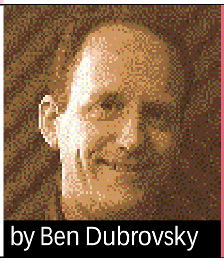
Mac & Windows Compatibility



by Ben Dubrovsky

While the Mac/PC wars rage on, multimedia authors (acting as a kind of Geneva convention) wrestle with the responsibiltiy of developing applications that will work on both platforms.

Ever hear the line, "Standards are such a good thing we ought to have lots of them?" Makers of paper, fan-belts, removable media, and SCSI cables all share a common problem with multimedia authors: what standard should we build for? In our world there are two primary standards, known colloquially as Max and Windeaux. As with smoke and fire, these standards invite a glance over the shoulder for the ensuing religious wars.

We, the authors, must rise above the fracas and build for both platforms. To do so, we must exchange all notions of what we think is right for the realities of a bifurcated user community.

But why are there differences, and what are they? As we all know, the Mac and Windows machines use different computer chips with different architectures and assembly languages. Dealing with differences in language is a relatively straightforward task that has a number of standard solutions. The differences between architectures are far more troublesome. Issues such as graphics displays, mouse input, and interrupts are handled in completely different ways. Therefore, we need different strategies for low-level problem solving on each platform.

The typical strategy for cross-platform development issues is to identify all programmable elements that relate specifically to the hardware or operating system in question and centralize them. Code centralization is critical; and cross-platform development is another example of why code should be written in layers. The bottom layer should deal with all system level issues in a centralized way. The rest of the program should ask the system interface layer to get things done. Then, the bottom layer would be the only one needing attention in a cross-platform development effort.

The key words to cross platform development are common interface and experimentation. Whatever the environment, be sure to test everything thoroughly. Use the following checklist to find red flags when developing cross-platform applications.

Illustration by Ruth Liebman.

File system

The file system is one of the most obvious areas of difference to most people. One the one hand, there is the eight-dot-three file naming convention for Windows, and the anything goes 32-character naming convention for the Mac.

Likewise, there is a different delimiter used in path names on the two platforms. Macs use a colon (:) and PCs use a back slash (\). If you're authoring for the Web, UNIX machines use a forward-slash (/).

Perhaps the most subtle difference between the platforms' file systems is the way in which text files are stored. Macs use just one character, the Carriage Return, (Control-M, or ASCII 13), while PCs use the Carriage Return followed by a Line Feed (Control-J, or ASCII 10).

What to do:

Use file names that follow the most restrictive eight-dot-three file naming convention from DOS. Keep the file delimiter stored in a global variable that gets based on the operating system being used. Pass all strings read out of a text file through a clean-up handler which strips out Carriage Returns, Line Feeds, and all other unwanted characters.

Display

Certainly one of the most annoying problems with cross-platform development is the issue of color palettes. Mac and Windows machines use different system palettes and use them in different ways. Macs allow users to change 254 of the 256 colors on a palette. Windows machines reserve the first and last eight colors in the palette, and Windows 95 uses two more colors on each end for its desktop. In addition, different Windows display cards have different gamma settings, thereby making the same graphic seem darker or lighter depending on the monitor.

What to do:

Keep in mind that any time a palette is changed, a palette flash will ensue. First, use 16-bit or greater color depth and avoid the problem altogether if appropriate. Second, use a Macintosh System palette and tolerate a palette flash on Windows machines when the program starts. Third, use one of the Windows system palettes, and push the trouble off to Mac users. Fourth, use one or many custom palettes and manage them carefully. In all of these cases, care must be taken when switching applications because the new application will also have its own palette.

There are only two ways to deal with the monitor gamma issue. The first is to write an XObject, DLL, or XTRA to change the particular monitor's gamma. The second is to cheat with palettes. For each palette used, create a couple of others by making all of the colors just a bit brighter. In either case, ask the user for a preference, and use that preference throughout. Whatever the choice, be careful. Palette barf left on the screen is ugly.

Fonts

Not only do Mac and Windows machines use different types of fonts, but the fonts have different names, different kerning, and different just-about-everything-else-you-can-imagine. To top it off, fonts often have heavy licensing fees associated with them.
What to do:
Again, be careful and conduct thorough investigations. Avoid rules of thumb because they might work for some fonts, but not others. Some programs, like Director, let you control the way fonts are mapped and how a font name and point size on one machine should be drawn on another. These mapping files can be adjusted, but experimentation is still the key. Manual adjustments can be made to line-height and character spacing as well.

Licensing is another issue, though not for bit-mapped artwork. The Mac has an advantage when it comes to distribution since some font manufacturers will allow bit-mapped fonts to be included in Director projectors if they are unseen resources. There is no such luck on the Windows side because fonts need to be installed in the system and can't be coupled with one particular application. Since fonts can be expensive to license, think twice before using one that needs to be installed.

Sound

Sound file formats are fairly consistent across machines. AIFF and WAV files will get the job done on both sides. The only issue of concern here is that of sound cards for the PCs. There are a number of cards out there, each with its own peculiarities.
What to do:
If you use the standard 8- or 16-bit sampling rates recorded at 11, 22, or 44 kHz, you shouldn't have a problem with playback. Problems occur when sounds must be mixed together during playback. Director fixes this byoffloading the poroblem from the soundcard and doing the mixing itself in software. Unfortunately this solution has a price in terms of a delay. If possible, try not to mix too many channels.

Digital video

Macs and PCs have different basic standards for digital video. How do we get QuickTime and AVI to work on both platforms?
What to do:
The news really isn't that bad—it's not impossible, just tedious. QuickTime and AVI are simply file formats, and file formats can be translated. The real problem is that we don't necessarily want a zillion bytes of digital video to exist in two different files. There are two dimensions to solving this issue. First is the file format, as mentioned above. The second is that of the codec, or compressor. QuickTime and AVI are simply large-scale data retrieval engines that grab bunches of data out of movie files. That data is then passed through a codec where it is converted into video.

First the file format issue. QuickTime for Windows is more reliable than AVI for Macintosh, so use the QuickTime file format. Also, keep in mind movie data that originates on Macintoshes must be flattened and single-forked before it can be used on a PC. There is a Mac utility called Convert to Movie or Movie Converter that will help you with this. In addition, Premiere lets you save movies, "Playable on Other Computers."

Compressors are the other issue. Some, for animation and graphics, already exist on both machines. They should be used for button animations or anything that is relatively small and needs to be seamlessly dropped into a background image. However, the compressor typically used for video is Cinepak on the Mac and Indeo on the PC. Fortunately, both Cinepak for Windows and Indeo for Mac compressors are available. Make a choice, and force one of your constituencies to use the other.

Authoring code

Lingo and other scripting languages are, as we've said in the past, programming languages. As such, they tend to take advantage of either the operating system or hardware being used.
What to do:
Write your Lingo or other authoring code in such a way as to keep all references to the operating system and the hardware in one place. This includes references to any subject discussed above. Just play it safe, and add an extra layer of code and checks. Even subtle things can cause headaches. For instance, the time of day might be returned differently on the two platforms, but if there is a central place to do all time getting, it will be easy to work around.

Code extensions

PCs and Macs run different code. That means that code extensions, XObjects, XCMDs, DLLs, and XTRAs, that work on one machine won't work on the other.
What to do:
There are two ways around this. You can either use third-party code or a custom code. There is only one real hope for third-party code, and that is if the manufacturer makes code extensions for all machines in question. If not, hope that another vendor makes a product similar enough for the unserviced platform and code around it in your authoring environment. For custom-written code extensions, get a good cross-platform C compiler, and write and test two versions of your code. DV

Ben Dubrovsky is a multimedia producer and programmer from Brookline, MA. Some of the ideas in this article will also appear in a forthcoming textbook from Prentice-Hall that he is co-authoring on Director Academic 5.0. Reach him at dubrovsky@dv.com.

For more information about this and other subjects check out our Web site at http://www.dv.com. Reach us at letters@dv.com or write to DV Letters, 411 Borel Ave., Suite 100, San Mateo, CA 94402 or fax 415-358-9865.