The Benefits of Using PowerPlant™
For Mac® OS and Mac OS X Development

Freescale Semiconductor
Author, radar pangaean

Document Number: POWERPLANTWP
Rev. 0
11/2005
A developer using PowerPlant™ can write code once that targets both Classic Mac® OS and Mac® OS X from a single code base. This means the application can reach a larger market more easily, and get to the market more quickly. This paper discusses these and other advantages to using PowerPlant as the underlying application framework for Mac OS.
1. **Targets Both Mac Operating Systems**

   Anyone creating a Mac OS application wants to get it into the hands of the largest number of customers. At present, and for some time to come, a significant percentage of Macintosh® owners will rely on Classic Mac OS. Others will use Mac OS X. If you want to reach the largest possible market, your application must work on both systems.

   Of the existing professionally supported frameworks, only PowerPlant enables a developer to target both Classic Mac OS and Mac OS X with a single code base. Any other choice requires you to create, test and support two completely different code bases for your application.

   PowerPlant provides the same classes, services and API for both Classic Mac OS and Mac OS X development. This benefit alone would be sufficient to choose PowerPlant over any other option, but it is only the first of many reasons that justify a decision to use PowerPlant.

2. **Benefits of Using Any Framework**

   When choosing a toolset for Mac OS development, it is prudent to select an application framework as one of those tools. All Mac OS applications that conform to Apple®’s user interface guidelines use similar interface elements and interact with the system using well-documented techniques. In most cases, the developer does not gain significant benefit by rewriting and verifying the code to manage this part of the application. Using an application framework that provides this support as a foundation for further development allows the developer to concentrate on writing the unique part of the application.

   Modern application frameworks provide the initial benefit of containing robust code that perform the common functions in a Mac OS application. Using a framework decreases the risk of introducing errors into the application, specifically in the areas of creating and managing the user interface, interacting with the event loop, and in any of the other areas supported by the framework. Because many other developers have also used the framework, those parts of your application have already been subjected to rigorous testing before you write your first line of custom code.

   Most developers prefer to use a commercially supported framework instead of one that is public domain. Using a non-supported framework can lead to disasters as Mac OS evolves. The possibility that the framework will not be upgraded to work with the new OS, or will lag too far behind the OS, makes the use of a non-commercial framework too risky for professional development projects.

3. **Comprehensive Framework**

   PowerPlant provides a complete framework for Mac OS application development. The first few levels of required event processing are managed totally by the application classes, and the appearance and first level of processing for all user interface elements are implemented fully by the associated classes.

   PowerPlant also provides a visual tool for designing the User Interface, Constructor. Use Constructor to design and layout the menus windows for your application, including the placement and appearance of controls, edit fields and any other interface elements contained within the windows, then incorporate them into your program with just a few lines of code. The simplicity of this process, especially when compared to building the user interface programmatically, encourages the developer to experiment with variations of the user interface until an optimum design is found.
4. Implements Mac OS Services Transparently

PowerPlant sits on top of the Classic or Carbon APIs, making system calls as necessary to allow your application to interact with the operating system. As Mac OS has evolved, PowerPlant has incorporated support for additional services, such as the use of the Navigation Services dialogs for opening/saving files into its suite of application classes. One of PowerPlant’s compelling advantages is that the applications that are developed with it will run in places where support for optional services is absent by testing for the presence of these services and making the appropriate function calls. As the operating system changes, your code can remain essentially unchanged. PowerPlant hides the intricacies and changes from you and often allows you to take advantage of new operating system features without having to write new code.

PowerPlant also eases the use of other optional services, such as Drag and Drop, by providing classes that simplify the programming interface and implement the first level of common processing necessary to use the desired feature. The developer does not need to master the intricacies of Drag and Drop processing to use it in an application. Instead, the developer uses the easier interface to Drag and Drop classes provided in PowerPlant. PowerPlant provides simplified front-ends to other OS services, such as printing and clipboard management.

5. Mature Product

PowerPlant is a mature product, having been used for many years as the basis for many commercial Mac OS applications. Using PowerPlant gives you the benefit of the testing that has been done by many developers who have already used the framework. This testing and widespread real-world use ensures that the PowerPlant classes are stable and perform as expected.

In addition to the standard classes, Freescale also provides some additional unsupported classes for optional use in your projects. Third-party vendors also provide commercial quality add-on classes for specialized functions.

Less official but no less important, many users of PowerPlant freely distribute useful custom classes of their own creation. These classes may be used in your application (after appropriate testing) or used as examples for how to approach your own custom solutions.

6. Evolving Product

Although PowerPlant is a mature product, no framework that supports an evolving OS is ever really finished. As Apple added new services to Classic Mac OS, PowerPlant evolved to support them. As we migrate to Mac OS X, CodeWarrior™ and PowerPlant have evolved and will continue to evolve to provide direct support for development in this new environment.

7. Well-Supported with an Active User Community

Direct technical support for PowerPlant is available from Freescale. Training in the use of the application framework, as well as many other aspects of development, are available at no cost on the web via CodeWarriorU.com. Freescale personnel also actively monitor Mac development newsgroups and public mailing lists and proactively respond to issues raised.

When you use PowerPlant for your development, you can rely on official support from Freescale via e-mail or by phone, as well as a large and helpful community of expert Mac OS developers who have used PowerPlant for years. Virtually every Mac OS developer uses CodeWarrior for their projects, and many also use PowerPlant. These users are almost always enthusiastic in their support for the use of PowerPlant and freely give their unofficial support and mentoring for users of the framework.

8. Conclusion

PowerPlant is the application framework of choice for the developer who wishes to target both Classic Mac OS and Mac OS X development. This allows the developer to write code once to target both operating systems and the entire Macintosh market quickly and easily. The popularity and reliability of PowerPlant gives the developer more resources when writing code: access to official and unofficial support networks, commercial and non-commercial classes that extend the framework, and knowledge that as the operating system changes PowerPlant will change to accommodate the latest technologies. Because PowerPlant handles retargeting your code to a new and changing operating system, programming with PowerPlant means that your code will remain at the leading edge of Mac OS programming.