

Larger Market Share! Greater Revenue! Universal Access!

Why Isn't Everyone Cross-Platform?

Everyone's software is not cross-platform for the same reason that everyone is not multilingual and using Quadra 950s - because it is difficult and expensive. The benefits are obvious but what about the pitfalls?

This survival guide is written to suggest specific areas you might want to think about if you are planning a cross-platform project. Even if you are currently not planning a cross platform effort, you might want to follow some of these guidelines to leave yourself the option later. This guide is for developers using higher level authoring systems such as Hypercard or Toolbook rather than C and is particularly directed to projects which contain a great deal of information such as text, pictures, and sounds.

Content Considerations

One great painful truth of multimedia authoring is that **tracking and maintaining the raw content** of a project is often a larger and more painful task than creating the engine and the structure which displays this data. Hundreds or thousands of hours can go into assembling and editing the raw content. Protecting that investment is critical to the profitability of a project since duplicating this effort can make a project's profits evaporate.

Perhaps the most obvious and yet most important content issue is a **usable naming convention**. Grim as it might appear, keeping names to eight characters (with, if needed, a 3 character extension following a period) will help avoid future problems. These characters should be 'vanilla' characters - the 26 letters of the alphabet (don't assume that differences in upper and lower case are significant), 0-9 and the underscore character '_'. You're allowed no special characters, no spaces, no formatting and no fun.

As you might guess, there is a lot of opportunity for creativity in trying to make a usable naming convention work within these guidelines. It helps if you think of it as a challenge rather than as a constraint. If DOS names on a Mac seem like death at an early age, you may want to use Mac names which would lend themselves to automatic conversion to DOS-tolerant names at a future date. These caveats for filenames also apply to directory structures with the additional gotcha that ISO-9660 directory structures can only be 8 levels deep.

After you have taken care of what your files are called, its time to worry about what is inside of them. Let's start with text:

If you are dealing with formatted text, you'll need translation capability between the two environments and authoring systems on each end which can handle these formats. Yet even plain text has problems which need to be addressed in a cross-platform project.

The first major consideration for moving between Mac and Windows/DOS is that **line feeds are required** for the text on the non-Mac side. Many word processors and translation programs, both public domain and commercial, can help with this step. A slightly more subtle problem can appear if you are using **special characters** such as 'ü' or 'ñ'. These characters are from the extended character set and **do NOT translate transparently**. As mentioned above there are utilities which can help with the remapping. Look for the ones which will allow you to do this type of translation in a batch mode. Such a feature is very handy when you have thousands of files to translate.

Translating graphics between platforms deserves its own guide. Be clear about the level of graphic support your product needs on both sides. All of the nail-biting about screen design, such as requiring or optimizing for such elements as screen size (512x342?,640x480?,640x400?) or color level (B&W,4?,16?,256?), are at least doubled as you move to a new platform. Do not assume any relationship between pixels and inches and don't assume that all pixels are square.

Palette control is a problem on single platform software requiring you to either standardize all your images or design around the 'flash' that can happen as the software changes palettes. The problem expands as you move to a two platform product. The best approach is to standardize on a set of tools and procedures which you have seen work (with your own eyes!) and then follow this standard religiously. Jumping between paint or translation programs can result in an unrealized and undesired multi-palette situation.

Keeping with a standard format and/or process is a good idea with video and audio. If you are digitizing video using Quicktime, **you should consider making all of your movies cross platform** (single fork) so they are usable unchanged on both platforms. If you are using a format other than Quicktime for audio, consider using a file format such as AIFF which is a wide-spread standard. This will allow you to convert it to either WAV or SND formats without an interim step. It can be a frustrating experience to have to translate many files into an interim format because there is no way to directly translate from file format A to C without going through B. Always strive to be in the middle of the road if you might need to move left or right.

Authoring System Considerations

Religious wars are fought over what authoring system is best. Ease of cross-platform authoring is one of many considerations. Some products have **binary compatible files** you can open directly on either platform. If you can live with the inherent **limitations** of these systems, buy a copy immediately. This is the easiest solution.

Next in line are systems which let you **author on one platform and have a player for the other**. Again, you will often have to trade off performance or cost for this convenience. Also, there may be system-specific constraints with a dual platform authoring environment. Yet this will still be easier than the lowest rung on the ladder.

The lowest level is using two **different systems** and trying to create two separate products which are visually and functionally as alike as possible. If you choose this solution, pay close attention to differences in the two systems. Can they both support multiple windows? Do they have the same color capability? Is I/O support equally capable? The differences you discover should play heavily in your design sessions. If you can create a design which works in both worlds which doesn't compromise your interface, you might save yourself hours of programming and headaches.

Hardware Considerations

Once your data and authoring software is cross-platform compatible, it is time to consider your hardware situation. If you have a large budget, then optimize for each platform. On the other hand, if you are working within a tight budget then duplicating expensive items such as large hard disks, tape backup drives and CD-ROM writers is probably not an option. You can avoid this problem by shopping carefully and making sure that **hardware is usable on both machines** or can be used remotely over a network.

Because of the huge size of multimedia projects, transporting files with floppies and "sneaker-net" is not really an option so give some thought to what you need for a **network**. Appletalk vs. Ethernet is a price vs. performance issue you will need to decide. If you only transfer items infrequently and in batches, you can get by with Appletalk and save yourself some money at the cost of some gritted teeth. Plan transfers for lunch time or overnight.

Further Resources

Many of the resources available are directed toward code level cross-platform issues such as the Bedrock system being developed by Apple and Symantec. However, there are place to find more information about general cross-platform concerns. You might want to look at the session on CD ROM Production and Optimization given at the World Wide Developers Conference this spring. This is one of the sessions that is available within the Members Only area on AppleLink in the WWDC folder. You can also contact some of the vendors of tools for information about their products (and the dirt on their competitors). Finally, participate in every multimedia forum on-line that you can. AppleLink has an excellent one as does Compuserve and America Online.

Final Thoughts

In closing, here is a hodge-podge of final thoughts. Whenever possible, try prototype designs AND processes. Build a tiny project with one converted picture and one sound rather than converting your 500 pictures and sounds and then trying to use them. Be rigorous about defining and following conversion procedures. Solve thorny design problems on the weaker platform and then elaborate on the more powerful platform as time allows. Create products which are flexible enough to exploit each platforms strengths. Don't force users on either side to learn the other platform's interface. Plan for a vertical learning curve.

Buy lots of aspirin.

© Apple Computer, 1993

Bryant Patten is a team leader at Bryten, inc. in Norwich, VT and has been producing educational software since 1984. Bryten's most recent cross-platform product is an interactive, college history text, *The Enduring Vision* published by DC Heath. He can be reached at (802) 649-3252 or AppleLink: BRYTEN

-----<<•>>-----

This information is provided by the product and/or service's developer, distributor or provider and constitutes neither an endorsement nor a recommendation by Apple Computer, Inc.. Apple assumes no responsibility with regard to selection, performance or use of this product/service. All understandings, agreements and warranties, if any, take place directly between the developer/provider and the prospective user. Prices are not set by Apple, but are established by the developer/provider.

-----<<•>>-----