

The
BeOS

BIBLIE

SCOT HACKER

WITH HENRY BORTMAN AND CHRIS HERBORTH



Peachpit Press

The BeOS Bible

by Scot Hacker

with Henry Bortman and Chris Herborth

Peachpit Press

1249 Eighth Street
Berkeley, CA 94710
(800) 283-9444
(510) 524-2178
(510) 524-2221 (fax)

Find us on the World Wide Web at: www.peachpit.com

Peachpit Press is a division of Addison Wesley Longman

Copyright © 1999 by Scot Hacker

Notice of rights

All rights reserved. No part of this chapter may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission of the publisher. For more information on getting permission for reprints and excerpts, contact Gary-Paul Prince at Peachpit Press.

Notice of liability

The information in this chapter is distributed on an "As is" basis, without warranty. While every precaution has been taken in the preparation of this chapter, neither the author nor Peachpit Press shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this chapter or by the computer software and hardware products described herein.

Trademarks

Be, BeOS, and the Be and BeOS logos are registered trademarks of Be Incorporated in the United States of America and other countries. All other products and company names mentioned in this book may be trademarks of their respective owners.

ISBN: 0-201-35377-6

Emulation

Types of Emulation • 4

SheepShaver (MacOS Runtime Environment) • 6

Emulators • 20

Game Emulators • 24

Chapter Summary • 25

In this Chapter

Types of Emulation • 4

SheepShaver

(MacOS Runtime Environment) • 6

System Requirements • 7

Acquiring SheepShaver • 7

Installing SheepShaver • 7

Launching SheepShaver • 8

SheepShaver Settings • 9

Memory Settings • 13

Display Modes • 14

Using SheepShaver to Transfer

Data and Files • 16

Quitting SheepShaver • 17

Known Problems • 18

Troubleshooting • 18

Emulators • 20

Bochs (x86) • 20

UAE (AmigaOS) • 21

Basilisk (Mac Classic) • 22

Open Bernie (Apple II+, IIe, IIGs) • 22

BeBeeb (Acorn BBC Micro) • 23

BeZX (Sinclair Spectrum) • 23

Frodo (Commodore 64) • 24

Game Emulators • 24

*BeMAME (Multiple Arcade
Machine Emulator)* • 24

*Snes9x (Super Nintendo
Entertainment System)* • 25

Chapter Summary • 25

When you use a computer, you don't think about the hardware you're running—you're immersed in a visual environment governed by the operating system and its applications. Whether you're deep in the dungeons of *Doom* or typing away in a BeatWare Writer document, your hardware fades into invisibility as the interface takes over. But the environment suggested by the operating system is never absolute—just because most of your apps conform to the general appearances and behaviors of the BeOS universe, that doesn't mean that all of them have to. When you're playing a full-screen game, for example, the game emulates another world within the context of BeOS.

But why stop with spaceports and mystical worlds? Why not emulate entire operating systems running on other types of hardware? BeOS emulators exist to let you run a copy of the AmigaOS or the MacOS inside a BeOS window, play Nintendo cartridge games in system RAM, or pretend that you're sitting at the helm of an ancient Sinclair Spectrum.

While some people experiment with emulators out of curiosity rather than necessity, a well-implemented emulator can save you from having to reboot by enabling you to accomplish tasks that normally can only be done in another system. This chapter offers only a brief overview of the emulators available for BeOS as of R4.0. Some of the emulators covered here (like SheepShaver, the MacOS emulator, which isn't truly an emulator at all) can be genuinely useful, while others (like BeBeeb, the Acorn Micro emulator) are probably around only for nostalgia's sake.



In this chapter, the term “host system” refers to the primary running operating system—the OS that booted the machine and that has ultimate control over the physical hardware. The term “alien system” refers to the OS being emulated, since it's being run outside of its natural home. There is no intended implication that some of these operating systems seem to have been designed by aliens.

Types of Emulation

Before we dig in, it's important to distinguish between true emulation, where the emulating software actually mimics another platform's hardware, and "runtime environments," where the original hardware is not emulated and a sort of "negotiating layer" is used instead.

What's the distinction? As seen in Figure 1, when the alien OS cannot use the machine's physical hardware, an emulator must fool it into thinking that the hardware it wants is actually present. But it's not enough to just emulate a physical hardware environment in software. If you want access to your disk drives, and want your monitor and keyboard to work, the emulator has to communicate hardware calls back to the host OS, which in turn sends them to the physical hardware. Understandably, this process consumes a lot of resources, which usually makes true emulation cumbersome and slow. In fact, it's theoretically impossible for a system to run in emulation as quickly as it does on its native hardware (however, older operating systems are often emulated on current hardware, making it possible to more than compensate for performance penalties).

On the other hand, if both operating systems expect to find the same physical hardware in the machine, there will be no reason to emulate the hardware. All that's required then is the presence of a software layer that knows how to manage traffic between the alien OS and the host OS. This layer is called a "runtime environment," and its job is only to negotiate communications,

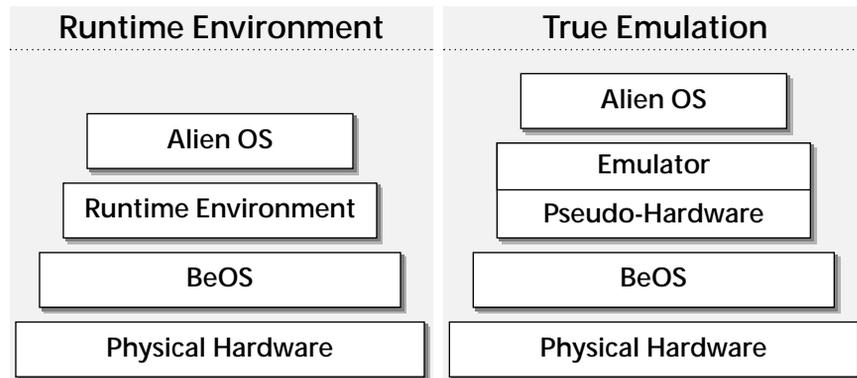


Figure 1

If the alien OS requires different hardware than that which is physically present in the machine, a true emulator is required to create the illusion of that hardware in system memory. If both BeOS and the alien OS are compatible with the physical hardware, emulation is not required. Instead, a "runtime environment" is used to negotiate between the two systems.

not to emulate hardware. As a result, runtime environments are faster than true emulators. A runtime environment is also sometimes referred to as a “virtual machine.”

To take an example, consider the Un*x Amiga Emulator (UAE), which lets you run AmigaOS within BeOS, regardless of whether you own a Mac or an x86 machine. Because the Amiga’s hardware is different from both the Mac and the x86, AmigaOS needs to have that hardware emulated; thus UAE is a true emulator. On the other hand, consider SheepShaver, which lets you boot and run MacOS in a BeOS window. Because both MacOS and BeOS know how to communicate with PowerPC hardware, hardware emulation is not necessary. Instead, MacOS can be made content to run inside of BeOS by providing it with a negotiating layer that knows how to turn MacOS system calls into BeOS system calls, so that you can, for example, copy and paste data back and forth from one operating system to the other. This is an example of a runtime environment, not a true emulator. It also explains why SheepShaver can run MacOS inside BeOS just about as fast as it would run natively.

Staying Legal

In most cases, emulators do not include the operating system being emulated; they only provide an environment in which that system can run. This means that you must already own a legal copy of that system. If the system being emulated is already on your own hard drive (as is the case when you have both MacOS and BeOS installed on one machine and use SheepShaver to run MacOS from BeOS), no problem. But in some cases the process is much more involved. With the Un*x Amiga Emulator, for example, you must make a copy of your Amiga’s ROM and transport it to your BeOS machine. In order to obtain an Amiga ROM, you must own an Amiga computer (or purchase the ROM separately through a legitimate vendor). The Web sites providing emulators won’t help you obtain ROMs, and neither will this book. In fact, it is for this reason that we weren’t able to test all of the emulators described in this chapter.

So does this imply that emulation is a shady business, or in any way illegal? That depends on how you go about it. In the case of older hardware where the original company has gone out of business, emulators and ROM images are sometimes traded freely on the Internet, perhaps with the assumption that no one cares (though you have a responsibility to find out). In the case of still-active companies, a strong distinction needs to be made between the emulator itself and the system code that runs inside it. Anyone can legally download and own an emulator, but using an OS or game ROM you don’t legitimately own is theft.

SheepShaver (MacOS Runtime Environment)

by Henry Bortman

SheepShaver is a runtime environment that allows you to use the MacOS side by side with BeOS. Of course it works only on MacOS systems supported by BeOS. It does not run on Intel boxes. SheepShaver appears as an application icon on the BeOS Desktop. When you launch it, it boots the MacOS within a BeOS window or workspace. Once the MacOS is active, you can run Mac applications and perform network activities just as you would directly from within the MacOS itself. SheepShaver is *not* a MacOS emulator. It runs the MacOS as a native PowerPC runtime environment. As a result, its performance is excellent (in full-screen mode; see *Display Modes*, below). For most activities, SheepShaver is as fast as running the MacOS directly on the same machine.

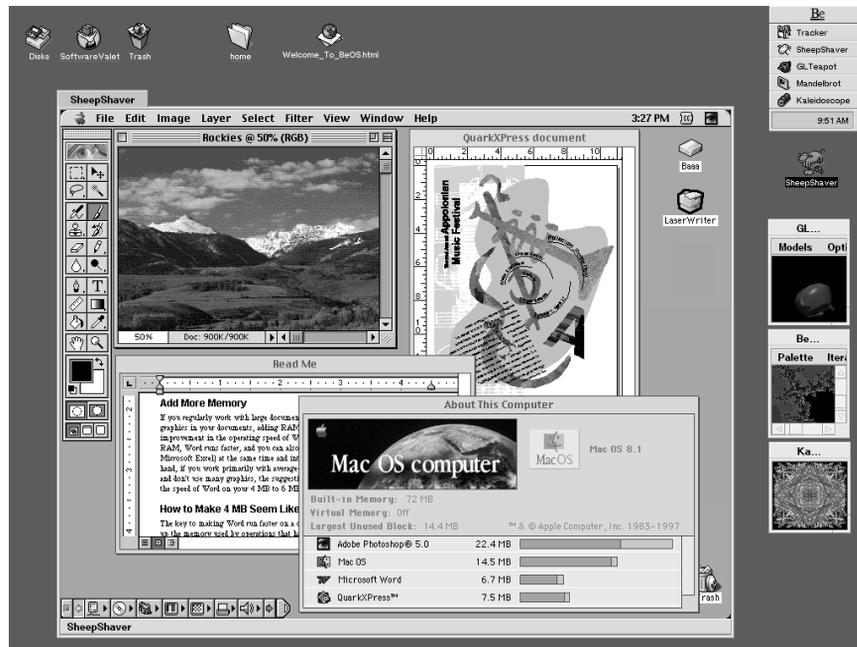


Figure 2

SheepShaver running the MacOS within a BeOS window; Microsoft Word 6.0.1, Photoshop 5.0, and QuarkXPress 4.0 are all active.

SheepShaver's creators, Christian Bauer and Marc Hellwig, claim that in theory any application that will run under the MacOS should run under SheepShaver as well. In practice, it's not that simple. As Figure 2 shows, I was able to get a number of popular applications, including Word 6.0.1, Photoshop 5.0, and QuarkXPress 4.0, to run successfully under SheepShaver.

I also succeeded in printing and in transferring files to and from another Mac with MacOS file sharing. That's the good news. The bad news is that it took several hours of tweaking to get everything to work (see *Known Problems* and *Troubleshooting*, below). And I never did get my modem to work properly. Be forewarned that software that accesses the Mac's hardware directly—many games fall into this category—is likely to cause problems.

System Requirements

SheepShaver 1.1 (the current version at the time of this writing) will run on any MacOS system that supports BeOS. This includes multiprocessor systems, although the MacOS applications that run within SheepShaver can't take advantage of the multiple processors.

SheepShaver supports versions 7.5.2 through 8.1 of the MacOS. Support for MacOS 8.5 is in the works. On the BeOS side, SheepShaver requires BeOS for PowerPC R3. (SheepShaver 2.0, when it becomes available, will require BeOS for PowerPC 4.0.)

To install SheepShaver, you will need 300K of free hard disk space on your BFS volume. To run SheepShaver, you will need a minimum of 16MB of physical RAM. Of course, you'll also have to have at least one MacOS partition available that contains a bootable version of the MacOS.

Acquiring SheepShaver

SheepShaver is available only as a download from BeDepot (www.bedepot.com). Both a trial version and a fully functional shareware version (\$50) are available. In the trial version, full-screen mode is disabled and an annoying message pops up on your screen every couple of minutes reminding you that you haven't paid yet. When you pay the shareware fee, you will receive by email a "key" file that enables full-screen mode and disables the warning messages. If you plan to use SheepShaver regularly, pay the shareware fee. You get *way* better performance in full-screen mode—not to mention the karmic benefits.

Installing SheepShaver

SheepShaver installation is simple: SoftwareValet does all the work for you. When you double-click the SheepShaver Trial.pkg icon, the BeOS will launch SoftwareValet. If this is the first time you've used SoftwareValet, it will present you with several configuration options (see Chapter 10, *System Tools and Utilities* for more on SoftwareValet). Once you get past this initial screen, you'll be given the opportunity to install SheepShaver.

By default, SheepShaver will be installed in the apps folder of your BeOS boot drive. Unless you have a good reason to change this location, stick with the default. SoftwareValet will also give you a choice between Standard and Minimal installs, but actually it doesn't make any difference which one you choose. If you've configured SoftwareValet to display installation previews, when you click the Begin button, a Preview Install window will open, showing you where SheepShaver's various files will be installed. Again, unless you know what you're doing and have a good reason for doing it, stick with the defaults.

When the installation finishes, a SheepShaver registration window will appear. See Chapter 10, *System Tools and Utilities*, for more information on registering software with SoftwareValet.

Launching SheepShaver

Performing a few setup tasks before launching SheepShaver for the first time will make things go more smoothly.

- Make sure that virtual memory is turned off in the MacOS Memory control panel. Don't worry, you don't need it anyway. SheepShaver uses BeOS's virtual memory system, which is superior to the Mac's.
- Disable all Mac extensions that you don't really need. SheepShaver is prone to extension conflicts. The fact that an extension works when you run the MacOS directly doesn't guarantee that that same extension will work under SheepShaver. The following extensions are known to cause problems: RAM Doubler, Speed Doubler, MacOS Licensing Extension, and 68K FPU emulator extensions (such as PowerFPU); but this is by no means a comprehensive list.
- Once you get SheepShaver working, you can start reenabling your favorite extensions. An extension-management utility like Conflict Catcher or Now Startup Manager is helpful for this task; the MacOS's built-in Extensions Manager will work, too, but it lacks some useful bells and whistles.
- If you intend to use AppleTalk network services from within SheepShaver, make sure that you have set up an Ethernet interface in the BeOS Network preferences panel (see Chapter 9, *Networking*, for details). Of course, if your Mac doesn't *have* Ethernet, you can't very well set up an Ethernet interface for it in BeOS. See *Network Settings*, below, for what to do in this case. LocalTalk networking is not supported.
- In BeOS, mount any Mac volumes that you will want access to from within SheepShaver. This includes removable cartridges.

Launch SheepShaver as you would any other BeOS application, by double-clicking its icon. Before the MacOS environment appears, SheepShaver displays a SheepShaver Settings panel, in which you can configure various aspects of the program's behavior. Details about these settings are described below in *SheepShaver Settings*.



The first time you launch SheepShaver, it will boot the MacOS from the first boot volume it finds, regardless of which volume is designated as the startup disk in the MacOS's Startup Disk control panel. If you want SheepShaver to boot from a specific MacOS volume, you must select that volume in the Startup Disk control panel *from within SheepShaver*.

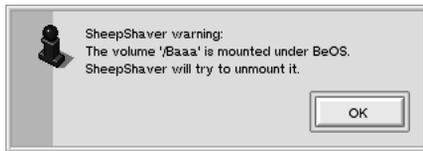


Figure 3

SheepShaver warns you before attempting to mount HFS volumes that are already mounted in BeOS.

If you have any Mac volumes mounted in BeOS that you have added to the SheepShaver volumes list, a warning similar to the one in Figure 3 will display for each volume before the SheepShaver window appears. If this warning appears, click the OK button (you don't have a choice). But be aware that SheepShaver is not always successful at automatically unmounting these volumes. And if SheepShaver can't unmount a volume, your Mac could hang. A safer

approach is to manually unmount HFS volumes from BeOS after you have added them to the SheepShaver Volumes list (see *Volumes Settings*, below).

The first time you run SheepShaver, a warning will appear asking you whether you want your network configuration modified to support Ethernet under SheepShaver. If you click the OK button, SheepShaver will automatically make the modification; if you click the Cancel button, SheepShaver will disable Ethernet.

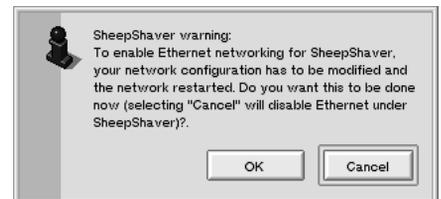


Figure 4

The first time you run SheepShaver, it will ask you if you want to modify the BeOS network configuration to support Ethernet under SheepShaver.

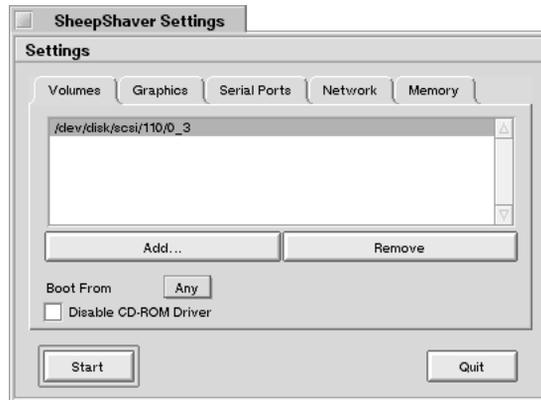
SheepShaver Settings

The SheepShaver Settings panel appears each time you launch SheepShaver. You'll probably configure most of these settings once and then leave them alone. In fact, many of them can simply be left at their default values. Before you can use SheepShaver, though, you'll have to visit the Volumes tab page. You'll probably want to tweak the settings on the Memory tab page as well.

Volumes Settings The Volumes tab lets you select which MacOS volumes you want to appear on the Mac Desktop in SheepShaver. Both HFS and HFS+ volumes are supported, as are removable volumes (Zip and Jaz

Figure 5

From the Volumes tab page in SheepShaver's Settings panel, you can select HFS and HFS+ volumes to be mounted on the Mac Desktop in SheepShaver.



cartridges, for example). Note, however, that you have to mount all volumes—including removables—*before* you launch SheepShaver. You can't mount a removable cartridge from *within* SheepShaver. This should be fixed in version 2.0.

Selecting Mac Volumes to Mount in SheepShaver When you first launch SheepShaver, the Settings panel will open with the Volumes tab page displayed. Its list of volumes will be empty. To add a volume to the list, click the Add... button. A BeOS File panel will appear, from which you can select a volume. Repeat this process for each volume that you want SheepShaver to mount. Note that before you can mount a volume in SheepShaver, you must first mount it in BeOS; Option+ click on the BeOS Desktop and choose Mount from the context menu.

Once you've added a volume to SheepShaver's list, go back and *unmount* it from the BeOS Desktop before you launch SheepShaver. It's a pain, but less of a pain than the alternative. As part of its startup process, SheepShaver tries automatically to unmount from BeOS any volumes that you've selected to appear in the MacOS. But this automatic process doesn't always succeed. If it fails, chances are good that your Mac will hang. If this occurs, the only way out is a manual restart.



While SheepShaver is running with HFS or HFS+ volumes mounted in its environment, do *not* attempt to mount those same volumes in BeOS. Doing so can cause disk corruption and unrecoverable data loss. *Never* click the "Mount all disks now" button in the Disk Mount Settings panel while SheepShaver is running!

To remove a volume from the SheepShaver volumes list, select the volume (you'll have to know its SCSI ID—the list doesn't display volume names), then click the Remove button

Selecting a SheepShaver Boot Volume The Boot From pop-up menu on this tab page lets you select a MacOS boot volume. When Any (the default value) is selected, SheepShaver will boot from the first boot volume it finds. When CD-ROM is selected, SheepShaver will attempt to boot from the disk in the CD-ROM drive.

Disabling SheepShaver's CD-ROM Driver SheepShaver includes a CD-ROM driver that should work for all CD-ROMs that shipped with MacOS systems supported by the BeOS. The SheepShaver CD-ROM driver supercedes whatever CD-ROM driver is installed in your MacOS System Folder (typically Apple CD-ROM or CD-ROM Toolkit). The Disable CD-ROM Driver checkbox on the Volumes tab page gives you a way to disable SheepShaver's CD-ROM driver. But since there's no other CD-ROM driver that works with SheepShaver, checking this box will disable your CD-ROM drive.



In most cases, you'll have no need to disable the CD-ROM driver. An exception: Disabling the driver will enable you to play audio CDs in BeOS while working in SheepShaver.

Graphics Settings SheepShaver provides two display modes: window and full-screen (see *Display Modes*, below, for more details). The Graphics settings tab page allows you to control the refresh rate of the SheepShaver window when SheepShaver is in window mode. It also allows you to turn off graphics acceleration. Other graphics-display settings, such as screen resolution, must be set from within SheepShaver itself.

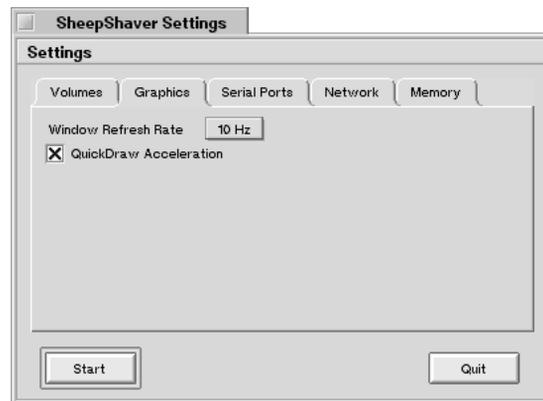


Figure 6

The SheepShaver Graphics Settings tab page provides partial control over how the MacOS environment is displayed.

Setting SheepShaver's Window Refresh Rate SheepShaver runs much more efficiently in full-screen mode. Performance and responsiveness are both far better in full-screen mode than in window mode. If, however, you decide to use it in window mode (and you have no choice if you haven't paid the shareware fee), you can change SheepShaver's refresh rate from the

Window Refresh Rate pop-up menu on the Graphics Settings tab page. But there's not much point. Although it's counterintuitive, *higher* refresh rates make SheepShaver behave more *slowly*, because it's spending its time drawing the contents of the SheepShaver window instead of running the MacOS. So stick with the default rate (which is around 5 or 6 Hz). Note that this refresh rate refers only to how frequently the MacOS environment inside the SheepShaver window is redrawn. Everything else in BeOS will continue to be refreshed at the rate set in the Screen preferences panel, which will probably be in the 65–75 Hz range.

Disabling QuickDraw Acceleration SheepShaver can take advantage of most of the BeOS graphics drivers for graphics cards supported by BeOS for PowerPC. These drivers provide QuickDraw acceleration. Accordingly, the QuickDraw Acceleration checkbox on the Graphics Settings tab page is selected by default. There's not likely to be any reason why you'd want to turn off QuickDraw acceleration, since it speeds up graphics display, but if you suspect that the driver for your graphics card may be causing a problem, you might want to try deselecting this option.

Serial Ports Settings These settings are here for people running SheepShaver on a BeBox. If you're running it on a MacOS system, you shouldn't need to change anything on this tab page.

Network Settings The Network Settings tab page presents you with the option to disable Ethernet. By default, this checkbox is unselected; that is, Ethernet is enabled. In most cases, you'll want to leave it that way. If, however, your Mac has no Ethernet capability, you'll probably want to select the Disable Ethernet option. Otherwise, every time you start up SheepShaver, you'll get a warning that there's a problem with your Ethernet connection. You'd think that software that's clever enough to allow one operating system to pretend to

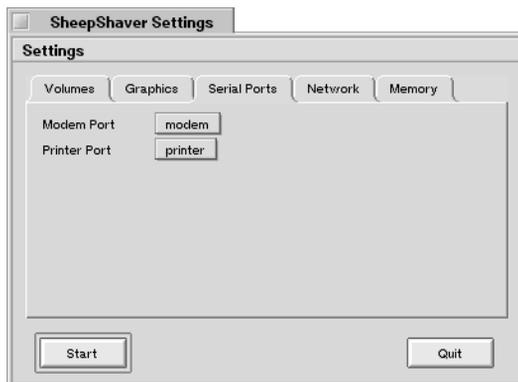


Figure 7

Users running SheepShaver on a Mac have no need to alter these serial port settings.

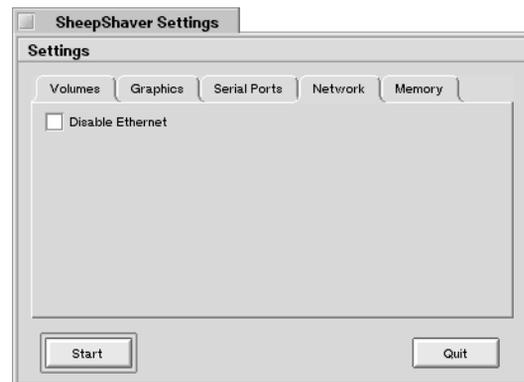


Figure 8

If your Mac has no Ethernet interface, select this checkbox. Otherwise, leave it unselected.

be another one would also be clever enough to figure out that a particular machine has no Ethernet interface, but apparently that's not the case.

Memory Settings

SheepShaver's Memory Settings tab page is where you establish how much memory will be available to the SheepShaver MacOS environment. Setting this memory allocation is simple: You just move a slider. Understanding what's going on, however, is a bit trickier.

BeOS has a fully functional demand-paged virtual memory (VM) system that is always enabled (see Chapter 9, *Preferences*, for details). It is part of the core OS. The MacOS, in contrast, has a suboptimal virtual memory system that is bolted onto the OS. The two VMs don't get along well. Hence the warning in the *Launching SheepShaver* section, above, to turn off the MacOS's virtual memory system before running SheepShaver. Although Apple generally recommends that you turn virtual memory *on* in the MacOS, you'll have nothing but trouble (and, in fact, you probably won't be able to launch SheepShaver without crashing) if you don't turn it *off*.

One of the advantages to Mac users of the MacOS's virtual memory is that when it is turned on, it enables applications to run in less memory than when it is turned off. You've probably noticed the messages in the Get Info windows of Mac applications that, when virtual memory is on, tell you how much more memory will be needed by the application if you turn virtual memory off. So when setting the RAM size for SheepShaver, remember that if you want to run several Mac applications at once, and you normally run the MacOS with virtual memory on, you're going to need more memory than you're used to needing for those apps. Fortunately, BeOS's virtual memory system is quite up to the task.

The minimum RAM size that you can set for SheepShaver is 8MB. You'll probably want significantly more, though. You wouldn't try to run a Mac with only 8MB of RAM (and virtual memory off, remember)—especially not if you wanted to use any Microsoft applications—so it doesn't make sense to run SheepShaver that way, either.

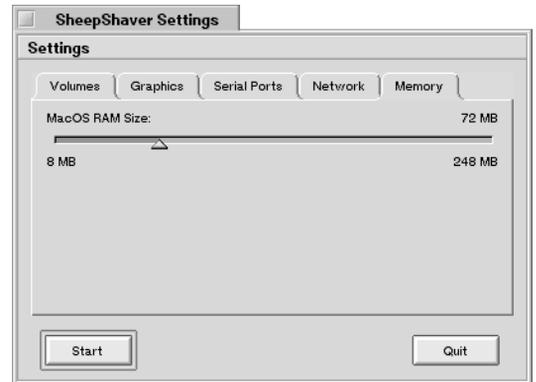


Figure 17.09

The SheepShaver Memory Settings tab page lets you establish the amount of memory that will be available in the SheepShaver Mac OS environment.

The maximum amount of memory you can allocate for SheepShaver depends on how you have BeOS virtual memory set up in the Virtual Memory preferences panel. When you run BeOS, it creates a swap file. The swap file's minimum size depends on how much physical RAM is installed in your computer, but it will be at least 48MB. If there is room available on your BeOS hard disk, you can set the swap file size to be much larger.

What shows up on the SheepShaver Memory Settings tab page as the maximum amount of RAM available for SheepShaver is equal to the size of your BeOS swap file minus 8MB. In the Figure 17. 09, for example, the BeOS swap file is 256MB, so the maximum RAM that can be allocated to SheepShaver is 248MB.

Display Modes

As stated previously, SheepShaver supports two display modes: window and full-screen. Performance in full-screen mode is far superior. In fact, when in full-screen mode, SheepShaver will respond as though you were running the MacOS without BeOS as an intermediary. Window mode is slower, jerkier, and less responsive.



Although the MacOS supports multiple monitors, SheepShaver doesn't. Having more than one monitor connected to your Mac when you run SheepShaver is not a problem, but only the monitor that contains the menu bar will be active.

Setting SheepShaver's Resolution and Refresh Rate SheepShaver supports six standard Mac screen resolutions, ranging from 640×480 to 1600×1200 pixels. At 640×480 and 800×600, both window mode and full-screen mode are available. At higher resolutions, only full-screen mode is available. In window mode, SheepShaver gives you a choice of several different refresh rates. In full-screen mode, the actual refresh rate is determined by the resolution of the workspace in which SheepShaver is running, which is set in the Screen preferences panel.

The first time you launch SheepShaver, it will come up in window mode with a resolution of 640×480 and a refresh rate of 5 Hz. Once SheepShaver is running, you can change resolutions from the resolution pop-up in the control strip or from the Monitors & Sound (or Monitors) control panel. (See *Window Mode*, below, for information on changing the window-mode refresh rate.)

Selecting any resolution that has an accompanying refresh rate of 70 Hz will put SheepShaver into full-screen mode. Selecting a resolution with a refresh rate of 5 Hz will change SheepShaver's resolution but will leave it in window

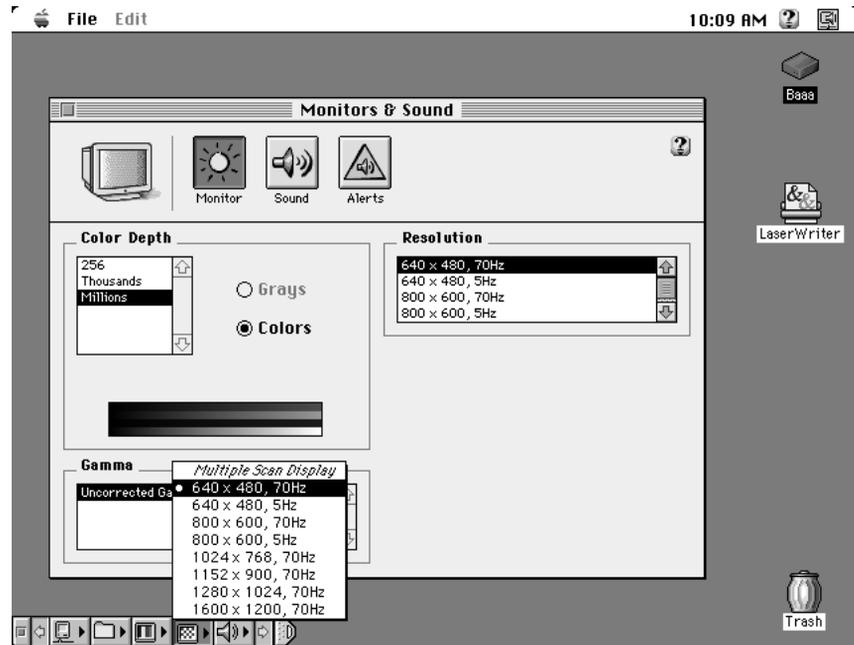


Figure 10

You can set SheepShaver's screen resolution and refresh rate from the Mac's control strip or from the Monitors & Sound control panel.

mode. Once in full-screen mode, you can change to window mode by selecting a screen resolution that is paired with a lower-than-70-Hz refresh rate.

Full-Screen Mode In full-screen mode, the SheepShaver MacOS environment takes over an entire BeOS workspace (whichever one is active when you launch SheepShaver). When you're in that workspace, it's as if you were running the MacOS directly. BeOS is nowhere in evidence.

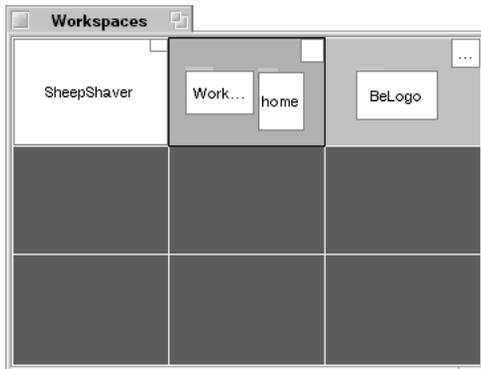


Figure 11

When SheepShaver is in full-screen mode, it takes over an entire BeOS workspace.

This, of course, raises the question of how one gets “back to BeOS” from SheepShaver. The answer is built into BeOS, which allows you to use keystroke combinations to move among workspaces. Each workspace has an internal numeric designation. The workspaces are numbered from left to right and top to bottom. Thus, in Figure 11, SheepShaver occupies workspace #1, and reading left to right in the top column, the next workspace is workspace #2; the next, workspace #3 with the Be logo window in it. In this example, if you were in SheepShaver, you could get to workspace #2 by pressing Command+F2, to

workspace #3 by pressing Command+F3, and so on. Command+F1 would bring you back to SheepShaver from any other workspace.

Alternately, you can return to BeOS by changing SheepShaver's resolution. If you select 640×480 or 800×600 with a 5 Hz refresh rate, SheepShaver will revert to window mode. You can change the refresh rate either from the Monitors & Sound control panel or from the resolution pop-up in the Control Strip.

Window Mode In window mode, the SheepShaver MacOS environment is contained within a BeOS window, as shown in Figure 2, on page 6. In most respects, SheepShaver behaves the same in window mode as it does in full-screen mode. Performance is not as good, however.



Figure 12

In window mode, you can set SheepShaver's refresh rate from the SheepShaver pop-up menu at the lower-left corner of the SheepShaver window.

Window mode provides several different screen refresh rates, ranging from 5 Hz to 60 Hz. The window-mode refresh rate can be set either in the SheepShaver Settings panel that appears when you launch SheepShaver or from within SheepShaver itself. Setting the refresh rate from the Settings panel is described above, in *Setting SheepShaver's Window Refresh Rate*. To set the refresh rate from within SheepShaver, click on the word "SheepShaver" in the lower-left corner of the SheepShaver window. From the menu that appears, select Refresh Rate; then select the desired rate from the secondary menu.

Although it might seem that higher refresh rates would result in improved responsiveness, in my experience, that wasn't the case. If you're going to run SheepShaver in window mode, you're probably best off leaving the refresh rate at the default of 5 Hz.

Using SheepShaver to Transfer Data and Files

The built-in FTP capability in BeOS provides a means of transferring files between BeOS and MacOS systems over Ethernet (see Chapter 9, *Networking*). SheepShaver opens up additional possibilities: cut, copy, and paste (for text); and Mac file sharing, for moving files from the SheepShaver MacOS environment directly to another Mac, without going through BeOS.

Cut, Copy, and Paste SheepShaver shares a clipboard with BeOS, but only for text. Other types of data are not supported. If, however, all you need to do is move some text from a SimpleText document to a StyledEdit document, or vice versa, it's a simple matter of cutting and pasting. Note that only ASCII text will be copied—typeface, style, point size, and other types of formatting information are not preserved.

File Sharing If you've used Mac file sharing before, you already know how to use it in SheepShaver. Working in SheepShaver is just like working directly on a Mac. As with a Mac, file sharing can go both ways. You can use the AppleShare client from the Chooser to log onto another Mac, or you can turn file sharing on in SheepShaver to allow other Macs to share your files.



Mac file sharing is a great solution if what you want to do is move Mac files from SheepShaver to another Mac. Unfortunately, moving files from SheepShaver to BeOS is more of a hassle. Because you can't have the same volume mounted in SheepShaver and BeOS at the same time, to move a file from a Mac volume mounted in SheepShaver to a BeOS volume, you have to quit SheepShaver, mount the Mac volume in BeOS, and then copy the file. Then you have to remember to *unmount* the Mac volume in BeOS before relaunching SheepShaver. This will be fixed in 2.0: SheepShaver will provide a BeOS icon on the Mac Desktop, making file transfer a much simpler proposition.

Floppy Disks According to SheepShaver's documentation, floppy disks are supported in the SheepShaver Mac OS environment. They do not, however, mount automatically when inserted into the Mac's floppy drive. To mount a floppy manually while in window mode, insert the disk, then select Mount Floppy from the SheepShaver menu in the lower-left corner of the SheepShaver window. To mount a floppy while in full-screen mode, insert the floppy, then press Ctrl+F1.

That's the theory. I was unable to get it to work in either mode. Other SheepShaver users have reported that floppy mounting works fine for them. If you're one of the unlucky ones, have faith: The problem should be resolved in SheepShaver 2.0.

Quitting SheepShaver

Unlike most BeOS applications, SheepShaver has no Quit menu item. In fact, SheepShaver has no BeOS menu at all. But it does have Mac menus. To quit SheepShaver, select Shut Down from the Mac's Special menu.

And, in case you're curious, selecting Restart from the Mac's Special menu while in SheepShaver will restart the SheepShaver MacOS environment, leaving the rest of BeOS as it was.



The SheepShaver MacOS environment is active only when SheepShaver's window (or workspace) is active. If you click outside the SheepShaver window (or, if SheepShaver is in full-screen mode, switch to a different workspace), all activity in the MacOS environment halts. When you reactivate SheepShaver, it resumes where it left off.

Known Problems

The following list of known incompatibilities and bugs in SheepShaver 1.1 (as noted in the SheepShaver documentation) is up to date as of press time. For the latest information, check out the SheepShaver Web site (www.sheepshaver.com).

- MacOS programs that access hardware directly are not supported.
- RAM Doubler, Speed Doubler, and 68K emulators such as PowerFPU are not supported.
- The QuickTime 2.5 Cinepak codec causes SheepShaver to crash.
- Sound doesn't work on Tanzania-based systems.
- Random crashes have been reported on some systems when using the serial ports.

Troubleshooting

SheepShaver can be very finicky. Your best bet is to start clean and be conservative. Don't assume that you're going to be able to use your existing MacOS system software setup without a hitch. More likely, you're going to have to create a custom System Folder for SheepShaver use.

When SheepShaver crashes, it's not pretty. In my experience, a SheepShaver crash usually brings down BeOS with it. (I know, BeOS memory protection is supposed to prevent such events from happening—but they happen anyway.) To get back to where you were in SheepShaver, you have to reboot *three* OSs: the MacOS, BeOS, and the SheepShaver MacOS environment. So you really want to do all you can to avoid this unpleasantness. You'll find an extension-management tool like Conflict Catcher or Now Startup Manager invaluable if you're having problems.

Here are some troubleshooting tactics you can try (in addition to the recommendations made in *Launching SheepShaver*, above):

- Use a different version of the MacOS. I tried 8.1, 7.6.1, and 7.6, all of which crashed repeatedly. I finally settled on 7.5.5 (the OS version that shipped with my Umax clone), which is pretty reliable. This isn't to say that you'll necessarily need to downgrade to 7.5.5; just that you may need to experiment with an older version of the MacOS to create a stable environment.
- Do a clean install of the OS. The first time you run SheepShaver, hold down the Shift key after you get past the SheepShaver Settings panel to disable all MacOS extensions. Once you've succeeded in getting to the

Mac Desktop, add your favorite extensions—one at a time if you’re timid; all at once if you’re the adventurous type.

- Reinstall SheepShaver. When you get a message asking if you want to replace existing files, say yes. I found that reinstalling SheepShaver once usually didn’t do the trick; two times generally seemed to work, though.
- Delete the `/boot/home/config/settings/NVRAMfile`.
- If you’re having a networking problem, make sure that Ethernet is configured in the BeOS Network preferences panel. Also, try disabling AppleTalk in the Network preferences panel.
- If you’re using TCP/IP from within SheepShaver, make sure you have different TCP/IP addresses set for the MacOS and BeOS.

Like Russian Dolls...

Soon after the release of SheepShaver, the satirical pseudo-news site BeDope (www.bedope.com) ran a contest to see just how many emulated or runtime environments could be hosted within one another. The winner of the contest would be the person who could create the deepest “hall of mirrors”—an OS within an OS within an OS, nested one inside the next like Russian dolls. To kick things off, BeDope ran a screenshot of BeOS running MacOS in a window via SheepShaver. In turn, MacOS was running a Windows 3.1 emulator, which in turn was running an Apple IIe emulator (which, by the way, was playing a game of *DigDug*). Since the game itself didn’t count, that was four. Who could do five? The prize offered was a Be T-shirt. Anyone who could stack up six nested systems would take home a Be polo shirt, “proven to make you look ten pounds lighter, ward off mosquitoes, code faster, and attract groupies in droves. Your mileage may vary.” To this day, nobody has stepped forward to claim that prize (the contest is still open!), but Be’s own Mikol Ryan did manage to get five nested systems going at once.



Figure 13

Emulators within emulators, nested like Russian dolls. Pictured: BeOS hosting MacOS, hosting Windows 3.1, hosting DOS, hosting CP/M (which is an antique precursor to DOS).

Emulators

The majority of emulators available for BeOS are true emulators, which make it possible to run operating systems and games designed for a wide variety of hardware platforms from your PowerPC or x86 BeOS machine.

Bochs (x86)

www.bochs.com

With the early success of SheepShaver enabling Mac users to boot MacOS from within BeOS, one might expect there to be a similar option to let BeOS users run Microsoft Windows inside a BeOS window. While nothing like SheepShaver exists for running Windows in BeOS just yet, it *is* possible to install an emulator called Bochs that re-creates enough of the x86 CPU, chipset, and BIOS information to run DOS, Windows 3.1, Windows 95/98, Minix, and potentially other x86-based operating systems from within BeOS (Linux is not currently on this list, though it should be possible to do).

Bochs runs on both PowerPC and x86 versions of BeOS. In fact, Bochs runs on many platforms—it was running on Sun SparcStations and other Unix hardware before it was ported to BeOS. But this raises an interesting question: A few pages ago I told you that true emulation was not necessary if both the host OS and the alien OS could use the same hardware. So why should it be necessary to emulate x86 hardware if you're already running on x86 hardware? It's a good question, but it's not the right question. Bochs is an x86 emulator, pure and simple. It's not an application meant specifically to let you run Windows programs from within BeOS. You may be able to do that with it, but that's not its sole purpose in life.



Figure 14

Bochs for BeOS booting Windows 95 from its disk image for the first time.

So what does it take to get Bochs up and running? As of this writing, quite a bit. In fact, we were not able to successfully get Bochs up and running at all during testing with early versions. However, Bochs, Inc. issued a press release during Comdex '98 stating their commitment to make Bochs for BeOS much easier to install and configure. Additionally, the company has promised great strides in performance for upcoming versions; keep an eye on www.bochs.com for updates. The commercial potential for a user-friendly Windows runtime environment or emulator for BeOS is quite high, so don't be surprised if some kind of "WinShaver" application appears in the future (either from Bochs or from another company).



Most of the emulators described in the rest of this chapter require a ROM image copied from the original hardware they're emulating. Instructions for obtaining this ROM image are different for each emulator, but generally involve using a special utility provided with the emulator itself. Read the included documentation carefully, and don't use ROMs from machines you don't legally own.

UAE (AmigaOS)

As those who have been following Be for a while know, the BeOS community includes more than its fair share of Amiga enthusiasts. BeOS has, in fact, been called "the new Amiga" by some—and for good reason. The BeOS focus on digital media, excellent design, and cutting-edge technologies does display a certain spiritual continuity with the Amiga's legacy. In the days of the BeBox, there were even more parallels to draw, since the Commodore Amiga was also a proprietary box with a digital-content-creation emphasis. For those still in love with their Amigas, or simply addicted to Amiga programs that lack a BeOS parallel, the Un*x Amiga Emulator (UAE) is capable of emulating the

hardware in Amiga models 500, 1000, and 2000.

UAE is available on BeWare, but before you start thinking that this is your big chance to see what it's like to run AmigaOS, you'll need to get legit. As noted earlier, UAE emulates Amiga hardware—it doesn't give you the operating



Figure 15

The Un*x Amiga Emulator running AmigaOS in a BeOS window.

system code to boot on that emulated hardware. In order to boot AmigaOS from within BeOS, you'll need to copy your Amiga's boot ROM over to your BeOS machine first. Don't have a spare Amiga sitting around? You'll have to either buy one or purchase the boot ROM on CD. More information is available at www.freiburg.linux.de/~uae/.

Basilisk (Mac Classic)

Much like UAE (which is by the same author, Christian Bauer), Basilisk requires a ROM image from the hardware it's meant to emulate, which in this case is the 512K Macintosh Classic. Basilisk is free of charge and can be downloaded from BeWare. Although it includes source code that you're welcome to explore, tweak, and recompile, it does come with a ready-to-run binary. All you have to do is place your ROM image in the same directory as Basilisk, make sure it's named `ClassicROM` and launch. If Basilisk finds the classic Mac ROM, MacOS will boot in the Basilisk window. It should be possible to boot Systems 5.0, 6.0.3, 6.0.8, 7.1, and 7.5. If you've got 266 MHz of total CPU, performance should be approximately equivalent to a real Mac Classic.

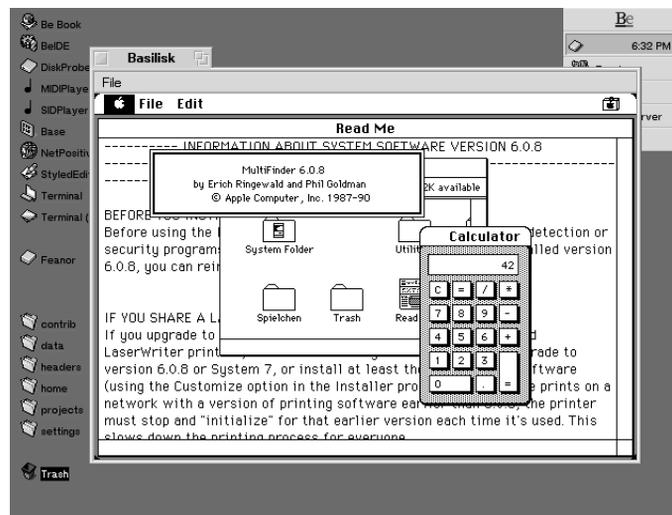


Figure 16
An early version of MacOS running in a Basilisk window, which emulates the hardware of the Mac Classic.

Open Bernie (Apple II+, IIe, IIgs)

If you've got an old Apple II+, IIe, or IIgs gathering dust in the garage, go grab its ROM, download a copy of Open Bernie, and fire up a pre-GUI, pre-MacOS, bona fide Apple computing environment like the one you used in high school all those years ago (OK, I've just dated myself). Bernie is known for its great performance as an emulator, can be run in window or full-screen modes, and supports stereo sound, virtual disk drives, and mouse/keyboard emulation.

BeBeeb (Acorn BBC Micro)

The history of computing is strewn with stories that seem odd given today's computing climate. In the early 80s, the British Broadcasting Corporation launched "The BBC Computer Literacy Project." They had a lot of ideas about what they wanted computers to do in the project (including artificial intelligence, hardware control, and graphics/sound), and since nothing on the computing landscape at the time quite fit the bill, they went shopping for a machine. To make a long story short, a company called Acorn created the BBC Micro for this project and ended up selling nearly a million units. These days, Acorn makes the StrongARM chip found in some PDAs, and has built a variety of "RISCPC" systems as well. But if you've got a BBC Micro you can copy the ROMs from, you'll be able to emulate its hardware in BeOS with BeBeeb, which is available from BeWare. Learn more about the history of the BBC Micro at www.nvg.ntnu.no/bbc/.

BeZX (Sinclair Spectrum)

You may never have heard of the classic Sinclair Spectrum, but believe it or not, there are still people walking this earth who swear that the Spectrum was the coolest computer ever made. Its creator, Clive Sinclair, has been one of the most prolific inventors of the century, and holds patents on everything from amplifiers to multimeters to pocket TVs to electric vehicles. Amazingly, some of his early computers sold for less than a couple hundred bucks. He sold millions of them. If you have a Spectrum and want to emulate it in BeOS, download BeZX from BeWare and follow the instructions in the README. For more information see Planet Sinclair at www.nvg.uni.t.no/spectrum/.



Figure 17
Emulating a Sinclair
Spectrum game
under BeOS.

Frodo (Commodore 64)

Raise your hand if you typed your first-ever line of BASIC into a Commodore 64. Millions of people did just that in the early 80s, and the C-64 is still a favorite among computer history buffs. The C-64 shipped with three audio channels, had great graphics (for the time), and ended up being a very popular gaming machine. Frodo, a C-64 emulator available for most operating systems, is free and can be downloaded from BeWare. Unlike the other emulators listed here, Frodo does not require you to obtain a Commodore boot ROM; just launch the emulator and you're in the operating system. Also included in the download are a number of test files and games you can run immediately, though you'll need to read the included documentation carefully if you're not already familiar with the Commodore command line.

Game Emulators

Unlike standard computer hardware, console games typically don't have disk drives, keyboards, or other devices necessary to Hoover out ROM images. The only available tool is the console's backup mechanism, which is intended for saving copies of your cartridges to hard disk in case the original cartridge is lost or damaged. Read the licensing agreement that came with your cartridge system to make sure you're allowed to do this.



Figure 18

Return to the grand old days of *Space Invaders* and *Ms. Pac Man* by playing classic arcade games in a BeOS window.

BeMAME (Multiple Arcade Machine Emulator)

If you spent half your junior high lunch money dropping quarter after quarter into ravenous *Space Invaders* or *Ms. Pac-Man* machines at the local 7-11, you may be unimpressed with the current state of arcade gaming. 24-bit color, thundering sound, 3D monsters, 17 joysticks, and two foot pedals ... who needs all of that nonsense? For some of us, the first arcade games are still the best. Fortunately, you can still enjoy many of those old arcade-style games by downloading a copy of BeMAME, which emulates as much of that early arcade hardware as possible. You'll have to do your own research to find playable game ROMs, but the emulator itself is absolutely free, and can be found on BeWare (PowerPC only at this writing, though an x86 version is expected soon). If you prefer your games

with an accompanying soundtrack, remember to download sound samples as well (they're usually available wherever ROMs are distributed). To get maximum use out of BeMAME, you'll probably want to print out the list of hotkeys, which you'll find in the README. I'll give you a hint: Press 3 to emulate the act of dropping a quarter into the slot.

Snes9x (Super Nintendo Entertainment System)

At this writing, the only game cartridge emulation system available for BeOS was Snes9x, a port of the Super Nintendo Entertainment System.

Unfortunately, this emulator is rather slow due to the fact that five separate chips have to be emulated in software. In BeOS, the problem is compounded by the single-threaded nature of the Super Nintendo programs. The documentation accompanying Snes9x includes a number of suggestions for ways to speed up performance, such as eliminating translucency and sound. However, more recent versions of the Snes9x emulator are reported to be up to 200% faster than were the early ports. Learn more about SNES emulation at www.snes9x.com



A PowerPC-only port of a Nintendo GameBoy emulator is also available on BeWare. However, it's quite outdated, and it appears that the original author is disallowing further ports due to abuse of his source code.

Client/Server Environments

While they don't fit into either the "true emulation" or "runtime environment" categories, there are a couple of other ways to run (or to appear to run) software from other operating systems under BeOS. Learn about the X-Windows server in Chapter 15, *Other Goodies*, and the Virtual Network Client in Chapter 8, *Networking*.

Chapter Summary

- There are two major types of emulation: "true emulation," where some kind of hardware is faked in the host system to satisfy the requirements of another operating system, and "runtime environments," where hardware does not need to be emulated, but a negotiation layer is needed between the host system and the alien system.
- Emulators emulate hardware, not software. After downloading an emulator, you'll still need to obtain a ROM image or other boot code to run inside the emulator, in addition to any actual application software you

want to run. You *must* be the legitimate owner of the ROMs you run within your emulators. Think of an emulator as if it were a real machine—purchasing it does not automatically get you the operating system or the applications to run on it.

- SheepShaver, a runtime environment that lets you boot MacOS inside a BeOS window, is currently the most sophisticated and user-friendly runtime environment available for BeOS.
- There is currently no available BeOS runtime environment for Windows or Linux. There is, however, an x86 emulator called Bochs, which was in an early stage of development at this writing but may become a viable way to run Windows software from within BeOS in the future. The market for a user-friendly Windows runtime environment is ripe.
- Emulators are currently available for quite a few historical hardware platforms, in addition to a couple of console game platforms.