
Card Initialization

When the user starts up a Macintosh II-family computer, the **Slot Manager** in ROM searches each slot for a declaration ROM. It identifies all of the sResource data structures in each declaration ROM and creates a table-the **Slot Resource Table**-that lists all of the sResource data structures currently available to the system. The **Slot Manager** then initializes the parameter RAM bytes reserved for each slot and executes the initialization code in the PrimaryInit record in the declaration ROM of each NuBus card.

The **Slot Manager** in the ROM of early Macintosh II computers (revision A of the ROM) can address NuBus cards only in 24-bit mode and can search for declaration ROMs only in the 1-megabyte (MB) slot space for each slot-that is, in addresses 0xFsxx xxxx, where s is the slot number. Because some NuBus cards have a declaration ROM that must be addressed with 32-bit addresses, not all NuBus cards can be addressed or even located by the **Slot Manager** in ROM revision A. The **Slot Manager** released with system 7.0 or later remedies this problem.

There are two versions of the **Slot Manager** that address NuBus cards in 32-bit mode: version 1 and version 2 (see [SVersion](#)). When version 1 of the **Slot Manager** is loaded into memory, it conducts a second search for declaration ROMs, this time addressing all of the slots in 32-bit mode. If it finds any new NuBus cards, the **Slot Manager** adds their sResource data structures to the Slot Resource Table and executes the code in the PrimaryInit records on those cards. Version 2 of the **Slot Manager** conducts only one search for declaration ROMs-before RAM patches are loaded into memory-and it executes PrimaryInit records at that time.

After RAM patches to the Operating System have been loaded from disk, either variant of the system 7.0 or later **Slot Manager** executes the code in any SecondaryInit records it finds in any of the declaration ROMs. It does *not* :

- reexecute any PrimaryInit records that were executed by the ROM-based **Slot Manager**
- reinitialize the parameter RAM values that were initialized by the ROM-based **Slot Manager**
- restore any sResource data structures that were loaded by the ROM-based **Slot Manager** and subsequently deleted by the PrimaryInit code

The **Slot Manager** executes the code in PrimaryInit records with interrupts disabled before Operating System patches have been loaded into RAM, but it executes SecondaryInit records with interrupts enabled after system patches have been loaded.

A SecondaryInit record has the same format as a PrimaryInit record. To include a SecondaryInit record in your declaration ROM, you must include a SecondaryInit field-a field with an identification (ID) number of 38-in the Board sResource data structure. (The Board sResource data structure is a type of sResource data structure that must be present in the firmware of every NuBus card that communicates with the computer. The format of a PrimaryInit record and the Board sResource data structure are described in *Designing Cards and Drivers for the Macintosh Family*, second edition.)

The system 7.0 or later **Slot Manager** allows you to disable a card temporarily until the SecondaryInit record is executed. The SecondaryInit record can then enable the card and complete initialization. You can use this feature, for example, to ensure that a card that requires **Color QuickDraw** is not used until after **Color QuickDraw** has been loaded into memory.