Supplemental Documentation

to Mark Minasi’s Steadier State

by Thomas Freedman

I am the IT guy for a local volunteer library that provides a small number of public use computers. Surprisingly, I didn’t discover Mark Minasi’s Steadier State until this year (2016), although I have conducted numerous searches related to the original Steady State program Micro$oft chose to discontinue, and the built-in version of it initially bundled with early prereleases of Windows 7. Our library is a non-profit organization that receives no subsidies from any government agency and the staff is comprised entirely of volunteers. The operating budget is generated from the sales of books, donations and the $1 and hour fee charged to use the computers.

I have been volunteering my IT services for the past 5 years and it has kept me quite busy dealing with issues brought on by patrons that inadvertently manage to infect the computers with viruses or do things that require my attention to fix, not to mention issues Micro$oft introduces. Besides myself, the library staff has no expertise or desire to manage these issues or help patrons with anything but the most basic computer problems.

Micro$oft doesn’t make it easy for organizations such as our library to run public use computers. I have intended to replace Microsoft Windows with Linux at some point, but most users here are so computer illiterate that such a replacement would be viewed with significant resistance. If such a change is to occur, it must take place gradually and by the user’s own prerogative. That’s only fitting for an all volunteer organization don’t you think?

Another hurdle I had to jump was convincing the library board to cover the cost of upgrading the Windows licenses from Professional to Ultimate. Fortunately this was not as difficult as I anticipated, due in part from the low cost of OEM licenses I obtained on ebay 2 years ago and the improvements to reliability and performance that results from running an OS that doesn’t accumulate crud in the user’s profile and system registry. It is to this last point that Mark Minasi’s Steadier State has been a huge advantage.

Although as originally designed Steadier State requires more time to boot into a steady state, Mr. Minasi’s generous work illustrated the concept very well of how to run Windows from a virtualized VHD filesystem, and paved the way for the changes I implemented to reduce the boot up time by eliminating the secondary [Win PE](https://en.wikipedia.org/wiki/Windows_Preinstallation_Environment) boot cycle his approach required.

The essential innovation was conceived by user cdob on [the reboot.pro](http://reboot.pro/), a forum where IT professionals and computer enthusiasts discuss issues and problems, mostly but not exclusively related to Microsoft Windows. Although [I came up with the general approach](http://reboot.pro/topic/21148-boot-vhd-or-winpe-with-grub2/#entry199203) of using the boot loader rather than [Win PE](https://en.wikipedia.org/wiki/Windows_Preinstallation_Environment) to replace the VHD files with fresh, pristine “template” versions, [it was user cdob that suggested](http://reboot.pro/topic/21148-boot-vhd-or-winpe-with-grub2/#entry199206) to use boot manager entries and alternate between them to achieve the same result. Based on that suggestion I wrote a shutdown script that toggles the default boot manger entry to test the idea. It worked perfectly, and from there I decided to revise Mark Minasi’s Steadier State scripts to take advantage of this new approach.

It was further [suggested by user “Wonko the Sane”](http://reboot.pro/topic/21148-boot-vhd-or-winpe-with-grub2/#entry199211) on the reboot.pro forum that I could use [iPXE’s “wimboot”](http://ipxe.org/wimboot) to load Win PE directly from the grub2 menu, which eliminated the need for a dedicated partition to hold Win PE files. Win PE is required for maintenance of the parent template VHD file, for example to merge in Windows updates or make other changes deemed necessary. So now you have the history and motivation for these changes. I sincerely hope you find them useful. You may contact me on reboot.pro via the user [thomnet](http://reboot.pro/user/67663-thomnet/).

What to do After Running PrepSSdisk.cmd

The remainder of this document describes the steps to take after preparing the target installation drive using Thomas Freedman’s **PrepSSdisk.cmd**. This guide covers two topics. In **Part** **1:** I cover how to deploy your image.vhd so it will work as Windows did on the original system, but with Steadier State functionality so that every reboot results in a fresh, rolled back Windows, and **Part 2:** (optional) how to install a Linux operating system in a dual boot configuration alongside the Windows Steadier State operating system.

Thomas Freedman’s disclaimer for this version of Steadier State:

I have tested this version of Steadier State on a variety of systems both physical and virtual, but **I offer no guarantees at all**… back up things you care about before committing to this.

Remember, there's no way that Microsoft's going to support this, but it's easy to make an image "legal" by adding it to an existing Windows 7 system as a second boot option, a version that boots from a VHD filesystem.

Assumptions:

1. You have read and understand the information provided in Mark Minasi’s original Steadier State zip file, including his Power Point presentation.
2. You have successfully prepped the target system using the **PrepSSdisk.cmd** found on the STDYR‑STATE install media, built using Mark Manasi’s original instructions for **buildpe.cmd**. The **PrepSSdisk.cmd** is a replacement for his **prepnewpc.cmd** script, and his **buildpe.cmd** script has been slightly modified to support Thomas Freedman’s new “rollback” algorithm.
3. You have also created an image.vhd file from a working installation of a Windows Ultimate or Enterprise operating system using the **cvt2vhd.cmd** script in Mark Manasi’s original documentation.
4. If you wish to dual boot Linux and Windows Steadier State, it is assumed that you are comfortable working in the Linux operating system. This document will provide the basic steps but by no means comprehensively covers the possible variations of every Linux distribution or installation scenario.
5. You are an experienced IT professional or are otherwise technically competent enough to perform the steps described herein. **NOTE:** installing Linux is completely optional, and does not affect the operation of Windows, except for how it is booted.

**Part 1 — Image.vhd Deployment and Testing Steadier State Operation**

Completing the Steadier State Windows installation requires only a few more steps. When these steps are finished you will have an installation of Windows that always reboots to the exact same state as the parent “template” image.vhd you captured using the **cvt2vhd.cmd** script. The process is as follows:

1. Copy the image.vhd file onto the root of the “VHD Files” volume, drive letter C:
2. Create a blankDif.vhd differencing file, also known as a “snapshot” file using diskpart:
   1. Boot your system with the STDYR-STATE USB or CD media
   2. diskpart> create vdisk file=C:\blankDif.vhd parent=C:\image.vhd
   3. diskpart> exit
3. Make 2 copies of the blankDif.vhd file (C:\bootDif1.vhd and C:\bootDif2.vhd)
4. Remove the STDYR-STATE media and reboot the system to start Windows
5. Complete the initial operating system setup if the image was sysprepped
6. Customize the image any way you like; install updates, programs, users etc.
7. Create a new script to be executed when Windows is shut down (do this right before final merge) :
   1. Start the group policy editor (run gpedit.msc)
   2. Navigate to Computer Configuration\Windows Settings\Scripts and double-click Shutdown
   3. Click the **Add…** button and enter newBcD.cmd and click **OK**, then **Apply**
   4. Copy D:\Other\newBcD.cmd (verify drive D: is the “VHD Files” volume) to: C:\Windows\System32\GroupPolicy\Machine\Scripts\Shutdown\newBcD.cmd
8. Insert the STDYR-STATE media and reboot into WinPE-SS
9. Type “merge” to update the parent image.vhd file with the changes done in steps 5, 6 and 7.
10. Remove the STDYR-STATE media and reboot to your new Steadier State Windows!

**Test It**

To verify correct rollback operation of Steadier State, do the following under Windows:

1. Login to an account
2. Change something obvious like the desktop background or add a folder to the desktop
3. Reboot and verify the changes you made are gone
4. Repeat steps 1 – 3 to verify the alternate boot entry also works properly
5. That’s it, testing complete!

**If Something Goes Wrong…**

First, review the contents of the “VHD Files” volume. Under Windows it should be drive D: and contain:

|  |  |  |  |
| --- | --- | --- | --- |
| bootDif1.vhd | blankDif.vhd | Boot (directory) | BOOTMGR |
| bootDif2.vhd | image.vhd | Other (directory) |  |

Note that the Boot directory contains many files, all but one of which are standard for any Windows installation. The one exception is the \Boot\osBootGuids.ini file that holds the GUID values of boot manager menu entries. This file contains only 4 lines, as in the following example:

[BootVHDs]

diff1={16161a75-3965-1106-8a05-806e6f6e6963}

diff2={96161a76-3965-1106-8a05-806e6f6e6963}

lastBooted={99999999-9999-9999-9999-999999999999} 🡨 temporary value set by PrepSSdisk.cmd

Please be aware that your first merge may require you to edit the lastBooted value in this file. That is because merge uses this ini file to determine which difference VHD file was actually booted last, and that may not be accurate until the newBcD.cmd script is permanently added to your Windows shutdown process. You might also require more than one reboot to get your parent “template” image.vhd setup the way you want it initially, which is why adding the shutdown script to Windows should be the last or near the last thing to be merged into your permanent image.vhd file.

If your system’s device configuration results in the “VHD Files” volume showing up under a different drive letter, edit the newBcD.cmd shutdown script and change the value for vhdDrive on line 21 ( **) else set vhdDrive=D:** ) to match the drive letter you see with Windows Explorer.

If that was already correct, change the value of the debug variable at the top of the newBcD.cmd shutdown script from false to true, and run the script in a cmd window. Each run of the script should alternate between:

1. Last GUID booted={GUID value A}

Last VHD booted=bootDif1.vhd

Next VHD to boot=bootDif2.vhd

Next GUID to boot={GUID value B}

1. Last GUID booted={GUID value B}

Last VHD booted=bootDif2.vhd

Next VHD to boot=bootDif1.vhd

Next GUID to boot={GUID value A}

Don’t forget to change the debug flag back to false. If the items printed alternate as described, the only other suggestion I have is to verify the GUID values in the D:\Boot\osBootGuids.ini file agree with the above. Good luck!

**Part 2 — Install Linux and the Customized Grub2 Menu**

NOTE: This section covers the topic of installing Linux and a custom grub2 menu to boot Windows, Linux and Win PE from a graphical grub2 theme. If you are a novice Linux user or are unfamiliar with Linux it is highly recommended that you carefully consider undertaking these steps. Remember, the process outlined in this section is entirely optional.

If you are a novice Linux user you may find this process a bit difficult. Installing Linux is not so difficult but the steps required to install the custom grub boot menu are non-standard and at some point should be improved to take advantage of the modularity built into Linux for customization of the grub bootloader. This very basic guide assumes you're installing the most recent Linux Mint distribution, version 18. The process to install other Linux distributions is similar.

Almost any modern Linux will work, though you may want to change the size of the 2 Linux partitions (defined at the top of **PrepSSdisk.cmd**) depending on the distribution (hereafter “distro”) you choose and your specific requirements. I highly recommend one of the Ubuntu family of distros, such as Kubuntu, Lubuntu or my personal favorite: Linux Mint. All have their merits, and the choice is yours to make. I would stick to a modern distro though, so that the grub of the distro will be compatible with the grub files that are copied to the STDYR-STATE media. Grub is constantly evolving and aspects of its’ graphical themes and password security may differ.

Should you choose a distro with a version of grub that isn’t compatible with the one included with this version of steadier state, you will have to manually update your grub to include the theme and grub.cfg files copied from the STDYR-STATE install media. If that is the course you are faced with consider incorporating those changes according to the conventions of customizing grub that utilize scripts under /etc/grub.d/ to provide a modular approach for grub2 customizations that are used to generate the actual grub.cfg file that dictates how the grub2 bootloader works. The major advantage for following those conventions is that your customizations will not be clobbered by Linux OS upgrades. If you don’t do this you risk losing your customizations when you upgrade Linux. We will address that concern via making backup copies of your grub2 configuration, so it can be restored and edited to include the upgraded elements. With caution stated let’s get on with it.

The first thing I recommend you do is backup the Master Boot Record (MBR) of the hard drive using the dd program. This can be done using a Linux installation media, which allows you not only to try out the distro but also install Linux, all from the same installation media. Begin by booting the Linux installation media containing a “live” version of your chosen Linux distro.

To make backup copies of the MBR open a terminal window and enter these commands:

sudo bash

dd if=/dev/sda of=/root/win.mbr bs=446 count=1

mount /dev/sda3 /mnt

cp /root/win.mbr /mnt/Boot/win.mbr

This will make 2 backups of the MBR, one in the Boot folder on the vhdDrive and the other in the root folder of the Linux filesystem. These may be useful in the future to restore the direct Windows boot, should you want to uninstall Linux or change the bootloader .

When finished don't close the terminal window as you'll need it later.

Begin the installation of Linux in the manor appropriate for your chosen distro. If you use a Ubuntu variant like Mint it's quite easy, all you need to do is double click the **Install Linux Mint** icon on the desktop. When you get to the **Installation type** screen showing a list of choices with **Something else** at the bottom, choose the Something else option and click continue. This will take you to the partition editor where you choose what each partition will be used for.

Select the 12GB partition (/dev/sda1) and click the **Change** button, and change the **Use as:** option to Ext4 journaling filesystem. Change the mount point to / and check the box to format the partition, and then click OK to continue.

Repeat the process for /dev/sda2, but choose **swap area** for the **Use as:** drop-down list and click OK. When you're ready to begin the actual OS installation click the **Install Now** button. Provide the requested user / machine info and wait for the installation to complete. When it is finished do NOT reboot, but choose **Continue testing** or just move the **Completed** message dialog window aside to get it out of the way.

Here's where it begins to get interesting. We'll need some files that the **PrepSSdisk** script copied to the “Other” folder on the vhdDrive. Return to the terminal window, and enter the following commands. Note that you may need to remount the sda3 partition if it was unmounted by your distro’s installation:

cd /boot

cp -R grub grubOriginal

cp -R /mnt/Other/grub/\* grub/ **.**

touch advanced

The next step is the most difficult and if you get it wrong the system will not boot, but don't worry, if you make a mistake you can boot the Linux install media again and correct what you need to fix. If necessary you can copy grub.cfg from the grubOriginal folder back to /boot/grub/grub.cfg and try again.

What needs to be done is edit the /boot/grub/grub.cfg file so it contains the same UUID, kernel version and initial ramdisk values found in the /boot/grubOriginal/grub.cfg file. This is tricky so take your time and double check your changes.

The key parts to focus on are highlighted below in red text in the updated grub.cfg file. Note that it rearranges the order of menu entries so that Windows will appear first. If you want to follow Linux conventions so that this customized grub.cfg can be generated by grub utilities using the scripts in /etc/grub.d/, pay attention to the BEGIN lines as well, which dictate the config script element under /etc/grub.d/ that control the contents of that section. The numbers in entries like 30\_os-prober determine the order of processing and inclusion in grub.cfg.

### BEGIN /etc/grub.d/30\_os-prober ###

menuentry 'Windows 7' --class windows --unrestricted --class os \

$menuentry\_id\_option 'osprober-chain-96FCCADBFCCAB4B1' {

savedefault

insmod part\_msdos

insmod ntfs

set root='hd0,msdos3'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos3 --hint-efi=hd0, \

msdos3 --hint-baremetal=ahci0, msdos3 C86C60A66C6090CC

else

search --no-floppy --fs-uuid --set=root C86C60A66C6090CC

fi

parttool ${root} hidden-

chainloader +1

}

### END /etc/grub.d/30\_os-prober ###

### BEGIN /etc/grub.d/10\_linux ###

menuentry 'Linux Mint 18 Cinnamon' --class linuxmint --unrestricted --class gnu-linux --class gnu --class os \

$menuentry\_id\_option 'gnulinux-simple-47db388f-fefa-4f47-8062-8733a9cd2f06' {

savedefault

insmod part\_msdos

insmod ext2

set root='hd0,msdos1'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0, \

msdos1 --hint-baremetal=ahci0,msdos1 47db388f-fefa-4f47-8062-8733a9cd2f06

else

search --no-floppy --fs-uuid --set=root 47db388f-fefa-4f47-8062-8733a9cd2f06

fi

linux /boot/vmlinuz-4.4.0-21-generic root=UUID=47db388f-fefa-4f47-8062-8733a9cd2f06 ro \

quiet splash $vt\_handoff

initrd /boot/initrd.img-4.4.0-21-generic

}

if [ -f /boot/advanced ]; then

submenu 'Advanced Boot Options' --class advanced --unrestricted \

$menuentry\_id\_option 'gnulinux-advanced-47db388f-fefa-4f47-8062-8733a9cd2f06' {

# This entry boots WinPE-SS from a WIM archive, for maintence

menuentry 'Admins Only!' --class winpess {

savedefault

insmod part\_msdos

insmod ntfs

set root='hd0,msdos1'

linux16 ($root)/boot/grub/wimboot

initrd16 \

newc:bcd:($root)/boot/WinPE-SS.bcd \

newc:boot.sdi:($root)/boot/WinPE-SS.sdi \

newc:boot.wim:($root)/boot/WinPE-SS.wim

}

menuentry 'Linux Mint 18 (recovery)' --class recovery --unrestricted --class gnu-linux --class gnu --class os \

$menuentry\_id\_option 'gnulinux-4.4.0-21-generic -advanced-47db388f-fefa-4f47-8062-8733a9cd2f06' {

savedefault

insmod part\_msdos

insmod ext2

set root='hd0,msdos1'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0, \

msdos1 --hint-baremetal=ahci0,msdos1 47db388f-fefa-4f47-8062-8733a9cd2f06

else

search --no-floppy --fs-uuid --set=root 47db388f-fefa-4f47-8062-8733a9cd2f06

fi

echo 'Loading Linux 4.4.0-21-generic ...'

linux /boot/vmlinuz-4.4.0-21-generic root=UUID=47db388f-fefa-4f47-8062-8733a9cd2f06 \

ro recovery nomodeset

echo 'Loading initial ramdisk ...'

initrd /boot/initrd.img-4.4.0-21-generic

}

}

fi

The last step is to copy the WinPE-SS files so that you can boot WinPE-SS directly from grub and not require the STDYR-STATE bootable USB or CD. The “Admins Only” menu item in the *Advanced Boot Options* section will only appear if a file by the name of **advanced** exists in the /boot/grub folder. To hide the Advanced submenu rename or delete the **advanced** file.

There may be a way to code the ability of grub to look for a file on a removable media device, but I ran out of time to figure out how. To boot WinPE-SS, either boot directly from the STDYR-STATE media or start Linux and make sure the **advanced** file exists in /boot/grub.

To install WinPE-SS so it can be booted without the STDYR-STATE media, insert the STDYR-STATE media and wait for it to appear on the desktop. Using a terminal running under Linux, enter these commands:

cp /media/mint/STDYR-STATE/BOOT/BCD /boot/grub/WinPE-SS.bcd

cp /media/mint/STDYR-STATE/BOOT/BOOT.SDI /boot/grub/WinPE-SS.sdi

cp /media/mint/STDYR-STATE/SOURCES/BOOT.WIM /boot/grub/WinPE-SS.wim

**Password Protection**

Note the lines at the top of the updated grub.cfg you copied from Other/grub :

# Prohibit patrons from editing grub menu entries or accessing the grub console

set superusers="root"

password root YourGrubPasswordHere

These lines prohibit the default grub2 behavior that allows anyone with access to the system to edit boot entries or open the grub console. You can also configure grub so that individual boot entries are password protected, as well as encrypt the password so it doesn’t appear in clear text within the grub.cfg file. There are probably other options you could employ to add password protection to grub, but I’ll leave that as an exercise for you, the reader. For more information on this topic refer to <https://help.ubuntu.com/community/Grub2/Passwords>, and be aware that document does not cover the necessity of using --unrestricted so users don’t need to authenticate just to boot a selection. They do to edit a boot entry and access the grub console but not to boot.

**Providing a Linux “Steadier State” Guest Account**

Depending on the Linux distro you choose this may already be included. Unfortunately that isn’t the case with the Linux Mint distro. However, the lightdm display manager can be installed on Mint to provide guest account functionality. The installation and setup of lightdm is beyond the scope of this brief set of instructions. A heads up about that is that Mint has some issues automounting USB drives under lightdm. I suggest you read the following as a place to start looking into these issues:

* <https://wiki.ubuntu.com/LightDM>
* <https://sites.google.com/site/easylinuxtipsproject/2>
* <http://serverfault.com/questions/766506/automount-usb-drives-with-systemd>

**Conclusion**

After completing the above steps you should be able to reboot, see a nice graphical grub2 menu with the Admins Only! Menu item and boot Windows 7 Ultimate, Linux Mint and WinPE-SS (Windows PE for Steadier State).

I sincerely hope you find Steadier State and these changes useful. Thanks for trying it out!