the next step:

```
list disk
```

4. Use the DiskPart tool to partition and format the drive and make it bootable. At the DiskPart prompt, type the following, replacing <disk_number> with the disk number of the drive you wish to use. Although multiple partitions may be used, the following example creates a single partition in order to keep the lab exercise as simple as possible.

```
select disk <disk number>
clean
create part pri
select part 1
active
format fs=ntfs quick
assign letter = C
exit
```

Note:

You can also script this process by saving these commands to a text file. To run the script from a command prompt, type:

```
diskpart /s <scriptname.txt>
```

3.2.2 Apply an Image to Your Device's Hard Drive

Next you apply your Standard 2011 image to your device's hard drive. After it is applied, you need to run the BCDBoot tool to configure the boot partition.

To apply your image and run BCDBoot:

1. Connect the USB drive that contains your saved Windows Embedded Standard 2011 image to your device. Apply the image to the hard drive by using ImageX by typing the command below at a command prompt. In the example below <drive1> refers to the drive where ImageX can be found and <drive2> refers to the drive where your Standard 2011 image WIM can be found:

```
<drive1>:\imagex.exe /apply <drive2>:\images\myimage.wim 1
C:
```

2. From the command prompt, run BCDBoot to update the BCD store. For example:

```
bcdboot.exe C:\windows /s C:
```

3.2.3 Shut Down the Destination Device

Now that your Windows Embedded Standard 2011 image is deployed, you can shut down the device by using the wpeutil shutdown option in Windows PE. From the Windows PE command prompt, type:

```
wpeutil shutdown
```

3.3 Deploy an Image Using Image Builder Wizard

After capturing a custom image to a .wim file by using ImageX, you can deploy this image by using Image Builder Wizard. Using this method, you can create customized setup disks that deploy the preconfigured image easily and quickly to your devices.

3.3.1 Start Image Builder Wizard with your Image

Choose one of the following four methods to start Image Builder Wizard with your image.

3.3.1.1 *Use the Deploy an Answer File or WIM button:*

1. Boot your device into Image Builder Wizard using the Windows Embedded Standard 2011 DVD or the utility disk you created in section 1.6.
2. On the Build an Image/Deploy an Answer File or WIM page, select Deploy an Answer File or WIM.
3. Browse to your WIM file and click Open.

3.3.1.2 *Use the /installfrom Switch:*

1. Do one of the following:
 - Boot into Image Builder Wizard from your utility disk and open a command prompt by either
 - On the Build an Image/Deploy an Answer File or WIM page, press Shift + F10.
 - On the Build an Image/Deploy an Answer File or WIM page, select Launch WinPE Command Prompt.
 - Boot into Windows PE from your utility disk.
2. Navigate to the directory that contains setup.exe
 - If you are booting from your Image Builder Wizard disk, look in <drive>:\sources
 - If you are booting from a Windows PE disk, make sure you can access your Image Builder Wizard files from either a network location or a USB drive. Navigate to that location and find setup.exe in the sources folder.

3. Type the following at the command prompt:

```
setup.exe /installfrom:<drive>:\mydevice.wim
```

3.3.1.3 *Use the InstallFrom Unattend Setting:*

1. Use Image Configuration Editor to configure the InstallFrom setting under Windows Embedded Edition/Setup/ImageInstall/OSImage
 - a. The Path setting is the location of your custom image WIM file.
 - b. You may enter Domain, Username and Password settings under Credentials if the path to your WIM is a network location.
 - c. Save your changes to an answer file.
2. Using the answer file you just created, do one of the following:
 - Boot into Image Builder Wizard from your utility disk. On the Build an Image/Deploy an Answer File or WIM page, press Shift + F10 to open a command prompt.
 - Boot into Windows PE from your utility disk.

3. Navigate to the directory that contains setup.exe
 - If you are booting from your Image Builder Wizard disk, look in <drive>:\sources
 - If you are booting from a Windows PE disk, make sure you can access your Image Builder Wizard files from either a network location or a USB drive. Navigate to that location and find setup.exe in the sources folder.
4. Type the following at a command prompt:

```
setup.exe /unattend:<path_to_your_answer_file>
```

3.3.1.4 *Replace the install.wim on Image Builder Wizard Media*

1. Do one of the following:
 - Follow the steps in section 1.6 to create an Image Builder Wizard USB drive.
 - Copy all files from the Image Builder Wizard DVD to a folder on your development computer.

Note:

In steps 2 through 6 below, E:\ refers to the root of your Image Builder Wizard files.

2. Make a backup copy of the install.wim file in E:\Sources
3. Copy mydevice.wim to E:\sources
4. Rename E:\sources\mydevice.wim to E:\sources\install.wim
5. If E:\ is your USB drive, you can use this drive to boot into Image Builder Wizard and automatically use your custom image as the default image to install.
6. If E:\ is your folder of files copied from your Image Builder Wizard DVD, you will need to create an ISO from these files to turn it back into a DVD. Example:

```
oscdimg -n -m -b"C:\Program Files\Windows Embedded  
Standard 2011\Tools\PETools\x86\boot\etfsboot.com" E:\  
C:\MyNewIBWDisk.iso
```

You may need to change this command to point to etfsboot.com correctly based on your installation directory and device architecture.

3.3.2 *Customize a Captured Image*

In addition to simply deploying custom images with Image Builder Wizard, you can also apply answer files and add additional languages to your custom image.

3.3.2.1 *Add a language:*

1. Choose one of the methods in section 3.3.1 to start Image Builder Wizard with your image.
2. When prompted to choose a language, select the language you would like to add. If the language was not already added to your image, it will be added. The selected language will also be set to the default system language.

3.3.2.2 *Apply an Answer File*

1. Do one of the following:
 - Boot into Image Builder Wizard from your utility disk and open a command prompt by either:
 - On the Build an Image/Deploy an Answer File or WIM page, press Shift + F10.
 - On the Build an Image/Deploy an Answer File or WIM page, select Launch WinPE Command Prompt.
 - Boot into Windows PE from your utility disk.
2. Navigate to the directory that contains setup.exe.
 - If you are booting from your Image Builder Wizard disk, look in <disk>:\sources.
 - If you are booting from a Windows PE disk, make sure that you can access your Image Builder Wizard files from either a network location or USB drive. Navigate to that location and find setup.exe in the \sources folder.
3. Type the following at the command prompt where z:\mydevice.wim and y:\myanswerfile.xml refer to the names and locations of your files:

```
setup.exe /installfrom:Z:\mydevice.wim  
/unattend:Y:\myanswerfile.xml
```

4 Service an Image

4.1 Service an Image with Image Configuration Editor and DISM

This lab consists of two procedures in which you apply updates to a Windows Embedded Standard 2011 image by using Image Configuration Editor and Deployment Image Servicing and Management (DISM). DISM is a command line tool that installs, uninstalls, configures and updates features and packages. In the first section, you will create a configuration set using Image Configuration Editor. Then you will install the updates by choosing either online or offline commands.

In order to complete this lab, you must have a bootable device with a custom image already installed. Using the image you installed in sections 2.1.1 or 2.2.1.

4.1.1 Create a Configuration Set using Image Configuration Editor

1. Open Image Configuration Editor, select your Distribution Share, and open an existing answer file or create a new answer file.

Notes:

If you are creating a new image, ensure that the Embedded Core package (Product\Operating System\Foundation Core\ Embedded Edition) is in you answer file. This package is added by default when you create a new answer file.

If you are adding a package to an existing image, remove the Embedded Core package from your answer file.

2. Select one or more packages to add to your existing image (for example, Windows Media Player found under Graphics and Multimedia).
3. [optional] Add the driver packages needed by your device in one of two ways:
 - Manually add driver packages from the Distribution Share Packages\Driver category.
 - Use a .pmq file by clicking File and then Import PMQ.
4. Resolve dependencies by:
 - a. Selecting Validate and then selecting Add Required Packages.
 - b. If there are any errors on the Validation tab of the Messages pane that read “Dependencies of the source package are not satisfied”, double-click the error message and use the Resolve Dependencies dialog box to satisfy all required package dependencies.
 - If you are asked to choose between two USB stacks, choose “Bootable Windows USB Stack” only if you are using the USB Boot EEF and enabling a USB bootable image (see section 5.3). Otherwise, choose “Standard Windows USB Stack”.
 - If you are asked to choose between two boot environment packages, choose “Embedded Windows 7 Boot Environment” only if you are installing HORM with EWF. Otherwise, choose “Windows 7 Boot Environment”.
 - c. There may also be additional warnings to resolve.
 - d. Repeat steps a through c until no dependency errors exist.

5. Create a configuration set by selecting Tools and then selecting Create Configuration Set.
6. Select the target folder for your configuration set. This is the temporary location that will be used to create your configuration set (for example, C:\myCS).
7. [optional] You can also use an \$OEM\$ folder (see section 2.2.3).
8. Click OK.
9. When you see the “Config set created successfully” message, click OK.
10. Your configuration set is now available at the target folder you specified.
11. You can also save the answer file for future use:
 - a. Select File then select Save Answer File
 - b. Enter a location and click Save.

4.1.2 Install using your Configuration Set

After you have created your configuration set, you can choose whether to install it online or offline.

4.1.2.1 Online Installation

Note:

For the CTP1 build, online installation is not available. Please continue on to Option 2: Offline Installation to complete this exercise.

In this step, you apply your Configuration Set Unattend created in Section 4.1.1 to your online image.

To apply your configuration set:

1. Boot your device.
2. From a command prompt on your device, apply the AutoUnattend.xml file created by Image Configuration Editor. Replace <file_path> with the drive letter, folder, and file name of the configuration set that you created in section 4.1.1 above.

```
DISM /online /Apply-unattend:<file_path>
```

3. Verify that the packages were added to the image by using the DISM /Get-Packages command. /Get-Packages does not return a list of .msu packages; instead, it shows the corresponding .cab package.

```
DISM /online /Get-packages
```

4.1.2.2 Offline Installation

If you choose the offline installation method, you can also choose whether to create a new image based on an existing image using Image Builder Wizard, or you can use DISM to apply your changes directly to an existing image.

If you are creating a new image with a configuration set, you can use Image Builder Wizard to consume the answer file in the configuration set. Use the steps outlined in this section to point Image Builder Wizard to an answer file. Be sure that you point Image Builder Wizard to your configuration set answer file (AutoUnattend.xml) and that the configuration set payload (the AutoUnattend_Files folder) is at the same location as the configuration set answer file.

4.1.2.2.1 To create a new image using Image Builder Wizard:

After you have created your configuration set in Image Configuration Editor, save the folder in a location accessible by Image Builder Wizard along with your existing image and Setup.exe (IBW). This could be a network share, a USB drive, or a floppy drive, depending on the hardware you are installing on.

1. Boot into IBW on your device using the DVD or the utility disk you created in section 1.6.
2. On the Build an Image/Deploy an Answer File or WIM page, select Deploy an Answer File or WIM.
3. In the browse to file dialog box, browse to your answer file and click Open.

Note:

If you stored your answer file on a network drive you will need to access it from Windows PE by first mapping a network drive by using the “wpeinit” and “net use” commands. Wpeinit initializes basic WinPE functions like networking. Please run wpeinit first, then wait approximately 1 minute before continuing.

Example:

```
wpeinit
net use z: \\MyNetworkShareLocation
```

4.1.2.2.2 To use DISM to apply an update to an existing image:

Use DISM to mount the .wim file that you created in step 3.1.:

1. Similar to ImageX, you use DISM to mount and unmount images for the purposes of offline servicing. From the command prompt, type:

```
MD C:\TAP\Mount
DISM /Mount-Wim /Wimfile:E:\images\myimage.wim
/index:1 /Mountdir:C:\TAP\Mount
```

2. From a command prompt on your technician computer, apply the Autounattend created by Image Configuration Editor. Replace <file_path> with the drive letter, folder, and file name of the configuration set you created in step 5 of the section 4.1.1.

```
DISM /image:C:\TAP\Mount /Apply-unattend:<file_path>
```

3. Verify that the packages were added to the image by using the DISM /Get-Packages command. /Get-Packages does not return a list of .msu packages; instead, it shows the corresponding .cab package.

```
DISM /image:C:\TAP\Mount /Get-packages
```

4. Unmount and commit the changes to the image. From the command prompt, type:

```
DISM /unmount-wim /Mountdir:C:\TAP\Mount /commit
```

4.1.3 Test your image

In this step, you redeploy your updated image and verify your updates.

To test your image:

1. Deploy your updated image from your USB drive. (Also see section 3, [Deploy an Image](#))
2. When the image is applied, remove all media (CD-ROM, USB hard drive, or floppy disk) and restart the computer.
3. Verify your updates. Depending on which updates you selected, you can verify from the OS that they were installed and are operating correctly.

4.2 Service an Image with DISM

In this lab, you update an existing image by using Deployment Image Servicing and Management (DISM). DISM supports updating both an offline image (.wim) and a subset of functions for updating an online image. DISM.exe is copied to your development computer as part of the Windows Embedded Studio installation and can be found in C:\Program Files\Windows Embedded Studio\Tools\.

4.2.1 Mount your image

In this step, you will use DISM to mount the image you captured in step 3.1 [Prepare and Capture an Image](#). Similar to ImageX, you can use DISM to mount and unmount images for offline servicing.

To mount an image using DISM, open a command prompt and type:

```
MD C:\TAP\Mount
DISM /Mount-Wim /WimFile:E:\images\myimage.wim /index:1
/Mountdir:C:\TAP\Mount
```

4.2.2 Add sample packages

In this step, select two sample packages to add to your existing image offline. The following sample packages are included with the Windows Embedded DVD in the \Updates folder.

Update Packages for 32-bit Configurations

KB	Feature set (Parent Package)	Component
600000	WinEmb-Accessibility	Microsoft-Windows-accessibilitycpl
600001	WinEmb-IE-Core	Microsoft-Windows-IE-Behaviors
600002	WinEmb-MediaPlayer	Microsoft-Windows-MediaPlayer-autoplay
600003	WinEmb-INF-EmbeddedCore-Drivers-Package	1394.inf
600004	Microsoft-Windows-Embedded-Features-Package	Microsoft-Windows-WBioSvc
600005	WinEmb-usbboot	embedded-usbstor.inf
600006	WinEmb-usb	usbstor.inf
600007	WinEmb-Shell-Basic	Microsoft-Windows-notepad

Update packages for 64-bit configurations

KB	Feature set (Parent Package)	Component
400000	WinEmb-Accessibility	Microsoft-Windows-accessibilitycpl
400001	WinEmb-IE-Core	Microsoft-Windows-IE-Behaviors

400002	WinEmb-MediaPlayer	Microsoft-Windows-MediaPlayer-autoplay
400003	WinEmb-INF-EmbeddedCore-Drivers-Package	1394.inf
400004	Microsoft-Windows-Embedded-Features-Package	Microsoft-Windows-WBioSrv
400005	WinEmb-usbboot	embedded-usbstor.inf
400006	WinEmb-usb	usbstor.inf
400007	WinEmb-Shell-Basic	Microsoft-Windows-notepad

4.2.3 Adding a Package Using DISM

1. From a command prompt on your device, add two sample packages to your mounted image by using the DISM /Add-Package command. For example:

```
DISM /image:C:\TAP\Mount /Add-Package
/PackagePath:D:\Updates\KB600000\V1.001\WinEmb6.1-KB600000-
x86.cab
```

2. Verify that the packages were added to the image by using the DISM /Get-Packages command. /Get-Packages does not return a list of .msu packages; instead, it shows the corresponding .cab package. For example:

```
DISM /image:C:\TAP\Mount /Get-packages
```

4.2.4 Unmount and Commit Changes

In this step, you will unmount and commit the changes to the image. From the command prompt, type:

```
DISM /unmount-wim /Mountdir:C:\TAP\Mount /commit
```

4.2.5 Redeploy and Verify

In this step, you will redeploy your updated image and verify that the updates were installed.

1. Deploy your updated image from your USB drive (see section 3 [Deploy an Image](#)).
2. When the image is applied, remove all media (CD-ROM, USB hard drive, or floppy disk) and restart the computer.
3. Use DISM to verify that the updates were installed.

```
DISM /online /get-packages
```

4.2.6 Add a Language Pack (optional)

If your manufacturing process dictates adding language support during deployment, you can add the appropriate language packs by using the DISM /Add-package options. In this step, you will add a language pack to your base Windows Embedded Standard 2011 image.

1. Make a copy of your base image so it is preserved. For example:

```
Copy E:\images\myimage.wim E:\images\myimage_lab.wim
```

2. From a command prompt on your device, mount the image to a directory:

```
DISM /Mount-Wim /WimFile:E:\images\myimage_lab.wim /Index:1
/mountdir:C:\TAP\Mount
```

3. Add a language package to your image offline by using DISM /Add-Package command. This example adds a language pack from the \Windows Embedded Studio\DS\Packages\LanguagePack.

```
DISM /image:C:\TAP\Mount /Add-Package
/PackagePath:C:\Program Files\Windows Embedded Standard
2011\DS\Packages\LanguagePack\x86~Microsoft-Windows-
Embedded-LanguagePack-Package~ZH-CN~~6.1.7053.0~1.0\LP.CAB
```

4. Verify that the language pack was added to the image by using the DISM /Get-Intl command. For example:

```
DISM /image:C:\TAP\Mount /Get-Intl
```

5. Unmount and commit the changes:

```
DISM /unmount-wim /MountDir:C:\TAP\Mount /Commit
```

4.3 Install Updates with WUSA

Note:

For the CTP1 build, WUSA cannot be used to install updates. Please skip this exercise.

To install updates using the Windows Update Standalone Installer (WUSA), you have to add the WinEmb-Servicing-WUSA Feature Set to your image.

Copy the updates (in .msu format) that you want to install to your device. Sample updates can be found on the DVD in the \Updates folder. After the update file is copied to your device, you can either simply double-click the .msu file or use the WUSA command-line options to install the update:

```
wusa.exe d:\WinEmb6.0-KB934307-x86.msu /quiet
```

4.4 Service an Image with Package Scanner

Package Scanner is a command-line application that helps to service Windows Embedded Standard 2011 devices. You can find the version of Package Scanner (Pkgscn.exe) for use on an x86 runtime in the C:\Program Files\Windows Embedded Standard Quebec\Tools\Image Configuration Editor\ directory (or in the directory where you installed Image Configuration Editor). You can find the version of Package Scanner (Pkgscn.exe) for use on an amd64 runtime in the C:\Program Files (x86)\Windows Embedded Standard 2011\Tools\x64 Apps\ directory. You must run the version of Package Scanner appropriate to the architecture of the image.

Run *pkgscn.exe* /? at a command prompt for a full list of command line options.

Package Scanner can be run offline or online against WES2011 images. By using the /Image: option, you can point Package Scanner to the root of an offline WES2011 installation (for example, if you attached the WES2011 device's hard disk to your development computer, or mounted the .wim file). Running Package Scanner without the /Image: option scans WES2011 images online. If you would like to run Package Scanner online, run CopyPkgScn.cmd from your Image Configuration Editor folder. If you are running as administrator, double-clicking CopyPkgScn will create a pkgscn folder, which you may copy to your target device. Running CopyPkgScn from a command prompt allows you to specify a target directory. The following examples assume Package Scanner is run online on the WES2011 image. Use /Image:D:\ (where D:\ is the location of your offline WES2011 image) to run Package Scanner offline.

Note:

If your development computer is running a 64-bit operating system, the paths listed above will instead be under the C:\Program Files (x86)\ directory.

4.4.1 Package Enumeration

Package Scanner can list the packages installed in an image by using the `/Get-Packages` command. This command is useful for determining what packages are in your image, and what updates have been applied. By using the `/ConfigSetDir:` option in addition to `/Get-Packages` you can save the list of installed packages into an answer file in a configuration set.

Example:

```
PkgScn.exe /Get-Packages /ConfigSetDir:E:\configset
```

4.4.2 Find Applicable Updates

Using the `/Find-Updates` command, Package Scanner can query your image and a folder containing update packages. Package Scanner will determine which updates are applicable to your image.

Use the `/PackagePath:` option to specify the folder of updates to scan (a directory for you to test with is included in the Updates folder on your Image Builder Wizard x86 and amd64 DVDs).

Use the `/ConfigSetDir` option to create a configuration set from the applicable updates.

Example:

```
PkgScn.exe /Find-Updates /PackagePath:F:\Updates  
/ConfigSetDir:E:\configset
```

You can then use DISM to apply this configuration set to your device by using the following command from your Windows Embedded Standard 2011 runtime image:

```
Dism.exe /online /apply-  
unattended:E:\ConfigSet\autounattend.xml
```

4.4.3 Scavenging

Note:

For CTP2, scavenging is not available as the functionality is being revised. Please skip this step.

Scavenging refers to the removal of packages on your image that are no longer being used. You can use the scavenging feature of PackageScanner to prevent unwanted files from reducing the free disk space on your device over time.

For example, scavenging might be useful if you were to install an update KB12345 which updates notepad.exe to version 2.1. Later, you install update KB12346, which updates notepad.exe to version 2.2. Notepad.exe version 2.1 still exists on your device so that if you want to roll back to version 2.1, nothing will break. However, if you stay with version 2.2, version 2.1 still takes up disk space.

To try out scavenging, install two versions of an update. A directory of sample version 1 and 2 updates for you to test with is included in the Updates folder on your Image Builder Wizard x86 and amd64 DVDs. You'll need to use DISM to first install version 1 of the update, then install version 2 of the update.

You can use the following DISM command to install updates.

```
dism.exe /online /add-package /packagepath:D:\Updates\  
\KB600000\V1.001\WinEmb6.1-KB600000-x86.cab
```

Install the first version of the update and then install the second version. Once both updates are installed, if you run the following command, you will find that both packages are installed, although the version 1 package is listed as superseded:

```
Dism.exe /online /get-packages
```

Now run:

```
Pkgsn.exe /Find-Scavenge /ConfigSetDir:e:\configset
```

Package Scanner will detect that version 1 of the update is not being used and is therefore eligible for scavenging. The configuration set created with the */ConfigSetDir* option contains an answer file that tells DISM to remove the unused packages.

You can apply the answer file using DISM by running the following command from your runtime image:

```
Dism.exe /online /apply-unattend:e:\ConfigSet\Autounattend.xml
```

5 Additional Windows Embedded Standard 2011 Labs

5.1 Suppress OOBE

The Windows out-of-box experience (OOBE) or “Windows Welcome” is part of the setup UI that appears at the final stage of the setup process, or on first boot after a captured image is deployed. This UI element is run online (booting from the target device) and is a stand-alone application that runs when an image is set up with Image Builder Wizard or deployed from a sysprepped .WIM file. In the OOBE UI, an end user can configure certain personalization settings, such as naming the computer, creating a user account, setting date and time, time zone, and so on. Many of these settings may not be applicable in embedded scenarios, and, therefore, you may wish to suppress the OOBE UI during setup.

In this lab, we suppress the OOBE UI that appears at the end of setup. Suppression of the OOBE UI is accomplished by providing the necessary information to the setup/OOBE process by using a properly populated answer file.

Some key points that you should consider when suppressing the OOBE UI are user account creation, EULA acceptance, and the product key. Although all the settings used in this section are required to suppress the OOBE UI, these settings — in particular — are required for your system to be usable. If you omit any of these settings from your answer file, not only will the OOBE UI appear, but your system might possibly be unusable.

5.1.1 Create an Answer File

Create an answer file that contains all the settings required to suppress OOBE UI:

1. Launch the Image Configuration Editor and create a new answer file or open an existing one.
2. Add the Embedded Core package to your answer file if you do not already have it.
3. Optional: Add any other feature-sets and packages to your answer file that you need.
4. Click on the package that contains the setting in the Answer File pane.
5. Change Filter View By to the configuration pass shown for each setting in the following table.
6. Click on the Value column and enter the example value for each setting.
7. When all the settings have been updated, save the file in a location accessible by Image Builder Wizard. This could be a network share, a USB drive, or a floppy disk, depending on the hardware you are installing on.

5.1.2 Using your answer file with Image Builder Wizard

You can now use the answer file you created in the last step to suppress OOBE. When setup runs the oobeSystem pass, the OOBE UI should be suppressed because all the information required by the OOBE UI is provided by the answer file. Note that there are several ways of deploying an image beyond the one listed below. See Section 3 for more information.

To use your answer file with Image Builder Wizard:

1. Boot into IBW on your device using the DVD or the utility disk you created in section 1.6.

2. On the Build an Image/Deploy an Answer File or WIM page, select Deploy an Answer File or WIM.
3. In the browse to file dialog box, browse to your answer file and click Open.

Note:

If you stored your answer file on a network drive you will need to access it from Windows PE by first mapping a network drive by using the “wpeinit” and “net use” commands. Wpeinit initializes basic WinPE functions like networking. Please run wpeinit first, then wait approximately 1 minute before continuing.

Example:

```
wpeinit  
net use z: \\MyNetworkShareLocation
```

5.2 Using a Custom Shell and Custom Branding

This section describes how to create a custom shell experience using all of the custom shell components. Each of these components can be used separately or in any combination. With custom shell you can:

- Hide boot screens
- Use Shell Launcher to reference a custom shell
- Remove Windows branding from startup screens
- Replace the background image for the startup screens
- Add message blockers

5.2.1 Hide boot screens

The kernel screens shown during startup and resume from hibernation can be suppressed by adding the package Hide Boot Screens to your runtime using Image Configuration Editor or Image Builder Wizard.

5.2.2 Use Shell Launcher

5.2.2.1 *To use Image Configuration Editor to reference a custom shell (preferred method):*

1. Determine which package(s), if any, your shell needs, and select those packages.
2. Select Command Prompt with Custom Shell as your shell.
3. Place your shell in the appropriate \$OEM\$ folder. (Refer to section 2.2.3 for additional \$OEM\$ folder information or Microsoft TechNet article [Designing a Distribution Share](#).)
4. Add the Shell Launcher component to the offline phase
5. Set the CustomShell setting to run your shell application.
6. If necessary, update *setupcomplete.cmd*. (See Microsoft TechNet article [Add a Custom Script to Windows Setup](#).)

5.2.2.2 *To use Image Builder Wizard to reference a custom shell:*

1. Determine which package(s), if any, your shell needs, and select those packages.
2. Select Command Shell as your shell.
3. Update *setupcomplete.cmd* to modify the HKLM\Software\Microsoft\Windows Embedded\EEF\Shell Launcher value to include the name of your shell, using WMI. There is no Image Builder Wizard mechanism to modify unattend.xml. (See Microsoft TechNet article [Add a Custom Script to Windows Setup](#).)
4. Place your shell in the appropriate \$OEM\$ folder. (Refer to section 2.2.3 for additional \$OEM\$ folder information or Microsoft TechNet article [Designing a Distribution Share](#).)

5.2.3 Remove Windows branding

There are two packages available for branding the startup screens:

- Windows Embedded Standard Startup Screens – shows the Windows Embedded Standard logo.
- Unbranded Startup Screens – shows no logo.

To remove the Windows branding, select the Unbranded Startup Screens package.

5.2.4 Replace the startup screen background image

This step allows you to replace the background image for the startup screens. This, combined with the step above, allows you to rebrand the startup to your custom requirements.

1. Add the Custom Logon Desktop Background Images package.
2. Use the path setting to point to a directory containing one or more of the files listed below.
3. File names must be:
 - backgroundDefault.jpg
 - background768×1280.jpg
 - background900×1440.jpg
 - background960×1280.jpg
 - background1024×1280.jpg
 - background1280×1024.jpg
 - background1024×768.jpg
 - background1280×960.jpg
 - background1600×1200.jpg
 - background1440×900.jpg
 - background1920×1200.jpg
 - background1280×768.jpg
 - background1360×768.jpg

Note:

You do not need to supply all the files. BackgroundDefault.jpg will be used if the corresponding resolution file is not available. The file MUST be less than 256kb. Providing the image file that matches your video resolution will improve the system performance.

4. Use Image Configuration Editor to create a configuration set or an IBW disk from the answer file.
5. Deploy using the configuration set or answer file.

5.2.5 Add Message Blockers

Windows Embedded Standard 2011 has two different message blocking options. They can be used separately or in unison.

5.2.5.1 Message Box Default Reply

This component will automatically respond to all message boxes using their default response. It is identical to the WES component (see MSDN Windows Embedded Standard library article [Message Box Default Reply](#) for more information). Message Box Default Reply is only supported for x86 systems.

5.2.5.2 Dialog Filter

This component allows you to define windows based on a set of parameters (Process name, Window Title, Dialog Type, list of buttons). For each window you can choose which button is used to dismiss the window. This component requires you to know which windows you want to block.

To create the configuration file, we have provided a Dialog Filter Editor that is installed on your development computer with the Windows Embedded Standard 2011 Toolkit. The editor can be used to create a configuration file on any Windows 7 or Windows Embedded Standard 2011 machines. To use it on your runtime, you will need to copy the application to the runtime. The files are located in the *Windows Embedded Standard 2011\EmbeddedSDK\bin* directory. Be sure you copy the *en-US* directory along with the executable.

The Dialog Filter Service is able to process windows as they are created and, in some cases, before they are completely initialized. While this helps eliminate flicker on the device display, in some cases the Dialog Filter Service may see different window information (title, class, process or button list) than what is visible on the screen and the Dialog Filter Editor. Most commonly, "Untitled –" may not be inserted into the window title before the Dialog Filter attempts to close the window. This appears mainly with applications that use documents and place the name in the title (such as Notepad).

If, for any reason, a window does not close as expected, open the log file as described below. This will provide the XML required to filter a window and which can be compared with the XML created by the Dialog Filter Editor.

5.2.5.3 Log File Method

The Windows event log can be used to get the XML data necessary for the configuration file. The configuration file is located at *ProgramData\Microsoft\DialogFilter\ConfigurationList.xml*.

To use the log, you must do the following:

1. Run the event viewer (*eventvwr.exe*)
2. Go to Applications and Services Log/Microsoft/Windows/Dialog Filter/Operational

3. Enable the log (either in the right window pane or by using the right mouse button on the Operational directory).

From this point on the Dialog Filter will log all windows it finds to the operational log. The information will include an XML snippet that can be copied to the configuration file.

An example is:

```
<?xml version="1.0" encoding="UTF-8"?>
<CL:dialogs xmlns:CL="urn:Dialogs">
<dialog>
<ProcessImageName>notepad.exe</ProcessImageName>
<Title>Untitled - Notepad</Title>
<Class>Notepad</Class>
<Buttons>
<Button>Line up</Button>
<Button>Line down</Button>
<Button>Column left</Button>
<Button>Column right</Button>
<Button>Minimize</Button>
<Button>Maximize</Button>
<Button>Close</Button>
</Buttons>
<Actions>
<Action>Close</Action>
</Actions>
</dialog>
</CL:dialogs>
```

5.2.5.4 To Add Message Blockers:

1. Add Message Box Default Reply (see MSDN Windows Embedded Standard library article [Message Box Default Reply](#)).
2. Add Dialog Filter
3. Use the path setting in Image Configuration Editor to point to a configuration file. The file must be named *ConfigurationList.xml*.
 - If you use Image Builder Wizard, install with Image Configuration Editor without a configuration set, or do not provide a configuration file, you can manually place the file in the directory *\ProgramData\Microsoft\DialogFilter* on the device runtime. The file must be named *ConfigurationList.xml*.

Note:

The XML file must be saved in Unicode format. This can be done using Notepad if you choose the save as option and choose utf-8.

5.3 Build a USB bootable Windows Embedded 2011 image

USB Boot is an Embedded Enabling Feature (EEF) that you can use to boot a Windows Embedded 2011 image from a USB Flash drive (UFD), such as a USB thumb drive or USB key, instead of the device hard drive. Choose one of the two options below for building a USB bootable image with Image Builder Wizard. Although slightly more complicated, the second option will run faster.

5.3.1 Install from Image Builder Wizard directly to UFD :

1. Make sure the USB drive you want to install to is plugged into the device that is currently running Image Builder Wizard. Also, make sure there are no files on the USB drive that you need because they may be lost during setup.
2. Run Image Builder Wizard from your chosen source (network share, DVD, or utility disk).
3. Continue through the wizard.
4. On the Please select Feature Packages to include in your image page:
 - a. Under Embedded Enabling Features select Bootable Windows USB Stack.
 - b. Add any additional packages you wish to include.
 - c. Click the Resolve button and resolve all dependency issues.
 - If asked to choose between Standard Windows USB Stack and Bootable Windows USB Stack make sure you leave only Bootable Windows USB Stack checked.
 - Continue resolving dependencies until complete.
5. Make certain the UFD is plugged into the device, then select Next to continue.
6. On the target drive selection screen, select the UFD target, partition and format it, then continue the installation process.
7. Allow setup to complete.
8. If this UFD will be deployed to another machine, follow the SysPrep instructions in section 3.1.1.

Note:

If you are using a UFD with WinPE to run IBW and your install target is a second UFD, after the file copy and when the reboot happens, remove the WinPE UFD from the system to prevent it from rebooting back into setup.

5.3.2 Install from Image Builder Wizard to Hard Drive, ImageX to UFD:

1. Run Image Builder Wizard from your chosen source (network share, DVD, or utility disk).
2. Continue through the wizard.
3. On the Please select Feature Packages to include in your image page:
 - a. Under Embedded Enabling Features select Bootable Windows USB Stack.
 - b. Add any additional packages you wish to include.

- c. Click the Resolve button and resolve all dependency issues.
 - If asked to choose between Standard Windows USB Stack and Bootable Windows USB Stack make sure you leave only Bootable Windows USB Stack checked.
 - Continue resolving dependencies until complete.
4. On the drive selection screen, select the first partition on the target drive, using the GUI tools to partition and format the drive as appropriate.
5. If this image will be deployed to another machine, follow the SysPrep instructions in step 3.1.1.
6. Capture the sysprepped image to a .wim file with the following command:

```
imagex /capture <src><dst> "<name>"
```
7. Insert the destination UFD.
8. Run Diskpart by doing the following:
 - a. From a Windows Vista, Windows 7 or Windows Preinstallation (Windows PE) 3.0 environment, run the DiskPart tool by typing the following at a command prompt:

```
diskpart
```

Note:
The version of the DiskPart tool provided by Windows Vista, Windows 7 and Windows PE 2.0, 2.1 and 3.0 supports the partitioning and formatting of a UFD as a bootable device. Previous versions of the DiskPart tool, including the version provided by Windows XP, do not fully support partitioning and formatting a UFD to be bootable and should not be used.
 - b. Use the DiskPart tool to determine the disk number and device size to be used for the next step by typing the following at the DiskPart prompt:

```
list disk
```
 - c. Use the DiskPart tool to partition and format the drive and make it bootable. At the DiskPart prompt, type the following, replacing <disk_number> with the disk number of the USB drive:

```
select disk <disk_number>
clean
create part pri
select part 1
active
format quick fs=ntfs label=embedded
assign
exit
```
9. Apply the image to the UFD by using ImageX with the following command:

```
imagex /apply <src> 1 <dst>
```
10. After the image is applied, insert the UFD into the target hardware.
11. Boot the computer, making certain the BIOS settings are set to boot to a USB device first.
12. Allow setup to complete.

5.4 Create Custom Templates for IBW

5.4.1 Creating Templates

In addition to the templates created by Microsoft, IBW supports user-created templates. Templates are essentially just answer files that contain no settings. You can create your own templates using Image Configuration Editor.

1. Start Image Configuration Editor.
2. On the File menu, select New Answer File.
3. Add desired feature packages to your answer file.
 - It is recommended that you not add any driver or language packages to your template. These can be added in IBW.
4. On the Validate menu, select Add Required Packages.
 - While you may also choose Add Required and Optional Packages, this makes removing packages from a template in IBW more difficult, as there are many more dependencies between optional packages.
5. In the Messages pane, you may see warnings and errors.
 - a. Click on each error and resolve feature package conflicts by selecting or deselecting packages.
 - d. Review warnings and resolve appropriately.
6. Remove unnecessary language packages (optional). Resolving dependencies may have added a language package to your template. As mentioned above, this is not necessary for IBW templates, as languages will be added by IBW automatically. If you leave the language package in the template, this language will be in every configuration built with IBW, no matter what the user selected.
 - a. Delete any language packages that were added.
 - b. On the Validate menu, select Validate Only.
 - c. Ensure that the language package is the only missing dependency.
7. Save your template.

5.4.2 Using Custom Templates in IBW

Once you have created a custom template, you need to make it available to IBW. Choose one of two options:

1. Use the Browse to template button in IBW's Choose the way you want to build your image page.
- OR
2. Put your template in IBW's Distribution Share
 - a. IBW searches for templates in DS\Templates\IBW
 - b. If you have created a bootable IBW USB disk, you can easily put your custom templates in this folder to have IBW detect and make them available.