

Sweet16 User Guide

Version 3.0.3

Copyright © 1996-2014 F.E. Systems and Eric Shepherd

Developed for Mac OS X by Eric Shepherd

Based upon Bernie][The Rescue by
André Horstmann, Henrik Gudat, and Urs Hochstrasser

Marinetti Link Layer by
Ewen Wannop

Version 3.0 tested by
Tony Diaz, Kelvin Sherlock, Sean Fahey, Sophia Shepherd, and Dain Neater

Previous versions tested by
Tony Diaz, Sean Fahey, Kirk Mitchell, Ewen Wannop, Carl Knoblock, Jon Co, Ryan Suenaga, Antoine Vignau, and David Wilson

Additional thanks to
My wife, Sarah, for being surprisingly supportive,
and my daughter Sophie for thinking this is really, really cool.

Dedicated to Ryan Suenaga

Table of Contents

Table of Contents	2
License	5
License for jansson	5
Sweet16 uses libwebsockets	5
Support	5
Compatibility	6
Getting Started	6
Creating a ROM Image File	6
ROM 01	
ROM 03	
Installing Sweet16	7
Working With Disk Images	7
Creating Disk Images	
The Mounted Disks Window	
Mounting Disk Images	
Automatically Mounting Disk Images	
Mounting CD-ROM Media	
Ejecting (Unmounting) Disk Images	
Supported Disk Image Formats	13
Starting Up For the First Time	14
Getting Startup Disks	
Installing an Operating System	15
Enabling Platform Integration	16
ReadMeFirst	
SweetDupe	
SweetLWPatch	
ToggleFullScr	
SweetPrinter	
Sweet16	
GetClipboard	

SendClipboard	
Moving Files Between the Apple IIgs and the Mac	19
Printing	19
Networking	20
Using Emupacks	22
Creating an Emupack	22
Using an Emupack	22
Preferences	23
Memory	23
CPU	24
Display	25
Storage	26
Other	26
Menu Reference	27
File menu	27
Edit menu	28
Setup menu	28
View Menu	29
Window Menu	30
Help Menu	30
The Sweet16 Debugger	31
Release Notes	33
Version 3.0.3 (April 6, 2014)	33
Version 3.0.2 (November 26, 2013)	33
Version 3.0.1 (August 12, 2013)	33
Version 3.0 (July 24, 2013)	34
Version 2.3.1 (July 31, 2011)	35
Version 2.2 (July 18, 2010)	36
Version 2.1.3 (August 29, 2009)	37
Version 2.1.2 (August 26, 2009)	37
Version 2.1.1 (August 4, 2009)	37
Version 2.1 (July 23, 2009)	38

Version 2.0 (July 23, 2009)	38
Version 2.0 Public Beta 2 / 2.0b18 (August 11, 2008)	40
Version 2.0 Public Beta 2 / 2.0b17 (August 10, 2008)	40
Version 2.0 Public Beta 1 (July 19, 2007)	41
Version 1.7.3 (September 12, 2006)	42
Version 1.7.2 (August 28, 2006)	42
Version 1.7.1 (August 18, 2006)	42
Version 1.7 (August 14, 2006)	44
Version 1.6.1 (August 1, 2006)	44
Version 1.6 (July 19, 2006)	44

License

Sweet16 is freeware. You may use it free of charge, but may not distribute altered versions of the software. Copyright is retained by F.E. Systems, and, in parts, Eric Shepherd.

Please note that as is the case with any software product, there is the possibility that bugs exist that could cause damage to or loss of data, no matter how carefully constructed and tested. Neither Eric Shepherd nor F.E. Systems is responsible if loss occurs; you use this software at your own risk.

License for jansson

Sweet16 uses the jansson JSON library for C, which is distributed under the MIT license, as follows:

Copyright (c) 2009-2012 Petri Lehtinen <petri@digip.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Sweet16 uses libwebsockets

Sweet16 makes use of the libwebsockets library to implement its support for WebSocket technology in its debugging features. See <http://libwebsockets.org> for more information about this library.

Support

The first place to go for up-to-date information about Sweet16—and to get technical and other support—is the Sweet16 web page at <http://sheppyware.net/software-mac/sweet16/>.

You may also contact Eric Shepherd, the developer, by email directly at sheppy@sheppyware.net.

Compatibility

Sweet16 3.0 requires a Macintosh running Mac OS X 10.6.8 or later. To run on older Macs (including PowerPC-based Macs), you will need an older version of Sweet16.

Getting Started

To run Sweet16, you need a copy of the Apple IIgs ROM from either a ROM 01 or ROM 3 Apple IIgs system. You'll also need appropriate disk images to start up from. This section will cover how to get these items.

Creating a ROM Image File

For legal reasons involving copyright law, you must extract a copy of the ROM from an Apple IIgs you personally own in order to legally use Sweet16. If you contact me asking for a ROM image, I'll delete your email and pretend you never asked.

The easiest way to create a ROM image file is to download *GS ROM Grabber* by Sarah Shepherd and run it on your real Apple IIgs to save a copy of your ROM to disk, then copy that to your Macintosh. You can get it at <http://www.sheppyware.net/apple-ii-software/gs-rom-grabber.html>.

To create your ROM image file by hand, start by booting your real Apple IIgs into ProDOS 8, or by booting into GS/OS and then launching BASIC.SYSTEM to get into Applesoft BASIC.

ROM 01

If your computer is a ROM 01 system, enter the following command lines (keep in mind that the "]" and "*" characters are prompts; don't type them):

```
]call -151
*0/800<FE/0.7FFFM
*BSAVE APPLE2GS.ROM,A$800,L$8000
*0/800<FE/8000.FFFFM
*BSAVE APPLE2GS.ROM,A$800,L$8000,B$8000
*0/800<FF/0.7FFFM
*BSAVE APPLE2GS.ROM,A$800,L$8000,B$10000
*0/800<FF/8000.FFFFM
*BSAVE APPLE2GS.ROM,A$800,L$8000,B$18000
```

ROM 03

If your computer is a ROM 3 system, enter the following lines (again, keeping in mind that "]" and "*" are prompts):

```

]call -151
*0/800<FC/0.7FFFM
*BSAVE APPLE2GS.ROM2,A$800,L$8000
*0/800<FC/8000.FFFFM
*BSAVE APPLE2GS.ROM2,A$800,L$8000,B$8000
*0/800<FD/0.7FFFM
*BSAVE APPLE2GS.ROM2,A$800,L$8000,B$10000
*0/800<FD/8000.FFFFM
*BSAVE APPLE2GS.ROM2,A$800,L$8000,B$18000
*0/800<FE/0.7FFFM
*BSAVE APPLE2GS.ROM2,A$800,L$8000,B$20000
*0/800<FE/8000.FFFFM
*BSAVE APPLE2GS.ROM2,A$800,L$8000,B$28000
*0/800<FF/0.7FFFM
*BSAVE APPLE2GS.ROM2,A$800,L$8000,B$30000
*0/800<FF/8000.FFFFM
*BSAVE APPLE2GS.ROM2,A$800,L$8000,B$38000

```

You now have an APPLE2GS.ROM or APPLE2GS.ROM2 file for use with Sweet16. Transfer this file to your Macintosh using FTP, email, or a utility like ADTPro (<http://adtpro.sourceforge.net/>) and put it in a convenient location on your Mac's hard disk.

☆ Tip ☆

It doesn't matter much whether you use a ROM 01 or a ROM 3 image, most of the time. There are a few programs that only work with one or the other, and [a few minor differences](#) in functionality between the two, but unless you have specific needs, either will do. You cannot, however, use a ROM 00 image.

Installing Sweet16

Sweet16 comes on a standard Mac disk image file. Simply mount the disk image, then drag the Sweet16 application file to your Applications folder, or to anywhere else you'd like to put it on your Mac. After that, you can eject the disk image.

Working With Disk Images

Instead of working with disks and drives like you would on a real Apple IIgs, Sweet16 uses **disk images**. This part of the guide covers things you can do with disk images in Sweet16.

Creating Disk Images

Unless you're only working with floppy disk images, or are starting up from a pre-made hard disk image (such as the one that comes with the Sweet16 Starter Kit), one of the first things you'll do with disk images in Sweet16 is to create one to use as your startup disk.

Creating a disk image is easy. Start by choosing the "Create New Disk Image..." option in Sweet16's File menu. You'll be presented with a dialog like this one, in which you configure and name your new disk image:



In addition to naming your image file, you have some options for configuring the image. First, you may choose the image format. This may be one of:

- **DiskCopy 4.2:** This is an older Apple image format which may only be used for floppy disk sized images (that is, 800 kB and 1.4 MB sizes). This format has the advantage of being checksummed; that is, images are protected against unnoticed errors by checking their contents against a calculated value based on those contents, but the format is very old and not used much anymore.
- **DiskCopy 6:** This is a newer Apple image format. This is a specific subset of Apple's current disk image file format which, on the Mac, is now known as the "read/write DiskCopy" format.
- **Universal Disk Image.** This format, also known as ".2mg" format, is a newer format specifically created for use in Apple II emulators. While it's theoretically the best format

for use in emulators, its incompatibility with other kinds of Mac software can be a drawback at times. In general, though, this is probably your best choice for ProDOS-formatted hard disk images.

- **ProDOS 5.25" Disk.** This is also known as ".po" format. This is a raw image format specifically intended for Apple II 5.25" disk images. While some emulators let you use this format for disks other than 5.25" images, Sweet16 only allows creating 5.25" images in this type (although it will read them in other sizes).

Once you've chosen your image file format, you need to choose the size of image to create from the popup menu of size options. Some image formats (DiskCopy 4.2 and ProDOS 5.25" Disk in particular) have restrictions on what sizes are permitted; other options will not be available. Otherwise, your size options are:

- 140K (standard 5.25" disk size)
- 800K (standard double-density 3.5" disk size)
- 1.4 MB (standard high-density 3.5" disk size)
- 5 MB
- 10 MB
- 20 MB
- 32 MB (the maximum size possible for ProDOS disks)
- 100 MB

Once you've given your new image a name, and have selected the image format and size, click "Create" to save your new image file.

☆ Tip ☆

When Sweet16 starts up, it automatically mounts all the images in the ~/Library/Application Support/Sweet16/Disk Images folder, in alphabetical order. If you make sure that the first image in that folder is a bootable one, your emulated IIgs will start up from that. That makes this folder the ideal place to save your new disk images a lot of the time.

Once the new disk image has been created, it will automatically mount into Sweet16. Sweet16 automatically formats all new floppy disk (1.4 MB or smaller) images as ProDOS disks. Your Apple IIgs will prompt you to format any other new disk images. And, of course, you can reformat floppy disk images to other formats if you wish.

The Mounted Disks Window

Choosing the “Sweet16 Disks” option in the Window menu opens Sweet16’s mounted disks window, which looks like this:



Each disk that’s currently mounted is listed with its image file’s name, the type of disk image file, and the size of the disk. The disk activity indicator light on each drive lights up to indicate activity. In addition, if the disk is locked, a small padlock icon is displayed in the lower-right corner of the drive’s icon.

Right-clicking on a disk in the mounted disks window presents a contextual menu with some options that you can use to manage your mounted disks:

- **Eject Disk:** Ejects the disk you right-clicked on.
- **Lock Disk:** Locks the disk you right-clicked on, making it read-only.
- **Eject All Disks:** Ejects all mounted disks.
- **Reveal in Finder:** Opens a Finder window with the disk image file you right-clicked on selected in it.

Drives numbered 1 through 12 are SmartPort devices; typically, these are hard disk images.

Drive numbers 13 and 14 are 3.5” floppy disk drives. Drive numbers 15 and 16 are 5.25” drives.

Mounting Disk Images

Mounting a disk is a more general way to say “inserting a disk” or “making a disk available”. There are several ways you can mount disk images in Sweet16:

- Drag one or more disk images onto the Sweet16 icon in the dock.
- Drag one or more disk images into Sweet16’s mounted disks window.
- Choose the “Mount Disk Image...” option in Sweet16’s File menu, then select the image file you want to mount.
- Choose an image file from the “Recent Disks” submenu in Sweet16’s File menu. This menu is automatically populated with your most recently-used disk image files.

☆ Important ☆

It’s extremely important not to mount the same disk image in two places at once: don’t mount the same image on both the Mac and the emulated Ilgs, for example, or in two different emulators at once. If two programs write to the same disk image at the same time, the contents will almost certainly become corrupted.

When Sweet16 mounts a disk image, it chooses the most appropriate available device to mount it into. 800 kB and 1.4 MB images are mounted in the 3.5” drives by default. 140 kB images are mounted in the 5.25” drives. All other image sizes are mounted as hard disks. If an appropriate open device isn’t available, an alert appears letting you know that the disk couldn’t be mounted.

☆ Tip ☆

When mounting a floppy disk image, you can mount it as a hard disk by holding down the Option key on your keyboard while doing so. By doing this, you can mount more of these disks at once.

This is very useful when installing multi-disk software (such as the six-disk Apple Ilgs System 6.0.1), or when playing multi-disk games, like *Space Ace*.

In addition, you can configure Sweet16 to default to mounting 3.5” disks as hard disks in the Preferences window’s storage panel.

When mounting 5.25” disk images, remember that the Apple Ilgs doesn’t necessarily automatically detect their availability. For example, in the Finder, you’ll need to double-

click on the .APPLEDISK5.25A or .APPLEDISK5.25B icon on the desktop to actually let the computer know there's a disk in the drive.

Automatically Mounting Disk Images

Usually you will have disks that you always want to have mounted when you first start Sweet16, such as your boot disk. Sweet16 will automatically mount any disk image files that are located in your Library/Application Support/Sweet16/Disk Images folder when it starts up. Each device is mounted, in alphabetical order, into an appropriate device. The first device mounted into the slot configured as your boot slot in the Apple IIgs control panel will be your startup disk.

As a general rule, you should make sure that your boot disk is the first image, alphabetically, in the disk images folder.

You can easily open the Disk Images folder for copying image files into and out of by choosing the "Open Disk Images Folder" in the File menu.

Mounting CD-ROM Media

Sweet16 lets you mount CD-ROM media into your emulated Apple IIgs. Simply insert the CD-ROM into your Mac, then choose the "Mount CD-ROM" option in the Sweet16 File menu to insert it into the Apple IIgs.

Sweet16 will then mount up to nine partitions from the CD into the emulated Apple IIgs. Any of those partitions for which you have File System Translators (FSTs) installed on the Apple IIgs will appear on your Apple IIgs desktop. This includes ProDOS, HFS, and ISO 9660 (High Sierra) formats.

When you eject the last CD-ROM partition from the Apple IIgs, the disc will be physically ejected from your Macintosh.

☆ Tip ☆

If you plan to use CD-ROM media on your Apple IIgs, be sure to install the appropriate FSTs onto your Apple IIgs startup disk. In particular, you'll probably want the High Sierra (ISO 9660) FST; this is the standard CD-ROM file system.

Ejecting (Unmounting) Disk Images

Similarly, there are several ways you can eject a disk on Sweet16. The best way is always to use the eject option provided by the operating system or your utility software on the emulated Apple IIgs, but there are also ways to do it directly within Sweet16's user interface.

- Drag the disk to the trash (or choose the Eject option) in the Apple IIgs Finder.

- Choose a disk to eject from the Eject submenu in Sweet16's File menu.
- Right-click on a disk in the online disks window in Sweet16, then choose the "Eject Disk" option from the disk's contextual menu.

There are also two ways to eject all disks at once:

- Right-click in the online disks window and choose the "Eject All Disks" option.
- Choose "Eject All" from the "Eject" submenu in Sweet16's File menu.

Supported Disk Image Formats

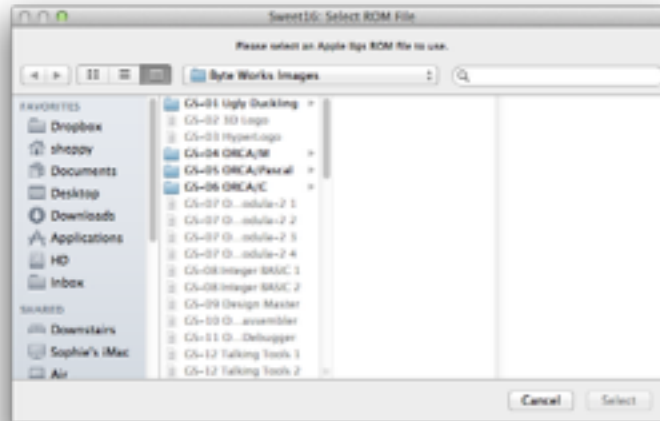
Sweet16 supports a number of popular disk image formats. Here's a look:

Device Type	Image Format	File Extension	Supported Sizes
3.5" disk	DiskCopy 4.1/4.2	.img, .image	800 kB, 1.4 MB
	DiskCopy 6	.dmg	800 kB, 1.4 MB
	Hard drive	.hdv	800 kB, 1.4 MB
	Raw data	.raw	800 kB, 1.4 MB
	Universal Disk Image	.2mg	800 kB, 1.4 MB
5.25" disk	DSK	.dsk	140 kB
	DOS 3.3 order	.do	140 kB
	ProDOS order	.po	140 kB
Hard disk	DiskCopy 6	.dmg	Any
	Hard drive	.hdv	Any
	Raw data	.raw	Any
	Universal Disk Image	.2mg	Any
	DOS 3.3 order	.do	Any
	ProDOS order	.po	Any

While Sweet16 supports mounting hard disk images in .do and .po format, it doesn't let you create them. These formats are intended for use as floppy disk images.

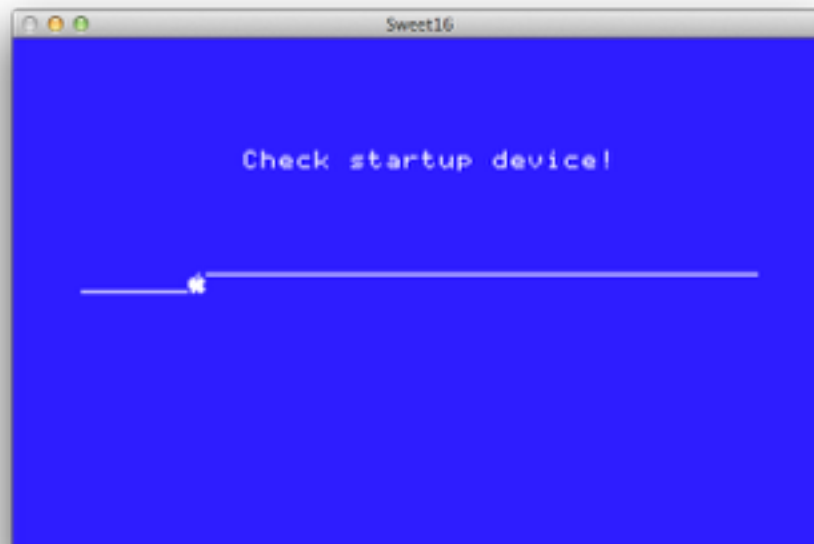
Starting Up For the First Time

Once you have a ROM image, you can enter the Apple II world by launching the Sweet16 application. You'll be prompted to choose a ROM file to use.



Locate the ROM file you created in the section “Creating a ROM Image File”, then click “Select”.

At that point, a large window will appear, containing the Apple IIgs video display. This screen should flash to blue, the Apple IIgs power-up screen should appear, the Apple IIgs system beep will sound, and then, after a moment, you should see the famous sliding apple screen:



This indicates that the Apple IIgs is successfully running... although it's not doing much yet.

If you don't get the sliding apple screen, then your ROM image is probably damaged. Try repeating the steps in "Creating a ROM Image File", carefully.

The sliding apple screen indicates that while the Apple IIgs is running correctly, you don't have any disks in any of the emulated drives from which it can start up. Getting some is the next step.

Getting Startup Disks

You can legally download copies of the Apple IIgs system disks from Syndicomm™ at <http://syndicomm.com/apple-ii-system-software/>. You can quickly get to this page by choosing "Download Apple II System Software" from Sweet16's Help menu.

There's an even easier way, though. Ewen Wannop provides a downloadable "Sweet16 Starter Kit" you can use. This includes a startup disk image you can boot from which not only has the latest Apple IIgs system software on it, but also some essential utilities, as well as an assortment of his own software.

Downloading the Sweet16 Starter Kit is as easy as choosing the "Download Sweet16 Starter Kit" option from the Help menu. Once you've downloaded the Starter Kit, check out its excellent documentation for details on what's included and how to use it as your Sweet16 startup disk.

Installing an Operating System

Installing system software works almost exactly as it does on a real Apple IIgs computer. You boot from the system software install disk and follow the same process as usual. The only difference is that you need to make sure you create a destination disk image (typically a 32 MB Universal Disk Image format disk) to install onto, then mount it, following the instructions in "[Creating Disk Images](#)".

☆ Tip ☆

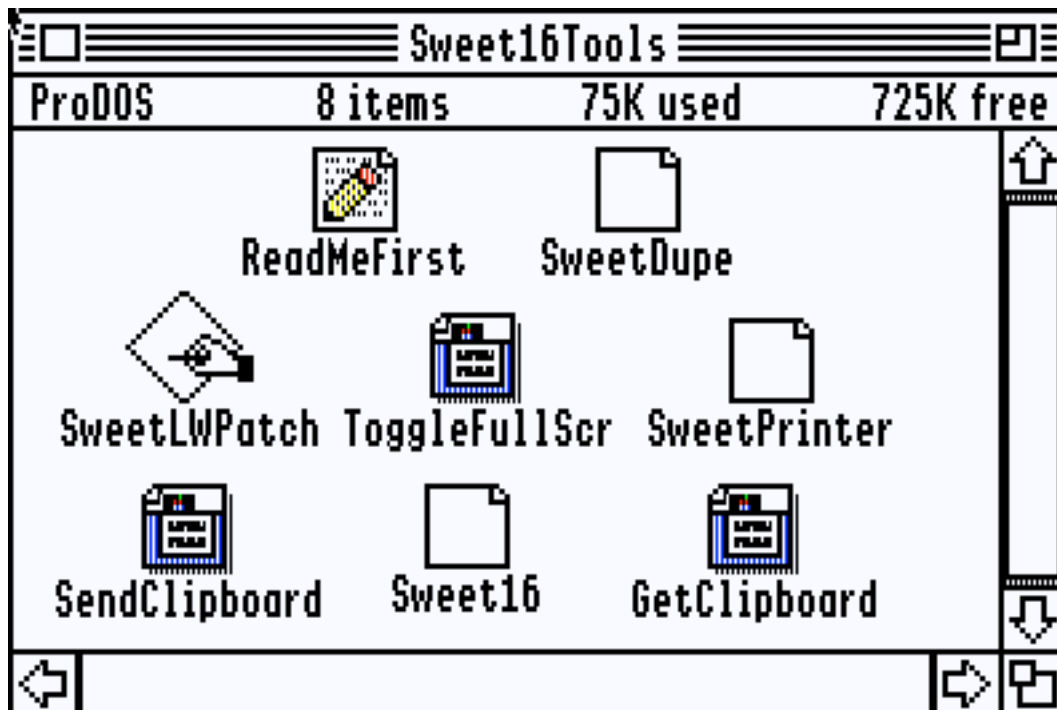
When installing Apple II software that comes on multiple disks (or disk images)—such as the system software—it's useful to know that Sweet16 lets you mount (or insert) more than two floppy disk sized images at a time.

Just hold down the Option key while you mount them and they'll be mounted as if they were tiny hard disks instead of as floppy disks. That lets you have up to a dozen of them mounted at once!

Enabling Platform Integration

If you've installed Apple IIgs System Software 6.0.1, you most likely will want to enable the special features available only when running under Sweet16. To do so, you'll need to install some optional system software add-ons. This section covers that.

The Apple IIgs system software add-ons that enable Sweet16 integration features are provided on a disk image that's integrated into Sweet16. You can mount it onto your emulated Apple IIgs by selecting the "Mount Tools Disk" option in Sweet16's Setup menu. This will mount a disk image called Sweet16Tools onto the IIgs desktop. In the Finder, it'll look something like this:



You don't **need** to install any of these, and some are more optional than others. Let's take a look at what each item is.

ReadMeFirst

This read-me document provides some information about the items on the tools disk.

SweetDupe

This extension provides support for dragging and dropping files between the Apple IIgs Finder and the Mac Finder. If you want to use that feature, you will need to install this extension. To do so, place a copy of it in your boot disk's System:System.Setup folder and restart the emulated Apple IIgs.

See "Moving Files Between the Apple IIgs and the Mac" for details on this functionality.

SweetLWPatch

This application is used to create a copy of the Apple LaserWriter driver that supports printing from Apple IIgs desktop applications through the Mac. To use this utility, follow these steps:

1. Use the Apple IIgs system software install disks to install the Apple LaserWriter driver onto your startup disk.
2. Run the SweetLWPatch utility.
3. Select the LaserWriter driver from your boot disk's System:Drivers folder; this folder should be displayed by default in the standard file chooser dialog box that appears on launch.
4. Enter a name for the patched copy of the driver. I recommend "SweetPrinter".
5. Once you've saved the patched driver, you can quit the SweetLWPatch utility.

At this point, you have a patched driver, but you're not ready to print yet. You'll also need to install the SweetPrinter extension; see below for details. Then see "Printing" for details on printing in Sweet16.

ToggleFullScr

This is a desk accessory that toggles Sweet16 between full screen mode and running in a window. If you wish to use it, place a copy of the file in your boot disk's System:Desk.Accs folder and restart the emulated system. Having done so, you'll see a new "Toggle Full Screen" option in the list of desk accessories in the Apple menu of your Apple IIgs desktop applications.

Choosing that option will switch you back and forth between Sweet16 running in a window and running on the full screen.

SweetPrinter

This extension provides a wide variety of Mac integration functionality. Once installed in your boot disk's System:System.Setup folder, this provides support for:

- In concert with the "SweetPrinter" driver, adds printing support.
- Adds a "Quit" button to the "It is now safe to turn off your Apple IIgs" shut-down dialog box.
- Adds support for using your Mac's mouse wheel in Apple IIgs applications.
- Lets Sweet16 show the name of the currently running Apple IIgs application in the title of Sweet16's main window.
- Provides other system software interfaces used by other Sweet16 add-ons.

Sweet16

This file is the Marinetti **link layer** for Sweet16. If you have the Marinetti TCP/IP stack installed, placing a copy of this file into the System:TCPIP folder will add support for TCP/IP networking to Sweet16. See “Networking” for details.

GetClipboard

This desk accessory (once installed into the System:Desk.Accs folder on your boot disk) will pull the current text clipping from the Mac into the Apple IIgs; this lets you copy and paste text from the Mac into Apple IIgs desktop applications.

☆ Important ☆

Unfortunately, AppleWorks GS does not use a standard text scrap, so Sweet16's clipboard support features do not work with AppleWorks GS.

In addition, due to other quirks in how AppleWorks GS is implemented, its user interface gets confused if you try to use SendClipboard and GetClipboard in AWGS, so you shouldn't try to do so.

Once you've installed any (or all) of these system integration add-ons, be sure to restart the emulated Apple IIgs to allow them to load up.

SendClipboard

This desk accessory (once installed into the System:Desk.Accs folder on your boot disk) will send the copy of the standard Apple IIgs clipboard to the Mac. This only works for text clippings at this time.

☆ Important ☆

Unfortunately, AppleWorks GS does not use a standard text scrap, so Sweet16's clipboard support features do not work with AppleWorks GS.

In addition, due to other quirks in how AppleWorks GS is implemented, its user interface gets confused if you try to use SendClipboard and GetClipboard in AWGS, so you shouldn't try to do so.

One more limitation to the SendClipboard and GetClipboard desk accessories: if there are any unexpected characters on the clipboard, the Apple IIgs may crash. Be sure that you only copy and paste text that uses the so-called MacRoman character set. This is

an extended form of ASCII that provides a number of special characters for a variety of western European languages.

Moving Files Between the Apple IIgs and the Mac

If you've installed both the SweetPrinter extension and the SweetDupe extension (see "[Enabling Platform Integration](#)"), copying files between the emulated Apple IIgs and your Mac is literally as easy as dragging them from one to the other.

To copy one or more files or folders from the Apple IIgs Finder to the Mac, select them in the Apple IIgs Finder and drag them out of the Apple IIgs video window to the destination folder window on your Mac.

☆ Important ☆

Items you wish to copy to the Mac may not be on the Finder desktop. They must be in a window.

Similarly, to copy files or folders from the Mac to the Apple IIgs, drag them from the Mac Finder to a Finder window on the emulated Apple IIgs.

If you don't drop the files on a Finder window, or you drop them in another application, a standard file dialog box will appear to ask you where to put the files and folders.

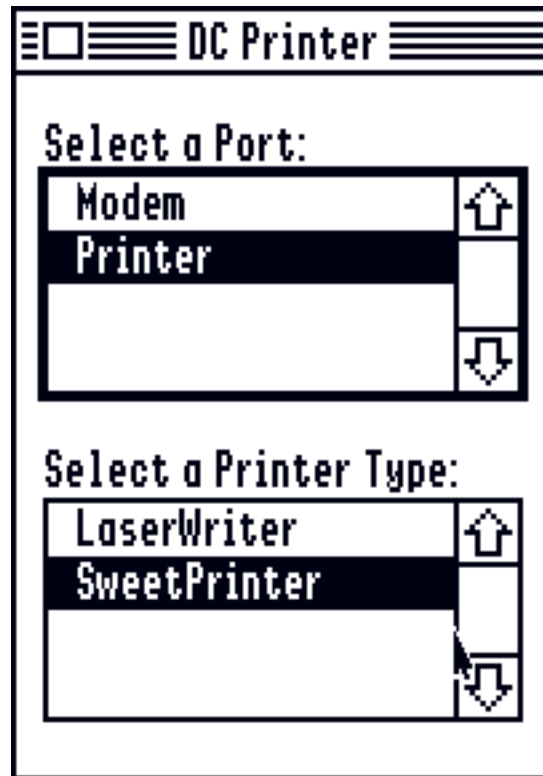
☆ Important ☆

Currently, Sweet16 can only receive dragged files when desktop applications are running on the emulated Apple IIgs.

Printing

If you've created the SweetPrinter driver and installed the SweetPrinter extension (see "[Enabling Platform Integration](#)"), you can print from Apple IIgs desktop applications to printers connected to your Macintosh. You can even print to PDF files! This works from most desktop applications, including AppleWorks GS and GraphicWriter III.

First, you need to select the SweetPrinter driver. To do so, open the Control Panels desk accessory from your Apple IIgs Apple menu and open the DC Printer control panel. You should see something like this:



Be sure to select “SweetPrinter” as the printer type. Once you’ve done this, any time you print from an Apple IIgs desktop application that uses the standard Print Manager print system, the output will be sent through to the Mac’s Preview application. Once the file is in Preview, you can print it, save it as a PDF, fax it, or otherwise manipulate it as you see fit.

This works by capturing the PostScript output from the patched LaserWriter driver and copying it to a temporary file on the Mac, which is then opened in Preview. Preview translates the PostScript into PDF format and presents it to you in a window.

Networking

While Sweet16 doesn’t support AppleTalk, you can use the free Marinetti TCP/IP system to add networking support to your emulated Apple IIgs. Visit <http://www.apple2.org/marinetti/> to download Marinetti, and for details on its use.

Once you’ve installed Marinetti, and have also installed the Sweet16 link layer as covered in “[Enabling Platform Integration](#)”, you can use Apple IIgs programs that rely on Marinetti for networking, just as you would on a real Apple IIgs.

To do so, open the TCP/IP control panel and select the “Sweet16” link layer.

Setup connection

Username: Primary Domain Name Server IP address:

Password: Secondary Domain Name Server IP address:

☐ Keep network alive every minutes ☐ Show connect messages

☒ Connect to network when GS/OS boots

Link layer:

You may also optionally choose to connect to the network automatically when GS/OS starts up, as shown in the screenshot above.

You don't need to set the Apple IIgs's IP address, configure DNS servers, or the like. Sweet16 provides an internal DHCP server and router. The Apple IIgs connects to that, and Sweet16, in turn, routes the transactions to other computers on the Internet.

☆ **Tip** ☆

It's perfectly normal for the IP addresses reported by the TCP/IP control panel on the Apple IIgs not to match numbers on your local network. These values are on Sweet16's internal, simulated network.

Once you've done all of this, you should be able to use almost any Apple IIgs program that uses Marinetti for networking.

☆ **Important** ☆

The method Sweet16 uses for enabling networking by creating a virtual router does not currently support "listening" for incoming network connections. That means you can't run servers (such as the Silver Platter web server) on your Apple IIgs.

That also means you'll need to use passive mode in FTP programs rather than the default active mode.

Using Emupacks

To make it easy for you to use multiple configurations of Sweet16, Sweet16 provides support for **Emupacks**. These are packages that bundle together a collection of disk images, preferences, battery RAM (that is, system settings from the Apple IIgs control panel), and Sweet16 preferences.

The best part: everything you do while running Sweet16 when launched from an Emupack is contained within the Emupack. Changes to Sweet16 preferences, control panel settings, and the contents of your disk images will be kept inside the Emupack. That makes it a great way to set up custom, launchable configurations of Sweet16 for sandboxing different tasks you do with your Apple IIgs.

Creating an Emupack

To create an Emupack, start by configuring Sweet16 just the way you want it:

- Mount any disk images you want. The Emupack you create will embed copies of the images that are listed in the mounted disks window (even if they're not currently visible in the IIgs Finder for any reason).
- Configure your Sweet16 preferences the way you want Sweet16 to behave when you start up using the Emupack.
- Use the Apple IIgs control panel to configure any Apple IIgs settings the way you want them to default when using the Emupack.

Once everything is set up just the way you want it, choose "Create Emupack..." in the Setup menu and save the new Emupack.

Using an Emupack

To use the saved configuration, you start up Sweet16 by double-clicking the Emupack in the Mac Finder. This will launch Sweet16 as configured.

There's an extra bonus to using Emupacks. Normally, you can only have one emulated Apple IIgs running at a time. However, if you install the "Open In New Sweet16 Instance" service included with Sweet16, you can have multiple sessions of Sweet16 at once, each running its own Emupack!

To install the service, find the file "Open In New Sweet16 Instance.workflow" on the Sweet16 install disk image and copy it to your ~/Library/Services folder.

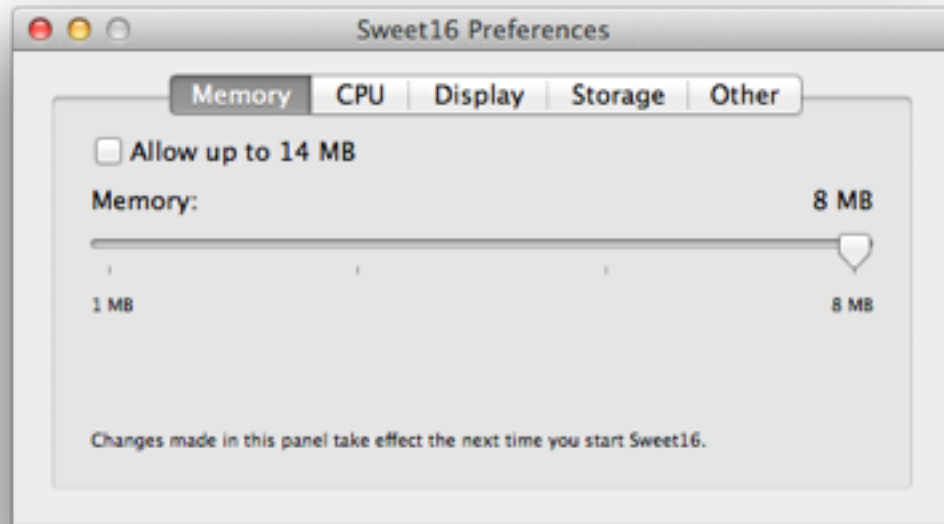
Then, you can open Emupacks in new Sweet16 instances by simply right-clicking on them and choosing "Open In New Sweet16 Instance" from the Services submenu (or from the Services submenu in the "Finder" menu).

Preferences

You can customize your Sweet16 experience a great deal using the preference window. The preferences are divided across five panels, depending on what they control.

Memory

The Memory panel lets you adjust the size of the Apple IIgs memory.



Allow up to 14 MB

This option, when enabled, patches the Apple IIgs ROM to let you use up to 14 MB of memory. You also have to actually select 14 MB using the memory size slider below.

Memory

This slider lets you select the amount of memory installed in the virtual Apple IIgs. This may range between 1 and 8 MB normally, or up to 14 MB if the “Allow up to 14 MB” option is enabled.

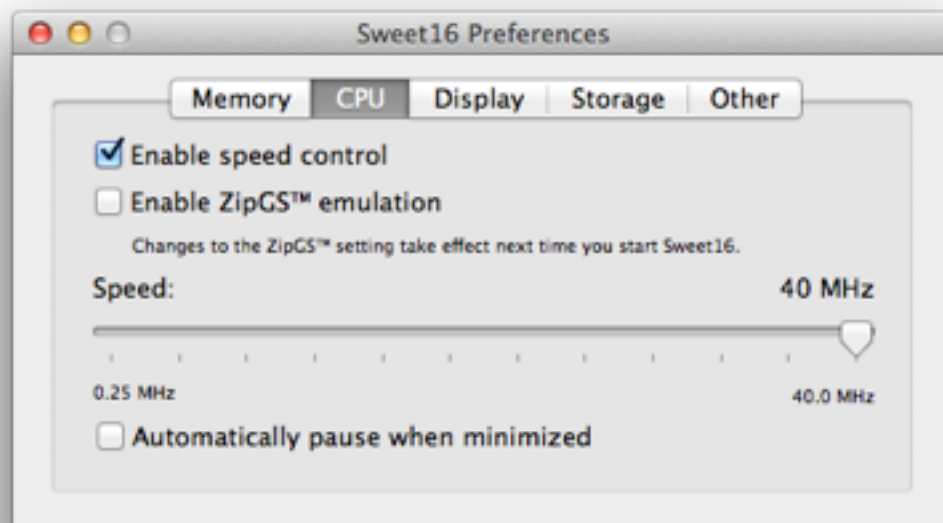
Changes to any of the options in this panel require a restart of Sweet16 before they take effect.

☆ **Important** ☆

A small number of applications do not work correctly with the “Allow up to 14 MB” option enabled, even if you have less than 14 MB allocated.

CPU

The CPU panel lets you configure the CPU emulation and related features.



Enable speed control

When this option is enabled, Sweet16 will do its best to make the emulated Apple IIgs run at specific speeds. If this option is disabled, Sweet16 will run as fast as it can on your Mac.

Enable ZipGS™ emulation

When this option is enabled, the emulated Apple IIgs acts as if it has a ZipGS™ accelerator card installed. This includes support for the fine-grained speed controls offered by this card. Changes to this option do not take effect until you quit and restart Sweet16.

Speed

This slider lets you set the speed of the emulated Apple IIgs. You may choose among an assortment of speeds between 0.25 MHz (a quarter of “normal” Apple II speed) all the way up to 40 MHz.

Automatically pause when minimized

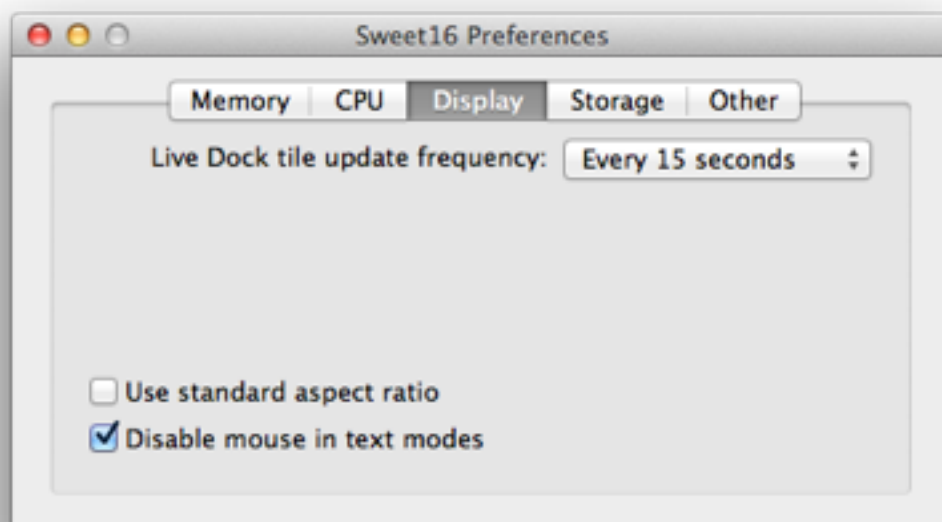
This option, when enabled, causes Sweet16 to automatically pause when you minimize Sweet16 into the Dock.

☆ Tip ☆

While it's fun to run your Apple IIgs at ludicrously high speeds, it can also reduce battery life on your laptop, and cause your laptop to get very warm. Usually it's best to enable speed control on laptops.

Display

The Display panel offers features for configuring the Apple IIgs display emulation.



Live Dock tile update frequency

Specifies how often Sweet16's Dock icon should update to reflect the state of the Apple IIgs screen.

Use standard aspect ratio

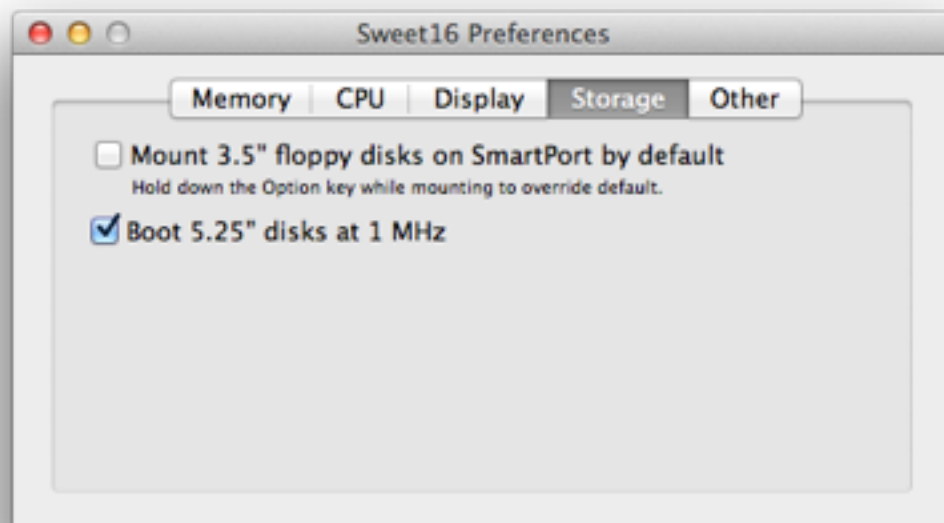
By default, Sweet16 uses a video window whose aspect ratio “compresses” the video vertically by a small amount, using square pixels. Enabling this option stretches the video window vertically to the normal Apple IIgs aspect ratio but results in slightly fuzzy pixels.

Disable mouse in text modes

When this option is enabled, the mouse is disabled while the Apple IIgs is in text mode. This can be handy if you use Apple II applications that try to support the mouse in text mode but don't do it very well. That can be particularly frustrating if you have a sensitive trackpad or high-speed mouse.

Storage

The Storage panel lets you configure storage options.



Mount 3.5" floppy disks on SmartPort by default

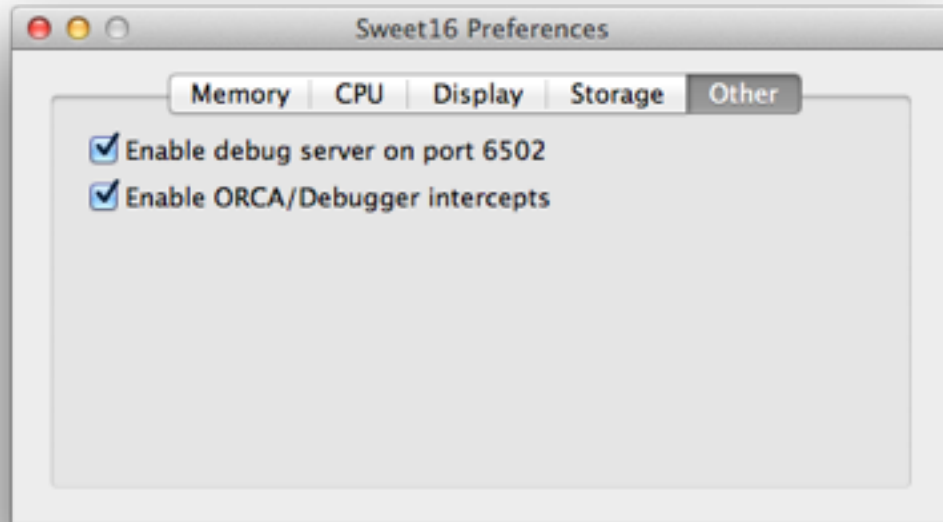
By default, 3.5" floppy disk images mount in the 3.5" drives in slot 5, where they act like floppy disks. You can enable this option to mount them on the SmartPort, as if they were hard disks, by default. In either case, holding down the Option key while mounting 3.5" floppy images reverses this setting.

Boot 5.25" disks at 1 MHz

Enabling this option causes the Apple IIgs to automatically slow to "Normal" speed—that is, the 1 MHz speed that 8-bit Apple II computers historically run at—when you boot from 5.25" disk images. This is handy when playing older games that are unplayably fast at higher speeds.

Other

The Other panel provides configuration options that don't fit into the other categories; these are mostly advanced options for developers, such as those controlling the integrated debugger features.



Enable debug server on port 6502

This enables the debugger features by activating the debug server, using network port 6502 on your Mac. See [“The Sweet16 Debugger”](#) for details.

Enable ORCA/Debugger intercepts

ORCA/Debugger by The Byte Works™ is an Apple IIgs debugger that uses the 65816 COP instruction to insert breakpoints and other information into program code. Sweet16 can intercept these instructions to use this information in its own debugger.

Menu Reference

Here you'll find an overview of all the menu options in Sweet16's menu bar and what they do.

File menu

Open Disk Images Folder

Opens Sweet16's Disk Images folder in the Mac Finder. This makes it easy to add images that will automatically be mounted when Sweet16 starts up. Just drop those images into this folder. Keep in mind that your boot disk will be the one whose name is first, alphabetically, in the directory.

Create New Disk Image...

Use this option to create a new disk image for Sweet16. See [“Creating Disk Images”](#) for details on how to use this feature.

Mount Disk Image...

Lets you select a disk image file to mount on your emulated Apple IIgs.

Mount CD-ROM

Attempts to amount the disk in your CD-ROM drive on the emulated Apple IIgs. See [“Mounting CD-ROM Media”](#) for details.

Recent Disks

The Recent Disks submenu lists the most recently accessed disks, so you can quickly get at them again. You can choose “Clear Recent Disks” to remove all disks from this list.

Eject

The Eject submenu lets you quickly eject one of the currently-mounted disks. Empty devices are grey and can’t be selected.

Close

Closes the frontmost window on the Mac; used for the mounted disks, preferences, and about windows primarily.

Edit menu**Paste**

Pastes the text currently on the Mac clipboard to the Apple IIgs by pretending to type it.

Capture Text Screen

Captures a copy of the text on the Apple IIgs’s current text screen to the Mac clipboard.

Capture Screen as Image

Copies the current Apple IIgs screen to the Mac clipboard as a graphic image.

Setup menu**Joystick Support**

Turns joystick support on and off. Sweet16 supports USB joysticks and gamepads. It will first look for a joystick; if one isn’t found, it will then look for a gamepad as a backup.

Mouse Support

Toggles mouse support on and off. See [“Display Preferences”](#) for a way to automatically disable the mouse in text modes.

Mute Sound

Toggles audio on and off.

Pause

Pauses the Apple IIgs. You can tell the Apple IIgs is paused because the screen is dimmed.

Reset

Simulates pressing Control-Reset.

Restart

Restarts the Apple IIgs; this simulates pressing Control-Command-Reset.

Open CDA Menu

Opens the Apple IIgs Classic Desk Accessories menu.

Mount Tools Disk

Mounts the integrated disk image that contains the Apple IIgs system extensions and desk accessories provided to help improve your Sweet16 experience. See “[Enabling Platform Integration](#)” for details on installing and using these features.

Choose ROM File...

Lets you select a new Apple IIgs ROM Image file to use with your Apple IIgs. This change will not take effect until you quit and restart Sweet16.

Save Emupack...

Saves the current Apple IIgs configuration (including control panel settings, ROM, mounted disks, and Sweet16 preferences) as an “Emupack.” This provides an all-in-one, wrapped package you can use to start a Sweet16 session. See “[Using Emupacks](#)” for details.

View Menu**Half Size**

Resizes the Apple IIgs video window to half its normal size.

Normal Size

Resizes the Apple IIgs video window to its normal size.

Double Size

Resizes the Apple IIgs video window to double its usual size.

Full Screen

Toggles Sweet16 between running in a window and using the full screen. You can press Control-Option-F to toggle back to running in a window. You can also use the full screen desk accessories provided for this.

Show System Speed

When enabled, this causes the current emulated Apple IIgs CPU speed, in megahertz, to be displayed in the top-right corner of the video window.

Window Menu

Minimize

Minimizes the frontmost window to the Dock.

Minimize All

Minimizes all Sweet16 windows to the Dock.

Zoom

Toggles between half-size and normal size video display.

Bring All to Front

Brings all Sweet16 windows in front of windows belonging to other applications.

Launch Debugger

Opens the built-in Sweet16 Debugger in your web browser. This requires that you enable the debugger functionality in the preferences; see the “[Other](#)” panel in the preferences documentation.

Online Disks Window

Toggles the Sweet16 disks window on and off.

Other Sweet16 windows will be displayed here as well, if they’re open.

Help Menu

Sweet16 User Guide

Opens this user guide.

Sweet16 Web Site

Opens the Sweet16 Web site in your browser.

Apple II Information Wiki

Opens the apple2.info web site, which is a great source of Apple II information.

A2Central.com: Apple II News

Opens the A2Central.com web site; this is a great Apple II news and information site.

Download Apple II System Software

Takes you to Syndicomm’s Apple II system software download page, where you can get system disk images for use in Sweet16.

Download Sweet16 Starter Kit

Downloads Ewen Wannop’s Sweet16 Starter Kit.

GNO Apple II File Archive

Takes you to the GNO file archive, where you can download an assortment of freely-distributable Apple II software.

Marinetti: TCP/IP Networking

Takes you to the web site for the Marinetti TCP/IP suite for the Apple IIgs. You'll need this if you want to use networking features.

The Sweet16 Debugger

Sweet16 offers support for a debugger protocol which can be connected to using modern Web browsers for controlling the Apple IIgs for the purpose of debugging code running on it.

☆ Tip ☆

Please note that the debugger is not finished yet. Expect to see a few 3.x releases of Sweet16 that enhance the functionality of the debugger. It was a late addition to Sweet16 3.0 and was not as finished as I'd like at launch.

If enabled (see the section on the [“Other”](#) panel), Sweet16 provides a debug server. This server uses WebSocket technology to offer a JSON-based network service on port 6502 for interacting with external debuggers or other utilities. A separate document describes this JSON-based API.

In general, however, the debugger API lets external programs access information about the state of the emulated Apple IIgs, fetch instructions, step the processor, and so forth.

This API can be used by any application you'd like to use to interface with Sweet16; however, Sweet16 also includes a built-in debugger, which is provided by connecting to <http://localhost:6502/index.html> in your Web browser. You can also get a PNG-format screen shot of your screen by accessing <http://localhost:6502/currentscreen.png>.

It's worth noting that this connection isn't limited to localhost; any computer that can access your computer can access the debugger's features.

There's also a silly sample program at <http://localhost:6502/share/index.html> which uses the debugger API and the screen shot capability to let you interact with the emulator from your Web browser. A special interface is presented to make it a little easier to use on mobile devices, as well; however, it's not pretty, and is very much a hack demo, so please don't try to use it for any real work.

The documentation for the debugger is intentionally sparse right now, as I expect it will change pretty rapidly as I continue to work on it.

At this time, the debugger offers the following features:

- You may pause and resume the emulator using the pause/resume button at the top-right corner of the debugger.

- Although the debugger connects automatically to Sweet16 running on your current computer by default when you open it, you may connect to Sweet16 on another computer by entering its hostname or IP address in the “Server” box at the top-right of the screen and clicking “Reconnect”.
- The “Emulator Info” button opens a dialog box with some basic information about the copy of Sweet16 you’re connected to, including its version number and build date and time.
- You can output text from Apple IIgs code to the console area of the debugger by loading a pointer to a C-style (null-terminated) string into the X and Y registers and then issuing a `COP $84` instruction. Be sure to include a linefeed (`$0A`) character to end lines if you want to start a new one. Each output is timestamped if it starts a new line.
- You can add and delete breakpoints using the interface along the left side of the debugger panel. Each breakpoint may have a short descriptive name to help you remember what it’s for. The “Enabled” button lets you toggle all breakpoints on and off. When a breakpoint is reached, you’ll see the instruction on which you broke in the code panel, highlighted in light green, as well as upcoming instructions in a light gray text.
- The debugger will also be activated on `BRK` instructions, as well as on ORCA breakpoints (if ORCA/Debugger intercepts have been enabled in the preferences).
- In addition, you can immediately enter stepping mode by clicking the “Break” button in the debugger’s code panel toolbar.
- Along the top of the debugger interface is a display of the values of all of the 65816 registers. These are only valid while stepping through code. When an instruction is executed while stepping that affects the value of a register, that register’s value is updated, and the register’s value turns red until the next instruction is run.
- You also see the individual bits of the processor status indicated with little boxes.
- The console box has a Clear button that erases the contents of the console area.
- Below the console box is the code panel. This is where single-stepping through the code occurs. To step, click the “Step In” button (there is no “Step Over” yet, but there will be). To skip the current instruction, click the Skip button. To resume normal execution, click “Continue.”
- Along the right edge of the debugger interface is the stack panel. While stepping through code, this shows you the current stack, with the address the stack pointer is currently pointing at at the top of the panel. Keep in mind that the stack pointer points one byte **past** the last byte on the stack!

A few special notes:

- At this time, there are no keyboard equivalents for the debugger interface. There will be.
- Clicking “Skip” when on a `BRK` instruction actually currently steps on into the `BRK` vector code instead of skipping the `BRK` entirely. This will be fixed soon.
- There’s no memory viewer yet. There will be.
- You can’t edit the contents of registers, memory, or the stack yet. That will change, too.

In general, there are some very neat things coming!

Release Notes

The following is a most-recent to oldest version history of Sweet16, with the new changes listed for each release.

Version 3.0.3 (April 6, 2014)

- All-new About box, rewritten in Cocoa, with additional features including license information, hexadecimal build number for added nerdiness, and more.
- Fixed bug in About box that could cause a crash if the build number couldn't be retrieved for some reason.
- Updated the code that draws the note in full-screen mode about how to get out of full screen mode to stop allocating new texture buffers 30x/sec without ever deallocating them. It only needed to do this once in the first place, and this bug was causing a massive leak of memory while in full-screen mode. Fixes [bug 158](#).
- Rebuilt the internal utilities that are bundled into the Sweet16 application that are used by the file drag-and-drop support; these were previously compiled to require a full 64-bit processor, and therefore would not work on an Intel Core Duo CPU (which is the oldest CPU on which Sweet16 can still be used).
- Updated a macro that auto-generates a lot of the emulation code to eliminate huge numbers of code warnings and in theory some possible run-time errors (although to be clear, no known problems were fixed by this).

Version 3.0.2 (November 26, 2013)

- Fixed the bug that was causing crashes on quit, finally. Took a long time to find this one; it was a sneaky devil!
- Made assorted bits of memory management code a little bit more robust.
- Added a "Clear All Breakpoints" button to the debugger.
- Updated the internal sound effect routine; the disk eject sound effects now work again (I don't know when they stopped working, but they had).
- Cleaned up bits of code related to memory management.
- The integrated utility programs are now code-signed as required by Xcode 5.
- Code signing of the application itself with my developer ID is now done more correctly.

Version 3.0.1 (August 12, 2013)

- Rearranged the order in which certain tasks are performed on shutdown to reduce the chances of hitting a Mac OS X bug that's causing crashes when freeing memory used for tracking certain disk image files.
- Updated the code that adds the Quit option to the "It is now safe to shut off your Apple IIgs" alert to not break if Marinetti presents its "You're shutting down with an active network connection" alert as well.
- Minor adjustments to the OpenGL startup code that may slightly improve performance on certain hardware.

- Rewrote the code that switches between full screen and windowed modes to use newer CGDisplay APIs. Full screen mode now works correctly on Macs with Retina displays.
- Fixed broken handling of error keys when being interpreted through low-level ADB access, such as in Wolfenstein 3D. Previously, key-up events were not being correctly generated for these keys.
- Added support to Sweet16 for a new “start” type for the “step” command. This is used by the included debugger to immediately break into the debugger when you click the new “Break” button.
- The debugger API “setRegisters” command now supports all the same registers “getRegisters” does, although this functionality is not yet used by the debugger itself. That should come soon!
- The Sweet16 build number is now displayed in the emulator info alert presented by the debugger, in parentheses after the version number. This value is also in a new “build” value reported by the debugger API’s “getEmulatorInfo” command.
- Console output is now only time-stamped at the beginning of new lines; you can no longer wind up with timestamps in mid-line if you don’t include a newline each time you send console output.
- Removed a JavaScript library from the debugger that I’m no longer planning to use.
- Sweet16 3.0.1 is now code-signed with my Apple Developer ID.
- When activating Sweet16 with a mouse click, the window clicked on is now correctly made the frontmost window, instead of just bringing the application to the foreground.

Version 3.0 (July 24, 2013)

- Sweet16 3.0 requires Mac OS X 10.6.8 or later, and as such is no longer compatible with PowerPC Macs.
- Implemented support for dragging files from the Mac into the GS Finder and vice versa; requires both the SweetPrinter and SweetDupe extensions to be installed on the emulated Ilgs.
- Implemented the first phases of the new Sweet16 debugger protocol, which is provided by a built-in WebSocket server that responds to special JSON-format commands, and sends its responses in JSON format as well. See “[The Sweet16 Debugger](#)” for details.
- There’s also a Web server internally, that’s running when the debugger server is running, which serves up the built-in Debugger for you to use in your browser.
- The internal Web server can also send a PNG image of the current Apple Ilgs screen if you ask it for a file called currentscreen.png; for example, to load it from a browser running on the same computer as Sweet16, visit <http://localhost:6502/currentscreen.png>.
- Created the new “Open In New Sweet16 Instance” service.
- Removed the old JavaScript-based scriptable debugger and the user interface for controlling and configuring it.
- The mounted disks window’s context menu now offers a “Reveal in Finder” option.
- Fixed a bug in the IWM code that could result in failure to correctly count the tracks on a 5.25” disk.

- Changed all references to “CleverPort” to “SmartPort” because... well... it was confusing people.
- Renamed the “Automatically pause when in Dock” preference to “Automatically pause when minimized.”
- Added an option to the Setup menu to let you select a ROM file. Finally!
- When the screen dims while the emulator is paused, it no longer fades all the way to black.
- Alert boxes are now presented using a thread-safe method that will reduce the potential for crashes being caused by trying to present an alert from threads that don’t have access to the user interface layer.
- Added an option to the Help menu to download Ewen Wannop’s “Sweet16 Starter Kit”.
- Updated many of the special internal COP intercepts to set the carry flag correctly on errors.
- Cleaned up a lot of compiler warnings; these were harmless but sloppy.
- A little clean-up work on the ADB emulation code.
- Showing and hiding of the Mac cursor is much more reliable as you mouse in and out of Sweet16.
- The click that activates Sweet16 after it’s been in the background is no longer passed along to the Ilgs. This should help reduce the number of accidental clicks on buttons and the like on the Ilgs.
- Mouse events handled by the emulator are no longer reported back to the Mac as “unhandled.”
- Significantly improved the Mac event handling code to be more efficient and less... quirky.
- Implemented support for detecting the COP codes inserted when ORCA compilers are configured to output code for ORCA/Debugger. Currently, this just results in some extra console output, but soon this will be used more by the Sweet16 Debugger.
- Console logging from Ilgs code (done using a special COP operand) is now directed to the debugger instead of to the Mac’s console.
- The about box now shows a build ID number next to the version number.
- The about box no longer wastefully requests two references to the application bundle.
- Updated all places that were using hard-coded buffer sizes for pathnames to use the PATH_MAX constant, so that they at least are guaranteed to be big enough.

Version 2.3.1 (July 31, 2011)

- Improved the vertical mouse wheel support to be more reliable.
- Added support for horizontal mouse wheel scrolling.
- The emulator speed overlay now moves properly when the video window is resized.

Version 2.3 (July 20, 2011)

- The mouse scroll wheel now operates by sending appropriate events to scrollable content to scroll it, instead of sending keyboard up and down.

- Clearing of system memory on a reset is now done like a real IIgs.
- The right-click button on your Mac mouse or trackpad is now delivered to the GS as a right-click.
- The “Clear Recent Disks” menu option no longer crashes.
- The “Get Clipboard from Mac” NDA has been replaced with a corrected copy that actually works.
- Rewrote all the sound code to use Core Audio instead of the archaic Sound Manager.
- Revamped the about box to look nicer.
- Performance of audio in select programs that use one-shot mode on the DOC has been improved (examples include Tass Times in Tonetown and Arkanoid). It’s not perfect but it’s better.
- Removed some dead code here and there, which reduced code size and improved performance.
- Classic 1-bit sound code is no longer being disabled when not in use; this prevents the first few moments of audio from being lost while it gets turned back on.
- While paused, the GS display is now dimmed to make this obvious.
- The CPU preferences panel now includes a note explaining that quitting and restarting Sweet16 is required to make changes to the ZipGS on/off setting take effect.
- Fixed a crash that was happening when playing the disk eject sound effect; this should finally resolve the last of the crashes happening when ejecting disks.
- Added a note to the screen border when in full screen mode that explains how to get out of full screen mode.
- Moved the system speed display into the video window, and its option to turn it on and off to the View menu.

Version 2.2.1 (September 16, 2010)

- The SmartPort emulation now correctly sets the screen hole at \$07F8 to \$C7 each time a device is accessed; this corrects some issues with disks not working as expected; for example, now you can boot ProDOS 8 and then run the GS/OS bootstrap version of the PRODOS file to switch to GS/OS.

Version 2.2 (July 18, 2010)

- Sweet16 now defaults to having speed control enabled, with the GS set to a 40 MHz emulated speed. This prevents laptops from getting hot while running Sweet16. You

can easily turn off speed control to let your GS fly, using the preferences' CPU options panel.

- You can now mount ProDOS-order (.po) disk images that are not floppy disk sizes.
- When paused, Sweet16 no longer continues to use lots of your Mac's processor capacity.
- Creating 1.4 MB disks using the Create Disk Image feature now creates disk images that are actually 1.4 MB, rather than 1.6 MB.

Version 2.1.3 (August 29, 2009)

- Massive speed improvement (especially on PowerPC) achieved by optimizing the main CPU thread loop.
- Fixed a problem preventing the speed control feature from working on PowerPC.
- Control-H no longer resets the Apple IIgs. Oops.
- Removed some stray debug output.
- Rewrote the Recent Disks code so the Recent Disks submenu now only shows file names instead of full URLs to the files in the list. This makes it much more user-friendly.

Version 2.1.2 (August 26, 2009)

- Fixed several errors in the French keyboard support.
- Creating any size disk image, including floppy disk sizes, now has the same speed gain as creating hard disk images got in Sweet16 2.1.1.
- Removed an obsolete error message.
- Fixed an error message that was coming up blank when trying to create invalidly-sized DiskCopy 4.2 images.
- The disk image size now tracks correctly in the "Create Disk Image" window, so the size you get actually matches the size you picked all the time (there were cases in which that wouldn't necessarily be true).

Version 2.1.1 (August 4, 2009)

- Reworded the alert that appears when trying to mount a CD-ROM when no CD-ROM is found, since the same alert is used both when there is no CD-ROM media available as well as when there is no CD-ROM drive available.
- Memory no longer gets leaked when controls in the Create Disk Image and Create Emupack file panels are clicked on.

- The Create Disk Image panel now lets you create DiskCopy 4.2 images at 1.4 MB.
- The Create Disk Image panel's size popup no longer wanders out of view when you resize the panel.
- Enabled some new optimizations that noticeably improves performance on PowerPC systems.
- Removed some unnecessary debug output.
- The SpiderMonkey JavaScript runtime is now weak-linked, so that Sweet16 really does now run on PowerPC G3 systems (albeit without debugger support).
- The Create Disk Image panel now lets you choose a disk image format on Mac OS X 10.4; previously this popup window didn't do anything prior to Mac OS X 10.5.
- Creating disk images of even 1 MB sizes (5 MB, 10 MB, 32 MB, 100 MB, for example) now goes much, much faster, since instead of writing out 512 bytes at a time, these are written 1 MB at a time. A future update will provide a similar performance improvement for floppy size image creation.
- Clicks in the video window are no longer ignored when the System Information window is open.
- Removed some unnecessary mutual exclusion code in the networking support, speeding up networking substantially.
- Rewrote the video timing code to be more precise.
- The main event loop now does sleeps instead of busy-loops, dramatically reducing wasted CPU cycles; Sweet16 plays much more nicely with other running processes now.

Version 2.1 (July 23, 2009)

- Added a new View menu, with new half-size, normal size, and double size options.
- Moved the Full Screen option to the View menu.
- Fixed a serious memory corrupting/crashing bug in the Get Clipboard NDA.
- Improved the robustness of the Sweet16 side of the clipboard code.

Version 2.0 (July 23, 2009)

- Added to the help menu links to several useful Apple II related web sites.
- Improved the algorithm for detecting DiskCopy and DiskCopy 6 images to hopefully work better.

- Fixed a bug that would cause a crash if an error occurred while trying to allocate the buffers for full screen mode; now full screen mode is simply disabled if this happens.
- Instead of giving up and ejecting the CD-ROM if an error occurs, now the first nine partitions are all tried, and if any of them succeed, the disk is considered mounted. This lets it work with disks that have “free space” and unrecognized partitions.
- Removed some dead code.
- Fixed the EMUBYTE softswitch to correctly return \$16 as the product ID; this has been broken for a very long time.
- Rewrote the networking code. The new networking code is both faster and no longer requires annoying authentication and so forth to use it. There is a new link layer that must be installed to use the updated networking code successfully.
- Removed Sparkle support for automatic updating; the Carbon version of this is a fourth-party patch and I decided I’d rather not worry about it.
- The debugger should now be disabled on G3 systems; I still haven’t figured out why the debugger makes Sweet16 crash on launch on G3 systems, so at least now Sweet16 should work, but the debugger will silently do nothing.
- Fixed the data displayed in the Finder’s Get Info window to correctly show Sweet16’s version number where you expect it to be.
- Gamepads are now supported for game control.
- Fixed a bug that caused screenshots captured under certain circumstances to have garbage at the bottom.
- Disk images that are exactly 819,200 bytes long are now assumed to be 800K raw disk images, regardless of their type and/or extension.
- The disk eject code has been updated to no longer crash while trying to redraw the online disks window.
- Added support for, and implemented, two new NDAs: GetClipboard and SendClipboard; these let you copy the Mac clipboard to the IIGs clipboard and vice versa.
- Added Emupacks—self contained packages including disk images, ROM, and preferences that can all be opened and enabled at once, simply by double-clicking the Emupack in the Mac Finder.
- Fixed the preference for disabling the mouse in text modes so it actually works.

- Removed the window that says Sweet16 is recalculating DiskCopy 4 checksums; it didn't stay onscreen long enough to be helpful, and would sometimes crash because it was being opened from background threads.
- The Close option in the File menu is now disabled if the video window is frontmost, since you can't close it.

Version 2.0 Public Beta 2 / 2.0b18 (August 11, 2008)

- Rebuilt the integrated JavaScript support to work on both PowerPC G4 and G5 processors. G3 support should be forthcoming.

Version 2.0 Public Beta 2 / 2.0b17 (August 10, 2008)

- If no joystick is connected, the joystick routines now report that they're dead center.
- If the memory for the screen shot buffer can't be allocated, screen shots are disabled, instead of letting it try and crash.
- You can now paste text from the Mac clipboard into the Apple IIgs. Only ASCII characters are supported at present.
- Added a System Information window to show the CPU speed.
- Programmers can access the OpenGL buffer used to draw the Apple IIgs screen in banks \$E2 through \$E9 of Apple IIgs memory. This is a 640x400 pixel buffer in 16-bit color.
- The Sparkle framework is now used to provide automatic updates. I don't know how well this will work yet.
- Added mutex to fix problems with crashes adding and removing disks to the online disks windows.
- Added a new Toggle Full Screen NDA so you can toggle full screen within the IIgs environment.
- Fixed a crash that occurred when changing the window title from the CPU thread; it is now done in the UI thread.
- Fixed a problem with color palettes that affected colors in DreamGrafix, Wolfenstein 3D, and other applications.
- You can now mount disk images by dragging them to the application's icon.
- Added joystick support; USB joysticks are now supported (but not gamepads).
- Fixed a crash that occurred when ejecting disks by moving the resize of the online disks window into the UI thread.

- If GS/OS is shut down and you're at the "It's now safe to shut off your Apple IIgs" screen, quitting Sweet16 no longer asks for confirmation, since it's safe to quit.
- Added a new preference that lets Sweet16 automatically switch to 1 MHz when booting 5.25" disk images.
- If the first character of a disk image name you're creating isn't a letter, it's replaced with an "x" when naming the ProDOS volume written onto it.
- Implemented TCP/IP networking support over Ethernet.
- Strings are now in the localizable.strings file so they can be localized.
- Replaced the application icon with something less dorky looking.
- Added a sweet new debugger.
- Fixed the title of the About window to get rid of some junk characters.
- You can now mount ISO format disk images; these may require the High Sierra (ISO) FST to be installed, depending on the format of the contents.
- You can now mount CD-ROM disks into Sweet16. Sweet16 will attempt to mount all partitions on the disc. Ejecting the disk from the emulator physically ejects it from the computer.
- Added the "Open CDA Menu" option to the Setup menu; this lets you open the CDA menu if you don't remember the keys to do it, or if they conflict with some other software running on your Mac.

Version 2.0 Public Beta 1 (July 19, 2007)

- Graphics are now done using OpenGL for added performance.
- Added routines that Apple IIgs code can call to aid in debugging.
- Added support for setting the video window's title bar to indicate the name of the running Apple IIgs application.
- Added the Storage panel to preferences, which includes settings for whether to mount 3.5" disk images on CleverPort or IWM by default, as well as a preference for whether to automatically set the IIgs to 1 MHz when booting 5.25" disk images.
- Added support for full screen mode.
- You can now resize the video window to any size; just hold down the option key to make the resize box appear.
- Now using Carbon events for improved performance and capabilities.

- Improved showing and hiding of the Mac mouse cursor.
- You can now switch between 16:10 aspect ratio and 4:3. 16:10 looks better, but 4:3 is the standard Apple IIgs aspect ratio for the screen, so things are the correct shape.
- Fixed a bug that was causing files to be created if they didn't exist, which was causing problems if your ROM was missing (it would create an empty ROM file and start having weird errors).
- DiskCopy 4.2 and Universal Disk Image files are now recognized by their file type and creator codes, which makes mounting these easier if they're missing their extensions.
- Universal Disk Image files are now recognized by their contents, and can therefore be mounted even if their type, creator, and extension are wrong.
- HDRV images are now recognized by type and creator.
- The mouse scroll wheel now sends up and down arrow key events to the IIgs.

Version 1.7.3 (September 12, 2006)

- Fixed timing problems with time-based interrupts such as the quarter-second and 1-second interrupts, which makes time-based games tend to work much better.
- Choosing the Dvorak and French Canadian keyboard layouts in the Control Panel no longer crashes Sweet16, and should actually provide the correct keyboard layout.
- Removed some unneeded data.
- Speed control now works correctly on PowerPC machines; it had previously been fixed only on Intel processors.

Version 1.7.2 (August 28, 2006)

- You can now write to 5.25" disk images.
- You can now drag multiple disk images to the disks window to mount them all at once.

Version 1.7.1 (August 18, 2006)

- Holding down the option key while mounting 800K disk images now mounts them as CleverPort devices instead of as IWM floppy disks; this lets you mount lots of 3.5" disk images at once, which makes installing the system software, for example, incredibly easy.
- Updated the URL in the documentation that's presented for where to go to download system disks.
- Aliases to files now resolve correctly, so you can drop aliases to your disk images into the Disk Images folder, instead of moving the images themselves.

- The Reset command in the Setup menu now works correctly.
- Optimized more PowerPC code, resulting in a small additional performance boost over version 1.7.
- Fixed a bug that prevented Recent Disks from mounting.
- Items in the Eject submenu now display the name of the disk correctly; this broke in version 1.7 due to the UTF-8 change.
- Ejecting a disk then inserting a new one now properly redraws the contents of the disks window if it's visible. Previously, it was possible for the contents of the window to become somewhat confused in this situation.
- Fixed a bug that caused the mouse cursor to bounce around the screen when Default Folder X is installed. This also slightly improves the response performance of the mouse in Sweet16.
- Fixed a couple of problems that caused crashes when clicking in the disks window under certain circumstances.
- The Control-Command-M keyboard combination now properly issues the Mount Disk Image command.
- Fixed a memory leak when initializing the Recent Disks menu.
- Added a "Clear Recent Disks" command to the Recent Disks menu. This empties the recent disks list.
- Fixed a bug that caused new DiskCopy 4.2 images to have invalid names in their DiskCopy 4.2 headers.
- Fixed a bug that caused the progress window used when creating disk images or calculating the checksum on a DiskCopy 4.2 image to crash.
- Fixed HTML markup problem in the documentation that made the instructions on copying the ROM from a IIgs unreadable.
- Fixed the error dialog box that appears if you try to mount too many disks (or too many of a given type) so it actually contains text.
- Fixed the "allow 14 MB" checkbox and the other controls in the Memory panel in the Preferences window to stay in sync properly.
- Fixed the memory size preference so it resets to 8 MB if you turn off the "allow 14 MB" checkbox with the memory size set above 8 MB.

Version 1.7 (August 14, 2006)

- Pathnames are now kept in UTF-8 format instead of in ASCII, which improves compatibility for non-English users.
- Added a preference to automatically cause the mouse to be disabled when in text modes.
- Fixed the speed control checkbox so that toggling it takes effect immediately instead of usually requiring you to quit and restart Sweet16.
- Optimized performance on PowerPC systems, improving speed by up to 30%.
- DiskCopy 4.2 images created and marked as being Mac format disks will now work in Sweet16.
- Sound effects are now played when ejecting floppy disk images.
- The font in the disks window is now Lucida Grande.

Version 1.6.1 (August 1, 2006)

- The Disks window no longer disappears when Sweet16 isn't in the foreground, which makes it an easier target for drag & drop of disk images.
- DiskCopy 4.2 disk images are now recognized for mounting even if they don't have an extension.
- Fixed a problem that caused an empty error dialog box if you tried to mount a disk image that's already mounted by Sweet16.
- Dragging disk images into the disks window now displays the fancy arrow with a green plus cursor.
- Enabled some optimizations when compiling that I hadn't been using before. These may or may not have any noticeable impact on performance, but it probably doesn't hurt.
- By default, the speed used when you turn on speed control is now 2.6 MHz instead of the strange 0.5 MHz it was set to before.

Version 1.6 (July 19, 2006)

- Initial Mac OS X release.