

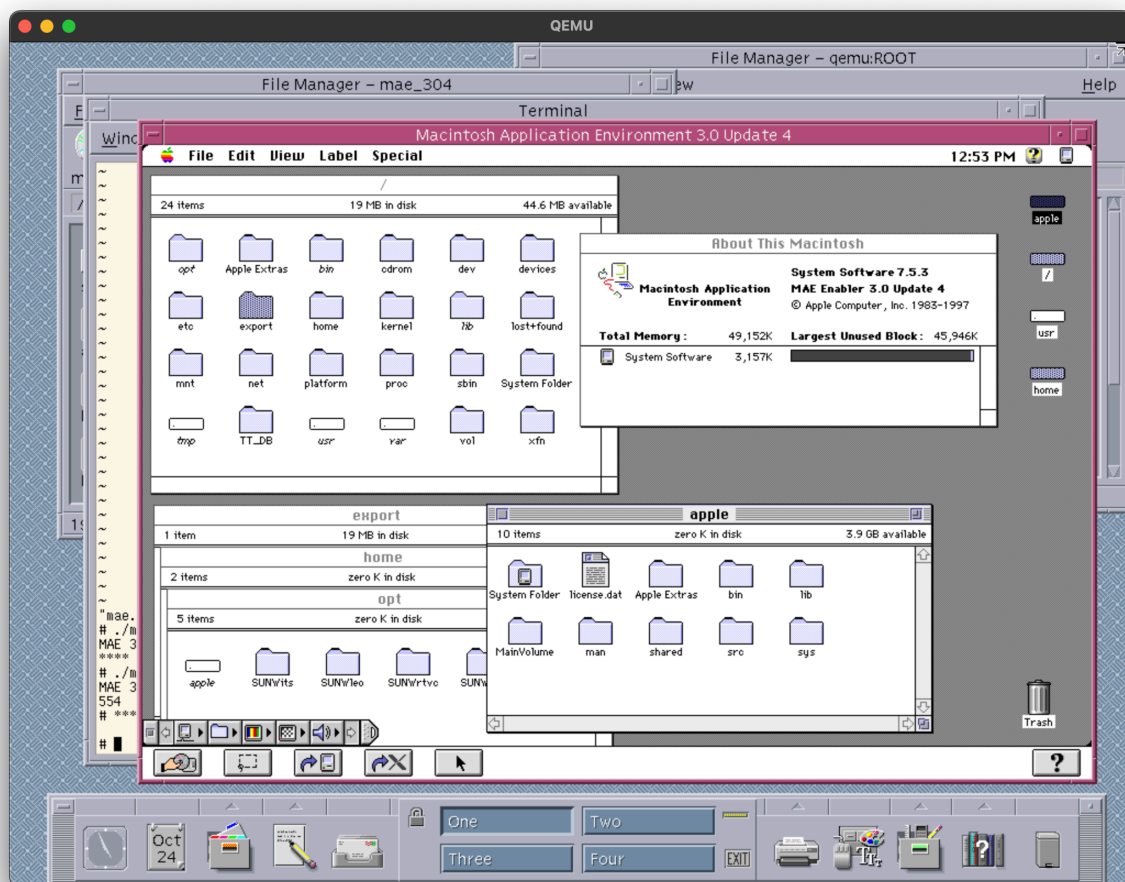
Let's get Macintosh Application Environment running in QEMU !

Let's get Macintosh Application Environment running on Solaris,
in QEMU, with AppleTalk !

Setup Guide

by Ken McLeod (@_the_cloud@mastodon.social)

v1.0, 2025-10-25



Introduction

The Macintosh Application Environment, typically abbreviated as MAE, was one of Apple's most niche software products. It did not even run on any computer that Apple made! This was strictly for Sun's SPARC-based machines running Solaris, or Hewlett Packard's PA-RISC-based systems running HP-UX, both of which were considered Serious Unix Workstations in the 1990s. The purpose of

Let's get Macintosh Application Environment running in QEMU !

MAE was to emulate a 68K Macintosh in a resizable window on those machines, allowing Mac applications to run on Unix. Data could be easily copied and pasted between the host workstation and the Mac environment. What's more, the Macintosh Finder provided seamless access to mounted volumes on the host machine, as if they were Macintosh disks. This blending of Unix and Mac environments was reminiscent of A/UX, Apple's first version of Unix introduced in 1988. It helped that many of the same people who worked on A/UX were involved in the development of MAE. Unfortunately, Apple was in dire financial straits at the time, and MAE was one of many products that were axed by 1998 as the company sought to narrow its focus.

There have been several articles written up by folks who have gotten MAE running on actual PA-RISC retro hardware [1],[2]. The second one in particular is an incredibly deep dive into MAE, and worth the long read. And at least one other person has gotten MAE running in a virtual machine under QEMU [3]. But we want to go one step further, and get AppleTalk working so that MAE can see zones and connect to other systems over the worldwide AppleTalk network that is #GlobalTalk [4].

Solaris 2.6 is a good choice to host MAE since that release came out not long after MAE 3.0, and it's known to work with QEMU according to their testing [5]. Networking support for STREAMS is already built into Solaris. So, let's get the Macintosh Application Environment running on Solaris, in QEMU, with AppleTalk connectivity!

Prerequisites

We will need the following:

- an installation of QEMU, specifically the `qemu-system-sparc` and `qemu-img` programs
- the Solaris 2.6 SPARC installer disk image (`solaris_2.6_sparc.iso`) [6]
- the MAE software (`MAE-3.0.4.tar.gz`) [7]
- software to create an ISO disk image that can be read by Solaris

QEMU can be installed with MacPorts or Homebrew; this guide assumes you have already done that or can search for further instructions. The specific version of QEMU you use is not important, but should be recent (9.0.0 or later is recommended.)

Let's get Macintosh Application Environment running in QEMU !

The `hdiutil` command is built into macOS and we will use that to create an ISO image from the MAE software; on Windows, a program called *InfraRecorder* can be used.

1. Build a virtual SPARCstation 5

The first step is to prepare our virtual hard disk. Every hard disk that Solaris will use has to be tagged with a special format and disk label before it is recognized. Using instructions also found here ([\[8\]](#),[\[9\]](#)), we'll first create a sparse image file for this disk with plenty of room to grow if needed:

```
gemu-img create -f qcow2 Solaris26.qcow2 36G
```

Then we initially run the emulator with both this hard disk and the installation CD attached, so we can format and label the new disk. (Note: this command is all one line.)

```
gemu-system-sparc -M SS-5 -m 128 -drive
file=Solaris26.qcow2,bus=0,unit=0,media=disk -drive
file=solaris_2.6_sparc.iso,bus=0,unit=2,media=cdrom,readonly=
on
```

This will boot into an OpenBIOS '>' prompt. Since there's no valid hard disk yet, we have to tell it to boot from the CD in verbose single user mode by typing:

```
boot cdrom:d -vs
```



Let's get Macintosh Application Environment running in QEMU !

At the next prompt, enter the drvconfig command:

```
drvconfig
```

At the next prompt, enter the disks command:

```
disks
```

Start the formatting procedure by entering the format command:

```
format
```



```
QEMU
cpu0: FMI,MB86904 (mid 0 impl 0x0 ver 0x4 clock 170 MHz)
mem = 131072K (0x8000000)
avail mem = 10964736
Ethernet address = 52:54:0:12:34:56
root nexus = SUNW,SPARCStation-5
iommu0 at root: obio 0x10000000
sbu0 at iommu0: obio 0x10001000
espdma0 at sbu0: SBus slot 5 0x8400000
esp0: esp-options=0x46
esp0 at espdma0: SBus slot 5 0x8800000 sparc ipl 4
esp0 is /iommu0,10000000/sbu0,10001000/espdma05,8400000/esp05,8800000
sd0 at esp0: target 0 lun 0
sd0 is /iommu0,10000000/sbu0,10001000/espdma05,8400000/esp05,8800000/sd0,0
WARNING: /iommu0,10000000/sbu0,10001000/espdma05,8400000/esp05,8800000/sd0,0 (sd0):
        corrupt label - wrong magic number
Vendor 'QEMU', product 'QEMU', 75497472 512 byte blocks
sd2 at esp0: target 2 lun 0
sd2 is /iommu0,10000000/sbu0,10001000/espdma05,8400000/esp05,8800000/sd2,0
root on /iommu0,10000000/sbu0,10001000/espdma05,8400000/esp05,8800000/sd2,0:b fstype ufs
obio0 at root
zs0 at obio0: obio 0x1000000 sparc ipl 12
zs0 is /obio/zs0,100000
zs1 at obio0: obio 0x0 sparc ipl 12
zs1 is /obio/zs0,0
SUNW,tcx0 at sbu0: SBus slot 3 0x8000000 and SBus slot 3 0x2000000 and SBus slot 3 0x4000000 and SBus slot 3 0x6000000 and SBus slot 3 0xa000000 and SBus slot 3 0xc000000 and SBus slot 3 0xe000000 and SBus slot 3 0x701000 and SBus slot 3 0x200000 and SBus slot 3 0x200000 SBus level 5 sparc ipl 9
SUNW,tcx0 is /iommu0,10000000/sbu0,10001000/SUNW,tcx03,300000
tcx0: revision 0, screen 1024x768
Configuring devices...
fdintr: nobody sleeping (c0 0 0)
fdintr: nobody sleeping (c0 0 0)
ledma0 at sbu0: SBus slot 5 0x8400010
le0 at ledma0: SBus slot 5 0x8c00000 sparc ipl 6
le0 is /iommu0,10000000/sbu0,10001000/ledma05,8400010/le05,8c00000
sbusmem0 is /iommu0,10000000/sbu0,10001000/sbusmem0,0
sbusmem1 at sbu0: SBus slot 1 0x0
sbusmem1 is /iommu0,10000000/sbu0,10001000/sbusmem01,0
sbusmem2 at sbu0: SBus slot 2 0x0
sbusmem2 is /iommu0,10000000/sbu0,10001000/sbusmem02,0
sbusmem3 at sbu0: SBus slot 3 0x0
sbusmem3 is /iommu0,10000000/sbu0,10001000/sbusmem03,0
sbusmem4 at sbu0: SBus slot 4 0x0
sbusmem4 is /iommu0,10000000/sbu0,10001000/sbusmem04,0
sbusmem5 at sbu0: SBus slot 5 0x0
sbusmem5 is /iommu0,10000000/sbu0,10001000/sbusmem05,0
pseudo-device: winlock0
winlock0 is /pseudo/winlock0
pseudo-device: l1c10
l1c10 is /pseudo/l1c10
|
INIT: SINGLE USER MODE
# drvconfig
fdintr: nobody sleeping (c0 0 0)
fdintr: nobody sleeping (c0 0 0)
pseudo-device: lockstat0
lockstat0 is /pseudo/lockstat0
SUNW,CS42310 at sbu0: SBus slot 4 0xc000000 SBus level 5 sparc ipl 9
SUNW,CS42310 is /iommu0,10000000/sbu0,10001000/SUNW,CS423104,c000000
drvconfig: /etc/path_to_inst.69: Read-only file system
drvconfig: Warning: Failed to update '/etc/path_to_inst'
# disks
# format
Searching for disks... WARNING: /iommu0,10000000/sbu0,10001000/espdma05,8400000/esp05,8800000/sd0,0 (sd0):
        corrupt label - wrong magic number
Vendor 'QEMU', product 'QEMU', 75497472 512 byte blocks
done

AVAILABLE DISK SELECTIONS:
  0. c0t0d0 (drive type unknown)
    /iommu0,10000000/sbu0,10001000/espdma05,8400000/esp05,8800000/sd0,0
Specify disk (enter its number):
```

There should only be one hard drive to format, so enter its number (0). The next steps will ask for the disk parameters. We will give our disk the following parameters, and just hit return to accept defaults for all other parameters.

Let's get Macintosh Application Environment running in QEMU !

Disk type: 16

Number of data cylinders: 24620

Number of heads: 27

Number of data sectors/track: 107

Disk type name: Qemu36GB



```
# disks
# format
Searching for disks...WARNING: /ionmu00,10000000/sbus00,10001000/espdma05,84000000/esp05,88000000/sd00,0 (sd0):
corrupt label - wrong magic number
Vendor 'QEMU', product 'QEMU', 75497472 512 byte blocks
done

AVAILABLE DISK SELECTIONS:
 0. c0t0d0 (drive type unknown)
   /ionmu00,10000000/sbus00,10001000/espdma05,84000000/esp05,88000000/sd00,0
Specify disk (enter its number): 0

AVAILABLE DRIVE TYPES:
 0. Auto configure
 1. Quantum ProDrive 80S
 2. Quantum ProDrive 105S
 3. CDC Wren IU 94171-344
 4. SUN0104
 5. SUN0207
 6. SUN0327
 7. SUN0340
 8. SUN0424
 9. SUN0535
10. SUN0669
11. SUN1.0C
12. SUN1.0S
13. SUN1.3G
14. SUN2.1G
15. SUN2.9C
16. other
Specify disk type (enter its number): 16
Enter number of data cylinders: 24620
Enter number of alternate cylinders[21]:
Enter number of physical cylinders[24622]:
Enter number of heads: 27
Enter physical number of heads[default]:
Enter number of data sectors/track: 107
Enter number of physical sectors/track[default]:
Enter rpm of drive[3600]:
Enter format time[default]:
Enter cylinder skew[default]:
Enter track skew[default]:
Enter tracks per zone[default]:
Enter alternate tracks[default]:
Enter alternate sectors[default]:
Enter cache control[default]:
Enter prefetch threshold[default]:
Enter minimum prefetch[default]:
Enter maximum prefetch[default]:
Enter disk type name (remember quotes): Qemu36GB
selecting c0t0d0
[disk formatted]

FORMAT MENU:
disk      - select a disk
type      - select (define) a disk type
partition - select (define) a partition table
current   - describe the current disk
format    - format and analyze the disk
repair    - repair a defective sector
label     - write label to the disk
analyze   - surface analysis
defect    - defect list management
backup    - search for backup labels
verify    - read and display labels
save      - save new disk/partition definitions
inquiry   - show vendor, product and revision
volname   - set 8-character volume name
!cmd      - execute <cmd>, then return
quit      -

Format> █
```

At the format menu, type the label command to label the disk:

label

Note this produces a warning about a corrupt label the first time. Type label again, which should succeed the second time.

label

Let's get Macintosh Application Environment running in QEMU !

```
QEMU
Specify disk (enter its number): 0

AVAILABLE DRIVE TYPES:
 0. Auto configure
 1. Quantum ProDrive 803
 2. Quantum ProDrive 105S
 3. CDC Wren IU 94171-344
 4. SUN0104
 5. SUN0207
 6. SUN0327
 7. SUN0340
 8. SUN0424
 9. SUN0535
10. SUN0669
11. SUN1.0C
12. SUN1.0S
13. SUN1.3G
14. SUN2.1C
15. SUN2.9C
16. other
Specify disk type (enter its number): 16
Enter number of data cylinders: 24620
Enter number of alternate cylinders[2]:
Enter number of physical cylinders[24622]:
Enter number of heads: 27
Enter physical number of heads[default]:
Enter number of data sectors/track: 107
Enter number of physical sectors/track[default]:
Enter rpm of drive[3600]:
Enter format time[default]:
Enter cylinder skew[default]:
Enter track skew[default]:
Enter tracks per zone[default]:
Enter alternate tracks[default]:
Enter alternate sectors[default]:
Enter cache control[default]:
Enter prefetch threshold[default]:
Enter minimum prefetch[default]:
Enter maximum prefetch[default]:
Enter disk type name (remember quotes): Qemu36GB
selecting c0t0d0
[ disk formatted ]

FORMAT MENU:
 disk          - select a disk
  type         - select (define) a disk type
 partition    - select (define) a partition table
 current      - describe the current disk
 format       - format and analyze the disk
 repair       - repair a defective sector
 label        - write label to the disk
 analyze      - surface analysis
 defect       - defect list management
 backup       - search for backup labels
 verify       - read and display labels
 save         - save new disk/partition definitions
 inquiry      - show vendor, product and revision
 volname      - set 8-character volume name
 !cmd         - execute <cmd>, then return
Format> label
Ready to label disk, continue? y
WARNING: /iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,8800000/sd@0,0 (sd0):
corrupt label - wrong magic number

Vendor 'QEMU', product 'QEMU', 75497472 512 byte blocks
(Qemu36GB cyl 24620 alt 2 hd 27 sec 107)
Format> label
Ready to label disk, continue? y
Format> quit
# reboot
```

Now that the drive has been formatted and labeled, type `quit` to exit the format program, and then `reboot` to proceed with installation.

On reboot, we should be back at the OpenBIOS '>' prompt. We'll boot from the CD again, but this time in normal mode:

```
boot cdrom:d
```

And after a bit, we should see the Solaris installer's welcome screen. At this point, you can use the excellent instructions found here [\[10\]](#) to go through each installer pane and set up your Solaris system. **(A future version of this guide may incorporate this step-by-step information directly.)** Pay close attention to the manual partitioning step, which creates separate partitions for `/`, `/var`, `/opt`, `swap`, `/usr`, and `/usr/local`, using the size values that he suggests.

Once the virtual Solaris machine is up and running, we are ready to start the MAE installation process.

Let's get Macintosh Application Environment running in QEMU !

2. Create a MAE installer image

On your modern computer, extract the `MAE-3.0.4.tar.gz` file. You can just double-click the file on macOS, or you can use this command line:

```
tar xvf MAE-3.0.4.tar.gz
```

Inside the `MAE-3.0.4` directory, there are several other tar files to be extracted:

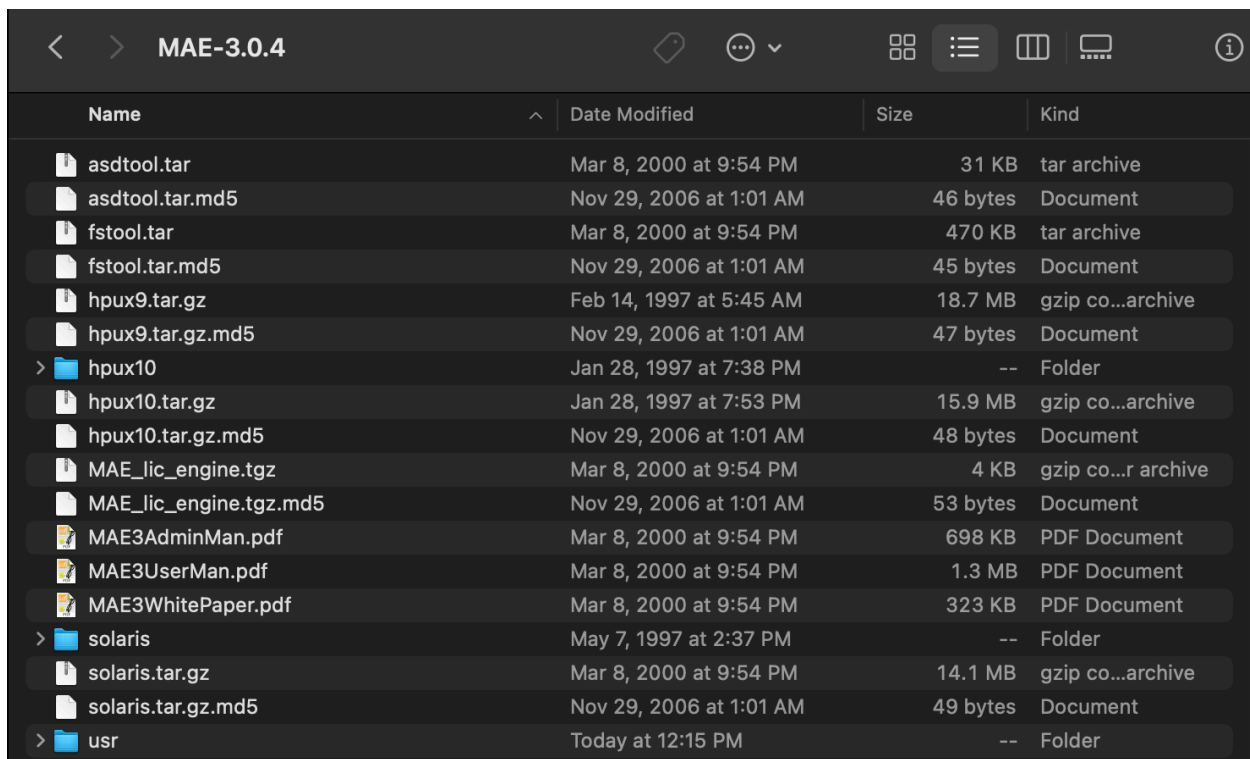
```
cd MAE-3.0.4
```

```
tar xzf MAE_lic_engine.tgz
```

```
tar xzf hpux10.tar.gz
```

```
tar xzf solaris.tar.gz
```

You should now have a directory containing the following items:



Name	Date Modified	Size	Kind
asdtool.tar	Mar 8, 2000 at 9:54 PM	31 KB	tar archive
asdtool.tar.md5	Nov 29, 2006 at 1:01 AM	46 bytes	Document
fstool.tar	Mar 8, 2000 at 9:54 PM	470 KB	tar archive
fstool.tar.md5	Nov 29, 2006 at 1:01 AM	45 bytes	Document
hpux9.tar.gz	Feb 14, 1997 at 5:45 AM	18.7 MB	gzip co...rchive
hpux9.tar.gz.md5	Nov 29, 2006 at 1:01 AM	47 bytes	Document
> hpux10	Jan 28, 1997 at 7:38 PM	--	Folder
hpux10.tar.gz	Jan 28, 1997 at 7:53 PM	15.9 MB	gzip co...rchive
hpux10.tar.gz.md5	Nov 29, 2006 at 1:01 AM	48 bytes	Document
MAE_lic_engine.tgz	Mar 8, 2000 at 9:54 PM	4 KB	gzip co...r archive
MAE_lic_engine.tgz.md5	Nov 29, 2006 at 1:01 AM	53 bytes	Document
MAE3AdminMan.pdf	Mar 8, 2000 at 9:54 PM	698 KB	PDF Document
MAE3UserMan.pdf	Mar 8, 2000 at 9:54 PM	1.3 MB	PDF Document
MAE3WhitePaper.pdf	Mar 8, 2000 at 9:54 PM	323 KB	PDF Document
> solaris	May 7, 1997 at 2:37 PM	--	Folder
solaris.tar.gz	Mar 8, 2000 at 9:54 PM	14.1 MB	gzip co...rchive
solaris.tar.gz.md5	Nov 29, 2006 at 1:01 AM	49 bytes	Document
> usr	Today at 12:15 PM	--	Folder

Now we'll turn this directory into an ISO format CD image that can be mounted in our Solaris machine. To do that, we'll use the macOS `hdiutil` command.

Replace the full pathnames used in this example with the actual path to the `MAE-3.0.4` directory on your machine.

Let's get Macintosh Application Environment running in QEMU !

```
hdiutil create -srcfolder /Volumes/Data/qemu/MAE-3.0.4
-format UDTO -fs UDF /Volumes/Data/qemu/mae304.cdr
```

The above command takes the contents of a folder (the `-srcfolder` parameter) and creates a CD image with the Mac's "Universal Disk Format" file system, writing the output to the `.cdr` file you specify.

Now we need to convert this UDF CD image to ISO format, so it can be read by Solaris. We must add a Joliet file system, so it can have long filenames up to 32 characters.

```
hdiutil makehybrid -iso -joliet -o /Volumes/Data/qemu/mae304.iso /Volumes/Data/qemu/mae304.cdr
```

We don't need the temporary .cdr file anymore, so let's delete it now:

```
rm mae304.cdr
```

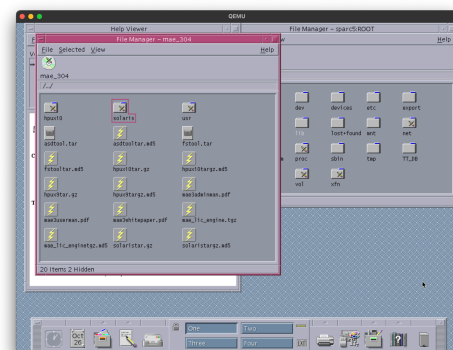
3. Mount the MAE installer image

Since we're finished installing Solaris from CD and we want to install MAE now, we'll modify our QEMU command to use the ISO image we just created. (Note: this command is all one line.)

```
qemu-system-sparc -M SS-5 -m 128 -drive
file=Solaris26.qcow2,bus=0,unit=0,media=disk -drive
file=mae304.iso,bus=0,unit=2,media=cdrom,readonly=on
```

After running this command and logging into Solaris as the root user, we can see a File Manager window open for the mae_304 CD-ROM. That's as expected. But wait a minute, something is wrong here. The window is showing all-lowercase filenames! That will cause the MAE installer to be unable to find any of its packages, because its installation scripts are case-sensitive and expect uppercase names. So we'll have to unmount and re-mount the CD using the Solaris Terminal app.

File Manager window for the MAE installation CD-ROM, with all-lowercase filenames! Oops.



Let's get Macintosh Application Environment running in QEMU !

Open the Terminal and type the `mount` command with no arguments. This will output a list of all mounted filesystems and their mountpoints. We are interested in the `cdrom` entry, which looks like this:

```
/cdrom/mae_304 on /vol/dev/dsk/c0t2d0/mae_304
```

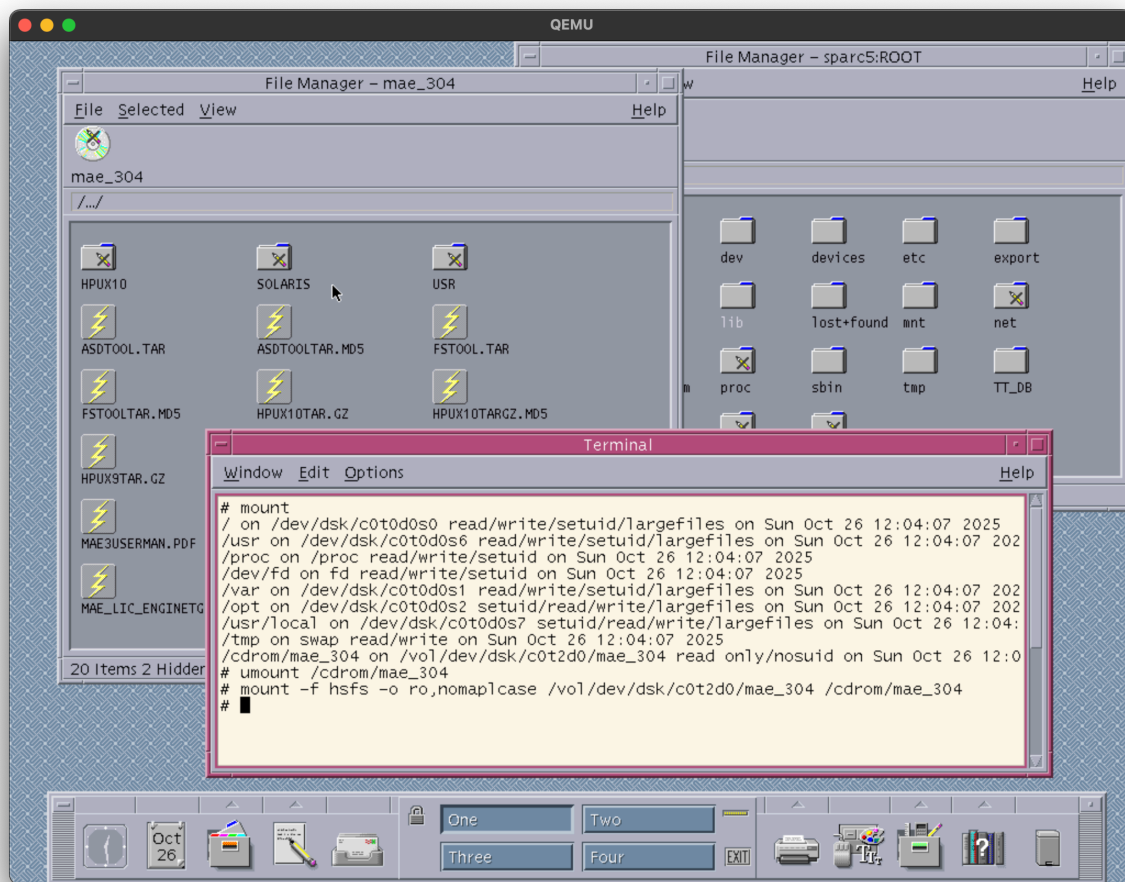
Unmount the CD-ROM by typing the following:

```
umount /cdrom/mae_304
```

Now re-mount the CD-ROM by typing the `mount` command with the following parameters, which we noted previously. The `nomaplcse` part avoids mapping everything to lowercase. (Note that the command is all one line.)

```
mount -f hsfs -o ro,nomaplcse /vol/dev/dsk/c0t2d0/mae_304 /cdrom/mae_304
```

You should now see uppercase filenames displayed in the CD window.



Let's get Macintosh Application Environment running in QEMU !

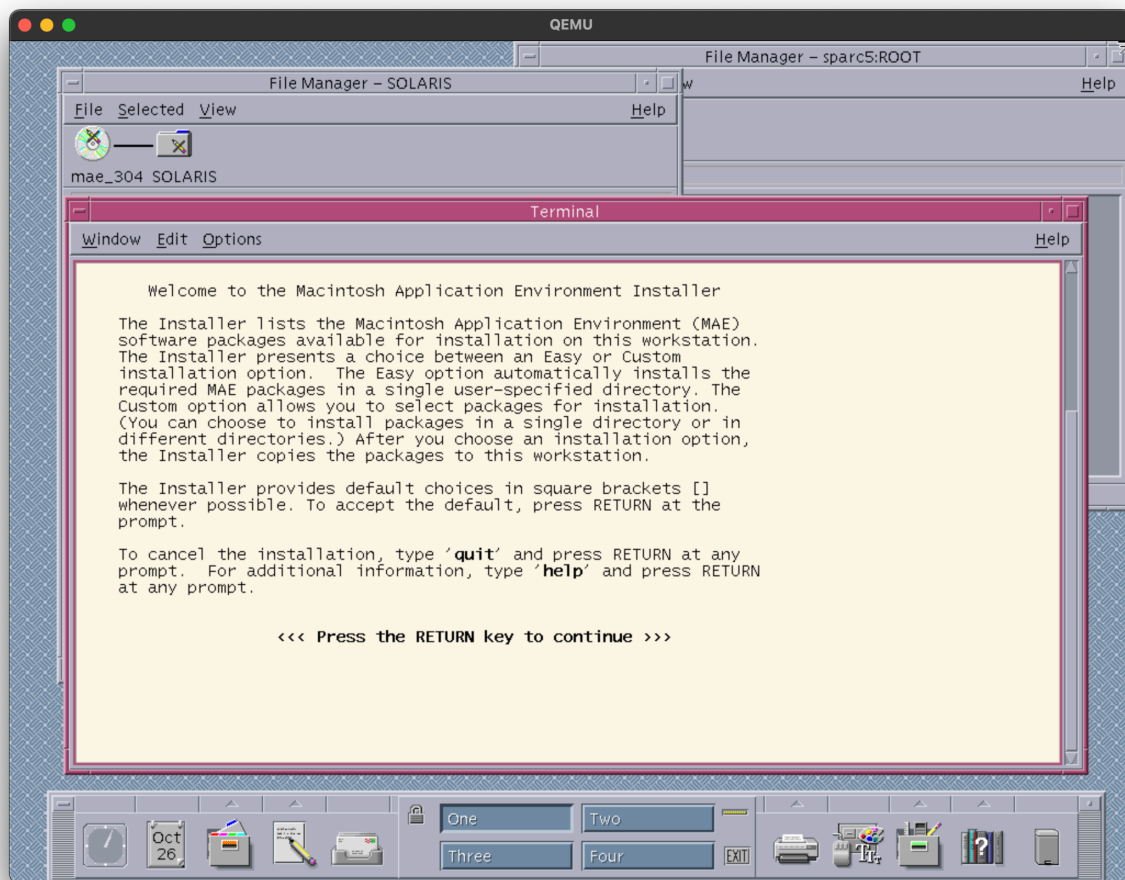
4. Install MAE

Open a Terminal window and set the current directory to the SOLARIS directory inside the MAE CD-ROM:

```
cd /cdrom/mae_304/SOLARIS
```

Start the installer by typing:

```
./INSTALL.SH
```



Press the Return key to advance to the next screen, selecting the Easy install option and the default location (`/opt/apple`) when prompted. The installer will tell you that the machine must be rebooted before AppleTalk will be available. At the end, the option to reboot immediately should default to "no", and return you to the shell instead of rebooting.

Let's get Macintosh Application Environment running in QEMU !

Install the license file

MAE requires that a valid license file, `license.dat`, be present in order to run the software. Back in the day, you would need to obtain this license file from Apple directly, or from a FLEXlm license management server in your organization which handled site licensing. However, Apple no longer provides any way to obtain a new license. A license file which was valid back in 1998 can be dropped into place and used, as long as we can manage to convince MAE that we're really living in the past. This is the job of an additional library that we can find on our MAE installation CD.

First, let's use the Solaris Terminal to unmount and re-mount the CD so its filenames are lowercase again:

```
umount /cdrom/mae_304

mount -f hsfs -o ro /vol/dev/dsk/c0t2d0/mae_304 /cdrom/mae_304
```

Now we copy the pieces in the `/cdrom/mae_304/usr/local/apple` directory to the `apple` directory where MAE is installed, using the following command:

```
cp -pR /cdrom/mae_304/usr/local/apple /opt
```

Also copy the `src` directory, which contains the dynamic library that will turn back the clock.

```
cp -pR /cdrom/mae_304/usr/local/src /opt
```

Create a MAE startup script

Two example scripts are provided in `/opt/apple/bin`. We'll make a copy of `mae.sh`, change it to be writable, and edit our script so we can use it to easily start up MAE. (If you aren't comfortable using the `vi` editor, Solaris also provides a Text Editor application whose icon can be found at the bottom of the screen.)

```
cd /opt/apple/bin

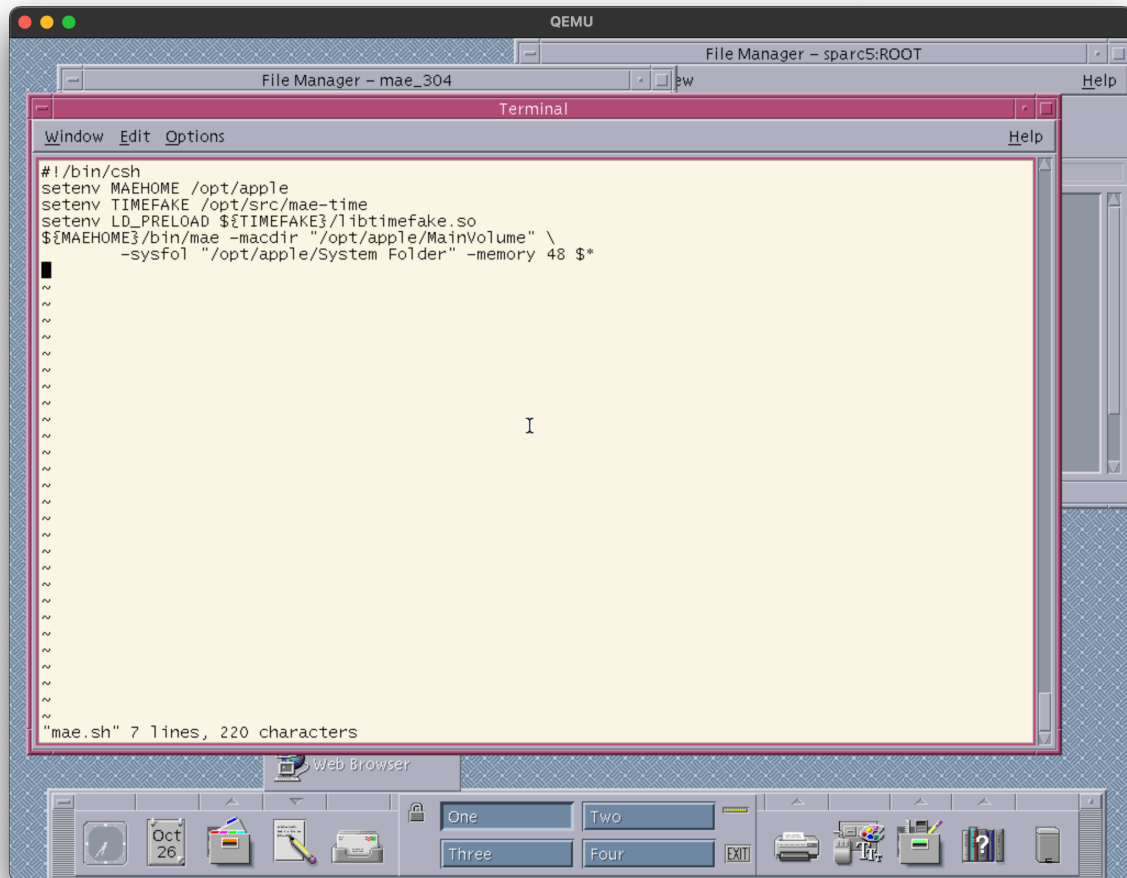
mv mae.sh mae_orig.sh && cp mae_orig.sh mae.sh

chmod u+w mae.sh

vi mae.sh
```

Let's get Macintosh Application Environment running in QEMU !

Edit the contents of the `mae.sh` script so the pathnames begin with `/opt` rather than `/usr/local`, and be sure to add double-quotes around pathnames that contain spaces. Speaking of that, you'll want to make sure there is a space in `System Folder` on the last line. Your script should look like this:



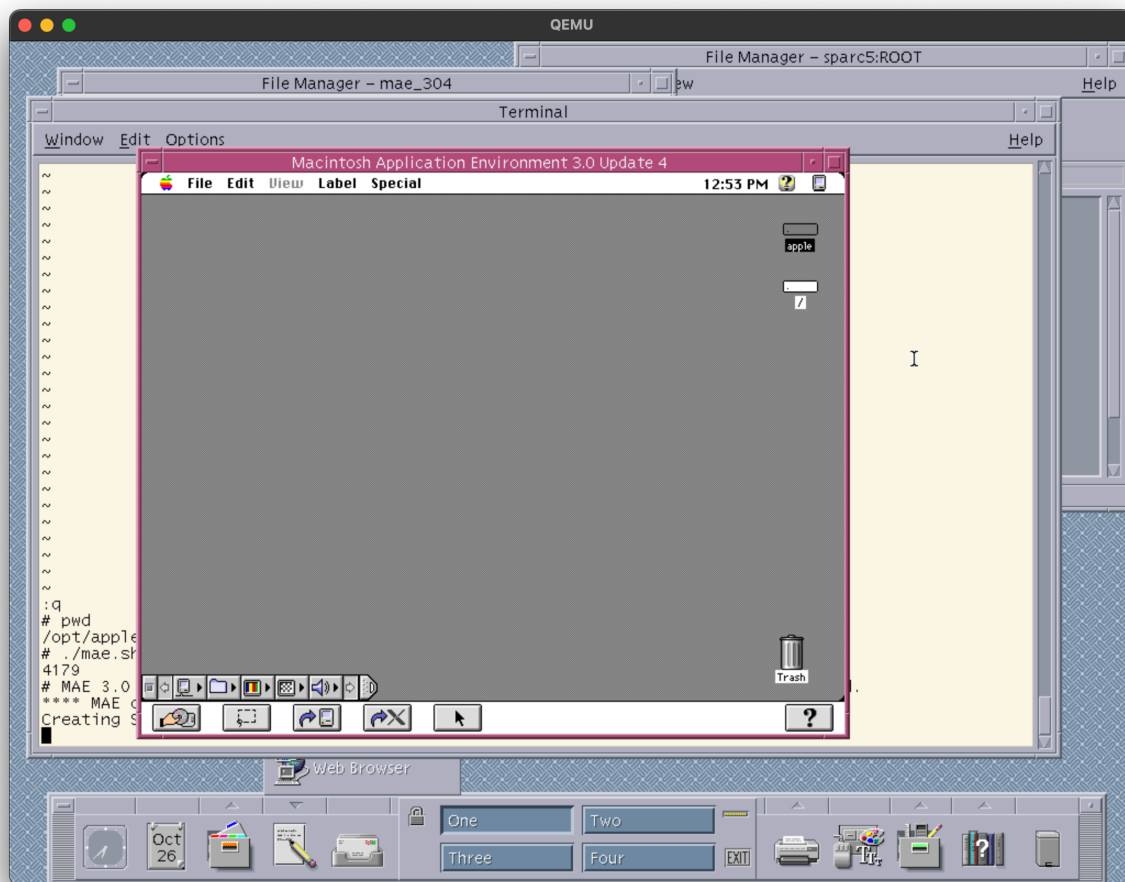
Start up MAE by running the script. (In all future sessions, you'll use this script to run MAE; if you try to run the `mae` binary directly, it will notice that the license has expired.) We'll add an ampersand to the end of our command, so the process will detach and allow us to continue using Terminal.

```
./mae.sh &
```

The Macintosh Application Environment should immediately open a new window and prompt you to accept its license agreement. Click Accept to continue.

If all went well, you should now be looking at a Macintosh desktop!

Let's get Macintosh Application Environment running in QEMU !



The Macintosh Application Environment, running in its own window.

5. Configure AppleTalk connectivity

We chose Solaris 2.6 to host MAE in part because we know that the AppleTalk kernel modules implemented by MAE were designed to work with STREAMS networking as it existed at that time. Later versions of Solaris may not work with those kernel modules. But first, let's verify that AppleTalk got installed correctly.

In the Solaris Terminal, the following command should show that ddp and lap are present under /dev/appletalk (after rebooting at least once since installation):

```
ls -la /dev/appletalk
```

If there is no /dev/appletalk present on your Solaris system, it may be "too new" to support AppleTalk. There seems to be a potential cutoff after Solaris 8 or 9.

Configure Solaris for the internal network

When we initially configured Solaris networking, we gave it the default QEMU IP address of 10.0.2.15, which is behind a NAT (network address translation) layer that QEMU provides. In practical terms, that means no other computers can see our Solaris machine on the network. Normally that's fine for TCP/IP connections made through the NAT, but we want AppleTalk packets to be passed as well, and it is not an IP protocol. So we must give our virtual machine a "real" IP address.

There are three files we need to edit on Solaris to make this happen: `/etc/hosts`, `/etc/defaultrouter`, and `/etc/resolv.conf`. These respectively hold our static IP address, the gateway address, and DNS server address. Note that you may need to use `chmod u+w` to make some of these files writeable first.

In `/etc/hosts`, I changed 10.0.2.15 to 192.168.11.98. This is an address on my local network which I know is not in use by other machines and is not part of the dynamically-assigned DHCP range. Pick a suitable static IP which can be assigned to this virtual machine on your local network.

In `/etc/defaultrouter`, I changed 10.0.2.2 to 192.168.11.1. This is the gateway (router) address on my local network. Use the gateway address for your own local network here.

In `/etc/resolv.conf`, I changed 10.0.2.3 to 192.168.11.11. This is the address of a DNS server on my local network. You should use your ISP's DNS server address here. In a pinch, you can use 1.1.1.1 or 8.8.8.8, which are publicly available DNS servers.

Once you've edited the three files, shut down your Solaris machine for the moment:

```
shutdown -h now
```

Configure QEMU for bridged networking

To get our Solaris virtual machine on the network, we'll change our previous QEMU command to specify a vmnet-bridged network interface:

```
sudo qemu-system-sparc -M SS-5 -m 128 -drive  
file=Solaris26.qcow2,bus=0,unit=0,media=disk -drive  
file=mae304.iso,bus=0,unit=2,media=cdrom,readonly=on -nic  
vmnet-bridged,ifname=en4,mac=52:54:00:f0:0d:03
```

Let's get Macintosh Application Environment running in QEMU !

The most important part of this network specification is the `ifname`, because that is the Ethernet interface on your *host computer*. Your computer will effectively be sharing it with the virtual Solaris machine. On your host computer (the one that's running QEMU), run the command `ifconfig -a` and scroll through the output to identify the specific interface that connects you to the Internet. In my case, that interface is `en4`, but it will almost certainly be different for you (`en0`, `en1`, `en7`, etc.), depending on your physical network hardware.

The MAC address of the virtual network card should be specified to ensure you have a unique value that isn't used anywhere else on the network. Feel free to change it however you want, just ensure that it's different for each QEMU instance you run (in case you want to run more than one simultaneously), and isn't the same as your real computer's MAC address.

Also, notice that our command now has a `sudo` in front of it, so you will need to enter an administrator password when running QEMU. To bridge network interfaces, we must be running as root.

Almost there, have patience!

When you first bring up the Solaris virtual machine in QEMU after this reconfiguration, it may initially seem that AppleTalk is not working. Start up MAE, then open the Chooser, as well as the Network control panel. You'll immediately notice that they appear very slow to open, as they make AppleTalk calls that end up loading the kernel modules, look up zones, and so on. AppleTalk applications will be slow and seem as if they cannot find anything. Try opening and closing the Chooser a few more times until you are able to see local servers and printers.

If you still don't see zones or servers in the Chooser after several minutes, check the status of `appletalk` in the Solaris Terminal with this command:

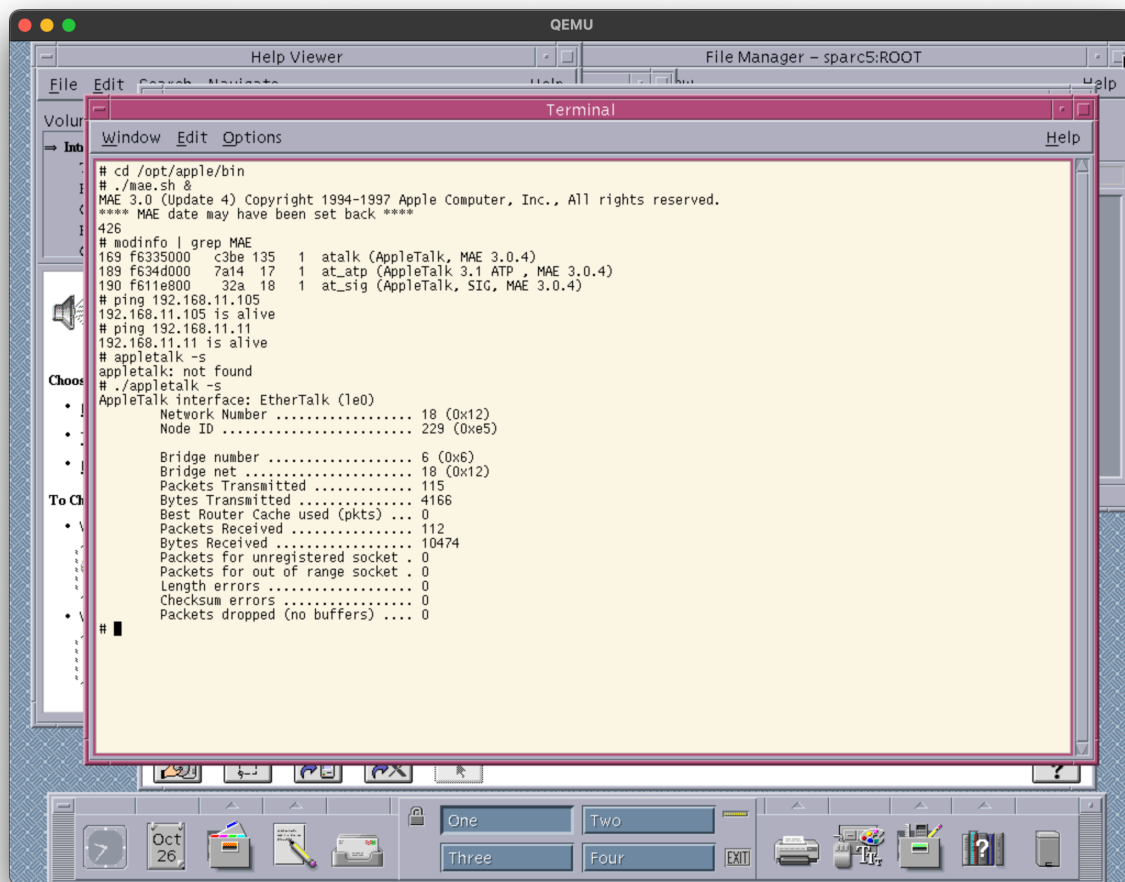
```
/opt/apple/bin/appletalk -s
```

This should show whether packets are being received in answer to AppleTalk queries. You can also check whether the kernel modules are loaded with:

```
/usr/sbin/modinfo | grep MAE
```

This should show whether any AppleTalk modules have been loaded. When this is working properly, the output should include `atalk`, `at_atp`, and `at_sig`.

Let's get Macintosh Application Environment running in QEMU !



Checking AppleTalk status in the Terminal with the appletalk command

If there are bytes being transmitted and received over EtherTalk, then you can be sure AppleTalk is working. It may take a reboot of the Solaris virtual machine to clear things up.

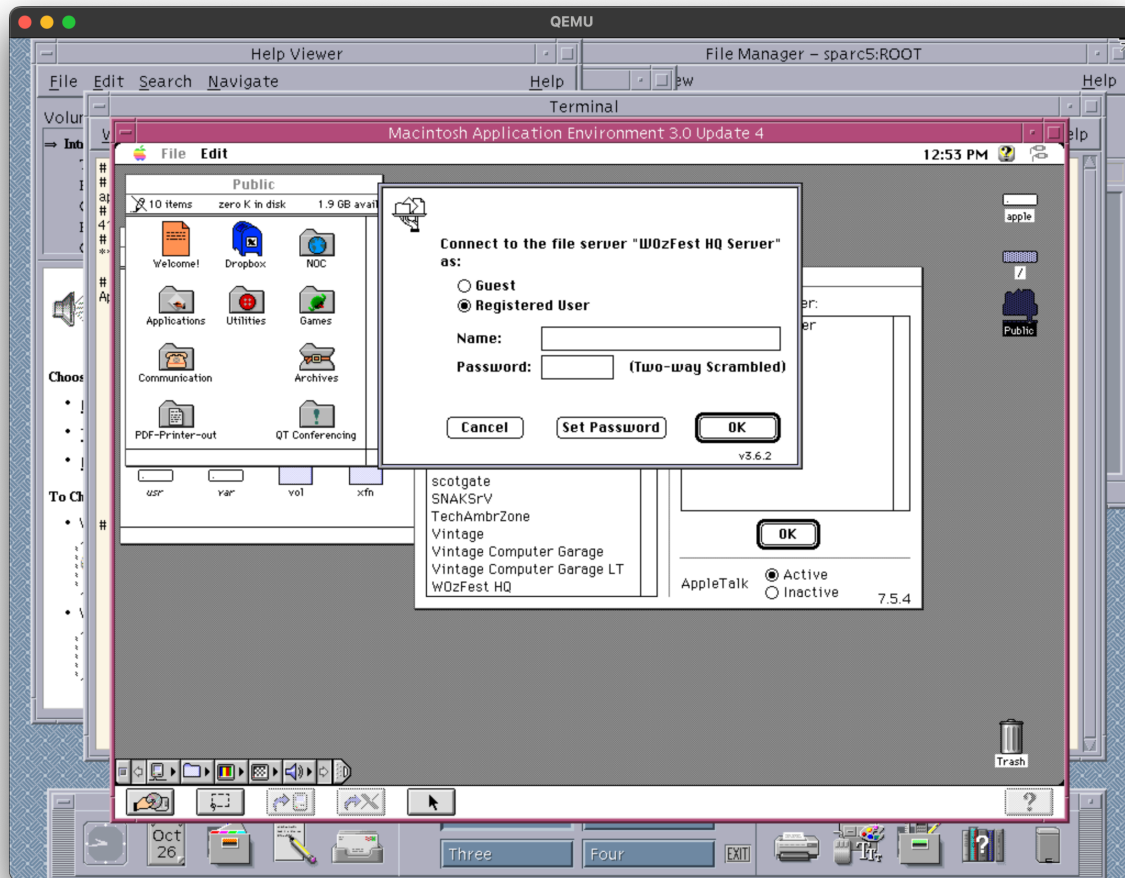
Eventually, you will start to see zones and servers and printers in MAE. Once you do, there is no further slowdown and MAE behaves just like any classic Macintosh which is connected to your network.

Credit where it's due

This guide was set into motion by a post from Dan ([@billgoats@bitbang.social](mailto:billgoats@bitbang.social)) on Mastodon, where he said, "I really do need to figure out how to get MAE AppleTalk networking to work at some point." It made me think about whether MAE could use AppleTalk in a virtual machine, and ultimately, it did.

Let's get Macintosh Application Environment running in QEMU !

Thanks to those intrepid retrocomputing adventurers who got Solaris and/or MAE running in QEMU before me, including neozeed (virtuallyfun.com), Newtonian (emaculation.com), and RetepV (retechvintage.blogspot.com).



Achievement unlocked: connecting to an AppleShare server in MAE

Footnotes

[1] https://www.reddit.com/r/VintageApple/comments/11v877r/most_absurd_marchintosh/

[2] <https://oldvcr.blogspot.com/2025/08/a-real-powerbook-macintosh-application.html>

[3] <https://www.emaculation.com/forum/viewtopic.php?t=10868>

[4] <https://marchintosh.com/globaltalk.html>

Let's get Macintosh Application Environment running in QEMU !

[5] <https://wiki.qemu.org/Documentation/Platforms/SPARC>

[6] <https://archive.org/details/solaris26desktop>

[7] <https://archive.org/details/mae-3.0.4>

[8] <https://virtuallyfun.com/2010/10/03/formatting-disks-for-solaris/>

[9] <https://retechvintage.blogspot.com/2024/10/your-own-sparcstation-5-workstation.html>

[10] <https://retechvintage.blogspot.com/2024/11/your-own-sparcstation-5-workstation.html>