

Introduction to I/O Kit Fundamentals

This document explains the terminology, concepts, architecture, and basic mechanisms of the I/O Kit, Apple's object-oriented framework for developing device drivers for OS X. It contains essential background information for anyone wanting to create device drivers for this platform.

Who Should Read This Document?

There are two general types of I/O Kit developers, and this document tries to be useful to both. The first type is the developer creating a device driver that is to be resident in the kernel; the second type is the application developer who is using an I/O Kit device interface to communicate with hardware. Some chapters contain information useful to both types of developers, and others contain information that is of interest only to writers of kernel-resident drivers.

Obviously there are things *I/O Kit Fundamentals* does not cover. It does not, for example, describe the use of the development tools or the use of specific driver programming interfaces. But it does help you to understand the hows and whys of the I/O Kit, enabling you to obtain the most value from the more specific documentation and examples.

Organization of This Document

I/O Kit Fundamentals gives a broad, conceptual description of the I/O Kit and device-driver development on OS X. It contains the following chapters:

- What Is the I/O Kit?
Describes the features and benefits of the I/O Kit, and also discusses the philosophy and decisions informing its design.
- Architectural Overview
Gives a high-level description of the I/O Kit's architecture, essential concepts, and basic mechanisms.
- The I/O Registry
Describes the I/O Registry, a dynamic database capturing the client/provider relationships among active driver objects.
- Driver and Device Matching
Explains the matching process by which the most appropriate client drivers are found for registered providers. It also summarizes the procedure processes in user space follow to find suitable devices and their drivers.
- The Base Classes
Describes the base classes that each driver object directly or indirectly inherits from. It includes discussions of object construction and disposal, driver objects as I/O Registry entries, and the driver life cycle.
- Handling Events
Explains the architecture and usage of work loops and event sources, mechanisms that the I/O Kit uses to process events such as interrupts and I/O requests in a protected single-threaded environment.

- Managing Data

Describes how to use memory cursors, memory descriptors, and related objects to handle I/O transfers. It also discusses how drivers should deal with hardware constraints, such as those imposed by DMA engines.

- Managing Power

Explains the concepts of OS X power management and describes different ways drivers can power-manage their devices.

- Managing Device Removal

Explains how to respond to device removal (hot-swapping).

- I/O Kit Family Reference

Displays a class hierarchy chart for each family and provides family-specific information that might differ from generic I/O Kit information.

- Base and Helper Class Hierarchy

Provides a class hierarchy chart for all I/O Kit classes that are not members of a specific family.

- Document Revision History

Lists changes to this document.

- Bibliography

Lists additional sources for information on OS X and related topics.

- Glossary

Defines key terms used in this document.

See Also

Once you've absorbed the information in *I/O Kit Fundamentals*, you should be able to forge ahead and actually create a device driver. Apple provides several documents and other sources of information to help you with your efforts:

- *IOKit Device Driver Design Guidelines* describes the general steps required to design, code, debug, and build a device driver that will be resident in the kernel.
- *Accessing Hardware From Applications* discusses how to use the I/O Kit's "device interface" feature; it also includes information on serial and storage I/O via BSD device files.
- *Kernel Extension Programming Topics* contains a collection of tutorials that introduce you to the development tools and take you through the steps required to create, debug, and package kernel extensions and I/O Kit drivers (a type of kernel extension). It also includes information on other aspects of kernel extensions.
- *Kernel Programming Guide* provides an overview of the architecture and components of the OS X kernel environment (Mach, BSD, networking, file systems, I/O Kit). All developers who intend to program in the kernel (including device-driver writers) should read this document.
- *Mac Technology Overview* provides an introduction to OS X as a whole, which is useful for developers new to the platform.

Of course, you can always browse the header files shipped with the I/O Kit, which are installed in `Kernel.framework/Headers/iokit` (kernel-resident) and `IOKit.framework/Headers` (device interface.)

You can also view developer documentation in Xcode. To do this, select Help from the Xcode menu and then click Show Documentation Window.

You can browse the BSD man pages for more information on BSD and POSIX APIs in two ways: You can type `man function_name` in a Terminal window (for example, `man gdb`) or you can view an HTML version at *OS X Man Pages*.

If you're ready to develop a universal binary version of a device driver to run in an Intel-based Macintosh, first read *Universal Binary Programming Guidelines, Second Edition*. Then, see *IOKit Device Driver Design Guidelines* for an overview of issues of particular interest to device driver developers. Related information that is specific to a particular device type is available in the documents listed at Hardware & Drivers Documentation.

Apple maintains several websites where developers can go for general and technical information on OS X.

- Apple Developer Connection Reference Library (<http://developer.apple.com/referencelibrary/index.html>) contains a comprehensive collection of technical resources, including documentation, sample code, and Technical Notes.
- Apple Developer Connection: OS X (<http://developer.apple.com/devcenter/macosx>) offers SDKs, release notes, product notes and news, and other resources and information related to OS X.
- The AppleCare Support site (<http://www.apple.com/support>) provides a search feature that enables you to locate technical articles, manuals, specifications, and discussions on OS X and other areas.