About Slot Manager And System 7.0

You need to use **Slot Manager** routines only if you are writing an application or device driver that must address a NuBus card directly. For example, you need to use the **Slot Manager** if you are writing a driver for a video card, but not if you only want to display information on a screen for which a device driver already exists. If you *do* have to use the **Slot Manager**, use the information here to see if any of the functions added to the **Slot Manager** by system 7.0 or later are of use to you.

There are two variations of the system 7.0 or later **Slot Manager**: version 1 and version 2. Version 1 of the **Slot Manager** is supplied with the system 7.0 or later System file on disk for use with Macintosh II-family computers that were designed and built before system 7.0 was available. Version 2 is included in the ROM of newer Macintosh II-family computers.

The **Slot Manager** polls the NuBus cards in the system and initializes the cards before patches to the Operating System are loaded from disk. Version 1 of the 7.0 or later **Slot Manager** polls all NuBus cards again in case any cards that must be addressed in 32-bit mode were not recognized by the older **Slot Manager**. It is not necessary for version 2 of the **Slot Manager** to poll the NuBus cards a second time. Both versions of the **Slot Manager** reinitialize all NuBus cards after RAM patches have been loaded, in case any card requires RAM patches to be available before the card is used. Other than this difference in initialization sequence, the two versions of the **Slot Manager** are identical.

You can use the **SVersion** function to determine whether the system 7.0 or later **Slot Manager** is available and, if it is available, whether it is version 1 or version 2. You cannot use the other System 7.0 or later routines if the system 7.0 or later **Slot Manager** is not available.

Once the system 7.0 or later **Slot Manager** has been loaded into memory, the **Slot Manager** no longer executes the **InitSDeclMgr**, **SInitSRsrcTable**, **SInitPRAMRecs**, and **SPrimaryInit** functions. Because these functions are used for card initialization and all initialization is handled by the Operating System, the availability of these functions should not affect your program. See **Card Initialization** for more information about the initialization of NuBus cards by the **Slot Manager**.

If you are writing a device driver, you need the book *Designing Cards and Drivers for the Macintosh Family,* second edition. You will also find useful information in **Device Manager** .

The **Slot Manager** provided with system 7.0 or later addresses NuBus cards in 32-bit mode to ensure that the Operating System recognizes all NuBus cards. It also performs card initialization in a fashion different from that used by older versions of the **Slot Manager** to ensure that all NuBus cards-including those that must be addressed in 32-bit mode-are initialized correctly.

Several new routines have been added to the **Slot Manager**. You can use the new **Slot Manager** routines to

- determine which version of the **Slot Manager** is available
- determine what sResource data structures are available

- get information about an sResource data structure, whether or not the sResource data structure is enabled
- get information about all sResource data structures that match the type of a specific sResource data structure
- enable and disable sResource data structures
- restore an sResource data structure that has been deleted from the Slot Resource Table

> **Note**: An sResource data structure is sometimes referred to as a *slot resource.* Note, however, that a slot resource is a data structure in the firmware of a NuBus card and not a type of Macintosh resource. The structure and content of an sResource data structure are described in detail in *Designing Cards and Drivers for the Macintosh Family,* second edition.

All of the **Slot Manager** routines use a data structure called the **Slot Manager** parameter block to exchange information with the **Slot Manager**. The **Slot Manager** parameter block has been modified to add a flag field that is used by two new **Slot Manager** routines: **SGetSRsrc** and **SGetTypeSRsrc**. These routines search for SResource data structures.