



# MAC OS X, Your Existing Application and Carbon

Mindfire Solutions

[www.mindfiresolutions.com](http://www.mindfiresolutions.com)

October 14, 2001

## **Abstract:**

*This article talks about the impact of OS X on existing applications, then goes on discussing the options available (specifically Carbon) for enabling them to run on OS X*

---

So, Mac OS X is finally there which many of us were eagerly awaiting for. But, wait a minute, what will happen to all those excellent applications written for Mac? “Are those going to run on OS X?” Well, the answer is Yes and No. Yes, because most of them will. No, because some of them may not run at all and even those, which will run, won’t be running in **True Native OS X Environment**.

OS X has a **Classic Environment** also called “Software Compatibility” environment, which makes it possible for the latest version of Mac OS 9, and all the applications capable of running on that version, to run on a Mac OS X system. This means user can still use his or her legacy applications until a complete transition to Mac OS X occurs. This classic environment more or less gives same look and feel as that of a "native" Mac OS 9 system.

To the Mac OS 9 operating system that OS X hosts, Classic appears as a new hardware platform. It implements hardware services using the Mac OS X kernel environment (particularly the *I/O Kit*). Hence, those native Mac OS 9 programs, which attempt to do anything directly at the lower layers of the system, will not run in the Classic environment. This means a number of different things but generally, programs that modify or rely on Mac OS internals below the hardware abstraction provided by the kernel environment will not work in the Classic environment.

Detailed discussion about Classic environment can be found at [http://gemma.apple.com/techpubs/macosx/Essentials/SystemOverview/InstallIntegrate/The Classic Application.html](http://gemma.apple.com/techpubs/macosx/Essentials/SystemOverview/InstallIntegrate/TheClassicApplication.html)

What the above means is that:

- A native OS 9 program may not run in classic at all, if it directly depends upon low-level Mac OS internals.



- Even if it runs, it will not be running under OS X environment rather it will run within classic environment running on a OS X system. For the user, at the bottom level, it means the same OS 9 like look and feel and no OS X new Aqua user interface.

The obvious next question, what, if we want to run our application in true OS X environment?

Mac OS X supports three development approaches, **Cocoa, Java & Carbon**. Cocoa being the native Mac OSX development framework, may be the first choice for from-scratch development. May be, because cocoa application will only run on OS X and not on the earlier version of Mac OS (which means that different versions will have to be maintained for OS X and OS 9). And for porting of an existing application to OS X, if not for the above reason of maintaining different versions, Cocoa is still not a viable solution due to the sheer amount of effort required in doing so.

Given the large existing application base for pre OS X and to make their transition to Mac OS X easier, Apple was bound to provide some solution other than Cocoa. And that solution is **Carbon**, which is most suited for porting of an existing application due to following facts:

- Since Carbon APIs are based on existing Mac OS APIs, porting effort required is minimal compared to the first option Cocoa.
- The Carbon applications can run on Mac OSX as well as previous versions of the Mac OS (8.1 or later), which means single code base to maintain.

The Carbon APIs can be used to write Mac OS X applications that also run on previous versions of the Mac OS (8.1 or later). While Carbon allows applications to take advantages of Mac OS X features such as multiprocessing support and the Aqua user interface, it is specifically designed to allow compatibility with older versions of the Mac OS. The Carbon APIs are based on existing Mac OS APIs. Because it includes most of the (about 95%) existing Mac OS APIs, which a typical application uses, converting to Carbon is a straightforward process.

Apple has provided tools and documentation to determine the changes needed to make (<http://developer.apple.com/carbon/>).

**Carbon APIs can be broadly classified in following categories:**

- Classic Mac OS APIs that can run on Mac OSX as it is
- Classic Mac OS APIs which slight modifications to work on OS X
- New APIs for both Mac OS X and Mac OS 8 & 9
- New APIs those are available on OS X only.



**Things missing from Carbon are:**

- Functions that are 68K-specific (like Trap table access or Patch Manager) or functions that allocate memory in the system heap are not allowed.
- Carbon environment was designed to be fully abstracted from hardware, so APIs, which access hardware directly, are missing from carbon.
- Old legacy functions, which has more modern replacements.

Carbon applications are linked with a single stub library at build time called *CarbonLibStub*. At runtime application links with appropriate carbon implementation found on the target platform

On Mac OS 8 and 9, the carbon implementation is there in *CarbonLib*. CarbonLib has the implementations for Carbon specific functions only and for other classic functions it just exports them from currently available system libraries like *InterfaceLib*.

On Mac OS X, the carbon application links dynamically to the *Carbon Framework*

**Carbonization of an existing application is a three-step process:**

- 1) Making essential changes to enable the application to run on Mac OS X.
- 2) Other Porting issues, which can affect the porting process depending on the complexity and level of the application.
- 3) Optimization steps, which are not necessary but should be considered so that the application can take best advantage of Mac OS X.

These issues are discussed in details in various documentations available at Apple's website.

**Further readings:** The documents that should be of interest for people going for carbonization are:

<a href="#">System Architecture</a>	(Apple)
<a href="#">Mac OS X System Overview (pdf)</a>	(Apple)
<a href="#">Carbon Porting Guide</a>	(Apple)
<a href="#">An Experience in Carbonization</a>	(Mindfire Solutions)

The “*Carbon Porting Guide.pdf*” is the detailed document giving step-by-step process and discussing the issues in converting a typical application to carbon application, whereas “*An Experience in Carbonization*” is a hand on experience of carbonization of a particular application, discussing in details the actual steps involved and issues faced during the process.



*Mindfire Solutions is an offshore software services company in India. Mindfire possesses expertise in multiple platforms, and has built a strong track record of delivery. Mindfire passionately believes in the power of porting and its many advantages for software product companies.*

*We have developed specialized techniques to make porting efficient and smooth, and to solve the issues specific to porting. We offer core development and QA/testing services for your porting requirements, as well as complete life-cycle support for porting.*

*If you want to explore the potential of porting, please drop us an email at [info@mindfiresolutions.com](mailto:info@mindfiresolutions.com). We will be glad to help you.*

*To know more about Mindfire Solutions, please visit us on [www.mindfiresolutions.com](http://www.mindfiresolutions.com)*

---