

Customizing WindowsPE 2.0 For Use With
Specific OEM, PnP Hardware Drivers, Menu
Items, and Paragon Scripts Using Paragon
Customization Scripts

Contents

Introduction	3
What is needed	4
Build WinPE DM Client using example scripts	4
Using build.cmd	5
Using build_pxe.cmd	5
Customization of existing wim image	5
Adding programs into wim image	6
Installing specific hardware drivers.....	7
Editing initial Deployment Manager startup settings	8
Editing Start Menu and Script Items.....	10
Configuring menu items and scripts	10
Generating a Script File (.psl file).....	10
Generating a Batch File That Executes the .psl Script File.....	11
Adding the New Batch File to the Menu	12
Creating the Windows PE ISO image (CustomPE.iso)	15
Preparing a folder for PXE boot.....	15

Introduction

This document is designed to show how to edit and customize a Windows PE CD based on Windows Vista for specific hardware drivers, menu items, and Paragon scripts using Microsoft OEM Preinstallation Kit. Also it describes how to prepare a folder containing all necessary binaries for WinPE 2.0 boot through PXE.

Due to licensing issues Paragon is not permitted to distribute any tools from Microsoft which would be used to create or edit Windows PE images so all the tools required for WinPE customization must already present on the customer's side. The following will now demonstrate several important issues in adding OEM and PNP drivers using Paragon Customization Scripts to a Windows PE wim image.

The Customization Package includes the following components:

- *winpe_custom* – directory with default Paragon components for the WinPE 2.0 Deployment Manager Client;
- *dm_custom.cmd* – the script that implements customization functionality;
- *makepxe.cmd* – a script for PXE boot directory structure preparation;
- *dm_custom_init.cmd* – “header” file with environment variables definitions, paths to OPKTools executables, etc. It is called both by *dm_custom.cmd* and *makepxe.cmd*, so any changes made to *dm_custom_init.cmd* will affect these two customization scripts;
- *build_image.cmd* – a sample script file demonstrating how to use *dm_custom.cmd*. This example can be used to make customized wim image containing custom drivers, programs, settings and so on as well as to prepare bootable ISO image with customized WinPE 2.0. It can be used to quickly prepare ready-to-use bootable ISO image with user-specified set of custom programs, drivers, settings, etc, as well as stand-alone wim images containing necessary custom set.
- *build_pxe.cmd* – a sample script file demonstrating how to use *makepxe.cmd*. This example can be used to quickly prepare ready-to-use PXE environment to boot WinPE with user-specified set of custom programs, drivers, settings, etc. Strings starting with “rem” inside this script contain several cases of use of *makepxe.cmd*;
- *build.cmd* – a sample script file demonstrating how to prepare PXE environment from default wim image, supplied with OPK or WAIK, using *dm_custom.cmd* and *makepxe.cmd*.
- *build_simple_restore.cmd* – a batch file to create WinPE CD/DVD for automatic restoration of the HDD or Partition. After building the WinPE CD/DVD the .pbf image of the HDD/Partition should be placed into the root on this CD/DVD with Boot Media Builder program.

What is needed

In order to get started the following things are required:

- Supplied from Paragon
 - o Customization scripts
 - o Windows PE directory structure containing the binary files to be added (by default it is named `winpe_custom`)
 - o Paragon Deployment Manager installed
- Not supplied from Paragon
 - o Tools for wim images and WinPE 2.0 maintaining (`imagex`, `peimg`, dll libraries and `wimfltr.sys` driver) and supplementary files for bootable images creation. All these components are included into Windows Automated Installation Kit (WAIC) or OEM Preinstallation Kit (OPK) packages, so you need to install one of these packages in order to successfully customize WinPE 2.0 images. For more information regarding these software kits please refer to Microsoft documentation.
 - o Wim image which will be used as a raw. You can use default image `winpe.wim` which is included both into OPK and WAIC and located in `C:\Program Files\Windows AIK\Tools\PETools\<platform>` (for WAIC) or `C:\Program Files\Vista OPK\Tools\PETools\<platform>` (for OPK), where `<platform>` is “x86” or “amd64”.

Build WinPE DM Client using example scripts

These scripts already have default settings which define paths for OPKtools, raw wim image and PXE boot directory. These parameters are defined by the following variables:

- Path to OPKTools: `rootPath` variable in `dm_custom_init.cmd`
- raw wim image: `%sourceISOFiles%\winpe.wim` in `build_image.cmd`
- PXE boot directory: `C:\Program Files\Paragon Software\Deployment Manager\PXEServer` in `build_pxe.cmd`

Default paths are:

- for OPK: `C:\Program Files\Vista OPK\`
- for WAIC: `C:\Program Files\Windows AIK\`
- for Deployment Manager: `C:\Program Files\Paragon Software\Deployment Manager\`
- for PXE boot directory: `C:\Program Files\Paragon Software\Deployment Manager\PXEServer\`

Using `build_image.cmd`

This image can be used to make WinPE 2.0 bootable ISO image with customized set of programs supplied from Paragon. It can be used to quickly prepare ready-to-use bootable ISO image with user-specified set of custom programs, drivers, settings, etc, as well as stand-alone wim images containing necessary custom set.

Having run this script, you will get a wim image (customized.wim) prepared from the wim image, supplied with OPK (or WAIK), and contents of *winpe_custom* directory supplied from Paragon.

Using build.cmd

Having run this script, you will get completely configured PXE environment containing a wim image prepared from the raw, supplied with OPK (or WAIK), and contents of *winpe_custom* directory supplied from Paragon.

Using build_pxe.cmd

This example demonstrates how to get included already existing wim image into PXE and quickly prepare ready-to-use PXE environment with WinPE 2.0.

Having run this script, you will get completely configured PXE environment containing a image a wim image (customized.wim) prepared from the wim image, supplied with OPK (or WAIK), and contents of *winpe_custom* directory supplied from Paragon.

You can modify all these scripts and scripts they run as you wish in order to accomplish various WinPE 2.0 customization tasks. Below some of use cases are described.

Customization of existing wim image

Using the Paragon Customization Pack you can add custom programs, applications and their settings, manually generated scripts, device drivers, start menu settings, WinPE settings an so on into already existing WimPE 2.0 wim-image. Use Cases below cover all of these tasks.

First of all, before starting a customization process, you need to set proper paths in the *rootPath*, *toolsDir*, *PEToolsDir*, *dllsDir* and *sourceISOFiles* variables of the *dm_custom_init.cmd* script. These environment variables are already defined, but you can tune them according to your needs and requirements. In the text of *dm_custom_init.cmd* each environment variable has a descriptive comment explaining what this variable does define. Nevertheless, we will comment some variables meaning:

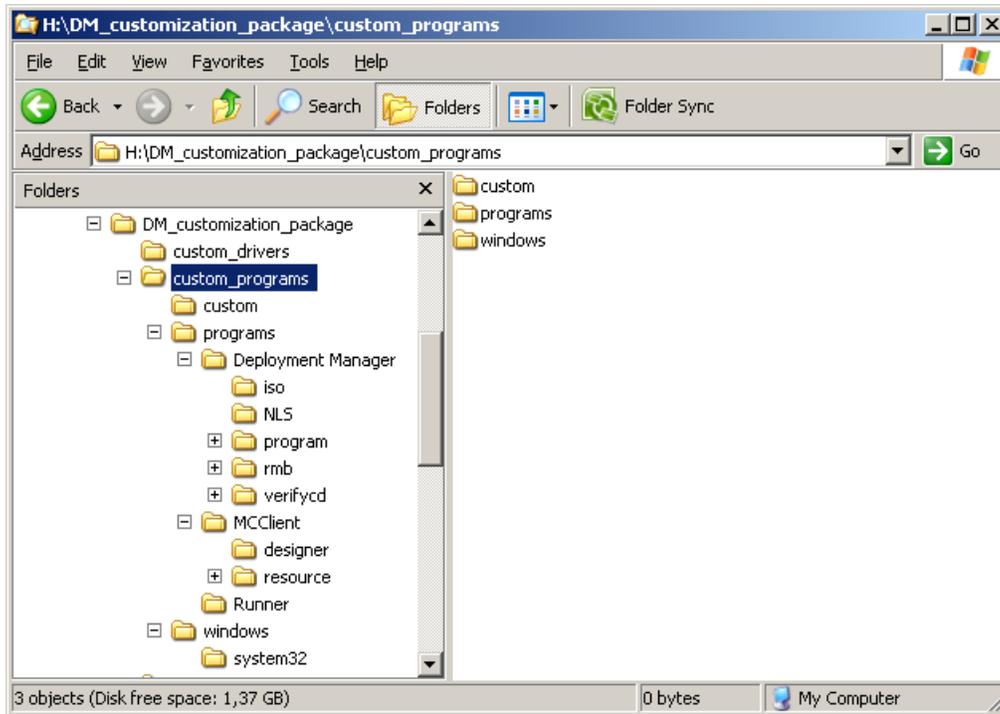
- *rootPath* – defines a root folder containing necessary tools. OPKTools and WAIK are usually installed into the “Program Files” folder, so if you have one of these software packages installed on your machine, *rootPath* may look like *C:\Program Files\Vista OPK\Tools* or *C:\Program Files\Windows AIK\Tools*.
- *toolsDir* – defines a path to *imagex.exe* utility;
- *PEToolsDir* – defines a path to *peimg.exe*;
- *dllsDir* – defines a path to a folder containing DLL libraries which are required for proper functioning of *imagex.exe* and *peimg.exe* utilities;
- *sourceISOFiles* – defines a path to a folder which contains necessary files for preparing ISO images and PXE environment. Also it contains a file *winpe.wim* which may be used by default as a raw in example scripts *build.cmd*, *build_image.cmd* and *build_pxe.cmd*;

Variables *toolsDir*, *PEToolsDir*, *dllsDir*, *sourceISOFiles* are pre-defined relatively to the “root path” of OPKTools, but is not obligatory and you can modify them according to your requirements. If you have installed WAIK or OPK, you can modify only the *rootPath* variable – all other variables will be set automatically due to their relativity to *rootPath*.

Adding programs into wim image

Create a folder and put into it all programs you need to be rolled into the WinPE image. All the files and directories from this folder will be copied into wim image saving subdirectories structure relatively to the wim image root.

On the screenshot below this folder is called *winpe_custom* and located in *DM_customization_package* folder on disk H, but this is not obligatory, and you can name and locate this folder as you want.



Run *dm_custom.cmd* with the following parameters:

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

where *raw.wim* is an image to be customized, *customized.wim* is an output image, and *winpe_custom* is a directory containing programs to be put into wim image. In other words, the formula *raw.wim + winpe_custom = customized.wim* is absolutely correct.

Note: Once /wim option is set, “raw” image will not be modified.

If you need to create bootable ISO image containing customized wim image use “/iso” parameter of *dm_custom.cmd*:

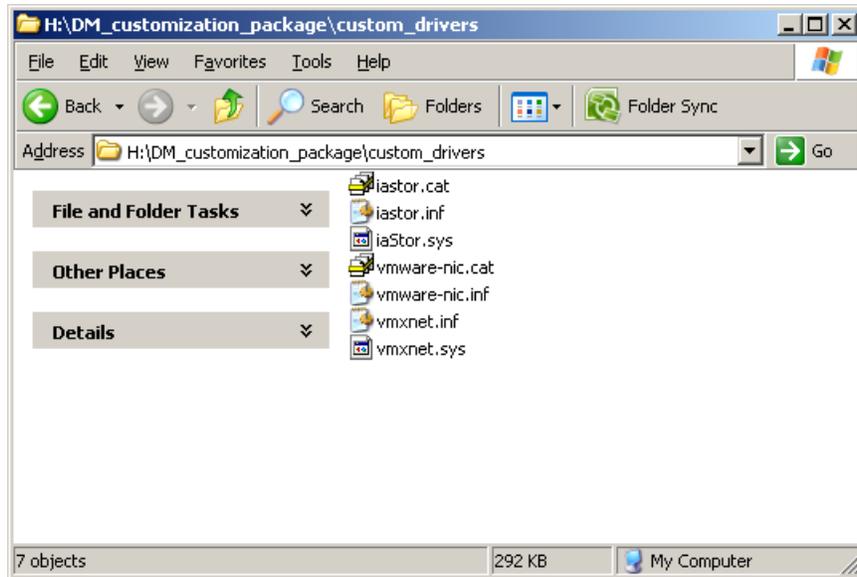
```
dm_custom.cmd raw.wim /p:winpe_custom /iso image.iso
```

Also /wim and /iso parameters may be combined in cases if you want to prepare both customized wim image (to put it into PXE environment, for instance) and bootable ISO image containing customized wim:

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

Installing specific hardware drivers

Create a folder and unpack driver package supplying from your device vendor into it. Your driver package should contain files like the one below. It is also possible that your disk contains multiple *.inf, *.sys, *.dll, *.din, and other files for different revisions of your hardware.



Run *dm_custom.cmd* with the following parameters:

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

This will inject the drivers into WinPE wim-image, so the drivers will be loaded during WinPE boot process. If you need only to copy drivers in order to load them manually after WinPE boot, use the following command line parameters for the script:

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

This will copy the whole folder with drivers into the root folder of the wim-image.

Note: If you need to prepare bootable ISO-image with WinPE you should use “/iso” command line parameter:

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

or

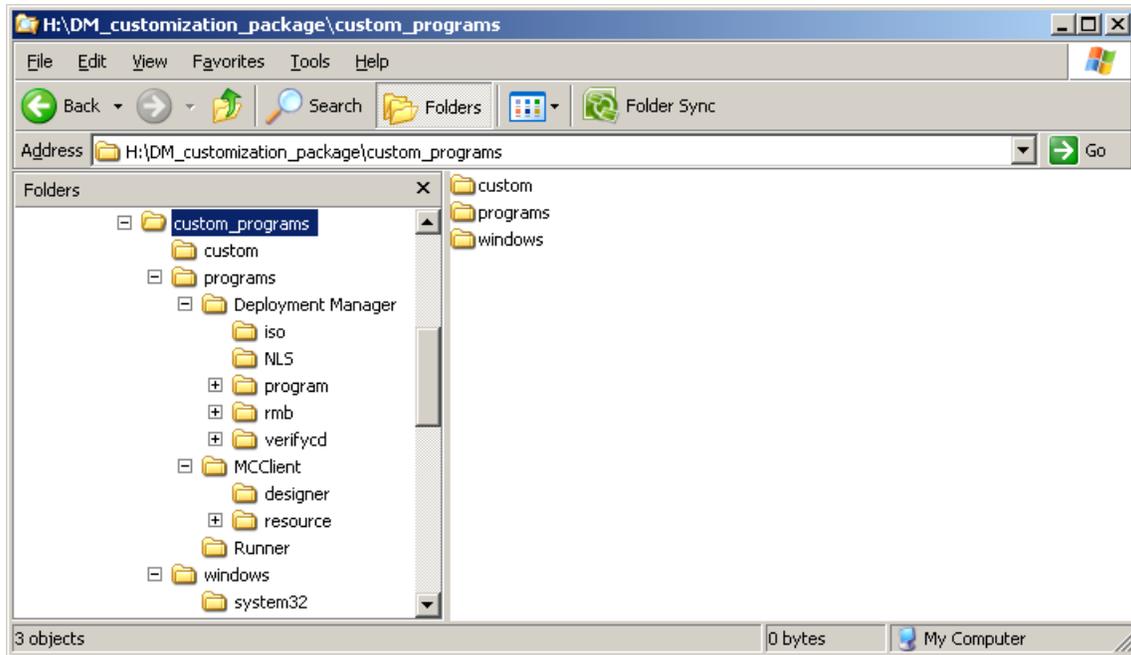
```
dm_custom.cmd winpe.wim /d:custom_drivers copy /iso image.iso
```

Editing initial Deployment Manager startup settings

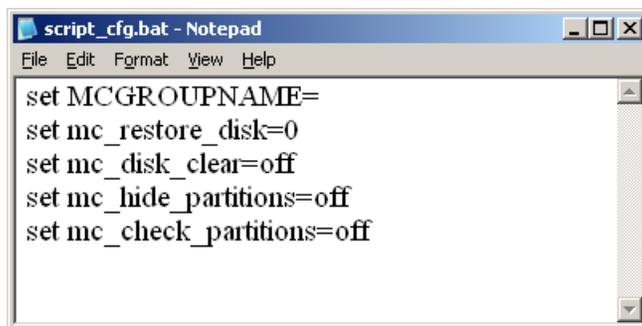
This section describes how to modify and tune settings which are used during Deployment Manager Client run. All definitions are made in the script *script_cfg.bat* which is located in **custom** directory of the Customization Package.

In order to edit the *script_cfg.bat* do the following:

1. Begin by unzipping the *DM_customization_package.zip* file to a location of your choice
2. Once that is done there should be a structure like the one below



3. Find a file `script_cfg.bat` in **custom** directory and open it in a text editor (notepad, for instance)



4. Set values according to your needs modifying values after the “=” symbol
5. Save the file (usually Ctrl+S or Shift+F12) and close the text editor

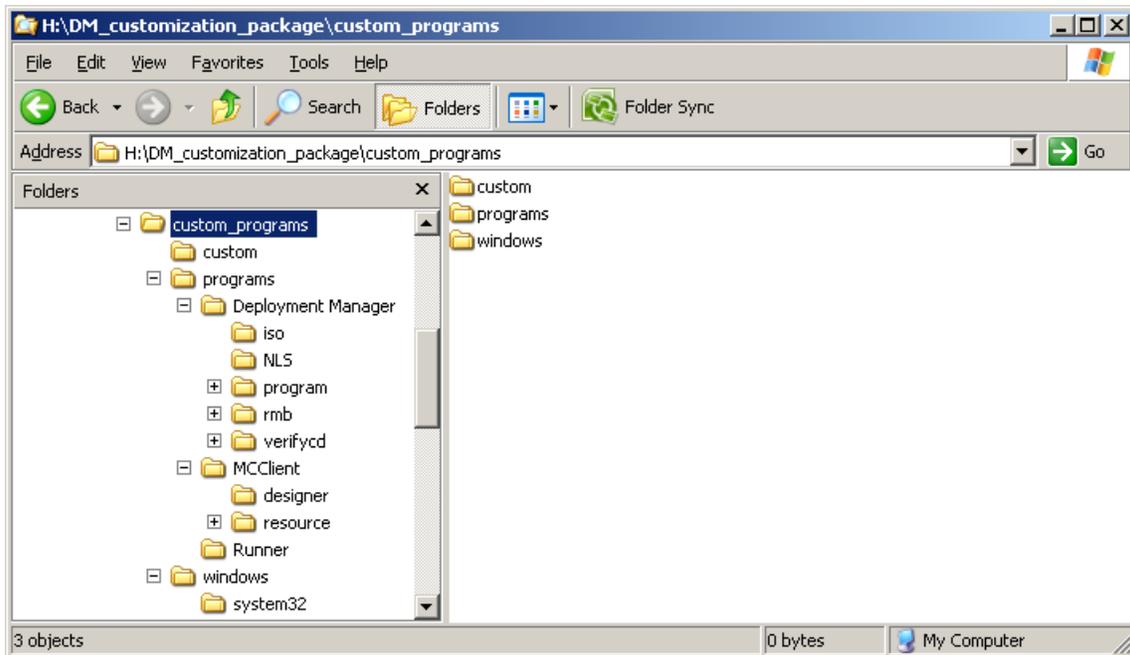
Default values of the variables which are already listed in this file:

Variable name	Allowed values	Pre-defined value	Comment
MCGROUPNAME	Any letters and numbers	<i>Not defined</i>	Session name
mc_restore_disk	0...128	0	Default HDD for deploy
mc_disk_clear	on, off	off	Clear HDD before deploying
mc_hide_partitions	on, off	off	Hide all partitions after deploy
mc_check_partitions	on, off	off	Check file system integrity after deploy

Editing Start Menu and Script Items

This section covers some important issues regarding adding menu items and Paragon scripts to the supplied start menu. The following will now demonstrate this.

1. Begin by unzipping the *DM_customization_package.zip* file to a location of your choice
2. Once that is done there should be a structure like the one below
3. With the files in place we can now start to add the hardware drivers of your choice



Configuring menu items and scripts

To control which items and scripts you would like to set up in your menu you have to edit a .cmd batch file which will be executed automatically at the end of the WinPE boot process. A short example shows you how to generate a .psl script file which controls the operation, create a batch which executes the script and then add this operation to the automatically started menu.

Note: Of course you can always replace the default menu by any other menu system designed by you yourself.

Generating a Script File (.psl file)

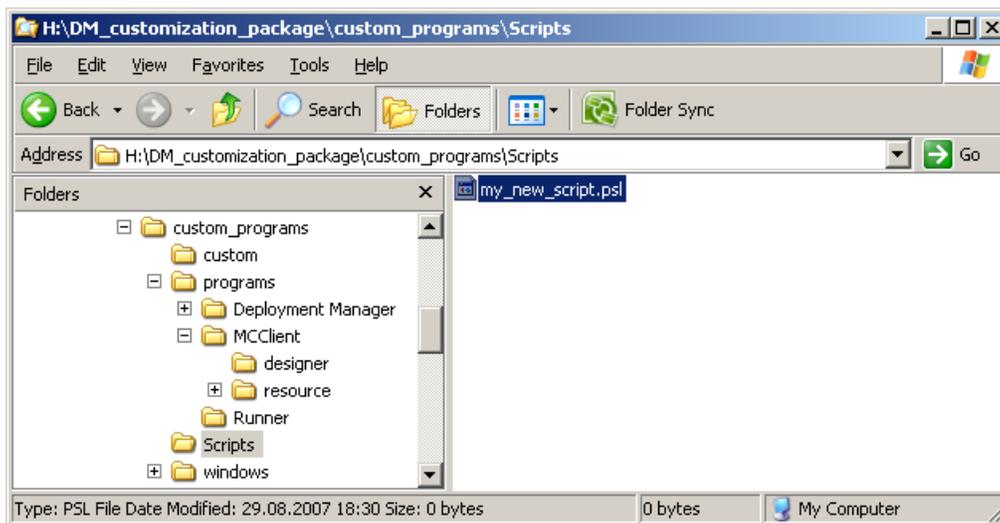
The simplest way to generate a script is to use the Paragon Windows GUI.

1. Create a directory which will contain scripts in the future (in our example we'll assume that this directory names "Scripts" and locates at the same dir as "Custom" and "MCClient")
2. Start the Paragon Windows installed part of the software

3. Set up the wanted operation by using a wizard
4. Instead of applying this operation with the green apply button, use the “Tools” -> “Generate Script...” like in the example below (your screen might look differently depending on which Paragon program is installed)..



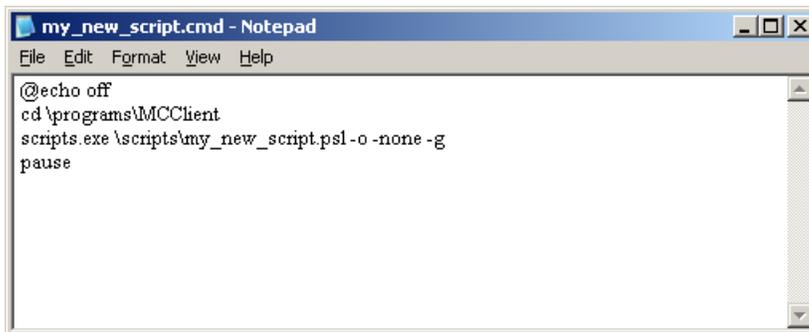
5. Now save this .psl file as my_new_script.psl in the folder *winpe_custom\Scripts* like you see in the image below.



Generating a Batch File That Executes the .psl Script File.

1. Now you need a batch file which will execute the previously created script. The following will now demonstrate this.
2. Enter into the Paragon *winpe_custom\Scripts* folder

3. Create new text file and open it in MS Notepad
4. Rename this file to *my_new_script.cmd* and edit it
5. Type a text as shown below (\programs\MCClient directory is not mandatory and used as an example. You can put any path to a folder containing scripts.exe)



```
my_new_script.cmd - Notepad
File Edit Format View Help
@echo off
cd \programs\MCClient
scripts.exe \scripts\my_new_script.ps1 -o -none -g
pause
```

6. Save the changes of this file and exit.

Adding the New Batch File to the Menu

At the end of the boot process WinPE always runs the program which path is defined by AppPath setting in windows\system32\winpeshl.ini file. Generally, this file contains path to a program which is considered as a “shell” in Windows PE. If this file is absent WinPE runs startnet.cmd script which is located at the same dir as winpeshl.ini – X:\windows\system32

In WinPE solutions from Paragon, Paragon Application Launcher is started after WinPE boot. This program has executable named *runner.exe* and uses ini-file *runner.ini* which is copied from *\custom\menu.cfg* file into the runner’s directory during WinPE startup. So, customization of menu items set brings to editing of the *menu.cfg* file.

With the *menu.cfg* it is possible to create your own menu items and subgroup items for use with your customized scripts and batches as seen in the below image but for the purpose of this demonstration you will only need to add a new subgroup item to the root item “Scripts”. The image below also shows how the start menu in general is built.

```
[Menu_0] <---- A. This is the root of the start menu
Name_0=Hard Disk Manager
Hint_0=Run Hard Disk Manager
Execute_0="%ParagonPath%\launcher.exe"
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= <---- These are line separators
Execute_2=-

Name_3=Scripts
Hint_3=Customized Scripts
Iconpath_3=scripts.ico
Execute_3=
SubItems_3=Scripts SubMenu <---- B. This is added to create
the submenu items of
the root item 'Scripts'.

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

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

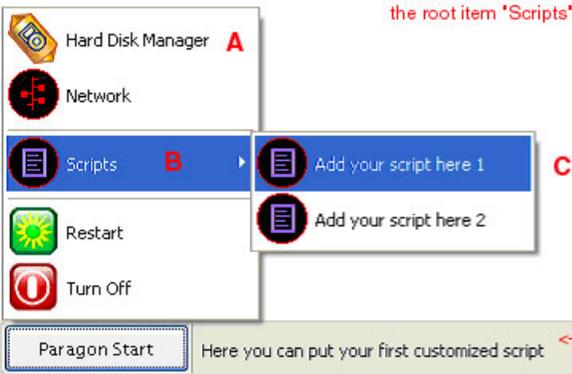
Name_6=Turn Off
Hint_6=Turn off the computer
Iconpath_6=turnoff.bmp
Execute_6=PowerOff

[Scripts SubMenu] <---- C. this group.
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 <---- Menu item name
Hint_1=Here you can put your second customized script
Iconpath_1=scripts.ico <---- Icon next to menu item name
Execute_1="%ScriptPath%\script2.cmd" <-- Change to your path
for example "x:\my.bat"
```

These are the submenu items of the root item 'Scripts'. These items are the commands or scripts to be executed in

Note: You can create new items and subgroups yourself by following this example. Be sure to follow the numbering sequence accordingly. For example, if the last number in the menu group is "Name_1", "Hint_1", "Iconpath_1", "Execute_1" then of course the new item that you will add will use "Name_2", "Hint_2", "Iconpath_2", "Execute_2"



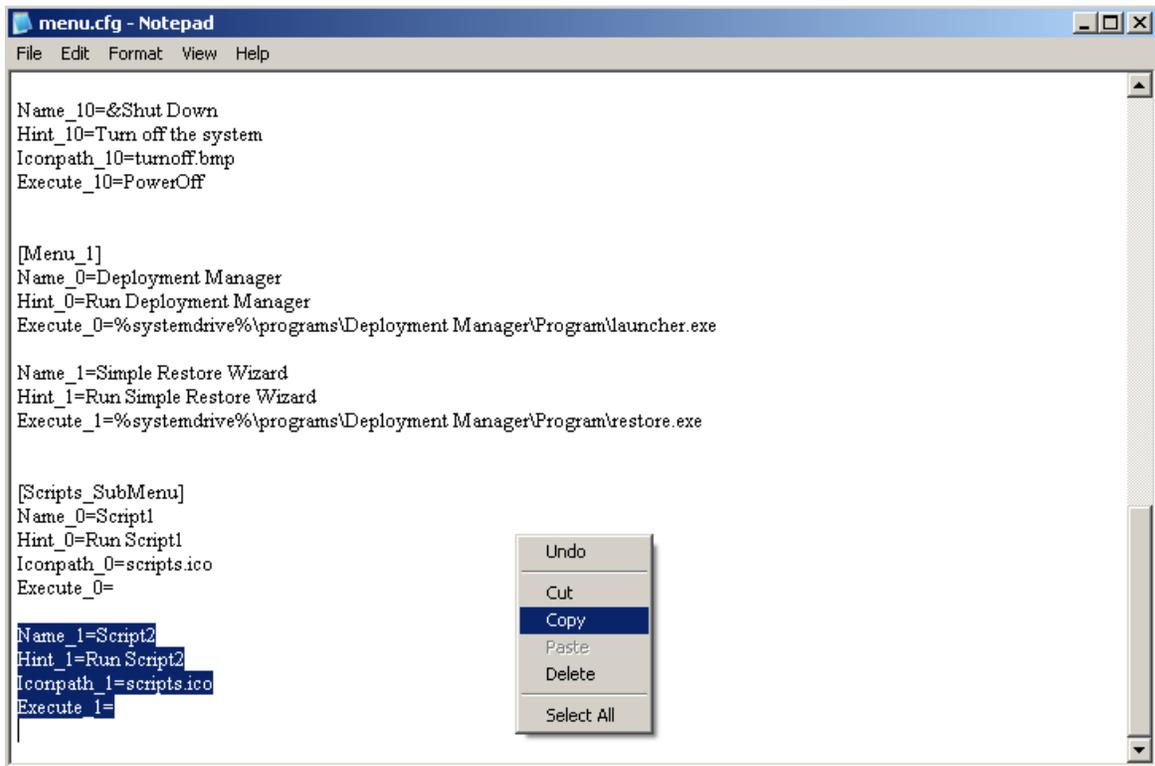
Paragon

Time 23:20:48

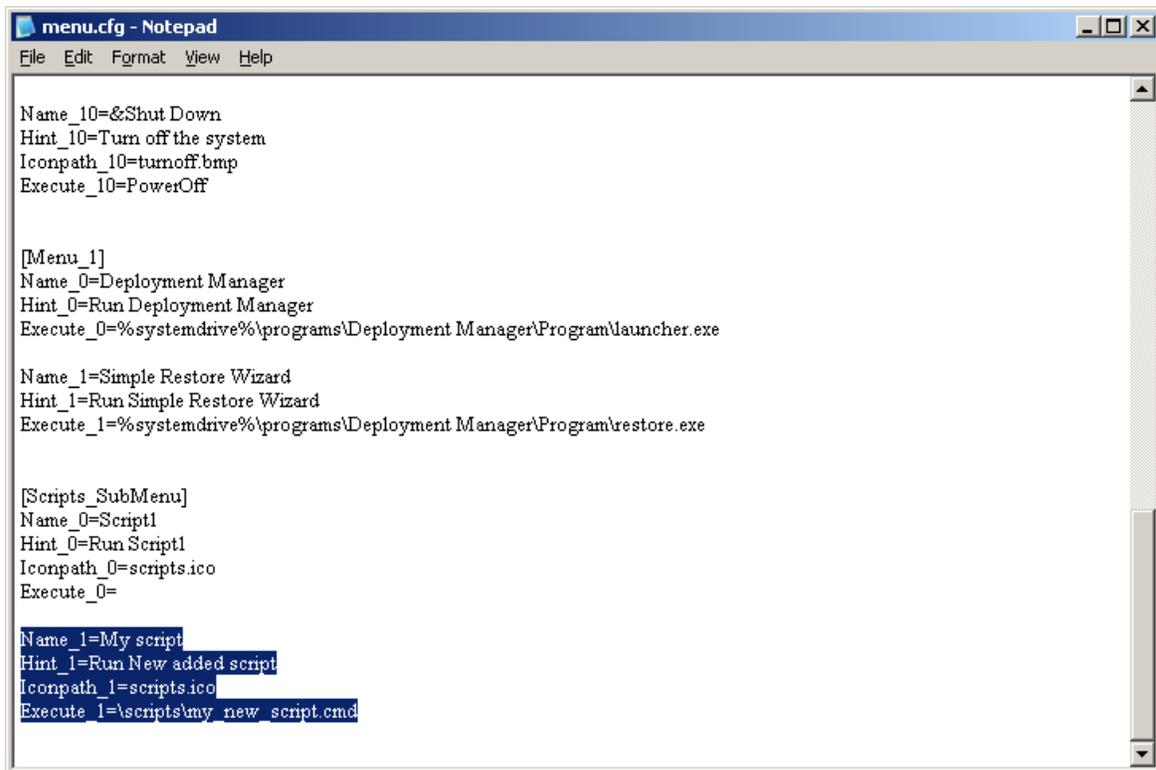
<---- This is the hint text which can be changed under the Hint_x option

It will now be demonstrated on adding the new batch file to the start menu under the “Scripts” group.

1. Find the menu file *menu.cfg* located in the *Paragon winpe_custom\custom* folder
2. Edit this file in a text editor and scroll down to the bottom to the [Scripts_SubMenu] section. At this point you can copy and paste the last item in that section to be used as a template as shown below:



3. Once the last item is copied over as a template you can then edit it at first by changing the Name_1, Hint_1, Iconpath_1, and Execute_1 to Name_2, Hint_2, Iconpath_2, and Execute_2. Once that is complete you will then only need to fill the value of the "Execute_2" parameter to the name of your new batch file "my_new_script.cmd". The below image shows this after it has been edited.



```
menu.cfg - Notepad
File Edit Format View Help

Name_10=&Shut Down
Hint_10=Turn off the system
Iconpath_10=turnoff.bmp
Execute_10=PowerOff

[Menu_1]
Name_0=Deployment Manager
Hint_0=Run Deployment Manager
Execute_0=%systemdrive%\programs\Deployment Manager\Program\launcher.exe

Name_1=Simple Restore Wizard
Hint_1=Run Simple Restore Wizard
Execute_1=%systemdrive%\programs\Deployment Manager\Program\restore.exe

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

Name_1=My script
Hint_1=Run New added script
Iconpath_1=scripts.ico
Execute_1=\scripts\my_new_script.cmd
```

4. You are now completed and can save this file. Now your menu is customized and the Paragon WinPE ISO can be generated.

Creating the Windows PE ISO image (CustomPE.iso)

Once you have all your scripts and batch files in place you can now generate your ISO image using the supplied *dm_custom.cmd*. See examples above.

Preparing a folder for PXE boot

This topic is described in detail in document "Creating WinPE 2.0 PXE Bootable Image" supplied with Paragon Deployment Manager customization package. Please refer to that document for more information.