Table of Contents

Introduction

Apple /// Ready-to-Run is a bundle of support files, docs, and scripts for MESS Apple /// emulation (http://www.mess.org/). Included is an HDD image with many popular Apple /// titles (source: http://apple3.org) preinstalled so that you can get started with your own virtual Apple /// as quickly as possible.

What's in the Box

- Apple /// Computer
  - 2 MHz 6502
  - 512K RAM Expansion
  - Internal Floppy Drive
  - (2) Joystick Ports (Demo: https://www.youtube.com/watch?v=W9pj7LUF6g8)
  - RS-232-C Port
    - Communications Emulation (Demo: https://www.youtube.com/watch?v=GuVev9AYGIc)
    - Printer Emulation (Demo: https://www.youtube.com/watch?v=ndU764dS23l)

- (3) External Floppy Drives
- CFFA2 Mass Storage Adapter in Slot 1
    - 128 MB Internal Hard Disk Drive (HDD) Partitioned into:
        - (2) 16MB Partitions:
            - `/BOS` (`.PROFILE`)
            - `/HOME` (`.CFIDE2`)
        - Space for 6 additional 16MB partitions available
- Thunderclock Card in Slot 2 for second resolution time
- Appli-Card 6 MHz z80 CP/M Adapter in Slot 3 for running those *other* business applications (and games!)
- AppleColor Monitor 100

Preinstalled Software on HDD

- BOS 1.0, the *most* advanced Apple /// Operating System
    - BOS Program Switcher
    - BOS Utilities
- SOS 1.3 System Utilities
- Backup ///
- Lazarus ///
- Access /// 3.2X (Demo: https://www.youtube.com/watch?v=GuVev9AYGIc)
- Access 3270 1.0
- Apple Writer ///
- VisiCalc Advanced (Demo: https://www.youtube.com/watch?v=eUx6TfEPDYA)
- /// E-Z Pieces

- Quick File ///
- Draw ON ///
- Apple /// Business Basic 1.23
- Apple /// Pascal 1.1
- Apple /// Fortran
- Apple /// COBOL
- CP/M 2.2 (2 MB Virtual HDD (`E:`) as file `/BOS/CPM1`)
    - WordStar 3.0
    - DBaseII 2.3B
    - Turbo Pascal 3.01A
    - Zork I: The Great Underground Empire
    - Zork II: The Wizard of Frobozz
    - Zork III: The Dungeon Master
    - SARGON 2.1

Installation

```
git clone https://github.com/datajerk/apple3rtr.git
```

or

Download https://github.com/datajerk/apple3rtr/archive/master.zip and extract (you'll want to rename `apple3rtr-master` to `apple3rtr`).

This document assumes that all work will be in the `apple3rtr` directory.

SDL and MESS Installation

OS/X

- SDL 2.x is required (http://www.libsdl.org/download-2.0.php)

  > Open `SDL2-2.x.x.dmg` and drag `SDL2.framework` to `/Library/Frameworks`

- SDLMESS 64-bit 0.155 or later: http://sdlmame.lngn.net

  > Extract and place in your path or in the `apple3rtr` directory.

  E.g.:

  ```
  cd apple3rtr
  curl http://sdlmame.lngn.net/mess0155-64bit.zip >mess0155-64bit.zip
  unzip mess0155-64bit.zip
  cd mess0155-64bit
  cp -a * ..
  cd ..
  rm -rf mess0155-64bit*
  ```

Windows

- SDL 2.x is required (http://www.libsdl.org/download-2.0.php)

- MESS 0.155 or later: http://www.mamedev.org/release.html--place in your path or in the `apple3rtr` directory.

- From the `apple3rtr` directory type:

  ```
  copy mess.ini.windows mess.ini
  ```

Linux

- Varies depending on distribution. SDL 2.x and MESS 0.155 or later required.

Roll Your Own (assumes OS/X or Linux)

- Install SDL 2.x.x development libraries (http://www.libsdl.org/download-2.0.php)

- Build from source:

  ```
  git clone https://github.com/mamedev/mame.git
  cd mame
  make TARGET=mess PTR64=1 -j4
  ```

  > Change `-j4` to `-j` *# of cores* to get faster builds.

Cold Booting

> If `mess64` is not in your path, then provide the full path name to `mess64`.
>
> OS/X and Linux users, if `mess64` is in `apple3rtr` then type `./mess64`.

1. Type on a single line (***but do not press RETURN--yet***):

   ```
   mess64 apple3 -skip_gameinfo -volume -24 -resolution 1024x768 -effect Scanlines0x4 -sl1 cffa2 -hard apple3.hd -sl2 thclock -sl3 applicard -ramsize 512k
   ```

   > Adjust `-volume` and `-resolution` to your taste. Volume 0 is the loudest.
   >
   > `-effect Scanlines0x4` is optional and only optimized for 1024x768.
   >
   > Smaller scanline options: `-resolution 768x576 -effect Scanlines75x3`
   >
   > Even smaller scanline options: `-resolution 512x384 -effect Scanlines75x2`

2. Append to the command line above *one* of the following floppy images and *then* press `RETURN`:

   > The `bosboot` and `cmpboot` images below have been modified to support the CFFA2 adapter.

- Apple /// Demo Disk:

  `-flop1 demodisk.dsk`

- BOS 1.0 (4) Disk III system with Apple /// preinstalled applications on HDD (`/BOS`):

  `-flop1 bosboot.dsk`

- CP/M 2.2 (4) Disk III system with CP/M preinstalled applications on HDD (`/BOS/CPM1` (`E:`)):

  > `A:` = `.D1` , `B:` = `.D2` , `C:` = `.D3` , `D:` = `.D4` , `E:` = `.PROFILE/CPM1`

  `-flop1 cpmboot.po`

- CFFA Utility to create or delete partitions (***DO NOT DELETE*** `/BOS` ***!***):

  `-flop1 cffautil_1.40a.dsk`

How to Operate Your Virtual Apple ///

All the keys on your keyboard *should* correspond to the correct Apple /// keys (`Left-Option/Alt` = `Open-Apple` and `Right-Option/Alt` = `Close-Apple`). The `DELETE` / `BACKSPACE` key has been remapped to the Apple /// `LEFT ARROW` key since there is no `DELETE` or `BACKSPACE` Apple /// key (each Apple /// application will determine how the `LEFT ARROW` behaves). The `F3` key has been mapped to `RESET` .

To remove or insert virtual floppies you will have to engage *PARTIAL Emulation* mode and then launch the *UI Menu*.

The default for MESS is to use the `INSERT` key to toggle *PARTIAL/FULL Emulation* mode and to use the `TAB` key to launch and dismiss the *UI Menu*. There are a few issues with the MESS defaults. First, not all keyboards have an `INSERT` key, and 2nd, while in *PARTIAL Emulation* mode keys sent for special functions such as launching the *UI Menu* are also sent to the Apple ///. This can contribute to a poor experience.

To address the aforementioned the *PARTIAL/FULL Emulation* mode, toggle has been mapped to the `F1` key, and the *UI Menu* to the `F2` key.

> OS/X users: If you have `F1` / `F2` set to control brightness (default OS/X behavior) then use `fn-F1` and `fn-F2` for `F1` and `F2` anytime you are instructed to use `F1` or `F2` .

To remove or insert a virtual floppy press `F1` to engage *PARTIAL Emulation* mode, then immediately press `F2` to access the *UI Menu*.

In the *UI Menu*, `DOWN ARROW` to the *File Manager*, press `RETURN`, select the drive, and then select the new floppy image. After you have made any and all changes in the *UI Menu* use `F2` to quickly return to the Apple ///, then `F1` to return to *FULL Emulation* mode.

> Avoid using `ESC` or *Return to System*. The last `RETURN` will get sent to the system when the *UI Menu* exits, and if not careful in *PARTIAL Emulation* mode an `ESC` will power off the Apple ///.

In summary:

1. `F1`, `F2` (or `fn-F1`, `F2`)
2. `ARROW` to *File Manager*
3. Select your images
4. `F2`, `F1` (or `fn-F2`, `F1`)

To power down your Apple ///: Quit any applications, and then select *Quit* from the BOS Program Switcher. If using CP/M, just quit your application first. For any other single disk application, just quit. Now you are ready to power down. To power down press `F1` (`fn-F1`) to enter *PARTIAL Emulation* mode, then press `ESC` to shutdown.

> While in *PARTIAL Emulation* mode many of the other keys on the keyboard will be setup to perform other functions, e.g. *P* will PAUSE the Apple ///. Refer to the MESS documentation (http://www.mess.org/mess/howto#using_mess) *and be careful!*

Known Issues

- Apple][emulation via BOS does not work. Use a proper Apple][emulation diskette image.

Support

- RTFMs: http://apple3.org and http://www.mess.org/mess/howto.
- Join and post a message to the Apple /// Facebook group (https://www.facebook.com/groups/appleiii/).
- Post a message to comp.sys.apple2.

- Post an issue on Github (https://github.com/datajerk/apple3rtr/issues).

More Info About MESS Apple ///

http://rbelmont.mameworld.info/

Advanced HOW-TOs

Using Meat-net to get data/programs in and out of the Apple ///

Getting things in and out of the Apple /// can be done with comm programs (see Get "online" with Access ///), VSDRIVE (see VSDRIVE), or by direct manipulation of apple3.hd (see Directly Manipulating apple3.hd). However, using floppy images is probably the easiest and most strait forward.

For SOS, AppleCommander works very well: http://applecommander.sourceforge.net. CiderPress for Windows/WINE also works: http://ciderpress.sourceforge.net.

For CP/M, use CPMTools (http://www.moria.de/~michael/cpmtools/) with a format type of `apple-do` for DOS and `apple-po` for ProDOS ordered disk images. OS/X users will need to install the CLI development tools to build. Windows users can download Bill Buckels binaries from http://www.cpm8680.com/cpmtools.

DOS order images should be suffixed with `.do` or `.dsk`, and ProDOS/SOS ordered disk images should be suffixed with `.po`. The incorrect suffix will confuse MESS.

See How to Operate Your Virtual Apple /// (above) for instructions on working with floppy images.

Fortran /// Q&D

1. Launch *Pascal ///*, edit and save your program, e.g. `/HOME/HELLO.F`

2. From *Pascal ///*, `X)ecute /BOS/FORTRAN/FORTRAN` *or* launch Fortran /// from the *BOS Menu*.

3. Follow the prompts, e.g. `/HOME/HELLO.F`, `/HOME/HELLO`

4. From *System Utilities* copy `/BOS/FORTRAN/SYSTEM.LIBRARY` to `/HOME/HELLO.LIB`

5. From *Pascal ///*, `X)ecute /HOME/HELLO`

Google *Apple III Fortran Reference Manual* for more details.

Get "online" with Access ///

The MESS 0.155 Apple /// driver now includes RS-232 support. To enable for communications use append: `-rs232 null_modem -bitb <target>`, where `<target>` is a socket descriptor (`socket.host:port`) for TCP-based bidirectional null-modem communications, e.g. `socket.127.0.0.1:2023`.

> If MESS crashes while starting with `-rs232 null_modem -bitb socket...`, then wait 30 seconds and try again.

As with physical null-modem serial-to-serial communications it is necessary to match the baud (300, 600, 1200, 2400, 4800, 9600, 115200) and protocol (e.g. 8-N-1) on both ends. For the MESS end of the null-modem connection this can be accomplished via the *System Configuration* menu in the *UI Menu*. The default is 9600, 8-N-1. For the Apple /// end, the settings may be available in the application (e.g. Access ///) or may require a change to the `.RS232` driver configuration block via the *System Configuration Program* (SCP). See the *Apple /// Standard Device Drivers Manual* for more details.

E.g., `/bin/bash` access from Access ///:

1. Install socat

   > E.g. MacPorts (https://www.macports.org/) users can type:
   >
   > ```
   > sudo port install socat
   > ```

2. Type:

   ```
   socat tcp-l:2023,reuseaddr,fork exec:/bin/bash,pty,setsid,setpgid,stderr,ctty &
   ```

   > ***IMPORTANT: POTENTIAL SECURITY RISK!!! USE A FIREWALL OR READ SOCAT DOCS ON HOW TO LIMIT ACCESS!!!***

3. Append to MESS command line:

```
-rs232 null_modem -bitb socket.127.0.0.1:2023
```

4. After shutting down MESS kill socat:

   ```
   kill %1
   ```

Video demo: https://www.youtube.com/watch?v=GuVev9AYGIc

VSDRIVE

VSDRIVE (Virtual Serial Drive) is an RS-232-based virtual drive hosted by a Mac or PC running ADTPro in *localhost* mode. Before continuing please read http://adtpro.sourceforge.net/vdrive.html to familiarize yourself with VSDRIVE.

VSDRIVE is a built-in feature of ADTPro. Only ADTPro-2.0.0 (http://sourceforge.net/projects/adtpro/files/adtpro/ADTPro-2.0.0) has been tested and is included with this distribution.

In the `disks` subdirectory of the included ADTPro-2.0.0 there are two zip files, `v800k.zip` and `v16m.zip`, containing two files, `Virtual.po` and `Virtual2.po`. The sizes are 800K and 16M respectively. You can replace them with any size ProDOS formatted volume as long as the volume is < 16MB (max recognized by SOS) and the names unchanged. You can create new virtual disks with AppleCommander (http://applecommander.sourceforge.net/) or CiderPress (http://ciderpress.sourceforge.net/).

> < 16MB is 32767 sectors (max). The provided 16MB images are 32767 sectors.

To start VSDRIVE:

1. Type: `cd ADTPro-2.0.0/disks`
2. Unzip `v800k.zip` for two 800K virtual disks or unzip `v16m.zip` for two 16M virtual disks.

   > OS/X and Linux users can type `unzip` *filename.zip*.

3. Type: `cd ..`
4. Type: `./adtpro.sh localhost &` (OS/X, Linux) or `adtpro localhost` (PC).
5. Type: `cd ..`

To use VSDRIVE standalone, append to the MESS command line:

```
-flop1 VDRIVE-2.0.0.DSK -rs232 null_modem -bitb socket.127.0.0.1:1977
```

To use VSDRIVE with BOS, CFFA2, etc..., append to the MESS command line:

```
-flop1 bosbootvsd.dsk -rs232 null_modem -bitb socket.127.0.0.1:1977
```

> ***IMPORTANT: IN ALL CASES, CHANGE THE BAUD RATE FROM THE UI MENU TO 115200 BEFORE ACCESSING .VSDRIVE or .VSDRIVE2!*** See Get "online" with Access /// about changing baud rates.
>
> If MESS crashes while starting with `-rs232 null_modem -bitb socket...`, then wait 30 seconds and try again.

The 16M `Virtual.po` and `Virtual2.po` will be mounted as `/V` and `/V2` respectively. The 800K `Virtual.po` and `Virtual2.po` will be mounted as `/VDRIVE.2.0.0` and `/VDRIVE2.2.0.0` respectively. Alternatively, `.VSDRIVE` and `.VSDRIVE2` can be used.

> The bosboot VSDRIVE disks have `.RS232` and `.PRINTER` removed. Serial printing and other non-VSDRIVE serial functions will be unavailable.

To use VSDRIVE with any other boot disks, use the *System Configuration Program* (SCP) to add the driver `VSDRIVE.A3DRVR` from `VDRIVE-2.0.0.DSK`. See the *Apple /// Standard Device Drivers Manual* for more details.

VSDRIVE performs slower than the CFFA2. The intended purpose is for large scale transfers *the Apple /// way*. Advanced users will want to use the direct methods below.

Video demo: https://www.youtube.com/watch?v=x-EU9nii2hM

Directly Manipulating apple3.hd

HOW-TOs Using Meat-net to get data/programs in and out of the Apple ///, Get "online" with Access ///, and VSDRIVE provide Apple /// "native" ways of transferring files (diskette, serial, and VSDRIVE). However, the fastest way, is to directly manipulate the HDD image from the host OS.

Prerequisites:

1. The `chdman` utility located in the MESS `tools` directory. Hopefully your MESS binary distribution includes this utility.

2. The `dd` command. Windows users will need to install/use cygwin (http://www.cygwin.com/) and run the following commands from a bash shell.

To extract the `BOS` and `HOME` volumes, type:

```
tools/chdman extracthd -f -i apple3.hd -o apple3.hd.image
dd if=apple3.hd.image of=bos.po bs=512 skip=1 count=32767
dd if=apple3.hd.image of=home.po bs=512 skip=32768 count=32767
```

The `chdman extracthd` command will extract the raw bits from the CHD formatted `apple3.hd` image into the file `apple3.hd.image`.

The two `dd` commands will create the files `bos.po` and `home.po`. Both are 16MB ProDOS (SOS) formatted images that can be directly manipulated with AppleCommander (http://applecommander.sourceforge.net/) or CiderPress (http://ciderpress.sourceforge.net/).

To create a new HD image from existing `bos.po` and `home.po` images:

1. Backup your files with:

   ```
   cp -f apple3.hd apple3.hd.backup
   cp -f apple3.hd.image apple3.hd.image.backup
   ```

2. Concatenate the partition table, `bos.po`, `home.po`, and the rest of the original image into a new image (The `(` and `)` are part of the command):

   ```
   (dd if=apple3.hd.image.backup bs=512 count=1; cat bos.po home.po; dd
    if=apple3.hd.image.backup bs=512 skip=65535) >apple3.hd.image
   ```

3. Create a new [replacement] CHD formatted HDD image:

   ```
   tools/chdman createhd -f -c none -i apple3.hd.image -o apple3.hd
   ```

4. Test your new image.

5. Cleanup (or not, it's a good to have a backup), type:

   ```
   rm apple3.hd.backup apple3.hd.image.backup apple3.hd.image bos.po ho
   me.po
   ```

Upgrade apple3.hd while preserving your HOME volume

Directly manipulating apple3.hd is a handy way to preserve your data with updated `apple3.hd` images included with this distribution, e.g.:

1. Dump current `HOME` volume:

   ```
   tools/chdman extracthd -f -i apple3.hd -o apple3.hd.image
   dd if=apple3.hd.image of=home.po bs=512 skip=32768 count=32767
   ```

2. Install new `apple3.hd` from distribution (or just extract new apple3rtr release)

3. Dump new `BOS` volume:

   ```
   tools/chdman extracthd -f -i apple3.hd -o apple3.hd.image
   dd if=apple3.hd.image of=bos.po bs=512 skip=1 count=32767
   ```

4. Combine into a new `apple3.hd` :

   ```
   cp -f apple3.hd apple3.hd.backup
   cp -f apple3.hd.image apple3.hd.image.backup
   (dd if=apple3.hd.image.backup bs=512 count=1; cat bos.po home.po; dd if=apple3.hd.image.backup bs=512 skip=65535) >apple3.hd.image
   tools/chdman createhd -f -c none -i apple3.hd.image -o apple3.hd
   ```

5. Clean up (or not):

   ```
   rm apple3.hd.backup apple3.hd.image.backup apple3.hd.image bos.po home.po
   ```

Directly Manipulating CPM1

The safest way to work with the CP/M HD image is from CP/M using floppy images to add or backup content. However, the advanced user, may find that a bit pedestrian.

Prerequisites:

1. ***CLEARLY UNDERSTAND*** Directly Manipulating apple3.hd (above).

2. Download and install CPMTools (http://www.moria.de/~michael/cpmtools/). OS/X users will need to install the CLI development tools to build. Windows users can download Bill Buckels binaries: http://www.cpm8680.com/cpmtools.

3. Edit CPMTools `diskdefs` and append:

   ```
   diskdef a3cpmhd
     seclen 128
     tracks 512
     sectrk 32
     blocksize 4096
     maxdir 512
     skew 0
     boottrk 2
     os 2.2
   end
   ```

   > If you create a different size CPM1 or CPM2, then you will need to use `STAT x:DSK:` where `x` is the HD image letter to get the geometry info required to build a CPMTools disk definition. E.g.:
   >
   > ```
   >     E: Drive Characteristics
   > 16320: 128 Byte Record Capacity
   >  2040: Kilobyte Drive  Capacity
   >   512: 32  Byte Directory Entries
   >     0: Checked  Directory Entries
   >   256: Records/ Extent
   >    32: Records/ Block
   >    32: Sectors/ Track
   >     2: Reserved Tracks
   > ```

4. Extract `CPM1` from `bos.po`:

   A. Follow Directly Manipulating apple3.hd and extract `bos.po`.

   B. Using AppleCommander or CiderPress extract `CPM1`, e.g. OS/X:

   ```
   alias applecmd='java -jar /Applications/AppleCommander.app/Contents/Resources/Java/AppleCommander.jar'
   applecmd -g bos.po cpm1 >CPM1
   ```

Directly Manipulate `CPM1`:

At this point you should have a 2097152 byte file called `CPM1`. Use the CPMTools commands `cpmls`, `cpmcp`, `cpmrm`, etc... with the `-f a3cpmhd` option to directly manipulate `CPM1`, e.g.:

Listing files from `CPM1`, type:

```
cpmls -f a3cpmhd CPM1
```

Extract files from `CPM1`, type:

```
cpmcp -f a3cpmhd cpm1 0:zork1.com zork1.com
```

Remove or Write/Update files to `CPM1`, type:

```
cpmrm -f a3cpmhd CPM1 0:sargon.com
cpmcp -f a3cpmhd CPM1 ~/Downloads/SARGON.COM 0:sargon.com
```

> You must remove the file first (if it exists).

Updating `bos.po` with `CPM1`:

1. Use AppleCommander or CiderPress to remove the old CPM1 and replace it with the new `CPM1`, e.g. OS/X:

   ```
   applecmd -d bos.po CPM1
   applecmd -p bos.po CPM1 PDA <CPM1
   ```

2. Follow Directly Manipulating apple3.hd to update/create `apple3.hd` with the updated `bos.po`.

   > An alternative to using Directly Manipulating apple3.hd would be VSDRIVE by copying `/BOS/CPM1` to `.VSDRIVE`, and then using AppleCommander or CiderPress to extract/replace `CPM1` from `Virtual.po`.

Printing

The MESS 0.155 Apple /// driver now includes RS-232 support. To enable for printer use append: `-rs232 null_modem -bitb <target>`, where `<target>` is a *filename* for the printer output, e.g. `printer.txt` .

As with physical null-modem serial-to-serial communications it is necessary to match the baud (300, 600, 1200, 2400, 4800, 9600, 115200) and protocol (e.g. 8-N-1) on both ends. For the MESS end of the null-modem connection this can be accomplished via the *System Configuration* menu in the *UI Menu*. The default is 9600, 8-N-1. For the Apple /// end, the settings are available in the `.PRINTER` driver configuration block via the *System Configuration Program* (SCP). See the *Apple /// Standard Device Drivers Manual* for more details.

> `bosboot.dsk` and `cpmboot.po` have `.PRINTER` set to 9600 8-N-1 to match the MESS default.

To capture printer output append to MESS command line:

```
-rs232 null_modem -bitb printer.txt
```

SOS Printing, use: `.PRINTER`

CP/M Printing, use: `LST:`

If given an option to append line feeds (LF) to carriage returns (CR), take it.

Video demo: https://www.youtube.com/watch?v=ndU764dS23I

Disable All Audio

Just append to command line:

```
-audiodriver none
```

Joystick

The MESS 0.155 Apple /// driver now inlcudes analog joystick support.

> Your average USB retro gamepad/stick will not work since Apple /// joysticks are analog. Get an analog stick.

Tested analog sticks:

- Classic Apple II joysticks with the following adapter:

https://www.tindie.com/products/option8/retroconnector-joystick-shield/

- Xbox 360

Nothing is required except to connect your joystick *before launching MESS*. Use the *Analog Controls* menu from the *UI Menu* to adjust the center and sensitivity as necessary.

Video demo: https://www.youtube.com/watch?v=W9pj7LUF6g8

Numeric Pad

No mystery here, but there are some applications that require a numeric keypad. E.g. Draw ON ///.

The following has been tested with OS/X MESS:

- Belkin YourType Bluetooth Wireless Keypad: (http://www.belkin.com/us/p/P-F8T067/)
- Logitech Wireless (not Bluetooth!) Solar Keyboard K750 for Mac: (http://www.logitech.com/en-us/product/wireless-solar-keyboard-k750-mac)

Boot from VSDRIVE

The latest versions of MESS and ADTPro have measurably increased the speed of VSDRIVE making it an option as your primary HD. "Why?", you ask. Answer: simple direct access to `/BOS` and `/HOME` as `Virtual.po` and `Virtual2.po`.

To make the switch from CFFA to VSDRIVE:

1. Follow Directly Manipulating apple3.hd to extract `bos.po` and `home.po`.
2. Move `bos.po` and `home.po` to `ADTPro-2.0.0/disks` as `Virtual.po` and `Virtual2.po`, e.g. type:

   ```
   mv bos.po ADTPro-2.0.0/disks/Virtual.po
   mv home.po ADTPro-2.0.0/disks/Virtual2.po
   ```

3. Start ADTPro Server, type:

   ```
   cd ADTPro-2.0.0
   ```

```
Mac/Linux: ./adtpro.sh localhost &
Windows: adtpro localhost
cd ..
```

4. Use the following boot image and RS232 settings:

   ```
   -flop1 bosbootvsdroot.dsk -rs232 null_modem -bitb socket.127.0.0.1:1977
   ```

   > ***IMPORTANT: THE BPS RATE MUST BE SET TO 115K. You've got about 5 seconds to set it on boot, however after setting the first time MESS will remember.***

> ADTPro must be restarted anytime there is an out-of-band, i.e. direct manipulation of `Virtual.po` or `Virtual2.po`. IOW, shutdown MESS, then ADTPro before changing `Virtual.po` or `Virtual2.po`.
>
> Not 100% stable. YMMV.

Using the boot Script (OS/X and Linux only)

If typing long command lines is too unpleasant for you, then consider using the `boot` script included with this distribution. `boot` provides an easy way to launch MESS Apple /// emulation with most of the aforementioned options listed in this document. `boot` will also attempt to coordinate the launching of other required services, e.g. `socat` and ADTPro.

> `boot` has not been agressively tested. YMMV.

To launch `boot` simply type: `./boot` from the `apple3rtr` directory.

If you do not pass any options to `./boot`, then you'll get the following output:

```
$ ./boot

Usage: boot [-h] [-sml] [-r prt|null|vsd] [-o port] [-acn] diskimage1 [diskimage2 diskimage3 ...]

diskimage1 (and only diskimage1) may use the following optional aliases:
```

```
    bos       = bosboot.dsk
    cpm       = cpmboot.po
    vsd       = bosvsdboot.dsk
    vsdroot   = bosvsdrootboot.dsk

    -h        display this help and exit
    -s        512x384 resolution with scanlines
    -m        768x576 resolution with scanlines
    -l        1024x768 resolution with scanlines
    -r prt    rs232 printer @ 9600 BPS outputing to printer.txt
    -r null   rs232 null modem @ 9600 BPS listening on "port", default 2
023
    -r full   rs232 null modem @ 115K BPS listening on "port", default 2
023
    -r vsd    rs232 null modem @ 115K BPS listening on "port", default 1
977
    -o port   override default port (requires -r), port = 0 will be rand
om
    -a        launch ADTPro on port specified by -r vsd or -o
    -c        launch socat on port specified by -r null or -o
    -n        no HD
```

Most users will just run `./boot bos`. This will boot up a virtual Apple /// using the following command:

```
mess64 apple3 -skip_gameinfo -volume -24 -sl1 cffa2 -hard apple3.hd -sl2
 thclock -sl3 applicard -ramsize 512k -resolution 768x576 -effect Scanli
nes75x3 -flop1 bosboot.dsk
```

`bos` is an alias for `bosboot.dsk`, other aliases are listed for your reference when invoking `./boot` without any parameters. You can of course supply any diskette image name, as well as the names for diskette images you'll like inserted in to virtual floppy drives `.D2`, `.D3`, and `.D4`.

You can modify the behavior with the options listed above. E.g. say you wanted to use BOS with VSDRIVE, simply type `./boot -a -r vsd vsd`, e.g.:

```
$ ./boot -a -r vsd vsd
ADTPro Server version 2.0.0

Running... mess64 apple3 -skip_gameinfo -volume -24 -sl1 cffa2 -hard app
```

```
le3.hd -sl2 thclock -sl3 applicard -ramsize 512k -resolution 768x576 -ef
fect Scanlines75x3 -flop1 bosbootvsd.dsk -rs232 null_modem -bitb socket.
127.0.0.1:1977
```

This command will launch ADTPro and connect MESS to it for VSDRIVE support from BOS. MESS will also be configured for 115K BPS serial speeds. After exiting MESS, ADTPro will be shutdown.

> You must have `Virtual.po` and `Virtual2.po` prepared first. If not setup, then ADTPro will create two blank 800K disk images.

The rest of the options should be fairly strait forward.

- `-s`, `-m`, `-l` are essentially *small*, *medium*, *large* display sizes (see the `boot` help output for the exact sizes). The default is *medium*.

- `-r` specifies RS-232 support and takes a single argument (`prt`, `null`, `full`, `vsd`) that is used to determine 9600 or 115K BPS as well as what port to connect to.

- If you are having problem with TCP sockets taking a long time to time out you can override the port number with `-o` *port number*. A port number of `0` (e.g. `-o 0`) will generate a random port using the range 1000-1999. `-o` requires `-r`.

- `-a` will launch ADTPro and configure it to listen to the port specified by `-r` and `-o`. `-a` and `-c` are mutually exclusive. ADTPro will be killed when MESS exits.

- `-c` will launch `socat` and configure it to listen to the port specified by `-r` and `-o`. `-c` and `-a` are mutually exclusive. `socat` will be killed when MESS exits.

  > It is not necessary to use `-a` or `-c` if you launch the services yourself.

- `-n` will not use the CFFA adapter excluding the use of `apple3.hd`