



PARAGON Technologie GmbH, Systemprogrammierung

Heinrich-von-Stephan-Str. 5c ● 79100 Freiburg, Germany

Tel. +49 (0) 761 59018201 ● Fax +49 (0) 761 59018130

Internet www.paragon-software.com ● E-mail sales@paragon-software.com

CUSTOMIZING WINPE 2.1 BOOTABLE IMAGES TO USE WITH SPECIFIC OEM, PNP HARDWARE DRIVERS, MENU ITEMS, AND PARAGON SCRIPTS

QUICK USER GUIDE

CONTENTS

Introduction	3
Package Contents	3
What's needed to Create a WinPE 2.1 Bootable Image	3
How to Create a WinPE 2.1 Bootable Image with Example Scripts	4
Using build_image.cmd	4
Using build.cmd	4
Using build_pxe.cmd.....	5
Customization of an Existing WIM Image	5
Adding Programs to a WIM Image	5
Adding Specific Hardware Drivers.....	6
Editing the Startup Menu and Adding Scripts	6
Generating a Script File (.ps1 file)	6
Generating a Batch File to Execute the Script File	8
Adding the New Batch File to the Startup Menu.....	8
Creating a WinPE ISO image (CustomPE.iso)	11
Preparing a Folder for the PXE Startup	11

INTRODUCTION

This document will help you customize a Windows PE CD based on Windows Server 2003 with Microsoft OEM Preinstallation Kit to use it with specific hardware drivers, menu items, and Paragon scripts. It also explains how to prepare a folder containing all the necessary binaries to boot a WinPE 2.1 image through PXE.

Due to the license stipulations, Paragon has no rights to distribute Microsoft software. That's why you need to get the required tools to build WinPE images yourself.

PACKAGE CONTENTS

Paragon WinPE Customization Package includes the following components:

- ❑ **winpe_custom** – a directory with the default product components to place to the resulted image;
- ❑ **custom.cmd** – a script to make from a raw WIM image supplied with OPK or WAIK, a custom WIM or ISO image containing custom drivers, programs, settings, etc.;
- ❑ **makepxe.cmd** – a script to make from a custom WIM image a PXE bootable image;
- ❑ **custom_init.cmd** – a “header” file with initial variables, a path to OPKTools, etc. It's called both by **custom.cmd** and **makepxe.cmd**, so any changes made to **custom_init.cmd** will affect these two customization scripts;
- ❑ **build_image.cmd** – a sample script file demonstrating how to prepare from a raw WIM image a custom WIM or ISO image containing custom drivers, programs, settings, etc.;
- ❑ **build.cmd** – a sample script file demonstrating how to prepare from a raw WIM image supplied with OPK or WAIK, a custom PXE bootable image by using **custom.cmd** and **makepxe.cmd**;
- ❑ **build_pxe.cmd** – a sample script file demonstrating how to prepare from a custom WIM image a PXE bootable image. Strings starting from “rem” inside this script show different scenarios of using **makepxe.cmd**.

WHAT'S NEEDED TO CREATE A WINPE 2.1 BOOTABLE IMAGE

In order to create a WinPE 2.1 based bootable image the following things are required:

- ❑ Supplied by Paragon
 - Customization scripts;
 - A WinPE directory structure containing all the necessary binary files (by default, it's named **winpe_custom**);
 - Paragon Network Module installed.
- ❑ Not supplied by Paragon

- All the necessary facilities to build WIM images based on WinPE 2.1 (imagex, peimg, dll libraries and the wimfltr.sys driver). You can find them all either in Windows Automated Installation Kit (WAIK) or OEM Preinstallation Kit (OPK), so you need to install one of these packages to successfully customize WinPE 2.1 images. To know more on the subject, please consult Microsoft documentation.
- A raw WIM image. You can use the default **winpe.wim** image included both in OPK and WAIK. For WAIK you can find it in **C:\Program Files\Windows AIK\Tools\PETools\<platform>**, while for OPK - **C:\Program Files\Windows OPK\Tools\PETools\<platform>**, where <platform> is "x86" or "amd64".

HOW TO CREATE A WINPE 2.1 BOOTABLE IMAGE WITH EXAMPLE SCRIPTS

As we've mentioned already, the customization package includes several example script files that demonstrate how to make a custom WIM, ISO or PXE bootable image:

- ❑ **build_image.cmd** – a sample script file demonstrating how to prepare from a raw **.wim** image a custom **.wim** or **ISO** image containing custom drivers, programs, settings, etc.;
- ❑ **build.cmd** – a sample script file demonstrating how to prepare from a raw **.wim** image supplied with OPK or WAIK, a custom PXE bootable image by using **custom.cmd** and **makepxe.cmd**;
- ❑ **build_pxe.cmd** – a sample script file demonstrating how to prepare from a custom **.wim** image a PXE bootable image. Strings starting from "rem" inside this script show different scenarios of using **makepxe.cmd**.

These scripts already contain default paths to OPKTools, a raw WIM image and a PXE boot directory, defined by the following variables:

- A path to OPKTools: the **rootPath** variable in **custom_init.cmd**;
- A raw WIM image: **%sourceISOFiles%\winpe.wim** in **build_image.cmd**;
- A PXE boot directory: **C:\Program Files\Paragon Software\Network Module\PXEServer** in **build_pxe.cmd**

The default paths are:

- For OPK: **C:\Program Files\Windows OPK**
- For WAIK: **C:\Program Files\Windows AIK**
- For the PXE boot directory: **C:\Program Files\Paragon Software\Network Module\PXEServer**

USING BUILD_IMAGE.CMD

After you've run this script, you will get a custom WIM or ISO image prepared from a raw WIM image supplied with OPK (or WAIK) plus contents of the **winpe_custom** directory delivered by Paragon.

USING BUILD.CMD

After you've run this script, you will get a custom PXE bootable image prepared from a raw WIM image supplied with OPK (or WAIK) plus contents of the **winpe_custom** directory delivered by Paragon.

USING BUILD_PXE.CMD

After you've run this script, you will get a custom PXE bootable image prepared from a custom WIM image.

CUSTOMIZATION OF AN EXISTING WIM IMAGE

Before starting customization of an existing WIM image, please make sure you've got correct paths defined by the **rootPath**, **toolsDir**, **PEToolsDir**, **dllsDir** and **sourceISOFiles** variables of **custom_init.cmd**.

In **custom_init.cmd** each variable has a description to explain what it defines:

- **rootPath** – defines a root folder of OPKTools or WAIK. By default it will be **C:\Program Files\Windows OPK\Tools** or **C:\Program Files\Windows AIK\Tools**;
- **toolsDir** – defines a path to the **imagex.exe** utility;
- **PEToolsDir** – defines a path to **peimg.exe**;
- **dllsDir** – defines a path to a folder with DLL libraries required for proper functioning of the **imagex.exe** and **peimg.exe** utilities;
- **sourceISOFiles** – defines a path to a folder containing all the necessary files to prepare an ISO or PXE image and PXE. It also contains a path to the **winpe.wim** file, a raw WIM image;

The **toolsDir**, **PEToolsDir**, **dllsDir**, **sourceISOFiles** variables are pre-defined relatively to the "root path" of OPKTools, which however can be modified according to your needs. So if you've installed WAIK or OPK not by their default paths, you can only change the **rootPath** variable, all the others will be defined accordingly.

ADDING PROGRAMS TO A WIM IMAGE

1. Create a new folder;
2. Copy into it programs you need to add to the resulted WinPE image. All files and directories of that folder will be copied to the WIM image keeping the subdirectories structure intact;
3. Run **custom.cmd** with the following parameters:

```
custom.cmd raw.wim /p:winpe_custom /wim customized.wim
```

Where:

- o **raw.wim** is an image to be customized;
- o **customized.wim** is an output image;
- o **winpe_custom** is a directory containing programs to copy to the resulted WIM image.



Once the `/wim` option is set, the raw image will not be modified.

If you need to create a bootable ISO image of the custom WIM image, please use the `/iso` parameter of `custom.cmd`:

```
custom.cmd raw.wim /p:winpe_custom /iso image.iso
```

You can also use the `/wim` and `/iso` parameters together to prepare images of both types:

```
custom.cmd raw.wim /p:winpe_custom /wim customized.wim /iso image.iso
```

ADDING SPECIFIC HARDWARE DRIVERS

1. Create a new folder;
2. Unpack into it a driver package supplied by a vendor of your device;
3. Run `custom.cmd` with the following parameters:

```
custom.cmd winpe.wim /d:custom_drivers inject
```

As a result your drivers will be added to the resulted WIM image and be automatically loaded during the WinPE boot process. In case you don't want to load the drivers automatically, please use the following command line parameters for the script:

```
custom.cmd winpe.wim /d:custom_drivers copy
```

If you need to create a bootable ISO image of the custom WIM image, please use the `/iso` parameter of `custom.cmd`:

```
custom.cmd winpe.wim /d:custom_drivers inject /iso image.iso
```

or

```
custom.cmd winpe.wim /d:custom_drivers copy /iso image.iso
```

EDITING THE STARTUP MENU AND ADDING SCRIPTS

To configure which additional items (program scripts) you'd like to have in the startup menu, you need to accomplish a number of actions.



You're free to replace the default startup menu by any of your choice.

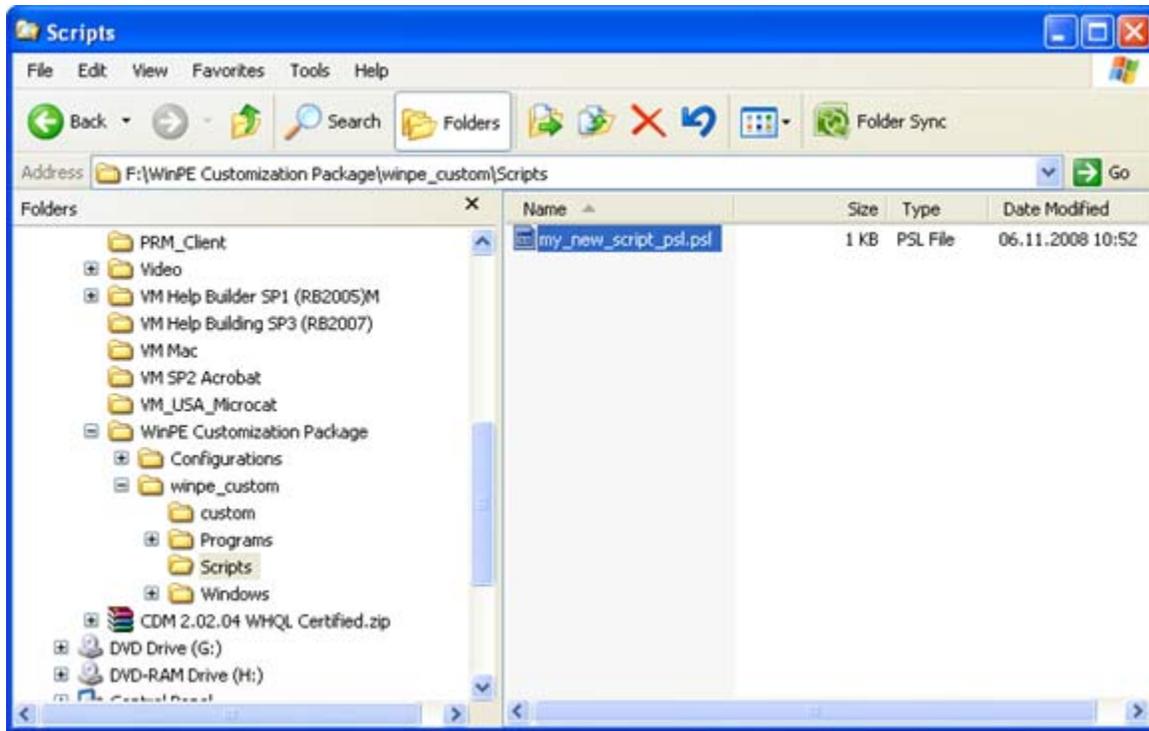
GENERATING A SCRIPT FILE (.PSL FILE)

In the current version of the product you cannot use scripts generated under Windows with custom WinPE 2.1 or Linux based bootable images. Thus you've got the option to make a script either with a custom Linux or WinPE 2.1 bootable image. So let's consider here how to make it with a custom WinPE 2.1 bootable image:

1. Create a directory to contain program scripts (in our case it's named "Scripts" and is located in the same directory as "Custom");
2. [Make a custom WinPE 2.1 bootable image](#);
3. Boot from the created image;
4. Launch Hard Disk Manager in the startup menu;
5. Make sure the virtual mode of execution is enabled;
6. Carry out with the program all operations you need to be scripted;
7. Call the Generate Script dialog in the Main Menu: **Tools > Generate Script...**;



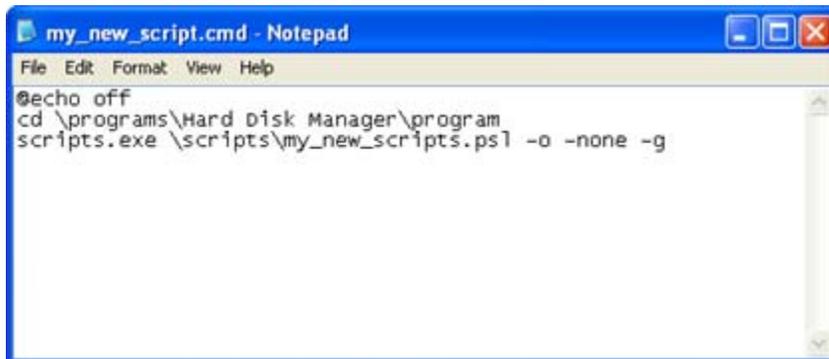
8. Save the resulted **.psl** file as **my_new_script.psl** in the **winpe_custom\Scripts** folder.



GENERATING A BATCH FILE TO EXECUTE THE SCRIPT FILE

Now you need a batch file to execute the previously created script:

1. Enter the **Paragon winpe_custom\Scripts** folder;
2. Create a new textual file and open it with MS Notepad;
3. Rename this file to **my_new_script.cmd** and carry out some editing:
 - Type a text as shown below (the **\programs\Hard Disk Manager\program** directory is not mandatory and is used as an example. You can use here any path to a folder with **scripts.exe**);



- Save the changes and exit MS Notepad.

ADDING THE NEW BATCH FILE TO THE STARTUP MENU

At the end of the startup process, WinPE runs a program which path is defined by the **AppPath** setting in the **Windows\System32\winpeshl.ini** file. Generally, this file contains a path to a program considered as a "shell" for WinPE. If it's missing, the **startnet.cmd** script will be run located in the same directory as **winpeshl.ini**, i.e. **X:\Windows\System32**.

In WinPE solutions delivered by Paragon, it's Paragon Application Launcher that is run after the WinPE startup. It's named **runner.exe** and uses the **runner.ini** file that is copied from **\custom\menu.cfg** to the runner's directory during the WinPE startup. So customization of the startup menu items implies editing of the **menu.cfg** file.

By editing **menu.cfg** you can create your own menu items to execute custom scripts as shown in the example below:

The image shows a split view. On the left is the **menu.cfg** file content, and on the right is a graphical representation of the startup menu it creates. Callouts explain various parts of the configuration and the resulting UI.

```
[Menu 0]
Name_0=Hard Disk Manager
Hint_0=Run Hard Disk Manager
Execute_0="%ParagonPath%\launcher.
Iconpath_0=launcher.ico

Name_1=Network
Hint_1=Mount a network share
Execute_1="%SystemRoot%\system32\mount.cmd"
Iconpath_1=network.ico

Name_2=-
Hint_2=-
Iconpath_2=-
Execute_2=-

Name_3=Scripts
Hint_3=Customized Scripts
Iconpath_3=scripts.ico
Execute_3=
SubItems_3=Scripts SubMenu

Name_4=-
Hint_4=-
Iconpath_4=-
Execute_4=-

Name_5=Restart
Hint_5=Restart the computer
Iconpath_5=restart.bmp
Execute_5=Exit

Name_6=Turn Off
Hint_6=Turn off the computer
Iconpath_6=turnoff.bmp
Execute_6=PowerOff

[Scripts SubMenu]
Name_0=Add your script here 1
Hint_0=Here you can put your first customized script
Iconpath_0=scripts.ico
Execute_0="%ScriptPath%\script1.cmd"

Name_1=Add your script here 2
Hint_1=Here you can put your second customized scrip
Iconpath_1=scripts.ico
Execute_1="%ScriptPath%\script2.cmd"
```

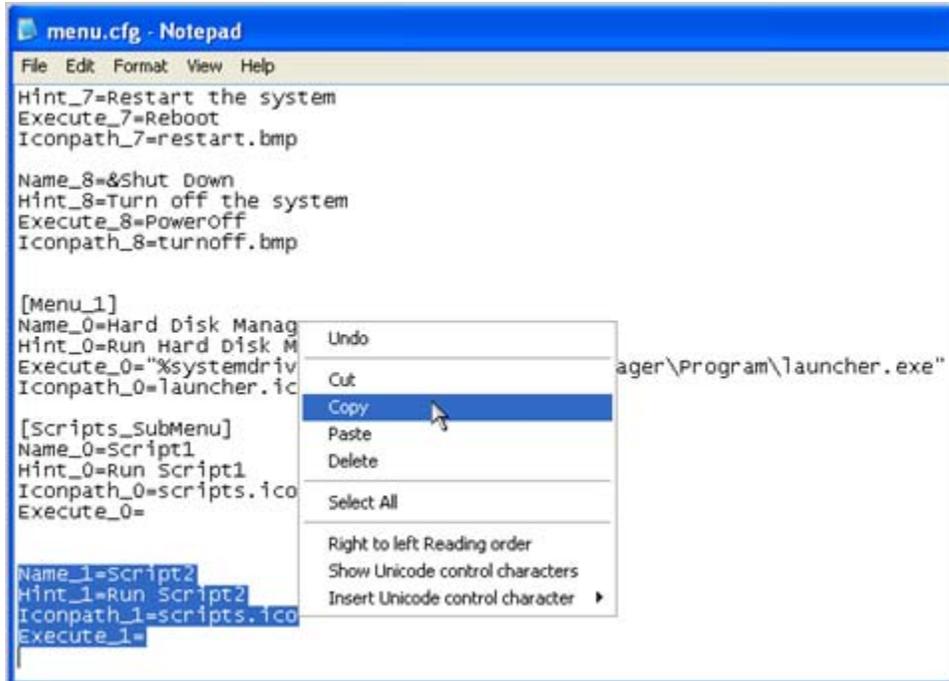
Callouts:

- A:** - This is root of the startup menu (points to [Menu 0])
- B:** - This has been added to create a submenu item of the "Scripts" menu (points to SubItems_3=Scripts SubMenu)
- C:** - These are submenu items of the root item "Scripts" (points to the [Scripts SubMenu] section)
- These are line separators** (points to Name_2=-)
- Name of the menu** (points to Name_0=Add your script here 1)
- Path to the batch file** (points to Execute_1="%ScriptPath%\script2.cmd")
- Icon of the menu item** (points to Iconpath_0=scripts.ico)

The graphical interface shows a menu with items: Hard Disk Manager (A), Network, Scripts (B), Restart, and Turn Off. The Scripts menu is expanded to show sub-items: Add your script here 1 (C) and Add your script here 2.

To add the newly created batch file to the "Scripts" group of the startup menu, please do the following:

1. Find **menu.cfg** located in the **Paragon winpe_custom\custom** folder;
2. Open this file with a text editor and scroll down up to the [Scripts_SubMenu] section;
3. Copy and paste the last item of that section to use as a template;



```

menu.cfg - Notepad
File Edit Format View Help
Hint_7=Restart the system
Execute_7=Reboot
Iconpath_7=restart.bmp

Name_8=&shut Down
Hint_8=Turn off the system
Execute_8=Poweroff
Iconpath_8=turnoff.bmp

[Menu_1]
Name_0=Hard Disk Manag
Hint_0=Run Hard Disk M
Execute_0="%systemdriv
Iconpath_0=launcher.ic

[Scripts_SubMenu]
Name_0=Script1
Hint_0=Run Script1
Iconpath_0=scripts.ico
Execute_0=

Name_1=Script2
Hint_1=Run Script2
Iconpath_1=scripts.ico
Execute_1=

```

4. Edit it by changing **Name_1**, **Hint_1**, **Iconpath_1**, and **Execute_1** to **Name_2**, **Hint_2**, **Iconpath_2**, and **Execute_2**;
5. Set value of the "**Execute_2**" parameter by providing a path to your new batch file "**my_new_script.cmd**";

```

menu.cfg - Notepad
File Edit Format View Help
Hint_7=Restart the system
Execute_7=Reboot
Iconpath_7=restart.bmp

Name_8=&shut down
Hint_8=Turn off the system
Execute_8=Poweroff
Iconpath_8=turnoff.bmp

[Menu_1]
Name_0=Hard Disk Manager
Hint_0=Run Hard Disk Manager
Execute_0="%systemdrive%\programs\Hard Disk Manager\Program\launcher.exe"
Iconpath_0=launcher.ico

[Scripts_submenu]
Name_0=Script1
Hint_0=Run Script1
Iconpath_0=scripts.ico
Execute_0=

Name_1=My script
Hint_1=Run the newly added script
Iconpath_1=scripts.ico
Execute_1=\scripts\my_new_script.cmd

```

6. Save the changes and close the editor. Your startup menu has been customized, so you can make a custom WinPE image.

CREATING A WINPE ISO IMAGE (CUSTOMPE.ISO)

Once you have all your scripts and batch files in place you [can generate a custom bootable image based on WinPE 2.1](#) by using **custom.cmd**.

PREPARING A FOLDER FOR THE PXE STARTUP

You can find all the necessary information on the subject in the "Creating WinPE 2.1 PXE Bootable Image" document that is also included into the package.