FEATURE

Test Lab Installations

Andreas Marx
AV-Test.org, University of Magdeburg, Germany

It's a familiar problem: the preparation of more than one computer in a test lab. PCs may have different hardware configurations, but there are several with exactly the same hardware. At *AV-Test.org*, for example, we have five identical P-III-800, five Athlon 1.3 GHz and two P-III-800 multiprocessor systems.

Of course, it should be easy to use different OS versions (like *Win 98* or *2000*) on one PC, but nobody wants to install all *Windows* versions on all computers, with long driver and software installation sessions.

PC Simulator

An interesting idea would be to use *VMware* (which can be downloaded from http://www.vmware.com/) in its most current version (3.0). *VMware* is a full PC simulator – you can run *Windows XP* from a *Linux* system, for example.

The other great advantage of *VMware* is that, with the exception of the CPU type, it simulates the hardware completely – for example the Ethernet adapter or PCI bus drivers. The current state of the simulated PC can be frozen in 'Suspend' mode at any time, during installation or normal work, and restored later with only a few seconds delay, making it faster than starting up a PC.

The simulated PC memory of this 'Suspend' mode will be written to disk and can be inspected using a file viewer. But the main advantage is that it is possible to copy the image files easily from one PC to another (with different hardware, processor and so on) and the simulated OS will not find any changes. This means that the system only has to be installed once and can be used as often as required on different hardware.

However, *VMware* is still a simulator and therefore slower than a normal PC. It is useful when answering customer support calls for inspecting problems using different customer operating systems, and it can be useful for small tests or even controlled virus replication. But for time-consuming tests, like larger virus scan sessions, *VMware* is too slow.

The Goals

Therefore, we still face the problem of different hardware configurations. The goal should be to have small installations, but which include all the necessary files of all the operating systems we need and which can be restored in less than one minute to test complex issues from a clean, consistent system. In this situation only the essential OS or *Office* components will be installed – viruses don't need a grammar or spell checker, for example.

At the same time, we don't want to boot from a floppy disk: not only is it much slower, but more important is the risk of 'forgotten' boot virus-infected disks after a test. Therefore, the boot sequence will be modified after all images are running satisfactorily.

On all of the different hardware the OS has to be installed at least once to avoid instability issues and driver problems. (For professional systems like *Win 2000*, tools exist to prevent these problems, but the instructions for their use would fill more than an entire issue of *Virus Bulletin*.)

Using a drive image software package such as *Ghost* the remaining PCs can be installed easily, changing only small parts like the computer name or the IP address (at *AV-Test.org* we do not use DHCP).

In our case, we need to install all operating systems on three PCs, because we have three different hardware configurations. However, we used all of the PCs for the installation session – three to install the German version, three for the English versions, two for documentation and the rest to install other language versions. It's quite easy to install these different language versions, since the screen messages mean exactly the same in the different languages.

Preparation

First, we ensure that all PCs with the same hardware have the same BIOS version with identical options. For example, we always disable APCI and other power save modes. We note all the changes we have made in the documentation so that the settings can be restored easily, if needed.

The next step is a run of fdisk of *Win 98* in the correct language version. All of our PCs use FAT16-only drives, because *Win NT* does not support FAT32 and *Win 98* cannot read NTFS partitions without additional tools and so on. However, all operating systems can use FAT16 with a limitation of 2 GB per partition. This does not present a problem, because we can add more of them – our system uses three partitions: one for the main OS installation, one for a few drivers we don't want to install from disks, plus tools for partition imaging as well as the swap file, and the last partition is reserved for image files of all operating systems.

Next, we format drive C: using a *Win 98* boot disk and the '/S' switch to ensure it's bootable. *Win 9x*-based systems will overwrite the boot loader using their own system, but *NT*-based operating systems will add a simple boot manager menu – we use this to be able to restore images easily without the need for floppy boot disks. We can start a DOS session and run *Ghost* after that.

If drives D: and E: are formatted, we continue to copy all necessary drivers to drive D: (as for the network adapter), as well as the drive image software, in this case *Ghost*. We also copy a file manager such as *Norton Commander for DOS* or the *Volkov Commander* onto the drive, so we cannot get lost in the command line completely.

As the first step, we will make an image file of the nearly empty, but bootable C: partition – we will use this for all later OS installations. After that, we can start to install the preferred OS (usually from a bootable installation CD).

Once the installation is completed and *Windows* shows a 'Welcome' screen, we start DOS (using the boot menu or a bootable floppy) and save this first installation to an image file. This is useful if we need to test something with a very clean installation.

Installation of Drivers and Programs

Next, we install all drivers and programs – this includes Service Packs for *NT*-based systems. However, for *Win 9x*-based systems we don't install much, because these systems tend to exist fully unpatched in the real world.

We do not install a newer version of *IE*, nor do we install new versions of *DirectX*, the *Media Player* and so on – mainly to save space. The only exception is *NT*, where we install *IE 4.01 SP1* which comes with the Option Pack and several other programs require it.

In addition, we install a few applications that we use very frequently, such as a small screenshot program, a file manager (I find I cannot live without *Windows Commander*, see http://www.wincmd.com/), *Acrobat Reader* for the inspection of documentation and *Winamp*.

Also we install a generic text-only printer (included in all *Windows* versions) so that we can save log files of scanners, if they do not provide a 'Save as' function. All applications can be accessed either by an icon on the desktop or using a hot key.

We make a note of all the changes and installations we have made as we go along, and at regular intervals we create new image files. We can use these image files if we find that a new driver causes problems, if we have changed the wrong registry keys or deleted too many files in the following steps. Alternatively, the image files could be used simply to test programs that do not run under our special test installations, although we have never encountered this situation.

Reducing images

Of course, we have at least 2 GB we can use on drive C:, but the smaller the installation, the smaller the image file and it will also take less time to be created or restored. Now it's time to reduce the size of the image file.

First, we can use the maximum compression for the image tool, for *Ghost* this is 'ghost -z9'. However, the size of a standard image file is still 110 MB for *Win 98*, up to 472 MB for *XP Pro* and most people stop here.

The first step should be to disable the Hilbernate mode: if enabled, a very large, hidden system file called hilberfil.sys can be found in the root of the C:\ directory. The system will save the current content of the memory to this file and all image programs will save the file, which is very time-consuming.

The next step is to change the swap file location – good image software will not store the swap file, but it's easier if we don't need to worry about the swap file.

As a third step, we investigate the files that, hopefully, we will not need any more on the systems, such as txt and readme files, bitmaps, wav, pnf, temp and log files, as well as all of *Windows* help files. If needed, we can still copy them in the correct folder from a network folder.

Windows 98

For *Windows 98* we install a boot menu by changing the msdos.sys file and placing the 'BootMenu=1' under the '[Options]' section so we can start the command line if needed to save or restore the image file. For this, a 'path' variable should be set in the autoexec.bat file to access the utilities directly.

Also, we delete all program starts in the 'Run' registry key under 'HKLM\Software\Microsoft\CurrentVersion\Run' except Systray.Exe – this means that the system starts much faster and no useless tasks will be started on every boot. We do the same for the 'RunOnce' key.

Windows ME

Under *Windows ME* the process is almost identical, but the 'PCHealth' task should not be deleted in the 'Run' key. Also it is not possible to start DOS directly, but we can use a simple trick to prevent the need for a boot disk. The program wininit.exe will be started at every boot if a file wininit.ini exists in the *Windows* installation directory.

If we want to *Ghost* a PC, we simply have to copy the image program to %windir%\wininit.exe (overwriting the existing one) and we have to create the ini file. After that, we can reboot the machine and *Ghost* will start automatically. We have created a simple batch file for this, and to reboot we use the command 'RunDll32.exe Shell32.dll,SHExitWindowsEx 0x2'.

Windows ME also creates a folder called '_Restore' and saves a lot of data here. In most cases, we do not need the data stored in this folder if we create a clean image. We can only delete the data under plain DOS, but it will save a lot of space. *Windows* will recreate the folder during the next boot-up and PC-Health can still be used. %windir%\Options\*.* can be deleted, too.

Windows NT

Windows NT (both *Server* and *Workstation*) is much smaller than *98* or *ME*. However, there is still some room for improvement. For example, we found all Help files twice in our installations – under ‘system32’ as well as in the correct ‘Help’ folder.

After the reinstallation of *SP 6a* and the *Security Fix Rollup Package* (*SRP*), we delete all folders like ‘$NTUninstall$’ (a backup of all modified data by Service Packs or Hotfixes will be saved here, but since we do not wish to uninstall them, we can delete it). It is also useful to clear the event log, not to save space, but to give us a better overview of what has happened if we start tests.

Windows 2000

Under *Windows 2000* the size of the image can be reduced significantly. First, all drivers will be stored in the ‘%windir%\Drive Cache\i386’ folder in a large cab file called driver.cab as well as one or two smaller cab files installed by a Service Pack. We should not simply delete these, but it is a good idea to store them on a (read-only) network drive.

All test systems can share these files – we only have to change a registry key which can be found by searching for ‘DriverCachePath’. After a reboot, we can delete the 56 MB files without any negative effects. Everything still works as it should, and *Windows* does not prompt for an installation CD if we want to install additional hardware.

In the next step, we should look at a folder called ‘%windir%\ServicePackFiles\i386’, which contains copies of all the Service Pack files that have been installed. After copying all files to a network drive, we should change the Registry key for ‘ServicePackSourcePath’, reboot, and delete about 160 MB.

Windows 2000 has a protection against the replacement of important system files. For this, a copy of the protected files will be stored at ‘%windir%\system32\dllcache’. We can delete the content of this folder too. If a program tries to replace a system file *Windows* will prompt for an installation CD and will not simply restore the original. I think it is much more useful to be alerted in this way if something goes wrong with a program. As with *NT*, it is useful to clear the event log to give us a clear overview of what is going on.

Windows XP

For *Windows XP* (both *Home* and *Pro*) the steps are nearly identical to those for *Windows 2000*. However, no Service Pack exists for *XP* at the moment and in our test installation only a few cat files can be found in the Dllcache, because *Windows* only uses this feature if the installation partition is larger than about 3 GB. This is also a good time to activate *XP* – it will remain activated if the image is installed on the other test computers.

After we have installed all *Windows* versions, we can install all the *Office* versions we need to test programs or replicate viruses. However, we only use *Windows 98* to install *Office*, because it’s both small enough and the fastest system to start.

Once we have finished all installations, we run a disk-fragmentation utility, if one is available. This does not save any space, but it does make the test systems more unified.

We can transfer all images to the other PCs, which have only an empty, but prepared HDD at the moment (‘fdisk’ has been run etc.). To do this, we copy an image of *Windows 98* (for all different configurations) to a bootable CD and restore the correct one. After this, we can start *Windows 98*, copy the rest of the images from a network drive to the local disk and *Ghost* them one after another.

To finalize all images, we change the label of the C: partition, the name of the computer and its IP address. But we also clean the ‘last used documents’ cache, copy our standard configuration file for WinCmd to the disk and create an image.

After all the steps, we have a *Win 98* image of 77 MB size and a Ghost time of less than 30 seconds. *Win ME* is 102 MB, *Win 2000 Pro* is about 141 MB and takes only one minute restore time.

We don’t use the GUI version of *Ghost*, but we have written a simple program that displays all the available image files and which starts *Ghost* using the command line switch ‘-clone,mode=pload,src=file.gho:1,dst=1:1 -rb -sure’, where ‘file.gho’ is the image file that will be written to the first partition of the first hard drive. The ‘-sure’ switch prevents an additional ‘are you sure?’ question and ‘-rb’ will reboot the computer after the image has been restored successfully.

Testing on Other Platforms

Of course, these are only our test systems for simple *Windows*-based product tests. For Server tests, we have additional hard disks where only one system is installed using NTFS partitions. We can exchange the hard disks easily, because they are all located in a mobile hard disk rack.

The same applies to other platforms like *Netware*, *Linux* or *FreeBSD*. We always try to reduce the size first and at least two FAT partitions can be found on the disks: one to be able to boot DOS without the need of an extra disk and one to store the *Ghost* images.

We needed about two complete weeks and three people to set up our lab with all *Windows*-based test environments, including the different *Office* installations. However, now this has been done, we find that it is a very efficient way to work, especially since it saves a lot of waiting time before a test can be started.