

Installing and running NextStep/OpenStep (Intel) on PC emulators

Tomaž Slivnik

DRAFT v0.2 (27 April 2006)

This note summarizes my experiences, and the experience of other people that I could find, with installing NextStep 3.3/Intel and OpenStep 4.2/Intel (in what follows, when referring to either of these, we shall use the abbreviation *Step) on the following emulators:

vmWare
Connectix/Microsoft Virtual PC
Qemu
Bochs
Parallels Workstation
Guest PC

Note that *Step are not supported guest operating system in any of these emulators.

Prerequisites

You will need

1) the following floppy images; these can be obtained at [2], or copied from the corresponding original *Step floppies by inserting the floppy into the floppy drive of a Unix machine and running "dd if=/dev/*fd-raw* of=*fd-image*" or "cp /dev/*fd-raw* *fd-image*". On *Step/{Intel, Motorola}, *fd-raw* is normally rfd0b. *Fd-image* is the name of the image file on which to store the image of the floppy. On *Step, both commands will terminate with an error, but will produce a valid floppy image.

(for NextStep 3.3)

NeXTStep 3.3 boot floppy	[3.3_Boot_Disk.floppyimage]
[FDI] in text to follow	
NeXTStep 3.3 core device drivers floppy	[3.3_Core_Drivers.floppyimage]
[FDD] in text to follow	
NeXTStep 3.3 beta device drivers floppy	[3.3_Beta_Drivers.floppyimage]
[FDDb1] in text to follow	
NeXTStep 3.3 device drivers floppy	[3.3_Driver_disk.floppyimage]
[FDDd] in text to follow (not needed)	
NeXTStep 3.3 additional device drivers floppy	[3.3_Addl_Drivers.floppyimage]
[FDDa] in text to follow (not needed)	
NeXTStep 3.3 PC card install floppy	[3.3_PCCard_Install.floppyimage]
[FDI2] in text to follow (not needed)	

(for OpenStep 4.2)

OpenStep 4.2 boot floppy	[4.2_Install_Disk.floppyimage]
[FDI] in text to follow	
OpenStep 4.2 device drivers floppy	[4.2_Driver_Disk.floppyimage]
[FDD] in text to follow	
OpenStep 4.2 beta device drivers floppy 1	[4.2_Beta_Drivers_1.floppyimage]
[FDDb1] in text to follow	
OpenStep 4.2 beta device drivers floppy 2	[4.2_Beta_Drivers_2.floppyimage]
[FDDb2] in text to follow (not needed)	

2) *Step installation CDs, or ISO images of them. In Mac OS X, the ISO image can be created from a CD by inserting the appropriate

CD, and running "dd if=/dev/diskns1 of=*cd-image-file.iso*" or "cp

`/dev/diskns1 cd-image-file.iso".`

NB1 I advise you to make several copies and compare them, as, at least on my computer, the CDs often do not read 100% correctly.

NB2 Make sure you copy `/dev/diskns1`, not `/dev/diskn`.

(for NextStep 3.3)

NeXTStep 3.3 User (for Intel Processors and NeXT Computers) CD
[CDU] in text to follow
NeXTStep 3.3 Developer CD
[CDD] in text to follow (optional)
Enterprise Objects Framework 1.1 CD
[CDEOF] in text to follow (optional)
NeXTStep 3.3 Updater (patch 2 for NeXTStep 3.3 + patch 1 for EOF 1.1) CD
[CDY2k] in text to follow

(for OpenStep 4.2)

OpenStep 4.2 User CD
[CDU] in text to follow
OpenStep 4.2 Developer CD
[CDD] in text to follow (optional)
OpenStep 4.2 Updater (patch 4) CD
[CDY2k], [CDEOF] in text to follow (this CD includes EOF2.1)

3) For some emulators, you will need 3rd party drivers. These can be downloaded from the links provided in the emulator-specific notes. If there is a working NeXT/Apple-supplied network driver for your emulator's virtual hardware, you can get these into your VM via the network; however, more often than not, you will have to transfer them into your VM via a virtual floppy or ISO image; create the image on another host, copy the driver files into it, then mount it on your VM and copy the files.

A floppy image can be created on Mac OS X as follows:

```
hdiutil create -size 1440k image-name.dmg -fs MS-DOS -fsargs "-f
1440"
hdiutil attach image-name.dmg -mountpoint tmp-mountpt
cp file1 ... filen tmp-mountpt
hdiutil detach tmp-mountpt
mv image-name.dmg floppy.ext
```

This should also work, but for me, it doesn't:

```
hdiutil create -size 1440k -srcfolder source-directory image-
name.dmg -fs MS-DOS -fsargs "-f 1440"
mv image-name.dmg floppy.ext
```

(where *ext* is the extension used by your VM; e.g. flp (vmWare), fdd (Parallels) etc.; all these image formats are the same thing and are simply a raw, sequential, block-by-block byte-by-byte copy of your floppy disk; the "floppyimage" files at [2] are also the same thing). floppy.*ext* now contains the appropriate floppy image. If working with floppy images, it's usually a good idea to back up the image so created, and work with a duplicate in your VM, as the VM-attached floppy image does occasionally get corrupted if things go wrong. Also, long (VFAT extension) filenames on floppy disk appear not to survive the trip from Mac OS X to *Step in the above procedure; only the 8.3 mangled filenames remain. You must copy files off the floppy image and rename them manually.

This is how to create an ISO CD image on Mac OS X:

```
hdiutil makehybrid -o cd-image.iso source-directory [-iso]
```

Using the CD ISO image has the advantage that it is never overwritten by the VM. Also, names are somewhat less mangled than on floppy images, although some mangling still occurs: file names are converted to lowercase, truncated to 30 characters, and all "."'s except the last one are removed. [I would like to hear from anyone who knows how to create a fully non-mangled CD image usable with *Step]

On Linux, [25] may be helpful to create floppy image, and other files (I have not tried).

Drivers usually come packaged into .tar.gz, .compressed etc. files. These must be uncompressed, and somewhere in the decompressed file hierarchy there will be a *.config file. To install this file on *Step, double click it; this will copy the *.config file into /usr/Devices, which is a link to /private/Drivers/i386; alternatively, you can also copy the *.config file there manually. Double-clicking on *.config will also launch Configure.app if it's not already running. Once the *.config file is installed, you may delete the driver package (.tar.gz / .compressed) and all the file it decompressed into.

Installation Procedure

General notes, *Step, all emulators. See also additional notes for your specific emulator.

1) Create a new virtual machine with:

- 32MB RAM or more (I have successfully used up to 132MB, all of it seen by OpenStep 4.2; [13a] reports using 256MB. I do not know what the upper limit is) But see [22b] which advises you may need to reconfigure your graphics driver not to use addresses <64MB if you're using more than 64MB RAM.

- a new IDE virtual hard disk

- [NextStep 3.3] 400MB-2GB in size. A full NextStep 3.3 installation including all language files, Developer & EOF 1.1 takes up 381MB. NextStep 3.3 installation procedure does not work with partitions over 2GB.

- [OpenStep 4.2] 600MB-4GB in size. A full OpenStep 4.2 installation including all language files, Developer & EOF 2.1 takes up 592MB. OpenStep 4.2 installation procedure does not work with partitions over 4GB.

If your virtual machine permits changing the boot order (e.g. in the BIOS setup program, or via command line options), set it to boot from the floppy drive first (on most VMs, this is the default).

Alternatively, you can set the boot order to hard disk first; in that case, you will on one occasion (in step 3)) have to tell *Step to boot from the floppy manually at boot prompt, but the rest of the installation will be slightly simpler.

2) Set up the virtual machine with [FDI] inserted into the floppy drive and the [CDU] ISO image (or physical CD) attached to the CD drive.

3) Start up the virtual machine & let it boot. It will boot off the floppy drive. Select your language & tell the installation program to start.

4) After a while, you will be prompted for the driver disk. Attach [FDD] to the floppy drive and press Return.

5) After a while, you will be asked which kind of SCSI controller the CD-ROM is attached to, and, once you have selected this, which SCSI controller the hard disk is attached to.

Attach [FDDb1] to the floppy drive, select the option to look for the driver on another disk, and press Return.

You want to select the "Primary/Secondary(Dual) EIDE and ATAPI Device Controllers" (NB NOT the "EIDE and ATAPI Device Controller") from the [FDDb1] floppy for both the hard disk and the CD-ROM. Note that you need the driver from the [FDDb1] floppy not the [FDD] floppy for both NextStep and OpenStep: the NextStep [FDD] EIDE driver does not support a dual EIDE interface at all, and the OpenStep [FDD] EIDE driver causes (for me at) least a kernel panic.

On NextStep 3.3, the installation routines won't let you select an EIDE driver for the CD-ROM controller, so select a bogus SCSI driver for that.

The EIDE driver you select for the hard disk controller will then be used to drive both the CD-ROM and the hard disk.

Now choose "Primary/Secondary(Dual) EIDE and ATAPI Device Controllers" (once on NextStep, twice on OpenStep).

[Notes: "Intel PIIX PCI EIDE/ATAPI Device Controller (Dev:7 Func:1 Bus:0)" would be a better choice for OpenStep 4.2, when supported by the hardware, but cannot be chosen: installation fails at the end of step 8) if this choice is made, because the *.table file is not successfully copied]

6) Tell the installation to proceed without loading any other drivers. *Step will boot, this time from the attached CD-ROM image.

7) You will be asked how to partition the hard disk. Allocate the whole disk to *Step.

8) *Step will format the virtual hard disk and copy over the files from the CD to the hard drive.

When it is done, it will ask you to remove the floppy and press return.

[sometimes, the "usage" message for the "mv" command is printed twice at this point, and sometimes it isn't. I don't know why but it seems to cause no harm.]

If your VM allows you to set the device boot order, set it to Hard Disk/CD-ROM/Floppy disk now & press return if necessary

(if you need to set this up in the BIOS setup, press return now and enter BIOS setup when the system reboots).

Otherwise, detach the floppy image & press return.

The system will reboot, this time (and, from now on, all remaining times) from the hard disk.

9) You will be asked to insert a floppy containing some drivers.

If asked for one of the following drivers, attach the [FDDb1] floppy and press "1 <Return>":

"Primary/Secondary(Dual) EIDE and ATAPI Device Controllers"
(the bogus SCSI adapter you selected in step 5))
[NextStep 3.3 only]

Otherwise, if asked for the following drivers, attach the [FDD] floppy and press "1 <Return>":

"EISABus"	[NextStep 3.3 only]
"PCMCIABus"	[NextStep 3.3 only]
"Intel824X0"	[NextStep 3.3 only]

The system will reboot.

10) You will be asked (this time by the GUI) to attach a floppy containing some drivers. Attach the same floppy for each driver as in step 9) and press

Return.

11) You will enter Configure.app. Unless otherwise noted in emulator-specific notes for your emulator, I recommend installing only a minimal driver configuration at this stage, i.e. only install these drivers:

- display: "Default VGA Adapter" driver, default 640x480 BW:2, 60Hz mode.
- no other drivers (network, SCSI, audio, etc.)

This will get you through the installation. In most cases, the optimal drivers for most functions are not available at this stage. If you wish to install more drivers now, it is best at least not to install networking yet - see step 14).

Click "Save". The system will reboot.

12) You will be presented with a list of packages to install. Select everything. Pay no attention to the "Disk Space Required" figure reported; it is incorrect. The correct number is 207.5MB (OpenStep 4.2).

13) When the packages have been installed, you are asked to press Return to restart.

If you have not set the hard disk to become before floppy disk in the VM's boot order, disconnect the floppy now.

Press Return.

14) The system will restart.

If you have installed network drivers in step 11), the system will hang for a while and then after a while, a monitor panel will come up saying

"No response from network configuration server. Type Control-C to start up computer without a network connection". This will always happen if you have a working network hardware/network drivers combination installed but the network not yet configured correctly (as per step 18)). Press Control-C. The system boot sequence will then complete.

15) You are presented with a panel to choose your language and keyboard. Select them and click OK. When asked to confirm your choices, click OK.

16) You will be automatically logged in as user "me". Run Preferences and set up a password for "me" and log out.

Then log in as "root" (the default root password is blank).

17) Install any additional drivers as suggested by the specific instructions for your emulator & save your configuration.

[NextStep 3.3] remove the bogus SCSI driver installed in step 5).

Don't reboot yet - especially if you've installed network drivers, as your network is not yet configured and you'll have to wait as per step 14).

18) Run HostManager.app. On main menu, click Local, and set it up as follows:

- NetInfo Binding: select "Use Local Domain Only".
- Time Standard: select "Ignore Network Time"
- Hostname: (choose your computer's name)
- Internet Address, Router, Netmask: (see the section for your particular emulator)

Click Set. You are given the message "Network type has switched to a non-NetInfo network. Disconnecting from all NetInfo parent domain servers."

Click OK.

You are given the message "Disabling \"broadcasthost\" as server for the parent NetInfo domain of this machine."

Click OK.

Click OK again. The system will reboot.

19) If you are going to install *Step Developer, do it now. Attach [CDD] to the CD-ROM drive. This contains the following package files:

- a) DeveloperTools.pkg
- b) DeveloperLibs.pkg
- c) DeveloperDoc.pkg
- d) ProfileLibs.pkg (OpenStep 4.2 only)
- e) GNUSource.pkg

Install any of those that you wish. [OpenStep 4.2 only?:] a) must be installed before b)

20) If you are going to install EOF, do it now. Attach [CDEOF] to the CD-ROM drive. This contains the following package files

[EOF 1.1 - for NextStep 3.2-3.3]:

- a) NEXTSTEPDeveloperPatch.pkg
- b) FoundationUserPatch.pkg
- c) EOUser.pkg
- d) EODeveloper.pkg

Install any of these you wish (a) is not needed for NextStep 3.3. b) and c) should only be installed together. d), if installed, should be installed last).

[EOF 2.1 - for OpenStep 4.2]:

- a) EO2User.pkg
- b) EO2Developer.pkg
- c) EO2SybaseAdaptor.pkg
- d) EO2OracleAdaptor.pkg
- e) EO2JOracleAdaptor.pkg
- f) EO2InformixAdaptor.pkg

Install any of these you wish (a) must be installed before any of the others, others can be installed in any order).

21) Install the Y2k patch. Attach [CDY2k]. This contains the following package files:

[NextStep 3.3 Y2k patch]

- a) NS33CISCUUserPatch2.pkg
- b) NS33DeveloperPatch2.pkg
- c) NS33RISCUUserPatch2.pkg

Install a) then, if you installed Developer tools, b). c) is not relevant. Then edit the scripts /NextDeveloper/Examples/EnterpriseObjects/PeopleDBScripts/installPEOPLE.sqloracle[67], changing all 2-digit-year dates to 4-digit ones.

[OpenStep 4.2 Y2k patch]

- a) OS42MachUserPatch4.pkg
- b) OS42MachDeveloperPatch4.pkg
- c) OS42MachProfileLibPatch4.pkg

Install a) then, if you installed Developer tools, b) then, if you installed Developer Profile Libraries, c). Reboot afterwards.

22) See section for your emulator to see if updating any installed drivers is appropriate at this stage.

General notes:

- USB support: as far as I know, there is no USB support on *Step. There were some efforts [5] to create an OpenStep USB driver, but I do not know where these efforts have gone. However, your emulator may emulate certain USB devices (e.g. a USB keyboard or mouse) and present it to your guest O.S. as another kind of device (e.g. a PS/2, bus or serial mouse or PS/2 keyboard).

If Things Get Sticky

*Step boot prompt accepts several options [1]. You may find the following useful. The command line format is as follows:

```
xx([d[,p]])kernel [-v] [-s] [-a]
[options]
```

where *XX*=hd for an IDE drive, sd for a SCSI drive, fd for a floppy drive, *d*=drive number (default=0), *p*=partition letter (default=a), *kernel* = kernel to use (default=mach_kernel). Typical examples would be

```
hd()mach_kernel
hd(0)mach_kernel
hd(0,a)mach_kernel
```

all of which boot from /mach_kernel on the first partition of the first EIDE disk in the system [I have never actually succeeded in booting off an IDE drive hd(*N*) with *N*≠0]. The meaning of the options is as follows:

```
-v = verbose boot
-s = single user boot
-a = ask user interactively for rootdev once the system boot sequence has
detected which disks are attached and where (implies -v)
```

Verbose boot is almost always useful when you're installing *Step for the first time on new (virtual) hardware as in many cases it tells you the information you need to determine what's gone wrong. *Options* are as follows:

```
config=configuration
maxmem=num-kbytes
rootdev=xxdp
"Boot Drivers" = "preboot-driver-list"
"Active Drivers" = "postboot-driver-list"
"Ask For Drivers" = ask
```

(other options are supported depending on what drivers you have installed, but will not be described here. See [6])

Above:

- *configuration* is either Instance0 (the currently selected configuration) or Default (the configuration of drivers you used when installing *Step, likely to be a more minimal configuration, but if this configuration still contains a bad driver, use the "Boot Drivers" and "Active Drivers" options). [further configurations can be created by creating files in ...]
- *num-kbytes* is the amount of RAM (in kiB) you want *Step to use
- *xxdp* is the raw device to mount on / after the system boots up (if, e.g., different from the drive you're booting from). *XX*, *d* and *p* are as above.
- *preboot-driver-list* is the list of drivers to load by the boot code [is this correct?]
- *postboot-driver-list* is the list of drivers to load by the kernel once loaded [is this correct?]

- *ask* is either Yes or No (default=No). If you specify Yes, you will be asked, before boot commences, to insert a floppy disk with additional drivers, and select drivers to load.

If you find you have installed a driver which makes your installation stop working, try booting using one of the following options:

```
config=Default
"Active Drivers" = "MOUSE VGA" "Boot Drivers" = "EISABus PCIBus
PCMCIABus PCIC PS2Keyboard Floppy Intel824X0 EIDE"
```

where *MOUSE* is either PS2Mouse, BusMouse or SerialMouse [NextStep 3.3]/SerialPointingDevice [OpenStep 4.2] (you may include more than one), as appropriate to your situation, or some variation of these.

Emulator-Specific Notes

vmWare

Summary

1) this emulator runs *Step well. There are now drivers which make the PS/2 mouse, networking, sound (playback, not recording) and high resolution colour graphics run well. The only thing currently not working is SCSI. [I have also only been able to get the sound to work in NextStep 3.3 so far, not yet in OpenStep 4.2 with the [4c] driver].

2) I have used the following versions of vmWare:

- vmWare Server 1.0.0.22088beta
- vmWare Server 1.0.0.22874beta

These are from the same generation of vmWare as vmWare Workstation 5.5 so these notes may be helpful on that emulator as well.

[11] and [12] have installed OpenStep 4.2 on vmWare 3.0 (MS Windows) and vmWare 3.0 Beta (Linux) successfully.

3) I have not succeeded in running more than one vmWare VM at once. Running two simultaneously caused instability in both VMs.

Setup

- step 1) use these settings:

- RAM: I've successfully used 32MB-132MB with OpenStep 4.2; [13a] has used 256MB. I do not know what the upper limit is.
- hard disk, CD-ROM: attach them to the IDE interfaces in one of the following combinations: HD/CD = IDE0:0/IDE0:1, IDE0:0/IDE1:0, IDE0:1/IDE0:0.
 - I have successfully used both fixed sized and dynamically growing disk images for the hard disk.
 - I have successfully used both ISO CD-ROM images and physical CD's in CD drives.

- SCSI adapters: use either BusLogic or LSILogic; neither will work. *Step is supposed to support a "BusLogic" adapter, but the emulated hardware/driver combination doesn't work. If you wish vmWare not to emulate either SCSI hardware, edit the VM's .vmx file after you've finished creating the VM and change the ``scsi0.present = "TRUE"`` line to ``scsi0.present = "FALSE"``.
- floppy disk: you will have to add a floppy drive manually to the VM by going to "VM -> Settings -> Hardware" tab and clicking "Add..."
- network connection: you have the following 3 options, depending on how you want to set up your network

- NetOpt-1) "Bridged. Connected directly to the physical network."
- NetOpt-2) "NAT: Used to share the host's IP address"

Netopt-3) "Host-only: A private network shared with the host"

This can be changed later if needed.

- if you want a sound adapter, you have to add it manually in the VM Settings/Hardware Tab. The emulated adapter is Ensoniq ES1371, but the one(s) you will see displayed will be the one you physically have in the host computer.

- guest OS: use Linux/Other Linux (I have not tried "Other"; [14] suggests using "FreeBSD" but this option is not available in my version of vmWare; [11] reports choosing anything but "Linux" on vmWare 3.0 causes the CD driver not to work correctly).

- go to Virtual Machine Settings/Options/Advanced and unclick "Run with debugging information". This will make it run MUCH faster (however, although

the installation itself seems to run much faster for me with this setting, if, once installed, I try to toggle this setting and measure the

difference in performance, the VM subjectively continues to run as fast as if this setting was set to "no debug" and I cannot measure any difference

in performance --- see Performance section. It is worth noting that I observe the same behaviour if I try to toggle the "run in full screen mode

at VM startup" setting - it doesn't seem to affect anything. The behaviour is as if vmWare kept its real preferences somewhere other than the .vmx

file, although the vmWare GUI shows the preferences as per the .vmx file. This mystery is, as of now, unexplained). Some versions of vmWare have the

ability to set this option disabled in the GUI but manually editing the .vmx file and adding the ``debug = "FALSE"`` line still makes the VM run faster even though the GUI shows debugging as enabled.

- [14] suggests adding these two lines to vmWare VM configuration file:

```
monitor_control.simmmu = TRUE
monitor_control.mmu_adaptive = TRUE
```

I have not tried these and don't know what they do.

- boot order in this emulator is set in the BIOS setup: press F2 while booting up to enter BIOS setup and go to "Boot" screen. Select the device

("Removable Devices" for floppy, "Hard Drive" for hard disk, "CD-ROM Drive" for CD-ROM, "Network boot" for network boot; some of these also have

sub-categories) whose position in boot order you wish to change and press "+" to bubble it up the boot order or "-" to bubble it down the boot order.

As long as you are working with a blank new EIDE drive (not doing a re-install), you can set the boot order to hard disk/floppy/CD-ROM when you

create the virtual machine, and never change it; it will work throughout the installation.

- step 17): download the [3a] SVGA driver & the [4a], [4b] & [4c] sound, network and PS/2 mouse drivers ([13a] also mentions the [4.5] sound driver but

all the links pointing to this driver that I can locate now appear to be defunct; also, there are the [3a] and [3b] display drivers but I have not

succeeded in getting either driver to work in any mode other than 640x480 BW:2; whichever mode I tell them to come up in, I get 640x480 BW:2. That was

on OpenStep 4.2 under vmWare/MS Windows. [3b] is supposed to be for vmWare/Linux

- try them, YMMV). These drivers work on both NextStep 3.3 and OpenStep 4.2 except:

- [4c] driver: I cannot get sound to work in OpenStep 4.2 (but OK on NextStep 3.3)

This will give you the files VMWareFB-1.0.0-b-I.config.compressed, VMMouse-

1.1.I.bs.tar.gz, VMXNet-1.1.I.bs.tar.gz, SoundBlaster16PCI-1.0.I.bs.tar.gz;

get these into your VM via a virtual floppy or ISO image.

- now configure the drivers in Configure.app:

- display:

- remove the "Default VGA Adapter" driver
- the vmWare driver is either auto-detected, or else click Add and select the

"VMWare SuperVGA Video Adapter(Dev:15 Func:0 Bus:0) (v0.00)"

(if using the [3a] or [3b] driver)

"VMWare Display Adapter by Atomic Objects(Dev:15 Func:0 Bus:0) (v1.00)"

(if using the [3c] driver)

driver from the list

- choose the appropriate display mode, e.g. 1024x768xRGB:888/32, 60Hz.

- mouse:

- remove the "PS/2-Style Mouse" driver. vmWare emulates a PS/2 mouse, but the combination of the emulated hardware and the standard *Step PS/2 mouse driver does not work very well together; it is not usable for general use: it results in erratic & random mouse movement which gets worse with time after each reboot. This appears to be a combination of a bug in the *Step PS/2 mouse driver and an incomplete implementation of PS/2 mouse hardware by vmWare. This problem has been reported with vmWare 3.x, 4.x & 5.x. The driver works just about well enough to complete the installation process, but it's best not to touch the mouse until you've installed the better driver in this step - hopefully if you leave the mouse alone, it won't generate any random mouse keypresses which will abort the installation or do damage (it has done for me on occasion).

- install the vmWare-specific driver; the driver is either auto-detected, or else click Add and select the "VMWare virtual mouse driver (v1.1)" driver from the presented list
- set the Desktop layout parameters, e.g. x=0 y=0 w=1024 h=768 for a 1024x768 display. Make sure you press return after entering each number, otherwise it does not appear to be remembered.

- audio:

- click Add and select "SoundBlaster 16 PCI(Dev:18 Func:0 Bus:0) (v1.0)" driver from the presented list

Notes:

- for me, this driver works in NextStep 3.3, but I have not succeeded in getting it to work in OpenStep 4.2.
- the following drivers do not appear to work on the versions of vmWare I have tried: "System Beep Driver", "Sound Blaster 8", "Sound Blaster 16". [11] reports that also on vmWare 3.0, NeXT-supplied "Sound Blaster 16" driver does not work

- network:

- click Add and select the "VMWare virtual network adapter(Dev:17 Func:0 Bus:0) (v1.1)" driver from the presented list

Notes: the NeXT-provided "AMD PCnet-32 PCI Ethernet Adapter (Dev:17 Func:0 Bus:0) (v4.00)" is supposed to be compatible with the vmWare-emulated network hardware, but it does not work reliably. [12] reports that people have reported packet losses with this driver. I can only get very limited performance out of this driver. If I run ping immediately after a reboot, it succeeds in sending out about 28-31 (never 32) ping packets, and then the whole interface hangs. But [11] reports successfully using this driver in vmWare 3.0 and [13a] in vmWare 3.1.1. The [4b] driver works flawlessly in the versions of vmWare I tested.

- EIDE: perform this step only on OpenStep 4.2; NextStep 3.3 does not have the required driver, so you must keep the "Primary/Secondary(Dual) EIDE/ATAPI Device Controller" driver.
- remove the "Primary/Secondary(Dual) EIDE/ATAPI Device Controller" driver
- click Add and select the "Intel PIIX PCI EIDE/ATAPI Device Controller" driver. Tick "Bus Master DMA" (& leave "Multiple Sectors" ticked).
- The performance is an order of magnitude better with this driver; see the Performance section for the comparative time measurements.

- SCSI: do not install a driver. The hardware emulated by vmWare & the driver automatically detected by *Step is "BusLogic PCI SCSI Adapter"

- once running, you have the following keystrokes:
 - Alt-Ctrl-Enter: toggle full screen mode
 - Alt-Ctrl: release mouse

Virtual PC

Summary

1) Virtual PC version up to 6.0.x were released by Connectix, versions 6.1.x and up were released by Microsoft.

2) Various people have reported the following experiences running *Step on Virtual PC:

- a) Virtual PC 4.2: [7] reports running NextStep 3.3 successfully on VPC4.2.
- b) Virtual PC 4.3.1: [11] reports running OpenStep 4.2 on VPC4.3.1 x86 extremely reliably, but very slowly - at a speed of about 10% of the speed of the host CPU. [11] reports VPC4.3.1 consuming a lot of CPU time when OpenStep runs idle. OpenStep clock apparently drifts by approximately 30 minutes/day. [11] reports that capturing floppy images via the menu does not work in Virtual PC 4.3.1, but dragging the floppy image over the floppy disc icon works. [11] reports some problems with the keyboard under Virtual PC 4.3.1. [11] reports not being able to read NeXT and Macintosh format floppies when running VPC4.3.1 on Windows 98 & Windows XP Professional, but says these are reported to work when run on Windows 2000 SR2. So you may have to copy any physical installation floppies into floppy image files before running the installation procedure.
- c) Virtual PC 5.0/5.2:
 - [11] reports OpenStep 4.2 does not run reliably on VPC 5.2/5.2 x86. "about 25% of" processes are reported to fail with Bus Error, Memory Fault or Segmentation Fault & the system is reported to be very unstable.
 - [7] and [8] report running *Step on VPC5.0 (Mac OS X) well with networking, sound and 32-bit graphics.
- d) Virtual PC 6:
 - [8] reports VPC6 (Mac OS X) will boot OpenStep 4.2, but processes will randomly crash at various points.
 - [9] reports successfully installing and running OpenStep 4.2 on VPC6.0.1 (Mac OS X 10.2.3). He recommends using verbose boot ["-v"] at *Step boot prompt; *Step apparently otherwise hangs sometimes.
- e) Virtual PC 7: [8] reports VPC7 (Mac OS X) will not boot OpenStep 4.2 at all; apparently, the IDE driver doesn't work.
- f) Virtual PC 2004:
 - [8] reports problems with VPC2004 x86. If booting with config=Default, [8] reports OpenStep starts booting and then fails with "Unexpected exception D"
 - [10] reports running NextStep 3.3 successfully on VPC2004 x86.

Setup

- step 1) use these settings:
 - guest OS:
 - [11] reports that on Virtual PC 4.3.1 it doesn't matter what type of Guest OS you select
 - [7] suggests using "Other" OS in Virtual PC 5.0(Mac OS X)
 - RAM:
 - [8] reports using 128MB successfully under Virtual PC 5.0(Mac OS X)
 - [9] reports that 32MB is the maximum that will work under Virtual PC 6.0.1(Mac OS X 10.2.3).
 - hard disk:
 - [7] reports that dynamically growing disk files do not work on Virtual PC 4.2
 - [9] recommends using a fixed size disk image
 - configure HD on IDE primary channel (IDE0:0)
 - configure CD-ROM on IDE secondary channel (IDE1:0). [11] reports that in Virtual PC 5.0/5.2 x86, CD-ROM does not work if connected

to the secondary EIDE controller; must be connected to the primary EIDE controller.

- network:

[7] advises to run NextStep 3.3 on Virtual PC 4.2, networking must be configured for "virtual, host and external". Advises against using NAT.

- step 11): you may install the following display driver, or you may install the "Default VGA Adapter" driver now, and remove it and install this one in step 17):

- display:

- [7] has used "S3 Generic PCI Display Driver" driver from [FDDb1] successfully on NextStep 3.3 under Virtual PC 4.2. The driver from

[FDDb1] must be used; the released driver crashes the kernel according to

[7].

- [11] has used "Diamond Stealth 3D (S3 ViRGE)" driver successfully on OpenStep 4.2 under Virtual PC 4.3.1 in 1280x1024, 1024x768 256/8 colour, 60Hz mode. He reports "S3 Generic Display" driver also works in 888/32 & 256/8 colour modes, with any refresh rate.

- [8] has used "#9 Motion 771 PCI Display Adapter (4MB modes)" successfully with OpenStep 4.2 under Virtual PC 5.0 (Mac OS X) using RGB 888/32 mode, but not using RGB 555/16 mode.

- [9] has used "S3 Generic PCI Display Driver" successfully, using display mode 1024 x 768, 60Hz, RGB:888/32, with OpenStep 4.2 under Virtual PC 6.0.1 (Mac OS X 10.2.3). 16-bit display modes reportedly do not work.

- step 17) install the following drivers:

- hard disk/CD-ROM: [11], [9] and [8] recommend removing the "Primary/Secondary (Dual) EIDE/ATAPI Controller" driver and using the "Intel PIIX PCI EIDE/ATAPI Device Controller (Dev:7 Func:1 Bus:0)" from [FDDb1] with OpenStep 4.2 under Virtual PC 4.3.1 / 5.0/6.0.1 (Mac OS X).

This apparently works much faster. Note that this driver is not available for NextStep 3.3. According to [9] you must manually delete

"/usr/Devices/EIDE.config" and manually copy the version of "EIDE.config" from [FDDb1] into "/usr/Devices" to use this driver.

Configure this driver by selecting [11],[9],[8] BusMaster DMA, and [11] Multiple Sectors (for faster emulated disk access).

- network:

- [7] recommends using "DECchip 21040 Based Adapter"/"DECchip 21041 Based Adapter" driver from [FDDb1] with NextStep 3.3 on Virtual PC 4.2.

He refers to using the "DECchip 21040" driver.

- [11], [9] and [8] recommend using "DECchip 21041 Based Adapter" with OpenStep 4.2 on Virtual PC 4.3.1 / 5.0/6.0.1 (Mac OS X)

- [11] reports that DECchip21140NetworkDriver.config driver does not work in "DECchip 21140 Based * Ethernet Adapter Driver" modes (the

Ethernet address is not read properly), but works in "DEC DE500 * Ethernet Adapter Driver" modes on OpenStep 4.2 under Virtual PC 5.0/5.2 x86.

- audio:

- [11] and [8] have used "SoundBlaster 16 (8 and 16 bit DMA)" with default settings successfully with OpenStep 4.2 under Virtual PC 4.3.1 / 5.0 (Mac OS X)

- SCSI:

- [11] reports that Virtual PC 4.3.1 does not support SCSI.

- COM ports:

- [11] reports success using these, and the TTY port server. But reports that data rates above 38400bps do not work.

- LPT ports:

- [11] reports that these are supported. I have seen reports that people have

tried printing on LPT ports but can't find a reference at the moment.

- step 18), [8] recommends using the following settings on Virtual PC 5.0 (Mac OS X) - the credit for discovering these is due entirely to [8]:

- Internet Address: 192.168.131.66
- Router: 192.168.131.254
- Netmask: ?

[8] recommends afterwards editing /etc/rc.net and adding the following lines just before the comment lines

```
"Finally, let nmserver know the fruits of our network configuration endeavour":
```

```
arp -s 192.168.131.254 00:00:0F:00:00:00
arp -s 192.168.131.252 00:00:0F:00:00:02
```

and creating the file /etc/resolv.conf containing the following lines:

```
domain next.com
nameserver 192.168.131.252
```

Parallels Workstation

Summary

See also [18a], [18b], [18c].

1) I have successfully installed OpenStep 4.2 on Parallels Workstation 2.1.1670 running on Windows XP. It runs well. This is the state of the emulated hardware/drivers:

- graphics: works well and very fast with high resolution colour graphics.
- mouse: mouse emulation is smooth and reliable, using the default *Step PS/2 mouse driver.
- sound: emulates a Realtek AC'97 sound adapter; AFAIK, there is no driver currently available.
- networking: emulates a Realtek RTL8029(AS)-based adapter; not supported by NeXT/Apple-supplied drivers, but this is compatible with the Novell Ethernet NE2000 PCI card, for which there is a 3rd party driver [17] available, which works very well.

- SCSI: ?
- EIDE: the emulated hardware is not compatible with the "Intel PIIX PCI EIDE/ATAPI Device Controller (Dev:7 Func:1 Bus:0)" EIDE driver, so the generic "Primary/Secondary(Dual) EIDE/ATAPI Device Controller" driver has to be used; the emulated EIDE performance is very poor.

"Full screen" mode appears to have a flaw; it exists spontaneously (goes to windowed mode) without user input from time to time, and frequently

Setup

- step 1) use these settings:

- RAM: I've successfully used 32MB with OpenStep 4.2
- hard disk, CD-ROM: attach them to the IDE interfaces in one of the following combinations: HD/CD = IDE0:0/IDE0:1, [IDE0:0/IDE1:0, IDE0:1/IDE0:0].
[I have not tried the latter two but they are likely to work]
I have successfully used [both fixed sized and] dynamically growing disk images for the hard disk.
I have successfully used both ISO CD-ROM images and physical CD's in CD drives.

- SCSI adapters: I have not tried this.
- network connection: you have the following 3 options, depending on how you want to set up your network

257 = 640x 480x8 (256)	272 = 640x 480x16 (5/5/5)
259 = 800x 600x8 (256)	275 = 800x 600x16 (5/5/5)
298 = 896x 672x8 (256)	299 = 896x 672x16 (5/5/5)
261 = 1024x 768x8 (256)	278 = 1024x 768x16 (5/5/5)
302 = 1152x 864x8 (256)	303 = 1152x 864x16 (5/5/5)
314 = 1280x 800x8 (256)	315 = 1200x 800x16 (5/5/5)
306 = 1280x 960x8 (256)	307 = 1280x 960x16 (5/5/5)
263 = 1280x1024x8 (256)	281 = 1280x1024x16 (5/5/5)
310 = 1400x1050x8 (256)	311 = 1400x1050x16 (5/5/5)
284 = 1600x1200x8 (256)	285 = 1600x1200x16 (5/5/5)

- once running, you have the following keystrokes:
 - Alt-Enter: toggle full screen mode
 - Alt-Ctrl: release mouse & keyboard

Qemu

Summary

1) I attempted installing OpenStep 4.2/Intel on Qemu 0.7.2 running on MS Windows. I have not been successful. What follows are notes on how far the installation is able to proceed.

- the PS/2 mouse does not work in Qemu 0.7.2 ([8] reports the same in Qemu 0.61). This makes it impossible to complete installation on this emulator.

A patch for the main distribution of Qemu 0.8.0 is available [15] which allows Qemu 0.8.0 to emulate a bus mouse. This has been reported to work, but it breaks sound emulation because the bus mouse IRQ 5 clashes with the same IRQ for Soundblaster. [16] has patched Soundblaster emulation to change it to IRQ 7 and this then apparently works again.

- Qemu emulates the NE2k network card. There is no driver for this card supplied on the NeXT/Apple driver floppies and *Step CDs. However, there is an open-source driver available at [17] which runs on OpenStep 4.2/Rhapsody.

2) [16] contains notes on successfully installing NextStep 3.1/Intel on Qemu 0.8.0, with working bus mouse, audio & (as far as I can tell, 640x480) video. [has anyone tried using the VESA driver from the OpenStep 4.2 Y2k patch CD?]

Setup

- step 1) proceeds as follows: first start up a shell (your favourite shell on Linux, or Command Prompt in Windows).

Change the directory to where Qemu is installed ("Program Files\Qemu" in MS Windows).

The new virtual disk image is created by

```
qemu-img create -f raw bootdisk.img <size>
```

The virtual machine is "created" at the same time the emulator is started via

```
qemu -L bios -m RAM -fda FDI -hda bootdisk.img -hdc CDU -boot a -localtime
```

(substitute the number of megabytes of RAM you want to have for *RAM*, the path to where you have put the [FDI] floppy

image for *FDI* and the path to where you have put the [CDU] ISO image for *CDU*.)

Note this attaches the hard disk to the IDE0:0 interface and the CD-ROM to the IDE1:0 interface. Qemu cannot

handle the CD-ROM image being attached to the IDE0:1 interface, as it then connects IDE1:0 to the physical CD-ROM and that confuses OpenStep.

Boot order in this emulator is set by specifying the drive to boot from in the "-boot" command line option.

- step 6) between steps 6 and 7, the system will become confused and will ask you what the root device is. Type "hd1".

- step 8) after you press Return in step 8, the system won't be able to boot. Quit emulator & restart emulation via

```
gemu.exe -L bios -m RAM -fda FDD -hda bootdisk.img -hdc CDU -boot c -  
localtime
```

(making all the substitutions as before and substituting the path to where you put the [FDD] floppy image for *FDD*).

- step 11) you will not be able to make any changes because the mouse won't work.

Bochs

Summary

1) I attempted installing OpenStep 4.2/Intel on Bochs 2.2.6. So far, I have been unable to get this to work. Bochs just hangs.

GuestPC

Summary

1) I do not have GuestPC and I have not seen any attempts published on the Internet of anybody trying to install *Step on GuestPC.

Hard Disk Partition Sizes

See [20], [21], [22a].

- SCSI disks: *Step operates on the disk as a whole; each such disk has an associated /dev/{,r}sdn name for it, for some $n=0,\dots,9$.

- EIDE disks: EIDE disks can either be presented to *Step raw (in which case *Step operates on the disk as a whole), or else they can be partitioned into up to 4 DOS-style fdisk partitions. *Step contains a utility (fdisk) which allows the creation of such partitions.

This utility is also run during the installation process. If the disk is (DOS-style) partitioned, there can be at most one *Step partition and it must be contained within the first 8GB of the disk (according to [23], once you apply either patch 3 or patch 4 (the Y2k patch) to OpenStep 4.2, the OpenStep fdisk-style partition can be located anywhere on an EIDE drive). *Step then operates on this partition alone. The *Step space on the EIDE disk has associated the /dev/{,r}hdn name for it for some $n=0,1,2,3$.

In either case, *Step's space on each disk can be partitioned into up to 7 Unix-style partitions (lettered a-g; this letter is appended to the name associated to the disk to give the partition's device name), each of which can be up to 2GB in size (NextStep 3.3-OpenStep 4.0)

or up to 4GB in size (OpenStep 4.1-OpenStep 4.2). The 8th (h-lettered) "partition" is special, and corresponds to the entire disk.

[I have only the references to go by to claim that the transition occurred between 4.0 and 4.1. Has anyone checked this]

There are two issues:

- a) what is the maximum size of your boot disk on which you first install *Step
- b) what is the maximum number and maximum sizes of additional virtual disks you can attach to your VM.

We first discuss issue b), because limitations imposed on b) also apply to a).

First, the number of disks.

EIDE: EIDE devices are connected to one of up to two EIDE host controllers IDE0 (primary) and IDE1 (secondary), each controller being able to run up to two devices, a master (IDEx:0) and a slave (IDEx:1). A slave device can only be connected to a host controller if a master device is also connected to the same controller. Thus, there can be a total of between 0 and 4 EIDE devices.

Upon boot-up, OpenStep 4.2 scans the devices in the following order: IDE0:0, IDE0:1, IDE1:0, IDE1:1. As a disk is found attached to the appropriate interface, a device of the form /dev/hd*n* is assigned to it (and the appropriate /dev/{,r}hd*n*[a-h] files created if necessary), *n* starting at 0 for the first disk found (whichever of the interfaces it is attached to) and being incremented for each subsequent disk found. /dev/hd0 and /dev/hd1 are always created even if only one disk is attached to the computer, but /dev/hd2 is only created if a 3rd disk is found, and /dev/hd3 is only created if a 4th disk is found. Once created, the files remain even after a reboot.

Equally, CD-ROMs attached to the EIDE interfaces are assigned devices /dev/sd*n* with *n*=0,1,2,3 in the same fashion. Note that normally /dev/sd*n* devices correspond to SCSI devices, so EIDE CD-ROMs get assigned device names as if they were SCSI devices. I'm not sure in what order the device file numbers are assigned if there are also real SCSI devices in the system.

OpenStep 4.2 will work with a variety of CD-ROM and hard disk devices attached to the EIDE interfaces (I have successfully used 1-4 EIDE hard disks, and 0-3 EIDE CD-ROMs). Not all configurations work. Any configuration in which the hard disks are assigned to the interfaces IDE0:0, IDE0:1, IDE1:0, IDE1:1, in order, works, i.e. the following configurations are known to work:

```
IDE IDE IDE IDE
0:0 0:1 1:0 1:1
--- --- --- ---
HD - - -
HD HD - -
HD HD HD -
HD HD HD HD

HD CD - -
HD - CD -
HD CD CD -
HD - CD CD
HD CD CD CD

HD HD CD -
HD HD CD CD

HD HD HD CD
```

Any configuration where no CD-ROMs are present and the hard drives are NOT assigned to the IDE0:0, IDE0:1, IDE1:0, IDE1:1 interfaces in that order, will NOT work. E.g., the following configurations, which might otherwise be useful, will not work:

```
IDE IDE IDE IDE
0:0 0:1 1:0 1:1
--- --- --- ---
HD - HD -
HD - HD HD
```

Configurations where hard disks are not assigned in order, and which include CD-ROMs, are more complicated; some work, some don't.

If you are going to be installing *Step on a new disk, it is best to use, if you can, only one hard disk and one CD-ROM and install any additional devices later.

SCSI: [What is the maximum number of virtual SCSI disks that can be connected? Is the limit 7 targets per bus as per SCSI-1, or 15 targets per bus as per SCSI-2? How many host adapters can be installed in a *Step VM? Is there a limit of 10 caused by there being only 10 device files /dev/rsd[0-9]a ? Are more such files, e.g. /dev/rsdaa, created if more SCSI disks are connected?]

Secondly, the maximum disk size:

EIDE disks: you can proceed in one of two ways: either use the whole disk as *Step space, or create one DOS fdisk-style partition on the EIDE drive and use it as *Step space.

The maximum EIDE disk size OpenStep 4.2pl4 (that's OpenStep 4.2 with Y2k patch applied)'s fdisk program can handle is 8032.5MB (=7.84423828125GB).

This strange number comes from the following limits:

- BIOS Int 13h interface max. number of heads is 255 (not 256)
- BIOS Int 13h interface max. number of sectors/track is 63 (not 64 - sectors are numbered using 6 bits, starting at sector number 1)
- BIOS Int 13h interface max. number of cylinders as 1024.

1024 cylinders x 255 heads x 63 sectors/track x 512 bytes/sector = 8032.5MB = 8GB * (255/256) * (63/64). Any larger, and fdisk fails with a segmentation fault.

Partitioning a secondary EIDE/SCSI disk, and maximum partition size

A SCSI disk, or the *Step space on an EIDE disk, is prepared for *Step use by:

- a) writing the disk label ("partition table" in modern parlance); this splits the *Step space into a number (1-7) of partitions
- b) initializing a new filesystem in (formatting) each partition

a) The disk label contains the partitioning information for the disk - the numbers, starting offsets, sizes of each partition, as well as various other UFS filesystem parameters for each partition; the exact list of parameters is given in the comments in the /etc/disktab file which describe the parameters in disktab.

The disk label is normally prepared by the /usr/etc/disk program which performs a function similar to DOS fdisk or Mac OS X pdisk.

Without parameters, /usr/etc/disk splits the disk into a number of partitions in some default way (I have seen it generate 1,2,3 or 4

partitions in this way; it seems that most of the time this may be to split the disk into partitions of equal size, as few as possible, as long as their common size is <= 4GB, but I've seen it try to create a single ~8GB partition too); you can also give it a parameter which specifies the first (a) partition size, in terms of the number of device-native blocks (& then it splits the rest of the disk up in some default way and it seems to me again that it uses the same algorithm to do so as above). If you want any other kind of setup, you need to set up an entry in /etc/disktab which describes the partition scheme, and then run /usr/etc/disk and tell it to use that /etc/disktab entry to partition the *Step space. This is tricky, as using incorrect /etc/disktab information can cause a kernel panic.

b) is normally also done by the /usr/etc/disk program, after it has performed step a). /usr/etc/disk calls /usr/etc/newfs. /usr/etc/newfs has one parameter: the name of the partition to format; all it does is read the disk label to determine the required formatting parameters, and invokes /etc/mkfs with those parameters on the partition.

So, once you have partitioned the disk, you can also manually run /usr/etc/newfs or /etc/mkfs to format each partition individually. You

may want to do this, e.g. if you do not want to change the *Step space partitioning but want to reformat just an individual partition.

The largest UFS partition I have successfully created using /usr/etc/disk under OpenStep 4.2 was made by taking a 8032.5MB disk, not partitioning it using fdisk, and creating a single partition on it using /usr/etc/disk. It was 8024.5MB in size. This appears to work fine until close to 4GB worth of data is written to the partition, at which point the filesystem reports that it is full and the system becomes unstable.

So, what is the largest Unix-style partition within the 8032.5MB *Step space that can be used without causing system instability? I don't know the answer. Even a 0.5GB partition behaves strangely if it is completely filled up; I successfully created a "553,762,880" byte file on a disk which supposedly only contained 524,288 1024-byte sectors (536,870,912 bytes), however, at least the system did not become unstable. My guess is that a 4GB (4,194,304 1024-byte sectors) partition is the largest that is safe to use.

You can also use BuildDisk.app to prepare secondary disks; this does much more than just partition the disk and format its partitions; it also copies over various files. BuildDisk operates on an entire SCSI or EIDE disk and can split it into 1 or 2 partitions only.

[it would be interesting to write a program like Mac OS X's pdisk which allows for more user-friendly creation of the disk label, or else at least writes out the appropriate /etc/disktab entry so that the disk label can be written using /usr/etc/disk. Once the disk label is written, partition formatting is done straightforwardly using /usr/etc/newfs program]

Boot disk:

- although the maximum size of *Step space on a secondary EIDE disk is 8032.5MB (even if it has to be partitioned into smaller partitions), the maximum size of *Step space (either the whole drive, or a fdisk partition) you can install OpenStep 4.2 in is 4GB (exactly), i.e. the boot drive can utilize no more than 4GB total. Attempting to install on a disk of size 4GB will succeed, but installing on a disk of size 4GB+1/2kB will cause a kernel panic in step 14).

If you attempt to install OpenStep 4.2 on an EIDE disk larger than 4GB, either you will not be able to select any packages to install in step 12) (if the disk size mod 4GB is less than the disk space required to install OpenStep), or, if you get past this point, a kernel panic will occur in step 14) as the system is booting up.

I have not tried installing OpenStep on a 4GB disk, then creating a 8032.5MB disk split into two partitions (4GB + rest), and then copying the 4GB disk onto the first partition of the 8032.5MB disk. If successful, this would allow the use of a 8032.5MB boot disk.

A technical note:

- every disk has a size (in 512-byte sectors), and two geometries: one which BIOS gives and fdisk sees, and another which /usr/etc/disk sees.

Here are some examples of disks created by vmWare and Parallels Workstation and their geometries:

Emulator + specified disk size	Actual disk size (sectors)	
"fdisk" geometry	"disk" geometry	(cyl x
trk x spt)	(cyl x trk x spt)	-----
vmWare 4GB	8388608	521 x
255 x 63 = 8369865	8322 x 16 x 63 = 8388576	
vmWare 4GB + 1/2kB	8388609	522 x
255 x 63 = 8385930	8322 x 16 x 63 = 8388576	
Parallels 4096MB	8389584	521 x

255 x 63 = 8369865 8323 x 16 x 63 = 8389584

It appears that:

- for large enough disks, the "fdisk" geometry is always X x 255 x 63, and the "disk" geometry is always Y x 16 x 63, in both vmWare and Parallels Workstation.

- vmWare creates a disk of size exactly as specified, to the last 1 sector, and then calculates "fdisk" and "disk" geometries to fit inside this disk, i.e. to each yield a number of sectors which is no greater than the total disk size

- Parallels creates a disk of size exactly CYL x TRK x SPT sectors, where CYL, TRK and SPT are as per the disk's "disk" geometry, and are chosen so that the disk is at least as large as the user-specified size.

Thus, a "4GB" vmWare disk is 4GB and OpenStep 4.2 installs fine on it, but a "4096MB" Parallels disk is actually approximately 4096.48MB in size, and thus installation of OpenStep 4.2 onto it fails.

Maximum file size:

- the maximum size of a file in OpenStep 4.2 is 2GB-1B ($2^{31}-1$ bytes).

Large Filesystems

While the maximum "native" disk space it is possible to use with OpenStep 4.2 is ($n \times 7 \times 4GB + 3 \times 8032.5MB + 4GB$), where n is the maximum number of SCSI (virtual) disks that can be attached to an *Step VM (I do not know the value of n, but in practice for all emulators I know about, it is 0 as SCSI is not supported), being able to run *Step efficiently in a virtual machine makes much more attractive the prospect of using the NFS file system to get around the 2GB/4GB filesystem size limit. This can be set up to run quite conveniently on a laptop, for example.

Namely, if OpenStep 4.2 is run in an emulator on a Linux (or other Unix) box, the native OS can have a large disk partition (or several such partitions) set aside for *Step use, and can be set up to NFS serve that partition. *Step can be set up with a host-only (or another type of) networking and to NFS-mount the partition(s) exported by the host OS. NFS version 2 uses a 256-bit file-handle + 32-bit file-relative byte offset addressing to access the files within the filesystem. Thus, a single machine can be set up in this way to run *Step with an apparent filesystem of size much larger than 4GB. I am not yet sure where the first limit is reached; perhaps it is imposed by the STATFS NFS RPC call which returns a 32-bit number of blocks in the filesystem. With a 512-byte block size, this would give a limit of perhaps 1TB or 2TB? The (2G-1)B maximum single file size limit also remains.

The NFS filesystem semantics are a little different from the UFS filesystem semantics, especially where permission checking is concerned, but people regularly use it to access their home directories in a networked environment, so it is reasonable to assume this should work well to run *Step in this way.

I have successfully used such setup (with another machine being the NFS server exporting a disk much larger than 4GB) a few times but not on a long term basis.

It would be interesting to evaluate the long-term stability, usability and performance of such a setup.

Performance

In this section, I shall list the results of performance benchmarks. It would be interesting to hear from others running other emulators/configurations as to how well each configuration runs so we can see which is the best emulator. For now, I ran the following benchmarks: they must all be compiled without optimization.

cpu: count up to 10,000 1,000,000 times

write: create a new file and write a buffer of 16384 NULs to it 8192 times. In each run, it was the first time the file was read after a reboot of

the VM to ensure it was not being read from cache
 write2: as per write, but use random data instead of NUL bytes.
 read: open a file of 128MB of NULs for reading and read 16384 bytes from it
 8192 times
 display: write 1,000 random 80x24 screenfuls to an 80x24 terminal (using
 default font size)
 net: ftp a 128MB file over a LAN in binary mode: ftp running on *Step,
 outgoing file transfer local host to switch (100Mbps wired) to Airport
 Extreme (100Mbps wired) to iBook G4 1.33GHz (54Mbps wireless)
 combo: find /NextDeveloper -print in an 80x24 Terminal.app window. A
 combination of disk read speed, CPU speed & display speed benchmark.

I did all the timings by an external timing device as well as *Step VM's clock as I
 wasn't sure how accurate the emulated time would be. It turns out it is pretty
 reliable, the absolute accuracy error for short term tests like these is
 unobservable. I ran each test 3 times; the table contains the mean and the st.dev.
 (the latter in faint gray).

In the table, all times given are in seconds, and I use the following
 abbreviations:

NS3.3 = NextStep 3.3
 OS4.2 = OpenStep 4.2

Dual = "Primary/Secondary(Dual) EIDE and ATAPI Device Controllers" EIDE driver
 PIIX = "Intel PIIX PCI EIDE/ATAPI Device Controller" EIDE driver

fix = fixed size EIDE disk image
 dyn = dynamically growing EIDE disk image

DBG = emulator/virtualizer running in debugging mode
 NDBG = emulator/virtualizer running in non-debugging mode

If anyone is planning to run these benchmarks, please contact me for the sources so
 that we all run the same benchmark. Also, these are rather rudimentary so
 suggestions for better benchmarks are sought.

Host: Packard Bell EasyNote M5 262, Intel Celeron 2.6GHz CPU, 256MB RAM, Fujitsu
 MHS2020AT 20GB 4200rpm/12ms HD, 1024x768 colour display,
 vmWare Server 1.0.0.22874beta

Guest	cpu	write	write2	read
OS4.2, PIIX,2GB dyn, 64MB RAM, NDBG	17	9		12
OS4.2, PIIX,2GB dyn, 64MB RAM, DBG	16.7	9.3		
OS4.2, Dual,2GB dyn, 64MB RAM, NDBG	17	62.7		
OS4.2, PIIX,1GB fix, 64MB RAM NDBG	16	47.7	48.3	
44.3	71.3	150.7		
63.3	Tomaž Slivnik			
NS3.3, Dual,1GB fix, 32MB RAM, NDBG				

Host: Packard Bell EasyNote M5 262, Intel Celeron 2.6GHz CPU, 256MB RAM, Fujitsu
 MHS2020AT 20GB 4200rpm/12ms HD, 1024x768 colour display,
 Parallels Workstation 2.1.1670

OS4.2, Dual,4GB dyn, 32MB RAM	17.2	205	218	
404	131	711.3		
52.3	Tomaž Slivnik			
OS4.2, Dual,1GB fix, 32MB RAM				

Host: Dell Inspiron 7000, Intel Pentium II 333MHz NB DIX MMC2, 64MB RAM, Fujitsu
 MHF2043AT 4.32GB 4200rpm/13ms HD, 1024x768 colour display,
 native

OS4.2, Dual	150.4	55.7	61.3	
			24	

Tomaž Slivnik

Host: iBook G4, PowerPC G4 1.33GHz CPU, 1.5GB RAM, Fujitsu MHV2100AT 100GB 4200rpm/HD, 1024x768 colour display, native

Mac OS X 10.4.6 68.3 8.9 9.9

0.9 Tomaž Slivnik

Host: iBook G3, PowerPC G3 600MHz CPU, 768MB RAM, Hitachi IC25 N080ATMR04-0 80GB 4200rpm/12ms HD, 1024x768 colour display, native

Mac OS X 10.4.5

Notes: despite the "display" performance figures above, Parallels Workstation display performance seems subjectively to be much better than vmWare's. It might be interesting to try running vmWare using the "VESA VBE 2.0 Display Driver" but that doesn't work, unfortunately.

Migrating Between Different Emulators

[this section is currently very incomplete]

Migrating a VM and its boot drive from one emulator to another is possible, provided the incoming emulator can read outgoing emulator's hard disk file. That is generally the case for flat/raw/fixed-sized disk images: these are all the same for vmWare, Parallels Workstation, Qemu. The flat file format consists of a raw copy of the disk's sectors, in sequence.

Most emulators also offer one or more compressed/dynamically growing disk formats; Bochs: sparse, growing; Qemu: cow, qcow, vmWare: vmdk, Parallels: hdd. Qemu can read/write vmWare's formats; I do not know about others. If enough disk space is available, each emulator [does that include Bochs?] is able to convert the compressed/dynamic disk image to and from the raw format.

Once the disk image is imported, the installed drivers will have to be changed, but that's not a problem. Any one of the following approaches should work:

1) within the original emulator, uninstall all the VM specific drivers and install generic ones (Default VGA, PS/2 Mouse, ...) just before migrating the VM.

2) boot the new VM up using a variation of the following parameters:

```
"Active Drivers" = "MOUSE VGA" "Boot Drivers" = "EISABus PCIBus  
PCMCIABus PCIC PS2Keyboard Floppy Intel824X0 EIDE"
```

where *MOUSE* is either PS2Mouse, BusMouse or SerialMouse [NextStep 3.3]/SerialPointingDevice [OpenStep 4.2], as appropriate to your situation. Then re-install the new drivers when the new VM boots up.

3) create a separate configuration file for each VM in /private/Devices/System.config. This allows for using the same boot disk image within several

emulators without any change, using only the

```
config = ...
```

option at boot time. [more details needed here]

Work Required

It would be interesting to update these notes for Rhapsody, but I don't have a copy. [19] suggests Rhapsody DR2 runs on Virtual PC 4 and vmWare.

Acknowledgements/Disclaimers

These notes come with no warranty. Use at your own risk.

This document grew out of some notes I originally prepared for my own purposes. I have released it in the hope that it will be useful to other people. If you feel I have left out interesting information, or missed out an important document, can answer some questions listed in document, or have a suggestion for improvements, I would like to hear about it. Finally, please note that this document is a draft; there are likely to be some very rough edges.

References

- [1] NeXT Computer, Inc., Installing and Configuring NEXTSTEP Release 3.3 for Intel Processors and NeXT Computers, November 1994.
- [2]
http://download.info.apple.com/Apple_Support_Area/Apple_Software_Updates/MultiCountry/Enterprise/nextstep/floppyimages/
http://download.info.apple.com/Apple_Support_Area/Apple_Software_Updates/MultiCountry/Enterprise/openstep/floppyimages/
http://mirror.apple.com/Mirrors/Apple_Software_Updates/MultiCountry/Enterprise/openstep/floppyimages/
http://mirror.apple.com/Mirrors/Apple_Software_Updates/MultiCountry/Enterprise/nextstep/floppyimages/
ftp://ftp.apple.com/Apple_Support_Area/Apple_Software_Updates/MultiCountry/Enterprise/openstep/floppyimages/
- [3a] Frederick Starck (fred@almonde.com)'s SVGA driver for OpenStep 4.2/Intel on VMWare/Windows.
<http://www.chez.com/fstark/VMWareSVGA.tar.gz>
- [3b] Peter Ramussen (rasmus@interactive-technology.com)'s SVGA driver for OpenStep 4.2/Intel on VMWare/Linux.
<http://www.moldus.org/~laurent/GNUstep/files/VMWareSVGA-linux.tar.gz>
- [3c] Bill Bereza (billbereza@atomicobject.com)'s SVGA driver for OpenStep 4.2/Intel on VMWare/Linux and VMWare/Windows.
<http://www.atomicobject.com/download/openstep-driver.html>
<http://atomicobject.com/pr-openstep-driver.page>
- [4a] Jens Heise, NeXTStep/OpenStep VMMouse driver
http://mitglied.lycos.de/Morgon_de/downloads/VMMouse-1.1.I.bs.tar.gz
- [4b] Jens Heise, NeXTStep/OpenStep VMXNet driver
http://mitglied.lycos.de/Morgon_de/downloads/VMXNet-1.1.I.bs.tar.gz
- [4c] Jens Heise, NeXTStep/OpenStep vmWare SoundBlaster16-PCI (ES1371) driver
http://mitglied.lycos.de/Morgon_de/downloads/SoundBlaster16PCI-1.0.I.bs.tar.gz
- [4.5] Stephen Brandon, University of Glasgow, Music Department, SoundBlaster driver for NextStep 3.3/OpenStep 4.2 with MIDI support
<http://www.peanuts.org/peanuts/NEXTSTEP/system/driver/audio/SBSoundmidi.I.b.tar.gz>
<ftp://ftp.music.gla.ac.uk/pub/NeXT/Drivers/SBSoundMidi.I.tar.gz>
- [5] Howard R. Cole, "Openstep USB driver project", comp.sys.next.programmer Usenet posting, 28 January 2000.
Howard R. Cole, "USB Audio Driver on OpenStep - Update", comp.sys.next.software Usenet posting, 9 November 2000.
Howard R. Cole, "USB beta Audio driver", comp.sys.next.software Usenet posting, 22 November 2000.
Howard R. Cole, "Can't access built-in USB ports under OpenStep", comp.sys.next.programmer Usenet posting, 30 November 2000.
- [6] Tomaž Slivnik, Which NeXT/Apple original and 3rd Party drivers are available

for NextStep/OpenStep?

- [7] Nicolais Leymann (?), NextStep 3.3/Intel on Virtual PC 4.2/5+ Installation
<http://www.nleymann.de/Nextstep/VirtualPC.htm>
<http://www.leymann.info/Nextstep/index.htm>
<http://www.nleymann.de/Nextstep/index.htm>
- [8] (name of author unknown), Running OpenStep under Virtual PC 5 for Mac
<http://ul.netgate.net/~snowcat/OpenStep.html>
<http://ng.netgate.net/~snowcat/OpenStep.html>
- [9] Chris Roehrig, Installing OpenStep 4.2 on Virtual PC 6
http://www.simski.org/next/software/openstep/info_os/OS4.2_on_VPC.txt
- [10] Tom ?, NEXTSTEP 3.3 on a Virtual PC 2004
<http://users.adelphia.net/~dialtr/next.html>
- [11] Ken Turner, OPENSTEP on Windows PC Emulators
<http://www.cs.stir.ac.uk/~kjt/software/nextstep/openstep-win.html>
- [12] Laurent Julliard (laurent@moldus.ort), Installation of OpenStep 4.2 in VMWare 3.0
http://www.moldus.org/~laurent/GNUstep/OS42_Install.html
- [13] Alejandro Diaz Infante, OPENSTEP and VMWare, NeXTeZine 01/2003
- [13a] Alejandro Diaz Infante (astroboy), HOW TO SETUP OPENSTEP IN VMWARE 3.X
<http://www.nextcomputers.org/forums/viewtopic.php?t=17>
- [14] Alex Protopopescu, Mach on vmWare?
http://software.groupbrowser.com/Mach_on_VMWare_-t33487.html
--- Note that Alex Protopopescu worked at vmWare, at least at the time of writing his posting
- [15] Michael Engel ("cuby"), OpenStep / Busmouse - correct patch
<http://www.mail-archive.com/qemu-devel@nongnu.org/msg03326.html>
- [16] neozeed, Installing Nextstep 3.3 in qemu on Windows XP
<http://qemu.dad-answers.com/viewtopic.php?p=4874#4874>
- [17] Gabor Sebestyen, NE2K OpenStep 4.2/Rhapsody driver
ftp://ftp.cs.tu-berlin.de/pub/NeXT/hardware/driver/intel/NE2K_0.91beta.README
ftp://ftp.cs.tu-berlin.de/pub/NeXT/hardware/driver/intel/NE2K_0.91beta.m.I.b.tar.gz
ftp://ftp.cs.tu-berlin.de/pub/NeXT/hardware/driver/intel/NE2K_0.91beta.m.I.s.tar.gz
- [18a] Graham Lee's blog
<http://iamleeg.blogspot.com/>
- [18b] Mac on Intel: NeXT's OPENSTEP up and running on a MacBook Pro
<http://www.maconintel.com/news.php?article=179>
- [18c] <http://www.flickr.com/photos/mrbill/sets/72057594112307196/>
- [19] <http://forum.osx86project.org/lofiversion/index.php/t2991.html>
- [20] comp.sys.next.misc, "FAQ Installing OpenStep on a laptop (Draft)" thread, 8 March 2000 onwards
Tomaž Slivnik, "FAQ Installing OpenStep on a laptop (Draft)", 8 March 2000
Uli Zappe, "Re: FAQ Installing OpenStep on a laptop (Draft)", 7 April 2000
Rex Dieter, "Re: FAQ Installing OpenStep on a laptop (Draft)", 10 April 2000
- [21] comp.sys.next.sysadmin, "max partition size" thread, 23 July 1997
Allen Mitchell, "Re: max partition size", 25 July 1997
- [22] Bernhard Scholz, The NEXTSTEP/OpenStep FAQ
<http://www.channelu.com/NeXT/NeXTFAQ-new/NeXTFAQ.toc.html>

[22a] <http://www.channelu.com/NeXT/NeXTFAQ-new/NeXTFAQ.173.html>

[22b] <http://www.channelu.com/NeXT/NeXTFAQ-new/NeXTFAQ.162.html>

[23] <http://docs.info.apple.com/article.html?artnum=70049>

[24] Most Common FAT Types
<http://oldfiles.org.uk/powerload/fat32/fat2.htm>

[25] <http://mtools.linux.lu/>
<http://www.gnu.org/software/mtools/mtools.html>
<http://freshmeat.net/projects/mtools/>