Virtual Machines & VMware, Part II

December 28, 2001
By: Jay Munro

VMware Workstation 3.0 for Windows operating systems shipped on November 5th, 2001, after a 2 month public beta trial. Web distribution is priced at $299, and packaged distribution at $329. While VMware Workstation 3.0 comes in two Host OS versions, one for Windows operating systems as the Host OS, and the other for Linux systems as the Host OS, we'll focus mostly on VMware Workstation 3.0 for Windows. Many of the same concepts apply to the Linux version, and you can still run Linux operating systems as Guest OSs under the Windows-hosted version.

Per VMware's FAQ: Here's a list of what's new or improved in Version 3.0

Enhancements to existing features include:

- Performance improvements: faster disk I/O, better video performance, faster CPU and networking performance, and smoother mouse movement
- Updated user interface on Windows hosts with better wizards for creating new virtual machines and adding devices to virtual machines
- Easier VMware Tools installation with improved SVGA driver
- Bigger virtual disks

New features include:

- Support for Windows XP
- Virtual machine list in the UI (Windows hosts only) make it easier to keep track of virtual machines
- Virtual machine names now allow users to create names for virtual machines, displayed in the Window title bar and elsewhere
- Automatic saving of configuration changes
- Network Address Translation to make it easy to connect to the Internet or other TCP/IP network using the host computer's dial-up networking connection, wireless LAN adapter, or Token Ring adapter
- USB device support
- Generic SCSI device support
- Connect virtual CD-ROM drives to either a physical drive or an ISO image file using the Configuration Editor
- Support for multiple networking configurations, making it easier to use a virtual machine on a notebook computer that is sometimes used in a docking station and sometimes used standalone
- Screen capture option on the File menu (Windows hosts only)

From our perspective, we believe Workstation 3.0 is a big improvement from Workstation 2.0, which had some technical problems and limitations, a non-intuitive user interface, and required users to edit config files at times to gain added features (much like configuring Linux). It also presented misleading messages occasionally. On Windows hosts, Workstation 3.0 brings a big change to the UI, with a cleaner more intuitive look, and more descriptive messages. The Linux host version retains a Workstation 2.0 user interface with changes to support new features. The help system has been expanded, with far fewer references to the Web (which was a real problem in Workstation 2.0).

VMware 2.0's UI was accused of being a bit of a wasteland of screen real estate. It had no clear way to see available virtual machine files. Additionally, the older method of starting a VM required you to go through the wizard or menu system, and the VM starting shell was just a large logo. The new version makes better use of the screen with a list of available VM files, and the option to open a new or existing VM. VMware 3.0

also has a "Windows XP" style UI, with a more rounded look and feel.

The "new virtual machine" wizard is similar in function to the earlier version, but cleaner in execution. It steps you quickly through creating a basic virtual machine. You can look at it as if you just put together a machine from parts yourself, with the only software being the BIOS. The wizard lets you take either a "typical" path, or a "custom" path. The typical path asks you to specify the guest operating system you will be installing, a directory or folder to contain it, and the type of networking to use. The custom path lets you tweak the memory setting, use an existing disk or create a new disk, set the virtual disk size, and set the disk type (IDE or SCSI). Once the VM is created, you can't change the hard drive size, though you can add drives, or change the disk file it uses.

One new feature that many users may overlook, but product reviewers, tech-writers and documentation professionals will love--is the built-in screen capture. This allows you to install a supported OS, including DOS based products, and get screen captures during times when you have no access to your machine--such as during the boot phase of an OS. From the File menu in the VMware Workstation shell, selecting "Screen Capture" grabs the screen and pops up a file save dialog. It can be stored anywhere on you system, though it goes to the VM directory by default.

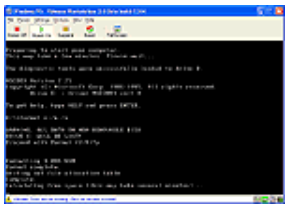
Installing an operating system is almost exactly like installing on a raw hard drive. VMware Workstation provides the virtual machine, along with a Phoenix BIOS v.4.0 release 6.0 that you can get in and tweak. Depending on the OS you're installing (see list of supported OS's), you'll need to have either a boot floppy, or if your PC supports it, a bootable CD. VMware has announced preconfigured Guest OS Kits to be launched in January 2002, but they were not available during our test period. VMware competitor, Connectix Virtual PC, includes PC DOS with their product, as well other operating systems for an extra cost. The bundled PC DOS helps jumpstart installing other operating systems, but we'll explain a work-around for Workstation a little later.

Guest Operating systems supported
Microsoft Windows
Windows XP Home and Professional (shipping versions only)
Windows 2000 Professional, Server, Advanced Server
Windows NT 4.0, SP 3 or later, workstation and server
Windows Me
Windows 98 and 98se
Windows 95 (all OSR releases)
Windows for Workgroups (3.11)
Windows 3.1
MS DOS 6
Linux
Linux-Mandrake 8.0
Red Hat 5.x, 6.x, 7.0, 7.1, 7.2
SuSE Linux 5.6, 6.x, 7.0,7.1, 7.2
Caldera OpenLinux 1.3, 2.x
TurboLinux 6.0
Corel Linux OS 1.0, 1.1, 1.2
FreeBSD
FreeBSD 2.2.x and 3.x

Operating systems supported by VMware 3.0 for Windows.

The most common way to create a guest VM is by using a virtual disk. The virtual disk looks like a hard

drive to the guest OS, but only takes as much space as it needs on your physical hard drive. It will grow as you install applications and data, but only to the limit you set when you created the VM. The virtual disk is unformatted and unpartitioned, so you'll either have to FDisk and Format your drive from a floppy before installing an OS, or install from a bootable CDROM. You can start by booting from a DOS or Windows 9x emergency floppy, or if running Windows NT or 2000, boot from the CDROM. Another way is to download a boot disk image from www.bootdisk.com.



click on image for full view

We first installed Windows 98 the hard way, using a physical boot floppy and the CD. Once the OS was installed, we reconfigured the Floppy drive A: to point to a blank virtual floppy. From within Windows 98, we went to Add/delete applications, and selected the startup disk tab, and clicked on the "create disk" button. This created a bootable floppy image, complete with the option of starting with or without CDROM support. The image was then saved and can be used for future operating system jump-starts.

Windows Me will also create a bootable floppy, but Microsoft in their infinite wisdom, removed the ability to do Format c:/s to create a bootable drive, limiting its usefulness for us in formatting our virtual hard drive. Just as a side note, for PC users who only know Windows, we recommend using the /u (unconditional) switch to keep Windows from doing a quick format or from saving any backup information.

Like any other Windows operating system, you can install Windows XP in a virtual machine. However, because of the Windows Product Activation scheme, you need to take a little extra care. Unless you have a site-licensed version of XP that doesn't use WPA, when setting up a virtual machine for XP, there are some precautions to take.

As Windows XP needs to be activated to run more than 30 days, if you're planning on using it in a VM, you need to be sure about your configuration before activation. Since you can change your VMs configuration at any time, you should take advantage of XP's grace period to work out the memory, drive, sound, and network settings. Additionally, you should install the VMware Tools before you activate. VMware Tools installs an improved video driver, which depending on other changes, could trigger a new XP activation.

Once you're happy with the configuration, go ahead and activate. We activated Windows XP Home edition in a VM without a hitch. While it does break the license agreement, you can move your Windows XP virtual machine files to another machine and run without reactivation. Because the virtual environment is completely encapsulated, Windows XP does not see the real hardware and doesn't know it has moved to another system. We tested our Windows XP Home Edition on three machines, a 700MHz laptop running Windows XP Professional, a 1GHz desktop running Windows 2000 Pro, and a 300MHz desktop running Windows 2000 Advanced Server. The roving VM worked fine on all machines, though it was a little slower on the 300MHz setup.

A key component of Workstation is the disk subsystem. VMware Workstation allows two main types of disk, Virtual and Raw. To a guest operating system, a virtual disk looks, smells, and tastes like a physical disk drive. An IDE virtual disk can be as large as 128GB, and a SCSI virtual disk can be up to 256GB in size. VMware Workstation creates a file for each 2GB of virtual disk capacity, which don't start at 2GB in size, but rather grow to their maximum size of 2GB. These disk files are transportable to other machines running VMware Workstation 3.0, though the latest format is not backward compatible. If you have older Workstation 2.0 files, you can update them to 3.0, either at installation (Workstation will search your drives) or as needed.

A raw disk uses a physical drive or partition. This allows you to run an existing operating system as a drive on your VM, and provides a way to use existing software. One application is on a dual boot machine--you can run an existing operating system in one partition, and a VMware Workstation VM in the other partition. You need to be careful though, as switching from a live host machine to a virtual machine is like pulling the drive and popping it into another machine. Depending on the operating system, booting under two different environments (host, virtual), could cause problems with one or the other. The BIOS, bus, video and most hardware is going to be different, so depending on the OS, it can cause problems. VMware documents some techniques that you can use to do it successfully.

VMware Workstation 2.0 had a 2GB virtual disk size limitation, so as a workaround, it supported a "plain" disk. Unlike a Raw disk, which maps a physical disk to a virtual machine, the plain disk is still a virtual drive. The plain disk let you create drives larger than 2GB, though it would pre-allocate all the space at once. If you wanted a 10GB drive, Workstation would create a 10GB virtual disk file. While it solved the problem of larger drives, it was wasteful. Workstation 3.0 drops support for creating plain disks, but can read existing version 2.0 plain drives.

Generic SCSI support new to Workstation 3.0 also allows SCSI and non-disk devices to be mapped directly into a virtual machine, and do not need to have drivers set up on the host machine.

One common usage for Workstation is to create multiple instances of a single operating system so you can change various settings, but start with the same base (I use this feature frequently for reviewing software). There are several ways to do this--make separate VMs and install fresh copies of the OS, make copies of an original VM file, or create a single file and reuse it without saving changes.

To help you chose how you want to work, Workstation lets you create Persistent, Non-persistent and Undoable drives. A persistent drive is the default mode and is the basic virtual drive, where any software additions, configuration changes, or file deletions are immediately applied. A persistent drive acts like any physical hard drive. You would use persistent mode to install applications and configure an operating system in a VM.

A non-persistent drive is the antithesis of the persistent drive, where anything you do in a session is discarded when you close the session. This is great for testing truly unstable software which can corrupt an operating system. Workstation maintains a Redo file during your session, and when you exit, it is deleted. The name "Redo" is somewhat of a misnomer, because the file is only used during the current session, and it contains the session changes to the VM, rather than having changes written directly to the VM file itself. The Redo file doesn’t exist beyond the current session.

The compromise between the two previous modes, and probably the best of both worlds is the Undoable mode. Undoable mode allows you to work in a session, and when exiting gives you the option to commit, discard, or save the changes that were made. During an Undoable mode session, Workstation keeps a Redo file like the non-persistent mode, but gives you the option of committing, discarding, or saving. If you commit, all your changes are applied to the base VM. If you discard, they are thrown away. However, if you save, the file itself is preserved and can be used in another session. When you power-up a VM session later, it gives you the option to commit, discard, or append the earlier change file. This allows you to do continue a previous day’s work without committing, or losing your changes.

Virtual Machine files can be created and copied to create a whole new VM. The .VMDK file, or image file, is the actual virtual drive containing the operating system and/or data and applications. Each virtual disk file has an accompanying configuration file, the .vmx file, which Workstation uses to access disk image files and setup the virtual hardware. When you first create a VM, the .vmx file contains the location of your virtual disk, memory size, and some basic hardware setup info (CDROM, floppy, network connections).

VMware configuration file of freshly created VM:

```
config.version = "6"
virtualHW.version = 2
guestOS = "win98"
displayName = "MyWin98Test"
```

```
ide0:0.present = "true"
ide0:0.fileName = "D:\VMware\guests\MyWin98Test\MyWin98Test.vmdk "
ide1:0.present = "true"
ide1:0.deviceType = "atapi-cdrom"
ide1:0.fileName = "auto detect"
floppy0.startConnected = "false"
memsize = "64"
ethernet0.present = "true"
```

As you add/configure ports, network and other hardware, the file will grow. Note the new items in the figure.

VMware configuration file with new devices added:

```
config.version = "6"
virtualHW.version = 2
guestOS = "win98"
displayName = "MyWin98Test"
ide0:0.present = "true"
ide0:0.fileName = "D:\VMware\guests\MyWin98Test\MyWin98Test.vmdk"
ide1:0.present = "true"
ide1:0.deviceType = "atapi-cdrom"
ide1:0.fileName = "auto detect"
floppy0.startConnected = "false"
memsize = "64"
ethernet0.present = "true"

draw = "gdi"
sound.present = "true"
sound.deviceNr = "0"

parallel0.present = "true"
parallel0.fileName = "LPT1"
```

While all of the information is configurable from within Workstation's configuration editor, it helps to understand the structure and content of .vmx files in case you need to move or copy a VM. If you copy a VM to a new folder, drive, or machine, you can manually change the path, or file name under "ide0:0.filename", or any other virtual drives to point to the new location. This prevents Workstation from popping up an error when you start. If you do get the error in Workstation's start window, you can open the configuration editor and edit or browse to the correct files. We found it quicker to edit the .vmx file in Notepad from the host system

When you have Workstation's window open, any file that's in the list will be locked, and you'll see a .lck file accompanying your vmx and vmdk files. Trying to open a file that has an associated .lck file will fail with a sharing error.

Though covered a bit above, we thought we'd convey how the redo log can be used to preserve unique configurations. You can create a series of rollback points that allow you to preserve a configuration, but make further changes that can be undone. While the technique is a little kludgy, when you want to make a breakpoint using an Undoable disk, you shut down your OS, and click Save when Workstation asks what you want to do with your changes. You then go back into the configuration editor and change the virtual disk's .vmdk entry to the .vmdk.REDO file, rather than the .vmdk file. The next time you start the VM, you'll be building another .REDO file, preserving the original in case you want to go back. When you commit your VM changes, it is written to the vmdk.redo (pointed to by your modification with the configuration editor). It gets a little messy because VMware designates REDO files by tacking a .redo extension on the original disk image file. This can result in files like windows98.vmdk.redo.redo.redo and so forth.

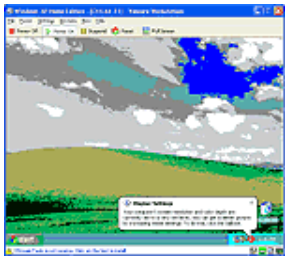
Looking at the competition, Connectix lets you create a new VM based on an existing file, and only records

the changes-called a Differencing disk. It's similar to how object-oriented code works--the main object contains the code, while instances and inherited objects only contain the added code and data. In Workstation, such capability is possible but is a little more manual. Pay attention, this gets involved... You start with an undoable disk image. When you power the VM off, choose save (not commit) for the changes. You then make a copy of the REDO log file (i.e. myVM.vmdk.REDO to ourVM.vmdk.REDO) and the .vmx config file (i.e. myVM.vmx to ourVM.vmx). In VMware's config editor, set the original VM to use the original .REDO file (myVM.vmdk.REDO). Next, you create a new VM and in the config editor, set the new VM to use the copy of the .REDO (ourVM.vmdk.REDO). You'll now have two VM files using the same root, though when you commit changes to the new VM, it doesn't affect the original VM.

VMware Workstation 3.0 maps your physical CDROM drive to a virtual device so you have access as you normally would. However, one of the cool things with Workstation is that you can map to a standard ISO CDROM disk image that you can create with WinImage, Nero Burning ROM, or most other CDROM burning utilities. By creating ISO CD images on disk, you can store often accessed CD's, like help or library disks on a server, and map to them as needed. You can map floppies and CDROM drives to your VM, and change where they point on the fly through the configuration editor in the Workstation shell.

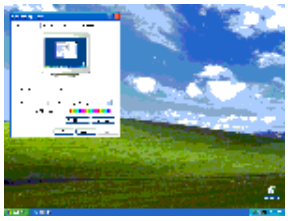
VMware Tools

A fresh installation of a Guest operating system may suffer from performance or usability problems initially. For example, Workstation loads a generic driver for the virtual machine's virtual graphics card. This driver is analogous to NT's VGA mode at 16 colors, 640x480, and looks like figure below. Switching mouse control from the VM to the host operating system requires you to press a hotkey combination. In both cases, performance of graphics display and mouse movements is far from optimal.



click on image for full view

To fix these problems and more, VMware recommends installing VMware Tools, a suite of utilities and drivers optimized for the virtual hardware. The VMware SVGA driver alone gives a noticeable video performance improvement, and mouse handling is snappier. You can now increase your resolution and color depth as shown in the figure below. Additionally, you can now move the mouse cursor in and out of the VM window, as well as do text cuts and pastes between the host and guest operating systems. The VMware SVGA driver installs into your guest OS like any display driver.



click on image for full view

Note that while video performance is vastly improved with the SVGA driver, the virtual video card does not support Direct Draw in a VM. This probably is most noticeable with games. However, since Workstation VMs are not recommended for real-time applications, it shouldn't be a problem. VMware made the decision not to support Direct Draw because of problems with supporting the vast array of DirectX versions, patches, and stability difficulties with DirectX.

Installing VMware Tools is easier with Workstation 3.0 than prior versions. VMware provides VMware Tools bundled in Workstation 3.0 for Windows 95, Windows 98, Windows Me, Windows NT 4.0, Windows 2000, and Windows XP, as well as Linux and FreeBSD guest operating systems. You first need to get your operating installed into a virtual machine. Once installed, you just click the Install VMware Tools option from the Settings menu. Workstation temporarily reconfigures your CDROM connection to a virtual install disk, and runs the Tools installation. It requires a reboot of the guest OS to activate. The reboot reconfigures the system to its original state, only now you have the tools installed. If for any reason you need to uninstall, just click on the Uninstall VMware Tools option under Settings.

The improved mouse handing does not work if you are at a DOS prompt, or if the guest system crashes. In these cases, you still need to use the hot key combination. We found though, in our testing that we were able to press the hot keys to get out of a crashed or hung VM, almost every time. The only time we had a problem was when something in the host OS caused a crash of the whole system. We found this feature impressive, as it reinforced the feeling of security and isolation that a virtual machine can offer when testing unstable software. The only time it didn't seem to work was when the host system hung due to problems with serial communications.

When you start a VMware Workstation virtual machine, the first thing you see is a screen with the Phoenix BIOS Power On Self Test (POST), memory check, and drive enumeration, just like a real PC. Yes, Virginia, this is a real BIOS. VMware licenses the Phoenix BIOS code for integration into their VM. During startup, you can press to get into the BIOS setup. Like a real PC with a Phoenix BIOS, you can set boot sequence, user and supervisor passwords, or power management. The BIOS configuration information is stored in the nvram file, emulating the CMOS memory of a real PC. If you modify a BIOS setting for a particular guest VM, be sure to include the nvram file if you move or copy the VM. Workstation makes a separate directory for each new VM by default, and encapsulates all required files in it. Consequently, the best way to guarantee that you have all the files is to move or copy the whole directory.

click on image for full view

Networking is one of the most interesting areas in the VM. Networking is installed whether or not your host OS is connected to a network. Networking can be configured in several ways--Bridged, NAT, and Host Only, as well as a Custom mode that gives more low level control. The configuration editor provides full control of configuring networking for a virtual machine.

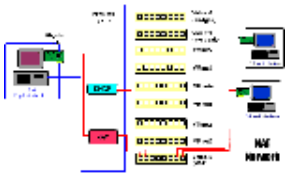
Up to three virtual network cards can be configured in each VM, and each appears to the Guest OS as an AMD PCNet II card-- your average, everyday generic Ethernet card. There are also nine virtual Ethernet switches, designated as VMnet0 - VMnet8, that exist within the PC. Workstation 2.0 only supported four virtual switches. Each switch can connect to one or more virtual network cards. By default, some of the VMnet switches have specific functions: VMnet0 is dedicated to bridged mode, VMnet1 is dedicated to Host Only mode, and VMnet8 is for NAT (Network Address Translation) mode. The others, VMnet2 - VMnet7, are available to be used by the VM when using Custom networking mode.

Bridged networking is the default (unless you opt for no networking), and it allows your VM to talk to the outside world via your host machine's network card. VMnet0 connects your virtual network card to the physical network card. While the VM shares the host's Ethernet connection, it appears as an entirely separate machine on the local Ethernet with its own MAC and TCP/IP address. For TCP/IP connections you can either set an IP address, or get a dynamic address from an outside DHCP server. If you can't allot more than one IP address for a physical machine (your IT guy may have something to say about this), then skip ahead to the NAT mode. Bridged mode by default connects the VM with VMnet0 which maps to the first available NIC automatically. If you have more than one physical network card in your Host machine, it is recommended to manually map your virtual switches to physical adapters.

click on image for full view

Wireless LAN cards are a problem in a Bridged setup and do not work. Previous versions of Workstation caused some confusion by attempting to use wireless (802.11b) Ethernet cards for bridged networking, but would not work, becoming a tech support nightmare. Workstation 3.0 now correctly detects a wireless Ethernet network card and blocks it from being configured for bridged networking. The reason for this is because of a security specification in the IEEE Ethernet 802.11 protocol that says a wireless adapter cannot send packets that have a different MAC address than its own. In bridged mode, the packets are created in the VM, and sent through the switch to the physical network card. Since a virtual machine has its own MAC address for each virtual NIC, the wireless adapter will not send packets other than those generated on the physical wireless card. However, the workaround for using a wireless network card is to use NAT mode.

NAT mode is new with Workstation 3.0. It allows the virtual machines to share the IP address of the host system. The Workstation DHCP server assigns IP addresses to the virtual network cards, and the NAT system translates packets between the host's network card and the guest OS. VMs using NAT connect via the VMnet8 switch. Workstation configures a separate NAT service that runs on the host OS and connects between the VMnet8 switch and a TCP/IP socket on the host OS. The host routes traffic from that socket to wherever needed, such as an Ethernet, Token Ring, or dialup connection. As mentioned above, NAT can work around problems like using a wireless network card. With the NAT service, the wireless Ethernet adapter just sees normal TCP/IP traffic from the host OS and no special bridge software trying to send packets with other MAC addresses.



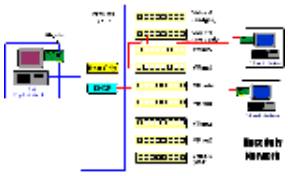
click on image for full view

For Host Only mode, VMware installs a virtual Ethernet adapter in the host OS that communicates with the VMnet1 switch. The host OS believes this is just another Ethernet adapter, and can be viewed from the standard Windows (host) network properties panel. Workstation also runs a virtual DHCP server connected to the VMnet1 switch. In this mode, the DHCP service will assign addresses to the VM's virtual Ethernet adapters (and actually the host OSs virtual Ethernet adapter as well) that are connected to the VMnet1 switch.

This allows communication between a virtual machine and the host operating system, but it is not routed to the outside world. Multiple virtual machines can talk to each other as well. Host Only networking also allows connection of virtual machines to outside networks through the host OS. This is done by configuring network routing, or even a firewall, on the host OS between a physical network and the Host Only virtual Ethernet adapter in the host OS.

Using Custom mode, you can directly specify the connection between a virtual switch and virtual Ethernet adapter. Virtual machines connected though the unused virtual Ethernet switches (VMnet2 through VMnet7) are completely isolated from the host OS or any local physical networks. This mode can be helpful in testing code that you want to isolate from a production or corporate network. You can create a complete network environment on a single machine, and neither the corporate network, nor your test setup knows the other exists.

The Workstation DHCP capability is a modified DHCP server that can only serve IP addresses to the VMnet switches. The DHCP server assigns IP addresses from a class C non-routable subnet. This is configured automatically at Workstation install time, and the installer checks to make sure the subnet is not already used by the host OS. The DHCP server normally only serves IP addresses to virtual machines connected to VMnet1 (Host Only) or VMnet8 (NAT), but can also be configured to serve IP addresses to the other virtual switches.



click on image for full view

Deciding which networking mode is basically a matter of whether you want the physical network to see the virtual network, or not. If you want the physical network to see the virtual network, use the Bridged mode. If you want an isolated virtual network without outside connectivity, use Host Only mode. And if you need outside connectivity, but want the virtual network invisible to the physical world, use NAT mode.

One area that Workstation has noticeable performance improvement is with the mouse and video handling as mentioned earlier. We found it snappier right off the bat, though only after loading the VMware Tools. However, any performance improvements are directly based on the speed of the machine and the amount of available memory. Much of our testing was performed on a 700MHz Pentium III laptop with 256MB of memory. Our Host operating system was Windows XP RTM (release version). We loaded up DOS 6.22, Windows 95 OSR2, 98SE, and XP Home Edition as guest operating systems.

When you initially setup a Virtual Machine, Workstation sets optimum settings for the guest you're planning on installing. The biggest key is the amount of memory. Too much, and the Host gets slow, too little and the guest is slow. VMware allocates more memory than Microsoft recommends in the VM setups. DOS gets 16MB (way beyond anything it'll ever use), while Windows 95, 98, and Me all get a 64MB allocation. Windows 2000 Pro gets 96MB, and Windows XP Pro gets 160MB, all of which are above Microsoft's "comfortable" recommendation.

Performance on applications can vary depending on the resources they use. If an application is compute-intensive, the performance hit from the VM is noticeable, but not too bad. If the application does a lot of network or disk I/O, then the performance penalty is greater. To see just how much of a difference exists between a physical machine and a virtual machine, we ran WinBench 99 v 2.0 on Windows 2000 Professional SP2, running under VMware 3.0 and compared it to native Windows 2000 SP2 performance on the same hardware.

Our test (real) machine was a PIII 1GHz Dell 4100 with 256MB of RAM, a WD 40GB 7200RPM drive, and

a 32MB nVidia GeForce2 MX video card. The VM was running with 176MB of RAM, the maximum recommended for our configuration, a 4GB virtual drive, and VMware Tools installed, which includes the VMware SVGA driver. To see how much difference using a Virtual Machine made, we ran our older CPUmark 99 and FPUmark 99 tests from WinBench 99, version 1.1. Although Ziff Davis has since dropped using CPUmark and FPUmark for system comparatives across processors, due to the tests not keeping up with CPU changes, they are still somewhat valid in a controlled and limited test scenario on a single machine. We expected CPU tests on the virtual machine to be somewhat slower than the native system, and we were correct--CPUmark on a virtual machine was 13% slower than on the real machine. With WinBench 99, version 2.0, we ran the Business Disk WinMark which showed a degradation of 16%. Since we were comparing the native nVidia driver in the Host with the virtual graphics card and VMware SVGA driver, we were looking at a big difference. Graphics tests measured with the Business Graphics WinMark were as expected, extremely slow, giving us 90% slower results with a Virtual Machine as compared to the host.

Since networking is system resource-intensive, VMware focused on improving their networking performance in version 3.0. To test network performance, we downloaded a 10MB file through a 100Mbps Ethernet network to our 700MHz PIII machine. We were surprised that in this test, the VM running in bridged mode was only 4% slower than the native machine.

During our testing, we used VMware Workstation 3.0 to evaluate a number of applications using Windows 98 and Windows Me in a virtual machine. We installed printer drivers and printed over a network. We even used the new USB support to grab images from a digital camera. We could change operating systems without reformatting the physical hard drive, and run NTFS in a VM, while on a FAT32 host. Once we put our VM into full screen mode, the environments were indistinguishable from Windows loaded directly on a physical machine. Some processes tended to run a little slower, but not to the point of being unusable. We think the advantages outweigh the tradeoffs in speed. We were able to use a clean OS setup for each application by just making a copy of the VM file, never touching (or mucking up the registry) the Host operating system. If one of the applications toasted our registry, or wouldn't uninstall, we just discarded the VM, and brought up a fresh copy.

Of course, not everything comes up roses, as we did run into some problems. With Acrobat 5, we had to delete the Intertrust DocBox DLL, as it would lockup when loading. We found that Adobe's latest Acrobat 5.0.5 update fixes the Intertrust bug. We could not run a Sony system disk restore into a virtual machine, mainly because it was looking for the Sony OEM BIOS, and though the VM was on a Sony Vaio, it only saw VMware's Phoenix BIOS. We mentioned the wireless network problem earlier that we needed to work around. We also found repaint errors in list boxes when we ran Namo WebEditor 5. However we ran Excel 2000, Word 2000, IE 5.5, IE 6, Outlook 2000, HomeSite 5, Authorware 6, DemoShield 7, and digital camera utilities with no problems at all.

(You should also check out our online message thread for some interesting inputs from Halo Four (one of our forum members), and Darryl Ramm from VMware regarding feature and compatibility issues.)

While VMware Workstation is not for gamers, or anyone who needs really fast, real-time graphics or animation, it will fit the bill for many applications. When used on a fast machine with a lot of memory, the time savings alone is worth it. For product test and review purposes not having to format and install when you need a new OS cuts a large part of a testing workload. If you add in the hardware costs of maintaining multiple machines, the $300 for VMware Workstation becomes a non-issue. If you're a developer or QA person who tests software, runs old applications, or wanders into dangerous cyber territory, then check out VMware. If you are a Windows who wants to experiment with Linux, VMware is a great tool to test the waters without requiring significant disk partitioning and setup changes. There are numerous other uses for VMware in the enterprise, for technical professionals, and in academic/learning environments. We highly recommend VMware for those who want to install, learn, and use multiple OSes on their existing hardware with the least amount of reconfiguration and hassles.

VMware Workstation for Windows 3.0
Web Distribution - $299, Packaged Distribution - $329
www.vmware.com
VMware, Inc.
3145 Porter Drive, Building F

Palo Alto, CA 94304 USA
Tel: 1-877-4VMWARE (650-475-5000 outside U.S.)
Fax: 650-475-5001