

Mac-Packaging Bits and Bites

*An Unofficial Guide for Authorware Program
Deployment on Macintosh Operating Systems*

Version 1.2

Updated August 19th, 2005



Prepared by:

Steve Gannon
GanTek Multimedia

steve@gantekmultimedia.com

DISCLAIMER:

This document is neither sponsored nor endorsed by Macromedia. Some of the information herein was gleaned from Macromedia's TechNotes, Authorware newsgroup discussions, and communications with other Authorware developers. Much of the content came about from trial and error while developing cross-platform pieces here at GanTek Multimedia. While we hope this document is of value to you, Steve Gannon and GanTek Multimedia make no assurances, either expressed or implied, as to the content's accuracy. By using any information in this document, you do so at your own risk.

CONTENTS

Introduction.....	1
Authorware Features Supported on the Mac	2
Authorware Features NOT Supported on the Mac	2
Deployment Options	3
Minimum System Requirements	3
Classic vs. OS X. What's the Difference?	3
Detecting the Operating System Currently Running	4
Steps to Package for Mac Playback	5
Testing in Classic Mode	7
Files You Need to Distribute	8
Xtras.....	8
XCMDs.....	9
Drive Names	10
Filenames	10
Search Paths	10
File Path Delimiter.....	11
Fonts.....	12
Installing Your Application on the Mac	14
LAN Deployment	14
Jumping From File to File	14
Graphics	16
Buttons	16
Movies	17
Audio	18
PDF Files	18
Changing a Program's File Association	19
Changing a Program's Icon	20
Memory.....	21
Printing.....	22
MoveCursor() Function	22
Key Names.....	22
Fade Transition	22
Target Area Interaction	23
Web Player Issues.....	23
Other Sources of Crashes.....	24
Other Suggestions for Correcting Problems	24
Crashes – Behind the Scenes	25
Creating a PC/Mac Hybrid	26
Need More AW Cross-Platform Help?.....	29
Version History.....	29

Introduction

The last release of Authorware to provide an Apple Macintosh development environment was version 4.x. However, later versions include a *Mac Packager*. This tool takes your packaged-without-runtime (.a7r file) Windows program and generates a program file that will run on the Mac. Not until Authorware 7 was playback directly supported under the newer Mac operating system (OS X).

The packaging process is generally straight-forward and quick for simple Authorware pieces. The more complex your application, the more involved the packaging process can become. This document outlines the packaging process, provides generally Mac deployment tips, and delves into some pitfalls to be aware of. The discussions offered here refer primarily to version 7.0.1 of Authorware Mac Packager-generated files. Also, this document makes some assumptions about your Authorware knowledge, including a working understanding of internal and custom variables, internal and external functions, etc.

Developing your piece fully on the PC and then porting it to the Mac is not recommended. You really should have a Mac on hand and test periodically along the way as you develop.* Otherwise, you may be in for some significant problems that may necessitate the rewriting of both Windows and Mac portions of your application. If you read the items presented in this document before beginning development, you may save yourself some frustration down the road. Most of these tips have come about from the “school of hard knocks”.

There are three Macintosh operating environments supported by Authorware 7:

- Mac OS X
- Classic Mode (Mac OS 9.x running on top of Mac OS X)
- Mac OS 8.6, 9.x (requires booting the computer with one of these operating systems)

Unless otherwise indicated, for purposes of this document we will refer to all supported Macintosh operating systems prior to ten as ‘Classic’.

* If you are packaging for Internet/intranet deployment, you *technically* don't need a Mac. The web-packaged files (.aam and .aas files) generated on the PC will work directly with the Mac Web Player. *Realistically*, you should have a Mac on hand to test along the way as you develop on the PC so you can compare the Mac's playback versus the PC's playback.

Authorware Features Supported on the Mac

The Authorware Mac Packager supports many of Authorware's features:

- JavaScript
- Alpha channel for graphics
- Anti-aliased text
- Most media formats (QuickTime, BMP, Flash, WAV, etc.)

Although we have not tested these at GanTek Multimedia, the Mac Packager Help file states that the following Knowledge Objects (KOs) are supported:

- Application
- Check Boxes
- Drag-and-Drop Question
- Hot Object Question
- Hot Spot Question
- Internal Quiz
- Launch Browser
- Movie Controller
- Multiple Choice Question
- Radio Buttons
- Short Answer Question
- Single Choice Question
- Slider
- Standalone Quiz
- True-or-False Question
- Web Player Security

Though not listed above, you should also find that the latest Accessibility Kit works well on the Mac too.

Authorware Features NOT Supported on the Mac

Authorware programs on the Mac do not support:

- DLLs
- U32s (i.e., no Windows Controls, WinAPI functions (e.g. SystemMessageBox), etc.)
- ActiveX controls
- Many third party Xtras
- Some specific Microsoft-developed Windows Media formats (e.g. .asf, .wmv).
- Object Linking and Embedding (OLE)
- WMF and EMF graphic files
- DVD icon

Deployment Options

The Mac Packager can package for local area networks (LANs), CD-ROM, local hard disk and the Macintosh Authorware Web Player. This document focuses primarily on CD and hard disk delivery.

Minimum System Requirements

To play back an Authorware piece packaged on the Mac you will need the following:

- Apple PowerMac G3 or higher
- One of the following Mac operating systems:
 1. Classic: OS 8.6, 9.x
 2. OS X: 10.1.5 or later
- 24 MB RAM

We recommend running Authorware pieces on Mac computers with at least 64 MB RAM (refer to the *Memory* section later in this document for more information).

Classic vs. OS X. What's the Difference?

In general terms, 'Classic' refers to Apple Computer's earlier Macintosh operating system. OS X refers to Apple's newer Macintosh operating system (version 10+). OS X is a completely different operating system than OS 9 and earlier versions, one that is based on UNIX (<http://www.apple.com/macosx/features/unix/>).

Prior to Authorware 7, the Mac Packager could only create pieces that would run on computers running a Classic operating system. In the first few releases of OS X, this Mac operating system would recognize if a program was developed for Classic and would load the Classic OS on top of OS X in order to run the piece if needed. Be aware that although the Classic operating system was included with many versions of OS X, it is now being phased out.

With the advent of OS X support in Authorware 7, you can now run an Authorware Mac Packaged piece directly in OS X without having to have Classic load first. The file generated by the Mac Packager is a 'carbonized' application. This means that it will work with OS 8 and 9 (Classic) as well as OS X.

Detecting the OS Currently Running

There are a couple of internal Authorware variables that can help you programmatically determine which operating system is running:

Machine returns the following based on the respective operating system:

- 1 = Macintosh Plus, SE, or Classic
- 2 = Macintosh or Performa system with color capability and a processor other than a 68000
- 3 = IBM PC or compatible computer
- 5 = Power Macintosh

...so, if `Machine=3`, then the program is running under a Windows operating system. On the other hand, if `Machine<>3`, then the program is running on a Mac.

OSVersion returns the version of the operating system being used. For example, `OSVersion` will return "Windows NT (5.1)" on Windows XP systems.

For the Mac, if you extract just the numeric portion of the value returned by `OSVersion`, your program can determine if the operating system is a Classic OS or a version of OS X.

```
if Machine <> 3 then -- running on a Mac
    if GetNumber(1, OSVersion) < 10 then
        -- running on a Classic operating system
    else
        -- running on an OS X operating system
    end if
end if
```

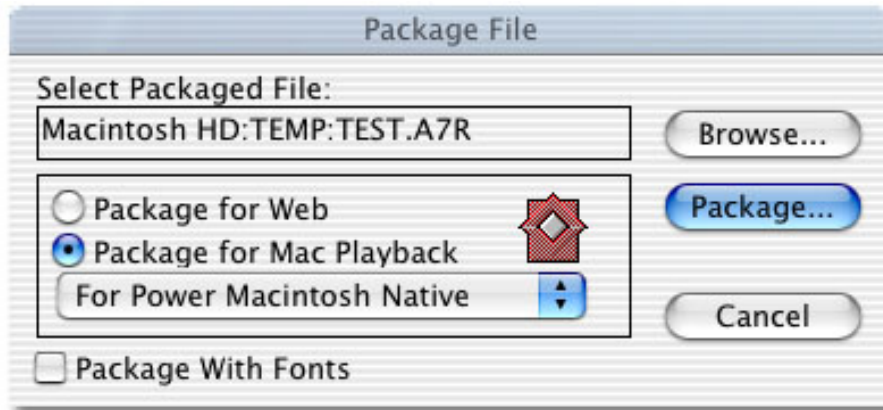
Steps to Package for Mac Playback

Below we have summarized the steps needed to package your Authorware program on the Mac:

1. Download and install the Mac Packager. The Mac Packager can be found here:
<http://www.macromedia.com/software/authorware/productinfo/macplayer/form/>

You will need your Authorware serial number to download the Packager. The Packager files are in a single compressed file. If possible, download this file directly to your Mac; alternately, you can download the compressed file to your PC and then move it to your Mac. Uncompress this file on your Mac and then run the installer. Do not attempt to uncompress and install the Mac Packager on your PC!

2. On your Windows-based PC, package your piece without runtime. The resulting file will end with .a7r. We recommend packaging all Authorware libraries internally into the a7r file. Refer to Authorware's online Help if you need assistance packaging files on the PC.
3. Create a folder on your Mac to hold your project. Copy the .a7r file to this folder on your Mac. Create an Xtras folder within this folder. **Do not copy the Xtras on your PC to your Mac!** The Mac version of the Xtras are located in the Xtras folder where you installed the Mac Packager. Copy just those Xtras needed by your piece from the Authorware Mac Packager's Xtras folder to the Xtras folder you just created. (Refer to the *Xtras* discussion later in this document.)
4. Launch the Mac Packager.
5. Click the Browse button to locate and select the a7r file you want to package.
6. Select Package for Mac Playback assuming you want to make a file to run off of a CD, hard disk, or LAN. Otherwise, select Package for Web if you want to make a web-packaged version of your AW program. (Remember, you can package for the Web on the PC using One Button Publishing; the resulting .aam and .aas files will work with the Mac Web Player). When you select Package for Mac Playback, you will be prompted to make a choice between Without Runtime or For Power Macintosh Native. In most cases, you will want to select For Power Macintosh Native. However, if you are packaging several pieces that are linked via JumpFile or JumpFileReturn, you may want to use Without Runtime so that only one piece includes the Authorware runtime. This will save disk space. If you're not sure, refer to *Packaging* in Authorware's online Help on the PC. You'll notice that the icon used for Package for Mac Playback does not look the same as the current Authorware 7 icon. This icon is left over from the Authorware 4 and 5 days. Just ignore it. (Refer to *Changing the Program's Icon* later in this document to learn how to change the icon assigned to your piece.)



The Mac Packager dialog.

7. One of the options on the Mac Packager dialog is to Package with Fonts. This feature is supposed to combine any special fonts you used in your piece with your .apm file. However, we have not seen it function with any version of the Mac Packager from Authorware 5-7. We'd be interested in learning if any developers have been able to get this to work with the Authorware 7 Mac Packager. We believe this is a leftover feature from Authorware 4 and earlier where users could develop directly on the Mac and is erroneously still included in the Mac Packager and Mac Packager Help file. (Refer to *Fonts* later in this document for a discussion of how to match fonts across platforms.)
8. Click Package.
9. In the dialog that appears, enter a name for your piece and specify a folder location. The Packager will create a file using the filename you assigned and will automatically append the .apm extension. In prior versions of Authorware, the Mac Packager added the extension '.pkg'. Unfortunately, the .pkg extension was used by other programs so the Mac didn't run Authorware's .pkg files properly. With the default .apm extension, this is no longer a problem. In fact, your piece should run fine if you remove the file extension altogether.
10. Click Save to package your piece.

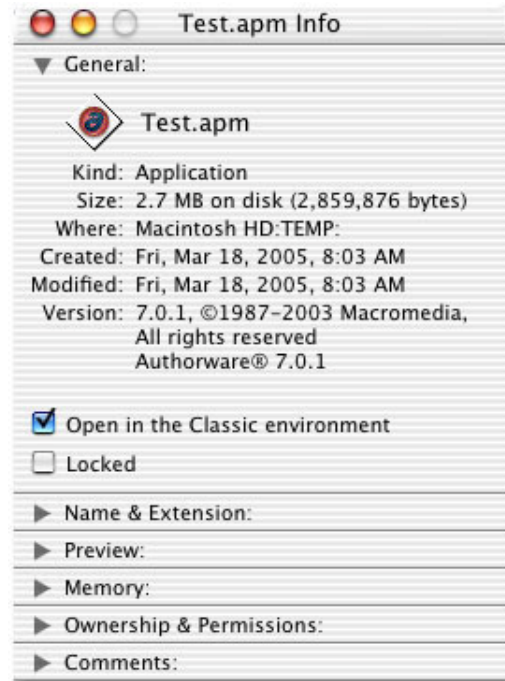
Testing in Classic Mode

If you are currently running OS X, there are two ways to test your apm program in Classic mode.

1. Reboot in OS 9:
 - a. Click the apple icon in the far left of the desktop's title bar.
 - b. Select System Preferences.
 - c. In the System category, launch Startup Disk (this icon is in the lower right corner of the System Preferences panel)
 - d. Click Mac OS 9.x... and click Restart.
 - e. Click Save and Restart.
 - f. Browse to your packaged Authorware file (.apm file) and double-click it to launch it.
2. Force the .apm file to open in Classic while still booted in OS X:
 - a. Browse to your packaged Authorware file (.apm file).
 - b. Single-click the .apm file.
 - c. Pull down the File menu and select Get Info.
 - d. Check the Open in the Classic environment checkbox.
 - e. Close the Get Info dialog.
 - f. Double-click the .apm file to launch it.

NOTES:

1. **Be sure to uncheck the 'Open in the Classic environment' checkbox before burning a CD!**
2. **You may need to increase the memory assigned to your .apm file before option 2 above will work. Refer to *Memory* later in this document.**



Check the 'Open in the Classic environment' checkbox in the 'Get Info' dialog if you want to open your application in Classic on top of OS X.

Files You Need to Distribute

There are several files you need to distribute with your piece in order for it to operate properly:

- Your Mac-packaged (.apm) file.
- Your Xtras folder (refer to *Xtras* section below).
- IMLLibCarbon (place alongside the .apm file).
- MacromediaRunTimeLib (place alongside the .apm file).

NOTE:

If you fail to include IMLLibCarbon and MacromediaRunTimeLib alongside your packaged (.apm) file, your program will fail to function!

Xtras

There are several things you need to know about Xtras on the Mac:

- **DO NOT** use Xtras from your Windows version of Authorware. Instead, get the required Xtras from the Xtras folder that was installed with the Mac Packager.
- Xtras on the Mac have a more 'readable' name. For example, the Xtra that allows playback of WAV audio on the Windows version of Authorware is called `wavread.x32`. On the Mac, this Xtra is called `WAV Reader` (no extension).
- The Xtras that are included with the Mac Packager have been *carbonized*. This means that the Xtras will work in both Classic and OS X environments.
- Just as with the PC version of your Authorware programs, you should endeavor to include only those Mac Xtras your piece requires. When an Authorware piece launches, it loads all of the Xtras within the Xtras folder whether your piece needs them or not. If you include all of the Xtras, you are unnecessarily causing your piece to load more slowly.
- Few third-party Xtras are designed to work on the Mac; even fewer have been modified to run under OS X. Fewer still have been carbonized so that a single Xtra will work under both Classic and OS X. Below is a list of some of the third-party Xtras for the Mac that have been modified to accommodate Authorware 7's support for OS X:

BudAPI - separate versions of the Xtra required for Classic and OS X (<http://www.mods.com.au>)

DirectXtras - several cross-platform Xtras available (<http://www.directxtras.com>)

Impressario - one Xtra works for both Classic and OS X (<http://www.inm.com>)

Printomatic - separate versions of the Xtra required for Classic and OS X (<http://www.printomatic.com>)

- If you attempt to include an OS X-specific third-party Xtra with a piece that is run on a Classic operating system, you'll simply get an error message if you're lucky, but sometimes the piece will crash. The same is true when you include a Classic-specific Xtra with a piece run under OS X. If you want to include a third-party Xtra that has separate versions of the Xtra for Classic vs. OS X, you have a couple of options:
 1. You can have an installation program install the appropriate set of Xtras based on the operating system currently being run.
 2. You can install two versions of your program and use a small router program to jump to the appropriate version so that the appropriate Xtras are loaded. You can create your own installation program using Authorware and a utility like BudAPI (<http://www.mods.com.au>). Use Authorware's `OSVersion` variable to determine which operating system is currently running (see *OSVersion* earlier in this document).

At GanTek Multimedia, we typically use option 2 because many Macs contain both Classic and OS X operating systems. Just because the user has Classic running when he/she installs the application doesn't necessarily mean he/she won't use OS X later to run it, or vice versa.

XCMDs

XCMDs are like U32s on Windows computers – these files contain a set of functions external to Authorware. Some of the more common XCMDs available in the past include:

- Cover.xcmd
- FTP.xcmd
- ODBC.xcmd.

The Help file that accompanies the Authorware 7 Mac Packager suggests that XCMDs function under OS X. Our experience is that they do not. Therefore, we recommend that you simply avoid using them. If you attempt to load a function from an XCMD when running your packaged .apm file on the Mac under OS X, your piece will likely crash. In fact, if you are upgrading an Authorware piece that used to run on the Mac and it now crashes, one of the first things you should check during the troubleshooting process is the inclusion of any XCMD functions in the piece.

You can include XCMDs in your piece but we recommend that you not call an XCMD function unless the user is running under Classic. Such as:

```
If Machine <> 3 then -- running on a Mac
    if GetNumber(1, OSVersion) < 10 then -- running on Mac Classic
        Cover() -- loaded from Cover.xcmd
    end if
end if
```

Drive Names

The Mac uses full text for drive naming where as the PC uses drive letters. For example, the primary Macintosh hard disk (comparable to C: on the PC) is generally called Macintosh HD on the Mac. However, this drive name could easily be changed. You need to be aware of this so you don't inadvertently attempt to install some files or create a folder on C:\ on the Mac.



Macintosh HD

Likewise, CD drives are not identified by a letter but instead by the name of the CD volume currently inserted in the CD drive.



Test Application

You can programmatically determine the name of the Mac drive your program is currently running from by parsing Authorware's FileLocation variable:

```
-- The code below extracts everything in the value currently  
-- stored in FileLocation, up to the first colon.  
Drive_Name := SubStr(FileLocation, 1, Find(":", FileLocation)-1)
```

NOTE:

To make the code above functional for both the PC and Mac, refer to the *Delimiter* topic on the next page of this document for some ideas.

If you need to know the names of the other drives on the Mac (such as in the case when you run an Authorware-developed installation program off a CD and need to know the name of the hard drive), you can use BudAPI's DiskList function (<http://www.mods.com.au>) to get a list of drive names and then use DiskInfo to determine the drive type.

Filenames

Like Windows, the Mac supports long filenames using most standard characters, including spaces. Be aware, however, that Classic operating systems (up to Mac OS 9.2) only support 31 characters in a filename. OS X supports filenames with up to 255 characters.

Search Paths

Authorware will not automatically convert a search path to the Macintosh equivalent. You must specify both Windows and Macintosh search paths in the Search Path box of the File Properties dialog box. If you use the SearchPath variable, be sure to do the same. Use a semicolon to separate Windows and Macintosh search paths.

File Path Delimiter

- On the PC, you specify file path delimiters using a backslash (\).
- In Mac Classic, file path delimiters are specified with a colon (:).
- Although file delimiters in UNIX operating systems (which Mac OS X is based on) is a forward slash (/), OS X uses the colon (:) for compatibility with Classic applications.

We recommend that you use a custom variable at the top of your flowline in which to assign the file delimiter, such as:

```
if Machine = 3 then -- running under Windows
    Delim := "\\\"
else                -- running on a Mac
    Delim := ":"
end if
```

Authorware's internal FileLocation variable knows what operating system delimiter to use automatically so you don't need to do anything with it. For example, on the PC, FileLocation may contain this path:

```
C:\\My_Application\\
```

...where as on the Mac, FileLocation may contain:

```
Macintosh HD:My_Application:
```

Note that FileLocation already contains the appropriate terminating delimiter (backslash or colon) no matter which operating system the piece is running on.

NOTE:

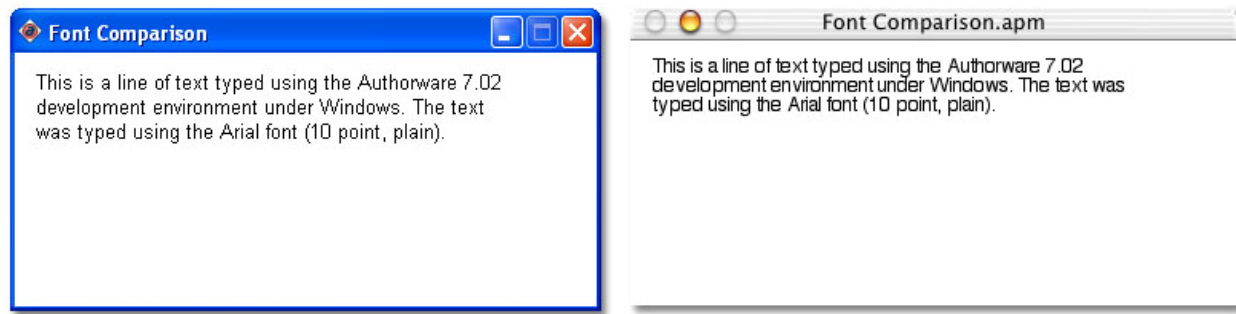
The double-backslash shown for the Windows examples above is used because a backslash in Authorware on Windows is a reserved character. Reserved characters must be 'escaped' with a preceding backslash so that Authorware knows to treat the following character as a literal character. You can learn more about reserved characters by reading the Authorware online Help on the PC.

So, if you wanted to reference an external graphic in a subfolder called My_Images, your code might look like this:

```
Image_To_Get := FileLocation ^ "My_Images" ^ Delim ^ "Image_01.gif"
```

Fonts

One of the most noticeable differences between the Windows and Mac environments when playing an Authorware piece is the screen resolution. On the Mac, you'll find that the vertical spacing between rows of text is smaller. Therefore, a paragraph of text on the Mac will take up less vertical space than on a Windows display (your text will look like it has been scrunched upward a little bit). Conversely, text characters require more horizontal space on a Mac display than on a PC. So a line of text is going to extend farther to the right on the Mac than on the PC.



Plain Arial 10 point text as it appears in Windows.

Plain Arial 10 point text as it appears on the Mac.
No font mapping was used in the file above.

Examples of fonts that work reasonably well on both platforms include Arial, Times New Roman and Verdana. Generally, you'll find that point sizes on the PC must be increased if they are to look comparable on the Mac. For example, an Arial font transfers to the Mac reasonably well using the point size increases shown in this table:

PC to Mac Point Sizes for Arial	
Windows	Mac
10	12
12	14
14	18
18	24
24	36

Authorware has accommodated these differences for many years by means of an editable font map file. The default font map file (a plain text file) is called FONTMAP.TXT. The intricacies of font mapping are beyond the scope of this document. You'll find a very detailed discussion of font mapping in Authorware's online Help on the PC as well as in the Mac Packager's Help file.

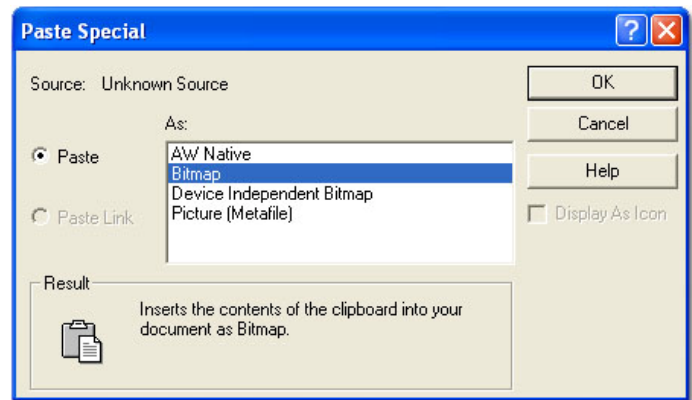
If you want to include a special Mac font with your piece, you will need to license the font, distribute it with your application, and install it on the Mac, preferably automating the process using a tool such as BudAPI (<http://www.mods.com.au>).

Be aware that Authorware's font mapping is still not going to provide you with a perfect match between platforms, mainly due to the difference in screen resolution between platforms. You will need to go through every screen on the Mac, noting where horizontal and vertical spacing, word wrapping, etc. are not meeting expectations. Then you'll need to go back and tweak those screens in Authorware on the PC.

One way you can achieve identical text appearance on both platforms is by turning your text into graphics. This is not quite as difficult as you might think. You may not even need a graphics program to accomplish this. Authorware already includes internal functionality for bitmapping text. On the PC, perform the following steps:

1. Make a backup of your Authorware code. Call it `Mac_Version.a7p` or whatever.
2. Open this backup file in Authorware.
3. Go to the first screen containing on-screen text.
4. Select your on-screen text by single-clicking on it (be sure the text itself is **NOT** highlighted...you just want the 'handles' to appear around the text).
5. Pull down the `Edit` menu and select `Copy`.

6. Pull down the `Edit` menu again and select `Paste Special...`
7. Select `Bitmap` and click `OK`.
8. Set the mode of this graphic text to transparent if necessary. However, if you need to do this, the quality of the text's appearance will vary based on the font, anti-aliasing, and colors used (pure white text set to transparent is not going to appear, obviously).



The 'Paste Special' dialog.

9. Position the graphic text (now a bitmap) directly on top of the existing text.
10. Pull down the `Modify` menu and select `Send to Back`.
11. Single-click on the actual text (now in front of the bitmapped text) and press the `Delete` key on your keyboard.

Of course this will not work for dynamic text (such as text read from external files during runtime or variable text like the user's name) but is otherwise worth considering. Although initially this may seem like an overwhelming task, you will be surprised how quickly you can accomplish these steps for dozens, or even a hundred or more, screens. If you want to consider this approach and you have not yet started developing, endeavor to keep the number of text display objects per screen to a minimum.

Installing Your Application on the Mac

There are several products available for preparing installation programs onto a Macintosh computer; below is a partial list:

Installanywhere	http://www.zerog.com/installanywhere-create-deploy-java-multiplatform-installers.html
Install Factory	http://www.installfactory.com/
Installer Maker	http://www.stuffit.com/installermaker/
MacInstallerBuilder	http://www.sdesoftware.com/products/macinstallerbuilder/
WISE	http://www.mindvision.com/

You can also develop your own installer using Authorware. Cross-platform third-party utilities such as BudAPI (<http://www.mods.com.au>) can greatly enhance the functionality that your Authorware-developed installer provides. GanTek Multimedia has a sample 'Setup' program available that makes use of the BudAPI Xtra. Although the program was designed for PC installations, it is relatively easy to modify for Mac-deployed applications. Download `setup.a6p` from <http://www.gantekmultimedia.com/download.htm> if interested. (This `.a6p` file can be opened in Authorware 7.02.)

LAN Deployment

When running a piece stored on an OS X local area network (LAN) server, you need to make the program file (`.apm` file) read-only. Be sure to set **ALL** permissions of the `.apm` file to read-only, including that of the 'Owner'. If you do not take this precaution, you may find that your piece cannot be run from two or more workstations concurrently.

Jumping From File to File

When jumping from one Authorware piece to another, you can use `JumpFile` or `JumpFileReturn`. The syntax for `JumpFile`/`JumpFileReturn` on the Mac is the same as it is on the PC:

```
JumpFile("filename"[, "variable1, variable2, ...", ["folder"]])
```

```
JumpFileReturn("filename"[, "variable1, variable2, ...", ["folder"]])
```

However, there is a slight limitation to be aware of. On the PC, you do not need to specify a file extension in order to jump between Authorware pieces. This is a valuable feature in that your code will function whether you are jumping between two unpackaged pieces in authoring mode, or two packaged pieces. But on the Mac, you need to include the file extension; otherwise, your program may have trouble locating the file. For example, if you use the following on the Mac:

```
JumpFileReturn(FileLocation ^ "MyProgram")
```

... the jump may fail. So instead, include the applicable file extension, such as:

```
JumpFileReturn(FileLocation ^ "MyProgram.a7r")
```

When jumping **out** between two packaged-with-runtime pieces or jumping out to a non-Authorware application, you need to use `JumpOut` or `JumpOutReturn`:

```
JumpOut("program" [, "document"] [, "creator type"])
```

```
JumpOutReturn("program" [, "document"] [, "creator type"])
```

The Macintosh requires the inclusion of a *creator type* as part of the function call. This parameter is not used on the PC so you may be unfamiliar with it. An Authorware file has the creator of APr7. Generally, it is safe to jump out from one Authorware piece to another without specifying the creator but you may want to include it just in case.

Also, when jumping out to another .apm file, you need to specify the .apm file name and path as both the `program` and `document` parameters. If you fail to do this, the Mac may ask you to locate the application it should use to launch the file with the .apm extension.

So, for example, use:

```
JumpOutReturn(FileLocation ^ "MyProgram.apm", FileLocation ^ "MyProgram.apm", "APr7")
```

Do not use:

```
JumpOutReturn(FileLocation ^ "MyProgram.apm")
```

When jumping out to a non-Authorware application, you **must** include the creator type. For example, Adobe Acrobat (PDF) documents displayed with the Adobe Acrobat Reader must be launched using the creator type of CARO. You do not need to specify the program name in this case, just the document name. Example:

```
JumpOutReturn("", FileLocation ^ "MyDocs:" ^ PDFFileName, "CARO")
```

So how do you know the creator type for a given application? There are several tools available for identifying the type and creator for a given file format. *FileTypeer* is one such utility: <http://dazuma.freeshell.org/filetyper/>

You can also find creator type lists on the Internet, such as:

<http://www.ntnu.no/itea.info/mactorget/pages/brukerstotte/infodocs/ct.html>.

Graphics

Authorware supports the following graphic formats on the Mac:

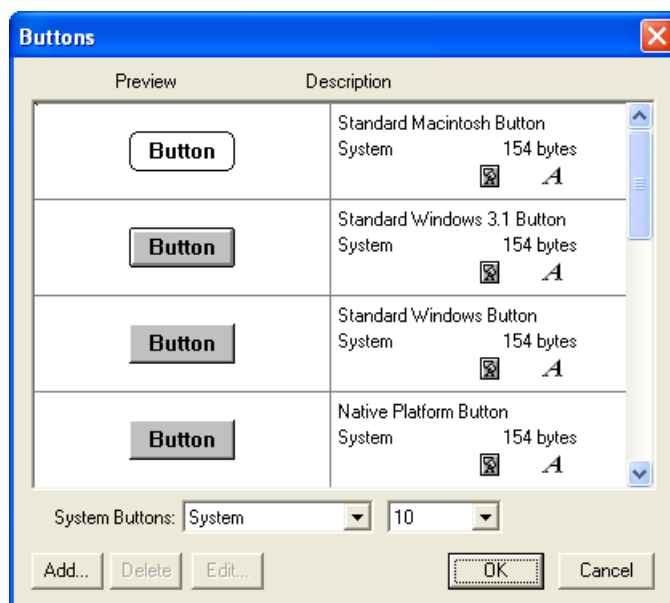
- BMP, DIB, RLE
- GIF
- JPEG
- LRG
- PSD
- PICT
- PNG
- TGA
- TIFF

On the PC, Authorware treats pure white (RGB: 255, 255, 255) as the color to remove when setting an image to matte or transparent. However, on the Mac, it appears (we don't know this for a fact) that the transparent color can be any color. Suppose you draw a borderless box and apply a fill pattern to it with the fill being gray and you erroneously set the box to transparent mode. On the PC, the box will appear with the gray fill pattern. But on the Mac, the box will not appear because the Mac has managed to set the gray to transparent. In short, if you see strange effects with your imported or drawn graphics on the Mac but these effects don't appear on the PC, check the mode setting of these graphics and change it accordingly.

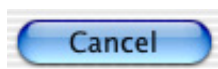
Buttons

You can import your own graphical buttons into Authorware's Button Editor. Stick to graphic formats that work well on the Mac (refer back to the *Graphics* section).

You can also use standard button styles included with Authorware. We recommend using the "Native Platform Button" when using standard button styles. Authorware will automatically show the standard Windows button style when your program is run on the PC and the Classic (white oval) style will automatically appear on the Mac.



The Button Editor's button selection dialog.



An OS X-style button; not available as a standard button style in the Button Editor.

Unfortunately, the new OS X style of button shown here will not appear if you select "Native Platform" or "Standard Macintosh"; the old style oval appears on the Mac instead.

Movies

Authorware on the Mac supports the following movie formats:

- Animated GIF
- BMP Sequences
- Flash (swf)
- MPEG (under Classic and via QuickTime sprite)
- QuickTime (.mov)

The Mac does not directly support AVI, WMV and RealMedia files. However, later versions of QuickTime can play AVI files. Depending on the version of QuickTime installed, some added support files may be needed. In this case, QuickTime may need to access the Internet to install these components before playback can occur.

Generally, when creating cross-platform pieces, many developers elect to use the QuickTime format because QuickTime is already available on Macs (QuickTime, after all, is an Apple Computer software product). Keep in mind though, many PCs do not have QuickTime pre-installed. If you digitize your movies in QuickTime's .mov file format, you will need to have the PC version of your Authorware program check for the existence of QuickTime (and possibly the version number) and either link to Apple's web site to download it or enter into Apple's free (but somewhat paperwork-intensive) license agreement and then distribute QuickTime on your CD. Use the QuickTime sprite icon to implement QuickTime movies in Authorware (select `Insert > Media > QuickTime...`).

Bitmap sequences may cause your piece to crash under OS X 10.0.0-10.3.3 if you set the mode to Transparent or Matted (Opaque should work fine). This is a known issue apparently unrelated to Authorware but instead due to a bug in OS X. Supposedly, it is fixed in OS X 10.3.4 though we haven't been able to test and confirm this. In any case, assuming OS X users will be running OS X 10.3.4 may not be the best approach. Instead, you may want to use a different supported movie format or set your Bitmap sequence's mode to Opaque. (You may need to render each bitmap over the portion of your application's background color or image.)

MPEG movies imported into the movie icon appear to playback fine under Classic. However, under OS X, you may find that an error message results, indicating that a driver is missing. Therefore, for pieces that must playback under OS X, we recommend using the QuickTime Asset Xtra to playback MPEG movies (use the `Insert > Media > QuickTime` option in Authorware). MPEG performance using the QuickTime Asset Xtra appears to work equally well under both Classic and OS X.

Audio

All Macs are sound-capable so if you want to include sound with your piece there's really no need to use Authorware's internal `SoundAvailable` variable to check for audio capability. It's important to do so on programs deployed on Windows machines, however, because if the user doesn't have a sound card installed and your program attempts to play a sound, the user will receive a somewhat cryptic error message.

The Mac supports the following sound formats:

- AIF/AIFF
- AIFF with IMA compression
- AIFF with MACE compression
- MP3
- PCM
- Shockwave Audio (SWA)
- WAV

Despite what the Mac Packager's Help file states, the Authorware 7 Mac Packager does not support Voxware (VOX) files under either Classic or OS X (see Macromedia's TechNote: http://www.macromedia.com/go/tn_18950).

PDF Files

If you want to embed one or more PDF files into your application, have a look at Integration New Media's *Impressario for Authorware Xtra* (<http://www.inm.com>). You will likely find that Impressario's supporting PDF library files require too much disk space for most applications deployed via the Internet or your intranet but this may not pose a problem for CD or hard disk delivery. We have tested this Xtra and have found it works extremely well cross-platform. Plus, it's one of the very few third party Xtras available where a single Xtra works under both Classic and OS X.

If you want to jump out to a PDF viewer, you can use Authorware's `JumpOutReturn` function. As noted earlier in this document, `JumpOut` and `JumpOutReturn` on the Mac require that a *creator type* be specified. For PDF files, the creator is `CARO`. So the syntax might look something like this:

```
JumpOutReturn("", FileLocation ^ "MyDocs:" ^ PDFFileName, "CARO")
```

In the past, end users would have needed Adobe's Acrobat Reader (now simply called 'Adobe Reader') in order to see the PDF document. In Classic, this generally still holds true. However, OS X ships with a utility called *Preview* that can display PDF files. For many users, Preview – rather than Adobe Reader – is the default PDF viewer. The downside to this is that Preview, to our knowledge, still does not support forms in PDF documents (you can see the form but you can't enter anything into the form fields).

You can determine which application on the Mac is associated with PDF documents by using the `baFindApp` function in the BudAPI Xtra (<http://www.mods.com.au>):

```
PDF_Viewer := baFindApp("CARO")
```

The value returned to the custom variable `PDF_Viewer` will be the full path to the program that can display a PDF document. If a null string is returned, no application could be found on the user's machine that could display a PDF.

When you jump out to a PDF document and Acrobat Reader displays the document, the user can elect to minimize Reader to return to the Authorware piece. Alternately, he/she can click the Authorware application's icon in the 'dock' to bring the Authorware program back to the front. In either case, if Reader is not closed, the next time a PDF document link is activated from within your application, it will not load because Reader already has a document open (this problem only occurs on the Mac, not on the PC). There are ways around this. Using BudAPI functions, you can perform the following steps in successive calls:

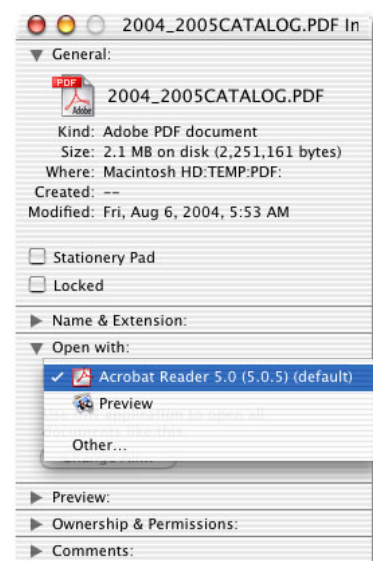
1. Wait for the Reader window to open;
2. Get Reader's window handle;
3. Wait for the Authorware application to regain focus;
4. And then close the Reader window programmatically via its window handle.

The code for this is beyond the scope of this article. If you would like this code (assuming you already have BudAPI's Xtra), feel free to contact Steve Gannon at steve@gantekmultimedia.com. You will also likely find it in the BudAPI forum archives (accessible from <http://www.mods.com.au>). Alternately, you can use AppleScript to close the Reader window on the Mac. For detailed instructions and a sample script file, see TechNote http://www.macromedia.com/go/tn_14695.

Changing a File's Program Association

Suppose you want to change the program used to open a file. As in the *PDF Files* discussion above, if you wanted to associate PDF files with Adobe Reader rather than OS X's Preview program, you can do so as follows:

1. Browse to any file with the extension `.pdf` on your Mac hard disk.
2. Single-click on the `.pdf` file.
3. Pull down the File menu. Select Get Info.
4. Expand the Open with: option.
5. Change Preview to Adobe's Acrobat Reader.
6. Close the Get Info dialog.



Changing the program used to open a file.

Changing a Program's Icon

There are a few utilities available for editing your program's .apm file icon (such as *Icon Machine*: <http://www.uncommonplace.com/shareware/iconmachine.html>). But you can also change the icon on the Mac without such a utility.

First, create your icon graphic using your preferred graphic editor (e.g. Photoshop). Although we typically think of icons being sized to 16x16, 32x32 or 48x48 pixels, on the Mac icons can be larger. Go ahead and make your graphic 128x128. The Mac will scale it to the various supported icon sizes automatically. Since this graphic will be used on the Mac, we recommend saving it as a PICT (.pct) file. If you develop the graphic on the PC, copy it over to the Mac. Now follow these steps:

1. Browse to your icon graphic (.pct) file.
2. Double-click the graphic to open it. On many Macs, the graphic may be displayed via the Simple Text application. (Note: Mac Classic may have to load on top of OS X before Simple Text opens and the graphic appears.)
3. Copy the graphic to the clipboard:
 - a. Pull down the `Edit` menu.
 - b. Select `Select All`.
 - c. Pull down the `Edit` menu again.
 - d. Select `Copy`.
4. Browse to your packaged Authorware file (.apm file).
5. Single-click on the .apm file.
6. Pull down the `File` menu. Select `Get Info`.
7. Single-click on the Authorware icon in the upper left corner of the dialog (it will become highlighted). Pull down the `Edit` menu and select `Paste`.
8. Close the `Get Info` dialog.

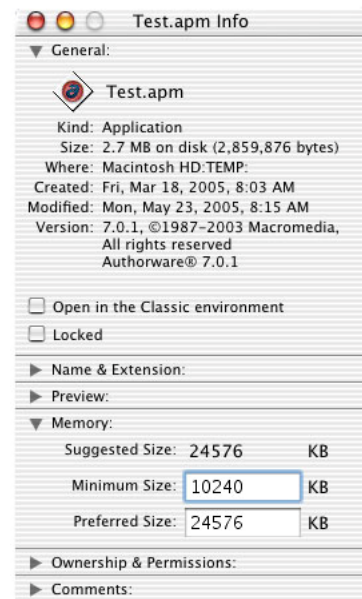


In the Get Info dialog, click on the icon and then paste a new image to change the icon.

Memory

OS X is based on UNIX (<http://www.apple.com/macosx/features/unix/>) so it handles memory differently than OS 8 and 9. If the Macs you are deploying to are running OS X, there is a good chance those Macs already have enough RAM installed to handle your application. OS 8 and 9 on the other hand assign a predefined amount of RAM to the application. If your program fails to run in Classic environments (either when the Mac is booted in OS 8 or 9 or when Classic is loaded on top of OS X), you can increase the amount of RAM assigned to the .apm file. This memory assignment will 'stick' when you deploy the application to other machines. Follow these steps to change the memory assigned to your piece.

1. Package your piece.
2. Since OS X does not use memory in the same way as Classic, in order to change the amount of memory assigned to your Authorware piece, you must first launch Classic on top of OS X. Alternately, you can reboot in OS 9.x as follows:
 - a. Click the apple icon in the far left of the desktop's title bar.
 - b. Select System Preferences.
 - c. In the System category, launch Startup Disk (this icon is in the lower right corner of the System Preferences panel)
 - d. Click Mac OS 9.x... and click Restart.
 - e. Click Save and Restart.
3. Browse to your packaged Authorware file (.apm file).
4. Single-click on the .apm file.
5. Pull down the File menu and select Get Info and then select Memory.
6. Increase the values in the Minimum Size and Preferred Size fields. As a rule of thumb, we generally use the value in the Preferred Size field for the Minimum Size and then double or triple the Preferred Size value.
7. Close the Get Info dialog.
8. If in step 2 above you rebooted in OS 9 rather than loaded Classic on top of OS X, you'll want to reboot back in OS X:
 - a. Click the apple icon in the far left of the desktop's title bar.
 - b. Select Control Panels and then select Startup Disk.
 - c. Select Macintosh HD: System to the left of Mac OS X 10.x and click Restart.



Use the Get Info dialog to increase memory to applications run in Classic.

We have also found that pre-OS X programs run more reliably when the computer is rebooted in OS 9 rather than trying to run Classic on top of OS X.

One other thing to try when experiencing memory errors (and all else fails) is to change the monitor's color depth setting from millions to thousands through the Control Panels.

Printing

Authorware's `PrintScreen()` function does not appear to work under OS X. No error messages appear; the function simply fails to do anything.

As an alternative, you may want to consider *Print-O-Matic* (www.printomatic.com). This cross-platform Xtra does a great job of printing formatted reports, charts, graphics, etc. It takes some effort to get the hang of the scripting (it's primarily designed for Director though Authorware is supported too), and you may find it a bit pricey, but this Xtra does work on Windows, Mac Classic, and Mac OS X. Printomatic requires a separate Xtra for each of these operating systems.

MoveCursor() Function

It has been reported that Authorware's `MoveCursor()` function does not work on the Mac. If you need this functionality, you'll need to use a cross-platform Xtra such as BudAPI (<http://www.mods.com.au>) to accomplish this. (BudAPI includes a similar function called `baPlaceCursor()`).

Key Names

1. In the past, Authorware mapped the Windows CTRL key to the Mac's CMD key but that no longer appears to be the case. When you want to use these keyboard alternatives, you may want to specify both, such as `CtrlX|CmdX`.
2. The ENTER key on the PC can be named `Enter` or `Return`. Look at this key on your PC keyboard and you'll likely see it labeled 'Enter'. On the Mac, this key is labeled 'Return'. We have found that the key name `Enter` may not always work on the Mac (such as when specifying `Enter` as a keyboard alternative to activating a button or as a text entry field's action key). Therefore, we recommend using `Return` as the key name for both the PC and the Mac.

Fade Transition

The 'Fade In' and 'Fade Out' transitions can cause OS X and OS 9 pieces to crash. Therefore, you should not use these particular transitions (see Tech Note http://www.macromedia.com/go/tn_19056).

Target Area Interaction

Macromedia acknowledges via TechNote http://www.macromedia.com/go/tn_19199 that there is a bug with Target Area interactions under OS X. If you set the “On Drop” option to “Put Back”, the object moved will just pop back to its original location instantly rather than animate back to its original position. The suggested workaround is to change the “On Drop” setting to “Leave at Destination” and then use a motion icon to animate the object back to its original position. This problem does not occur under Classic.

Web Player Issues

1. When running your program on OS 8 or 9, Authorware pieces that reference external media using a URL and that are packaged for the web will crash the Authorware Web Player. To work around this issue, download the media first. Refer to TechNote http://www.macromedia.com/go/tn_19263 for more information.
2. ReadURL function calls that include JavaScript in the URL parameter, such as `ReadURL("JavaScript:xyz", [timeout])`, will fail under both Classic and OS X, regardless of the browser used.
3. PostURL and Knowledge Track do not work in the Macintosh version of Internet Explorer (however, both should work in the Macintosh version of Netscape Navigator).
4. NetDownloadBackground may not work reliably in Internet Explorer on the Mac (however, it should function in Netscape Navigator).
5. Running a web-packaged piece in Internet Explorer on the Mac also has these limitations (none of these should be a problem with Netscape Navigator):
 - a. Performance may not be as good in Internet Explorer compared to Netscape.
 - b. You must use the EMBED tag in an HTML file to open an .aam file; you can't open the .aam directly in the browser.
 - c. You can't run a web-packaged piece from local media such as a hard disk or CD-ROM. It will only run from a web server.
 - d. Advanced Streamer will not function on the Mac's version of Internet Explorer.
6. On the Mac, the browser does not accept the inPlace window option in the <embed> tag of the HTML file used to launch your piece. Therefore, you cannot have your piece appear *within* the browser display. Instead, use onTop (or onTopMinimize). This will cause your piece to appear as a windows separate from the browser. (NOTE: This caveat was true for Classic .aam files; we have not verified that this problem still exists with the latest browsers under OS X. See TechNote: http://www.macromedia.com/go/tn_12541).

Other Sources of Crashes

Below is a list of some other possible causes of program crashes:

1. Corrupt Fonts:

- Try changing fonts.
- Try using fewer fonts.
- If you have DiamondSoft Font Reserve or Extensis Suitcase, be sure these applications are closed before running your application
- See Flash TechNote http://www.macromedia.com/go/tn_15830 to identify corrupt fonts.

2. Permissions Issues:

- If you're trying to write to a file, be sure the file has read/write privileges.
- If running from a LAN, be sure the .apm file is set to read-only.

3. Conflict With Other Applications

- Close any other open applications before running the program.

4. Preferences File:

- There is a preferences file generated for your Authorware .apm application. Under OS X, this file is located here:
Users:{UserName}:Library:Preferences:{ApplicationName}.plist. There's a chance this file may become corrupted. Move the file to a different directory before launching the Authorware program. Authorware will automatically create a new preferences file when your application is launched.

5. Extension Conflict in Classic:

- See Tech Note: www.macromedia.com/go/tn_16658.

Other Suggestions for Correcting Problems

The following courtesy of Michael Mizen, a frequent contributor to the Aware Listserve:

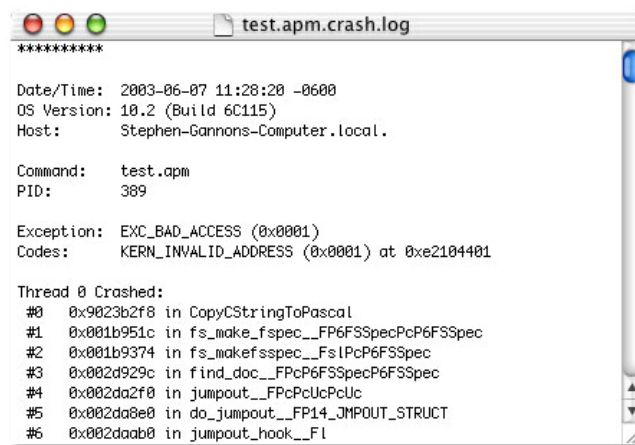
1. Remove the RunAPM file in the System Folder/Preferences location on OS 9. In OS X, do the same in user/library/application support/macromedia/authorware.
2. Rebuild the desktop. To do this, hold the Option+Command (apple logo) keys on startup.
3. Version 9.2.2 is the last release in the Classic series. Versions 9.2.0 and 9.2.1 had issues with OS X. If you have problems you can't rectify when running under OS 9, try upgrading to version 9.2.2 (free download from Apple's web site).

Crashes – Behind the Scenes

If you have heeded the suggestions and caveats in this document and still experience inexplicable crashes, you may be able to learn more about the errors being generated even if no error message appears. OS X includes an application called *Console* that monitors errors. To turn on 'crash logging', launch /Applications/Utilities/Console. Then, perform these steps:

1. Select Console and then Preferences.
2. Select the Crashes tab.
3. Check Log crash information in ~/Library/Logs/.
4. The log file is: /Users/{USERNAME}/Library/Logs/CrashReporter/{Authorware APM Filename}.crash.log.

You can open a .log file using a plain text editor to view all of the technical information about the crash (albeit, unless you're a Mac programmer, it's all rather cryptic).



Sample crash log.

Sometimes an error message will appear and provide you with an error code. You can search the Internet for information about the error code (be sure to include the negative sign, if applicable, when entering the error code in your search string). Apple provides lists of the more common error codes on these pages:

Codes -299 to -5553

<http://docs.info.apple.com/article.html?artnum=9804>

Codes 0 to -261

<http://docs.info.apple.com/article.html?artnum=9805>

Codes 1 to 32767

<http://docs.info.apple.com/article.html?artnum=9806>

Other Internet sites may be able to provide more detailed information about a specific code than what you might learn on Apple's site (e.g. <http://www.appleerrorcodes.com/>).

Creating a PC/Mac Hybrid CD

A 'hybrid' CD is generally one that contains a PC-formatted partition and a Mac-formatted partition on a single CD. With a hybrid CD, you can distribute a single CD that will run on both Windows and Mac operating systems. PC users see just the PC files and Mac users see just the Mac files.

In this section we'll explain how to create a hybrid CD that not only contains separate PC and Mac partitions but also a *shared* partition. A shared partition enables you to include files (such as external digital movie and audio files) that can be accessed by both the PC and Mac partition. Because external media typically occupies a large proportion of disc space compared to the application itself, the ability to share files across partitions can be very valuable.

Imagine you have 400 MB of QuickTime movies and 75 MB of WAV files. Imagine further that your application, including Xtras and other utilities, requires 5 MB. The total disc space required for each platform would be roughly 480 MB. If you made two sets of media files, one for the PC and another for the Mac, your disc space requirement would be approximately 960 MB, which exceeds the capacity of a standard 650 MB CD. You would need two separate CDs, one for the PC, another for the Mac platform. Distribution costs would rise considerably. But if you could share the 475 MB of movie and audio files, then you could fit all of the files for both platforms on one CD.

While it is possible to create a PC/Mac hybrid CD on the PC (see, for example, <http://www.macdisk.com/macimgen.php3>), hybrids are more often created on the Mac.

One of the more popular programs used to create hybrid CDs using the Mac is *Roxio's Toast* (<http://www.roxio.com/en/products/toast/index.jhtml>).

The steps below for creating a hybrid CD with a shared partition are specific to Toast 5 but you should find the process to be similar if you're using another version of Toast.

1. Create three folders on your Mac: one to hold the PC-only files; one to hold the Mac-only files; and one to hold any shared files.
2. Copy the PC-only files over to the PC-only folder on your Mac. Place the Mac-only files into the Mac-only folder. And place the shared files (typically documents and other external media files) in the shared folder. Remember, the Mac requires its own set of Xtras – **do not attempt to share the Xtras folder!**
3. Launch Toast.
4. Select the Other icon. 'Custom Hybrid' now appears near the top of the panel in the center of the Toast screen.
5. Click on Select ISO... The files for the PC partition will go in here. Drag the files from your PC-only folder into this panel (don't drag the PC-only folder itself...just the contents within it). NOTE: If you want the CD on the PC platform to auto-start (assuming the user has not disabled auto-start), be sure to include an appropriate `autorun.inf` file in the root of the PC partition.
6. Click on the title next to the CD icon and name it whatever name you wish to call the PC CD volume.
7. Click the Settings tab. Pull down the Naming list and change it to Joliet. In most cases, you should not need to change any other settings so just click Done.
8. In a moment you're going to work with the Mac side of the CD, but first, you need to create a *temporary partition* to hold the Mac and shared files. To do this, pull down the Utilities menu and select Create Temporary Partition. In the Name field, type the name you want to apply to the Mac CD volume; this is the name that users will see below the CD icon when they insert the CD into the Mac's CD drive. The size of the temporary partition should remain the default (650 MB assuming you're using a standard 650 MB CDR). The last field should indicate the name of your Mac hard drive. Click OK. It will take a minute or so for the temporary partition to be created. When the process is complete, you will see a CD icon on your desktop and it will be labeled with the name you assigned to it.
9. Now, browse to all of the folders/files within your Mac partition. Drag these files to the temporary partition CD icon on your desktop. **Do not drag the files to the Toast interface!** And do not drag the Mac-only folder itself...only the contents within it.
10. Next, browse to the folder on your Mac hard disk that contains all of the shared folders/files. Just as in step 9, drag these items to the temporary partition CD on your desktop.

11. In the Toast interface, click on `Select Mac . . .`. A dialog will appear prompting you to select the Mac partition for your CD. By default, the name of the temporary partition you created in step 8 should be highlighted; if another drive (such as your hard disk or another CD) is selected, click on the temporary partition.
12. Programs running under OS 8 and 9 (i.e., Classic) can be made to auto-start. (Unfortunately, you can't use Toast to make an OS X program auto-start.) If you want your Authorware .apm file to auto-start under Classic, check the `AutoStart` checkbox at the bottom of the Toast dialog. Browse to – and select – your .apm file being certain to select the .apm file in your temporary partition (the file selected for auto-starting must be at the root level of the temporary partition and the file name must have 11 or fewer characters). Click `Choose` to close the auto-start file browse dialog and then click `OK`.
13. Now we have the PC partition set up as well as the Mac partition. It's time to address the shared partition. This is how the shared partition works... the shared files are stored on the temporary partition for the Mac. You will place these in the PC partition. Toast will recognize that these files already exist in the Mac partition (from step 10) so Toast will only write them to the CD once but will expose them to both the Mac and PC partitions. To begin, click on `Select ISO . . .` once again in the Toast interface. Be sure the `Files` tab is selected. Now double-click the temporary partition CD icon on your desktop. Drag only the shared folders/files from the temporary partition onto the open `Files` panel of the Toast window. When you do this, you will see that the folders and files you want shared appear in the `Files` panel of the interface but that the disc space needed is 0 K to indicate that no extra disc space will be required for these files to be shared on the PC. Click `Done`.
14. Back at the main Toast screen, you should now see under the ISO category: `{X} MB + {Y} MB Shared Data'`. That's it! You're all set to record so insert a blank 650 MB CDR and press the `Record` button. Follow the instructions in the prompts that follow.

The CD that results from the steps above should work on the Mac and separately on the PC. When browsing the CD contents on the Mac, you should only see the Mac files plus the shared files. When browsing on the PC, you should only see the PC files plus the shared files.

Need More Authorware Cross-Platform Help?

Below is a list of some resources that may provide you with additional cross-platform assistance:

Authorware Developer Center: *Authorware Playback on the Mac*, by Michael Mizen:

http://www.macromedia.com/devnet/authorware/articles/mac_playback.html

Aware Listserve Archives:

<http://listserv.cc.kuleuven.ac.be/archives/aware.html>

Authorware Newsgroup:

<http://www.macromedia.com/cfusion/webforums/forum/index.cfm?forumid=10>

Authorware Support Center:

<http://www.macromedia.com/support/authorware/>

Apple Computer's Support Web Site:

<http://www.apple.com/support/>

Version History

Version 1.1

Movies:	Added discussion of Bitmap sequence issue.
Buttons:	This new topic added.
Web Player Issues:	Added some limitations with Internet Explorer on the Mac.

Version 1.2

Target Area Interact.:	Added discussion of bug in target area interactions.
Web Player Issues:	Added caveat regarding the inPlace window option on the Mac.
Movies:	Added limitation of playing MPEG movies via movie icon in OS X.

Questions, comments, or suggestions for additions to this document are encouraged. Please submit to: steve@gantekmultimedia.com.