New Technical Notes
Macintosh

®

Developer Support

NuBus Expansion Interface Q&As
Hardware

Revised by: Developer Support Center June 1993
Written by: Developer Support Center October 1990

This Technical Note contains a collection of Q&As relating to a specific topic—questions you've sent the Developer Support Center (DSC) along with answers from the DSC engineers. While DSC engineers have checked the Q&A content for accuracy, the Q&A Technical Notes don't have the editing and organization of other Technical Notes. The Q&A function is to get new technical information and updates to you quickly, saving the polish for when the information migrates into reference manuals.

Q&As are now included with Technical Notes to make access to technical updates easier for you. If you have comments or suggestions about Q&A content or distribution, please let us know by sending an AppleLink to DEVFEEDBACK. Apple Partners may send technical questions about Q&A content to DEVSUPPORT for resolution.

New Q&As for this month:
NuBus cards with Macintosh IIvx or IIvi back panels

NuBus cards with Macintosh IIvx or IIvi back panels
Date Written: 1/22/93
Last reviewed: 4/1/93

Our customers report that they're having trouble installing our NuBus boards into the new Macintosh IIvx or IIvi machine. The connector extender apparently interferes with the computer's back panel. We haven't had problems with other Macintosh models. Where can I find detailed diagrams of the back panels for these two machines?

___

The specs for the size of NuBus cards for the Macintosh haven't changed and are published in *Designing Cards and Drivers for the Macintosh Family,* 3rd edition. The V Series Developer Note in the Developer Notes folder on the Developer CD also provides Macintosh II case dimensions.

Previous models allowed more space than called for in the spec. In the V series (and any future machine using this new metal enclosure) the tolerances have been tightened up to exactly match the published spec. Unfortunately some boards have been designed in violation of the published spec and now have a much harder fit.

PowerBook 140 ROM writes to $0000

Date Written: 11/16/92
Last reviewed: 3/1/93

I just put EvenBetterBusError on my PowerBook, and it seems the PowerBook ROM writes to location $0. Why? This hampers bug testing on the PowerBook.

___

The current version of EvenBetterBusError (creation date April 8, 1991) isn't compatible with the PowerBook 140, 145, 160, 170, 180 and Duos because of the way the Power Manager implements "power cycling." When a power cycle ends, the Power Manager uses location 0 to restore the processor state.

EvenBetterBusError's author said he'll update his program so it doesn't check for writes to location 0 on PowerBooks that power cycle. Until EvenBetterBusError is updated, turn off "Rest Mode" when using EvenBetterBusError on PowerBooks that have power cycling. When Rest Mode is turned off, the unit will not power cycle the processor.

/PFW NuBus line impedance and other specifications

Date Written : 6/9/92
Last reviewed: 7/13/92

What are the voltage tolerances, duration, and input impedance of the NuBus /PFW signal?

___

Designing Cards and Drivers for the Macintosh Family (2nd edition, page 74 and 3rd edition page 110), "/PFW interaction with the power supply," has the following /PFW line specifications: "When /PFW is held between 3.0 and 6.8 volts for at least 1.5 seconds, the power supply turns on and the computer begins operating. Once the power supply turns on, its own +5 volt output holds /PFW high so it can continue operating. If /PFW is pulled below .6 volts, the power supply will turn off; /PFW should be held below .6 volts until the computer completely shuts down." All Apple's power supplies are specified to draw no more than .5 ma from this line.

Macintosh Quadra 700 & 950 maximum NuBus card size

Date Written : 5/27/92
Last reviewed: 9/15/92

Could you please tell me the maximum size a NuBus expansion card can be within a Quadra 700 and 950?

___

You can find the technical and physical specifications for NuBus cards by writing to IEEE.

You can also find CPU-specific specifications in Designing Cards and Drivers for the Macintosh Family 3rd edition, which you can purchase either from APDA (order #M7075/C) or from any good computer bookstore. Keep in mind, however, that those NuBus cards which exceed the IEEE specification may fit into some CPUs, but certainly not all. For example, there is a larger sized card which will fit into the Quarda 950, but it is technically not NuBus because these cards will not fit any true NuBus form factor. This form factor is documented in the

Developer Notes from APDA on the 900 and 950. Case in point, the 700 will only take NuBus cards, not the larger cards.

NuBus boot devices and seRefNum

Date Written : 3/10/92
Last reviewed: 8/1/92

This concerns NuBus cards that want to be boot devices. *Designing Cards & Drivers for the Macintosh Family* states that, given a boot sRsrc and the startup info set to the slot device (slot #, sRsrc ID, extDev), then the boot sRsrc is called twice. The first time, seBootState is set to 0 and the device is to be opened and added to the drive queue; seRefNum is to be set to the refNum of the desired device. Now comes the bad part: seRefNum is an 8-bit quantity at offset 20 (cf. Slots.h and SlotEqu.a). The next field is seNumDevices, a byte at offset 21, yet the Quadra and Macintosh IIcx ROMs pick up the refNum by doing a "MOVE.W 20(A1),D4" So the refNum used for a slot device is really the refNum in the upper 8 bits, followed by seNumDevices, which is typically garbage. Therefore, slot devices that return a refNum should actually do:

```
   *(short *)&se->seRefNum = refNum
```

Please confirm that this is what should be done.

___

We verified that the various Macintosh ROM sources treat seRefNum as they should, as 16 bits. Even though the interfaces declare a seNumDevices byte field right after seRefNum, the ROM code apparently doesn't use it at all. Therefore, consider seRefNum to be a real INTEGER, 16-bit quantity. Amazing, no one discovered this all these years...

Macintosh II NuBus prototyping cards

Date Written : 12/16/91
Last reviewed: 8/1/92

Are there Macintosh II NuBus prototyping cards with a hardware interface that can be used for developing declaration ROMs? What hardware/software support is there for using MPW to assemble and download code into the declaration ROM? Chapter 10 of Designing Cards and Drivers refers to a NuBus Test Card (NTC) and a SCSI-NuBus Test Card. How can I get these?

___

Creative Solutions sells a "Hurdler II" NuBus prototyping card via APDA (T0429LL/A, $199). There are others, but you'd need to take a look at the Macintosh Buyer's Guide to find them.

Software support for using MPW to assemble and download code into the declaration ROM can be found on the latest *Developer CD Series* disc in the Tools & Apps (Moof!): Devices & Hardware: NuBus/Slot Manager folder.

The NuBus Test Card and SCSI-NuBus Test Card are design examples that don't actually exist.

Using Mac GetPhysical call to translate NuBus address space

Date Written : 12/11/91
Last reviewed: 8/1/92

Is GetPhysical supposed to return an error when a valid NuBus address is used as an input to the GetPhysical call? If so, is there a foolproof way to determine (via a system call or by looking at the address) whether an address is a valid NuBus address? How can I translate a logical Nubus address to a physical one?

___

It’s true that the GetPhysical call currently can only deal with locked RAM it knows about. Your first step is deciding what addresses are actually not in RAM, and thus guaranteed to be physical addresses. Your first stop is the information regarding the NuBus address space in Designing Cards & Drivers, 2nd edition, pp 88-92. All you’ll need to do is leave an address unchanged if it’s greater than or equal to $4000 0000.

Your only other difficulty is addresses in physical RAM which appear to be in the NuBus space. Specifically, on the Macintosh IIci and IIsi, the onboard video is physically at address 0, but logically sits at a NuBus address allocated to slot B (or E in the case of the IIsi). This is difficult to deal with; basically, you’ve got to special-case the machines with this kind of onboard video and walk the GDevice list looking for a frame buffer in the appropriate slot, and if you find one, start manually translating addresses destined for that frame buffer to the correct physical address (which starts at address 0).

Should similar Macintosh LC and NuBus cards have same ID?

Date Written : 11/6/91
Last reviewed: 8/1/92

We’re considering a Macintosh LC card with the same functionality as our NuBus card. In creating the declaration ROM for this card, should we be requesting a new card ID from you or should we use the same ID as we do in the NuBus version?

___

Whether you request a new board ID is up to you. We argued about it here in DTS a bit:

Pro-New-ID: “You may someday need to tell the difference between the two boards; any design shortcuts that depend on ID equality are probably bad.”

Con-New-ID: “If they’re basically the same board, they oughta have the same ID.”

It really came down to: they’re not *exactly* the same board, and since board IDs are plentiful and free, you might as well request a unique ID for the new board.

NuBus cards that monitor multiple serial ports simultaneously

Date Written : 9/17/91
Last reviewed: 8/1/92

What NuBus cards are on the market to monitor multiple serial ports simultaneously? Do any of these come with source?

———

You should be able to use the Apple Serial NB Card, along with the Macintosh Communications Toolbox. The board ships with the Serial NB tool, which is the best way to access the card.

You can also check the Redgate Product Registry on AppleLink for the Creative Solutions serial card and other third-party products that might be other options for you.

NuBus 90 documentation
Date Written : 9/17/91
Last reviewed: 2/9/93

Where can we get NuBus 90 information?

———

NuBus 90 is specified in the following two documents published by the Institute of Electrical and Electronics Engineers (IEEE):

• IEEE Standard for a Simple 32-bit Backplane Bus: NuBus
• IEEE Std. P1196-R1990

You can obtain these documents by contacting IEEE at the following address:

IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854
908-981-0060

How to request advance NuBus product information from Apple
Date Written : 9/17/91
Last reviewed: 8/1/92

We develop NuBus products. How can we get advance information if Apple makes any NuBus changes?

———

Pre-release product information is managed by Apple’s Evangelism Group. To request advance information on future NuBus technology from Apple, please provide the following information either by AppleLink or letter addressed to MacDTS (the Hotline staff will forward your information to the appropriate Evangelist.):

• A detailed description of the product you intend to or are developing.

- Key distinguishing features of your product
- Resources you have to conduct the development including:
  - a description of your company's technical expertise
  - development experience and how it relates to the product
  - development you propose (include programming experience, products you've developed, etc.)
- How and where you intend to sell and distribute your product.

• Why you think this product warrants seeding.
• What does your product do uniquely with a new technology.

Your request for advance information or pre-release product seeding will be reviewed carefully by Evangelism. Because of the limited quantities of seed units, not all requests can be granted, nor can Apple seed for compatibility issues alone. Approximately three weeks after Apple receives your information, Evangelism will inform you of the decision regarding your request.

Apple Serial NuBus Card metal plate supplier
Date Written : 9/17/91
Last reviewed: 8/1/92

What vendor supplies the metal plate on the Apple Serial NB Card that holds the D-62 connector and covers the hole in the RF shielding?

___

North American Tool and Die (NATD) makes Apple's metal plates (shields). Their address is as follows:

North American Tool and Die
999 Beecher Street
San Leandro, CA 94577
(415) 352-5500

Before ordering the shield (part #805-5101), you must first obtain authorization through MacDTS.

Macintosh NuBus card development tools
Date Written : 6/25/91
Last reviewed: 8/1/92

Does a card exist with built-in circuits interfacing to NuBus that I can use to design my own custom part for my own interface card? Is there a chip set available to handle the NuBus protocol?

___

There are two levels of prebuilt NuBus support available to you, depending upon the level of control you wish to exercise over the interface of the design and how expandable and compatible you desire your card to be.

The first is to use the Macintosh Coprocessor Platform (MCP) prototyping kit available from APDA: Macintosh Coprocessor Platform Developers Kit (#M0793LL/B). The Macintosh

Coprocessor Platform, together with A/ROSE, is a generic hardware and software foundation to help developers create add-on cards and software applications for NuBus-compatible Macintosh computers. The card helps the developer quickly build a NuBus coprocessor prototype and ultimately helps speed time-to-market. This product contains the Macintosh Coprocessor Platform, A/ROSE (Apple Real-Time Operating System Environment) software, and the Macintosh Coprocessor Platform Developer's Guide. Its intent is to inform and assist the developer in creating an interface to the Macintosh II product family bus (NuBus).

The Macintosh Coprocessor Platform card is a prototyping card with over 26 square inches of space available for the developer to use. It has no input/output interface, but is a generic master/slave I/O processor. Affiliated I/O devices that the developer adds, such as RS232 ports or Token-Ring connectors, give the card access to the outside world. The Macintosh Coprocessor Platform includes a Motorola 68000 processor operating at 10 megahertz and has 512K of RAM. The NuBus interface provides a bus master interface to NuBus on the Macintosh II main logic board. The Macintosh Coprocessor Platform card acts as a slot device to the Macintosh II operating system, freeing the processor on the Macintosh II to perform other functions.

A/ROSE is a real-time multitasking operating system for smart cards and provides an intelligent peripheral-controller interface to NuBus on Macintosh II computers. It provides the operating system and core software services required by Macintosh Coprocessor Platform card by providing software services to smart card application programs. The code includes a collection of traps, interrupt handlers, and tasks that provide support for task naming, timing services, and inter- and intra-card communications via messages. The manual provides information about the Macintosh Coprocessor Platform and A/ROSE software. The manual also provides a general overview of the product as well as detailed information on both the hardware and software components of the product. In addition, there are sample programs and instructions on how to create applications using the services of the hardware and software. If you plan to redistribute any portion of the A/ROSE software, either within their own company or outside, you must sign a license agreement with Apple Software Licensing.

Note: Because of the card's design, access to the 6800 requires direct wiring to the 68000 socket.

For development with the Macintosh Coprocessor Platform you need a Macintosh II computer (with NuBus) running Macintosh System Software v. 6.0.2 or later, A/ROSE software and the Macintosh Coprocessor Platform, MPW v.2.0 or later, MPW C and/or MPW Assembler, and a Macintosh Coprocessor Platform NuBus card.

The second option is to use two NuBus interface chips designed by Texas Instruments specifically to aid in interfacing to the Macintosh NuBus. These chips are MCP NuBus Interface Controller SN74ACT2441 and MCP NuBus Address/Data Registered Transceiver SN74BCT2425. These chips are described and documented in the Texas Instruments NuBus Interface Products Data Book available from Texas Instruments

Macintosh IIsi MMU table bug fixed for System 7

Date Written : 6/20/91
Last reviewed: 8/1/92

A bug in the Macintosh IIsi MMU tables, which has been fixed for System 7, caused slot 9 to be shadowed in slot B. This shouldn't affect applications or drivers using the Slot

Manager to look for their cards IF they scan starting from slot 0 up to E, because a card in 9 will get seen in 9 and all will work OK. But if someone scans from E down to 0, the Slot Manager would think it found the card in B. That’s bad because the driver would install the SIntInstall interrupt routine for slot B, and the card would really interrupt on 9, causing a crash. Fortunately every example DTS has seen publishes indexes from 0 to E.

Macintosh bus or resource locking for NuBus access

Date Written : 7/30/91
Last reviewed: 8/1/92

Is there a way to do Macintosh-to-NuBus access with bus or resource locking but without the CAS CPU operation which uses a Read-Modify-Write cycle? Does a CPU Read-Modify-Write cycle do resource locking when the CPU accesses NuBus?

___

Macintosh II CPUs perform an Attention-Resource Lock cycle as a prelude to Read-Modify-Write cycles to NuBus addresses. There is no other way to initiate bus or resource locking from the Macintosh CPU. Macintosh CPUs do not support locking NuBus for longer tenures.

The only Read-Modify-Write instructions are TAS, CAS, and CAS2.

Use of these instructions is problematic, as they may introduce timing problems for certain CPUs (see the Macintosh Technical Note “Macintosh IIfx: The Inside Story”), and the operand address must always be previously cached in the MMU address translation cache or else a bus error results. They must be used with extreme care.

Byte lanes and 8-bit NuBus data transfers
Date Written : 5/7/91
Last reviewed: 8/1/92

Can we use the Macintosh SReadFHeader function and specify the appropriate byte lanes to send 8-bit data over the NuBus™ through QuickDraw?

___

I don’t think it’s a great idea to be 8 bits wide. But I think you are confusing byte lanes with the Macintosh actually talking to the board.

Byte lanes values serve _only_ to show where the declaration ROM is, not anything at all about how the rest of the card is laid out or accessed. Reading the header to figure out the byte lanes values is only used to figure out how to access the declaration ROM. The driver or application that locates the card (via the Slot Manager) is presumed to know how to talk to the card and so will do so in the most efficient manner.

Most card developers use a byte-wide declaration ROM and design the rest of the card for whatever makes the most sense. For example, video cards need some registers and frame buffering RAM. It makes the most sense to have the frame buffer be 32 bits wide for performance reasons. Because QuickDraw and the operating system don’t need to access the video card’s declaration ROM much at all (mostly during startup), it’s quite OK and cost efficient to have the declaration ROM be byte-wide (you can put it on any lane you like in this case, such as B/L 0). The registers are mapped typically to byte locations as well.

One thing to note is that QuickDraw (always) talks _directly_ to the frame buffer RAM on the card by doing long word accesses. This is done for speed reasons, because video buffers can get big and if you have a lot of video activity, QuickDraw has to push a lot of data around.

So, if you want to respond with an 8-bit RAM frame buffer port, when QuickDraw is accessing the video buffer on your card, your card is going to see the NuBus TM lines from the Macintosh showing long word accesses, and you will have to (slowly) fetch, and latch and hold each of the 4 bytes of the long word until you fill that long access (that is, all 32 bits) before ACKing. For this reason, no one to my knowledge has ever produced a video board that did not have a 32-bit wide frame buffer. Also, QuickDraw does Bit Field Extract instructions to the buffer, so that may slow things down some more.

NuBus board sResources needn't start with ID=1
Date Written : 3/11/91
Last reviewed: 8/1/92

Does the first item in the Macintosh NuBus™ sResource directory have to be a board sResource with an ID of 1? Is a declaration ROM in error if this is not the case?

—

There's no restriction that sResources must begin with a board sResource with an ID of 1. All our examples do start this way, but it's not necessary. The sResource IDs do need to be organized in ascending order. A special case restriction is that video board functional sResource IDs must begin at 128 (for the default mode) and continue sequentially, covering the remaining operating modes. Beyond this restriction, the board sResource can have any ID value.

Where to find NuBus declaration ROM sample source code
Date Written : 12/12/90
Last reviewed: 8/1/92

Where can I find a source code sample for NuBus declaration ROM creation?

___

You can find sample code for a NuBus declaration ROM in Issue 1 of *develop* and in the develop folder on the latest *Developer CD Series* disc.

Where to get Macintosh IIsi NuBus expansion card specs
Date Written : 12/11/90
Last reviewed: 8/1/92

What are the exact physical specifications for NuBus expansion cards for the Macintosh IIsi? I have heard that there were some slight differences from the physical specs published in Designing Cards and Drivers for the Macintosh Family.

___

Macintosh IIsi dimension restrictions for NuBus cards are documented in the Macintosh Technical Note "NuBus Physical Designs—Beware." (This was written before the Macintosh IIsi was released, but with the Macintosh IIsi in mind.)

Test and board exchange for early Macintosh II/NuBus problem

Date Written : 12/5/90
Last reviewed: 8/1/92

I seem to remember that the first few months' worth of Macintosh II systems shipped had some sort of NuBus problems, which Apple acknowledged and even had a free motherboard swap offer. These problems were subsequently fixed on later revisions. What was the real problem with those early Macintosh II systems? Is the free (or any other) board swap still available, and how does an end user take advantage of it? What were the serial numbers of the affected Macintosh units? What should we be telling our customers and dealers?

___

The Rev A Macintosh II mother boards had a problem with 32-bit addressing of NuBus space. If you have a card which has its ROM mapped into the top of the 16 MB space, then the card would not be seen. The other problem was with cards which had more than 1 MB of RAM.

The swap is still available as far was we know. The way to get the swap is to go into any dealer and ask for it. It may take a week for the dealer to get the parts, but they should be able to help any user.

We give out a tool that will determine if the board is an old version or not. The tool, "NuBusTester," is on the developer CD and you can give it to all of your users. The tool checks the machine and if the machine is one of the problem machines, then it will bring up a dialog which tells the user to see their dealer. If the machine is fine then the app does nothing. The tester application will tell the customer what to do.

NuBus RAM can be used as RAM disk driver
Date Written : 10/22/90
Last reviewed: 8/1/92

Can the Macintosh use my NuBus board RAM as additional system memory or as a RAM disk?

___

There is no way that the Macintosh operating system can recognize NuBus RAM and use it for applications. If you want to use it as a RAM disk, you'll need to write a RAM disk driver. (Refer to *Inside Macintosh* for more information about this.)

Macintosh board IDs and sResource equate info not available
Date Written : 11/17/89
Last reviewed: 8/1/92

Can I get a list of all Macintosh NuBus board IDs?

___

No, that information is confidential. We register board ID and functional sResource equates so developers don’t use equates that are already in use, but we can’t distribute the list because the database contains information on unreleased products.

However, even if the list could be distributed, any program that depended on the information in it would be obsolete as soon as a new board came out.

We recommend that you use the Slot Manager’s ability to find certain cards or functions. That way, you only need to write your code once, and it will work with newer boards. That’s why

QuickDraw can find video cards years after it was frozen in ROM. It does so by calling the Slot Manager and looking for boards that perform the QuickDraw-compatible video function.

How to register a Macintosh NuBus board

Date Written : 5/3/89
Last reviewed: 8/1/92

How do I register a Macintosh NuBus board?

___

You need to have the Developer Support Center assign ID values for the sRsrc_Type fields (Category, cType, DrvrSW, and DrvrSW) for sResources other than the board sResource (the sRsrc_Type fields for the board sResource are ALWAYS $0001, $0000, $0000, and $0000 for the Category, cType, DrvrSW, and DrvrSW fields respectively), as well as board IDs for the board sResource. AppleLink or mail the following information to MacDTS:

- Company name
- Company address
- Person to contact (name, mailstop, phone number, and electronic mail address)
- The official product name (or at least a code name for the board)
- What the board will do
- If the board will have a software driver
- If the board will have a software driver, if it will be in ROM
- If the board will have a software driver, if it will be compatible with one that has already been defined (for example, Apple's defined video driver)

We need this information to maintain our database of NuBus boards. The information you provide is kept confidential, and the database is not available outside of DTS.

X-Refs:
“Slot Manager,” *Inside Macintosh* Volume V
Designing Cards and Drivers for the Macintosh Family, Addison-Wesley

NuBus test for early 1-MB Macintosh systems

Date Written : 5/3/89
Last reviewed: 8/1/92

My NuBus card needs more than 1 MB of memory. How do I determine if my card has been installed in a Macintosh that can see more than 1 MB of memory?

___

Some early Macintosh II systems can’t use more than 1 MB of memory. There is an upgrade available for those machines.

“NuBus Tester,” available on AppleLink and on the current developer CD, tells you which version of Macintosh you are running on.

There are object files that you can link with so that you can call CheckNuBus from your own application. You can also just send out the application called "NuBus Tester," which checks the Macintosh and alerts you if you need to upgrade.

Do not try to make your own version of CheckNuBus—the dialog was set up by Apple's legal department.

How to upgrade earlier EtherTalk cards for Mac IIcx and IIci

Date Written : 11/17/89
Last reviewed: 8/1/92

Why am I having problems with EtherTalk and my Macintosh IIcx/IIci?

___

Revisions J and earlier of the Apple EtherTalk card have a timing problem with Macintosh IIcx and IIci units. Symptoms of the problem are intermittent and include system hangs on boot, serious system errors, and exceedingly poor network performance. Revisions K and later of the EtherTalk card have been modified so that they will not exhibit this problem.

This problem occurs because the capacitance of a six-slot NuBus was factored into the timing mechanism of the EtherTalk card. Consequently, the capacitance level in a three-slot NuBus is lower, thereby affecting adversely the EtherTalk card's timing.

As additional NuBus cards are installed into a system, the capacitance level is raised on the NuBus. As a result, the problem is less likely to present itself. The most likely IIcx and IIci candidates for having the symptoms mentioned above are units with a single EtherTalk card installed and no other NuBus cards in the remaining two slots. This is a configuration that would be commonly used for routers, file servers, mail servers, and so on.

If you need to exchange your EtherTalk card for a newer revision, please see your local authorized Apple dealer. To find a dealer near you, call toll-free 800-538-9696. You will be asked for your zip code, and you'll be given names and phone numbers of all dealers in your area.

Macintosh II NuBus slot numbers and addresses

Date Written : 11/17/89
Last reviewed: 8/1/92

What are the numbers and addresses of the NuBus slots in the Macintosh II class of machines?

___

If viewed from the front of the machines, from left to right:

```
Macintosh II     1,2,3,4,5,6 ($9, $A, $B, $C, $D, $E)
Macintosh IIx    1,2,3,4,5,6 ($9, $A, $B, $C, $D, $E)
Macintosh IIfx   1,2,3,4,5,6 ($9, $A, $B, $C, $D, $E)
Macintosh IIcx   1,2,3       ($9, $A, $B)
Macintosh IIci   4,5,6       ($C, $D, $E)
Macintosh IIsi   1           ($9)
```

Software should not assume anything about slot numbers. Programs should call the Slot Manager and index through the slots using the range 0 through E, or use Gestalt or the pslt resource.

NuBus board configuration ROMs and IDs

Date Written : 4/25/89
Last reviewed: 8/1/92

I am making a NuBus board and I don't think that I need a configuration ROM. Do I need to build a configuration ROM? If so, how do I get the sResource board ID? Why should board manufacturers have configuration ROMs?

___

Technically, you don't need one, but if you don't have at least a minimal configuration ROM (also known as a declaration ROM), you can't use the library of Macintosh Slot Manager calls.

Developer Technical Support assigns the board IDs (and sRsrc_Type IDs). These exist in order for your card to tell the world what functions your board can perform. In order for DTS to assign these values, you need to provide certain information about your board and your company. There is a HyperCard stack published by DTS that allows you to enter the required information (it also has a tutorial that explains the Slot Manager and configuration ROM concepts in more detail).

A configuration ROM is desirable for flexibility. The combination of the Slot Manager and configuration ROM exists to locate and identify your expansion card and allow the computer and higher-level applications and drivers to communicate with it.

To better understand this, let's look at the Slot Manager in more detail, and see why it is desirable for you, the developer, to take advantage of it.

The Slot Manager/configuration ROM architecture has two main functions. The first is to provide a mechanism to allow the user to insert a card in a slot and have the card be operational without having to set any switches, worry about which slot to put the card in, or install any special software.

The second is to free an application from being dependent on a particular type of hardware. That is, the goal is to allow an application to be insulated from the hardware by being able to locate underlying, intermediate driver software that will "know" about and talk to the specific hardware. This frees the application to be able to deal with higher-level functions without being tied to a given set of hardware. It allows for different revisions of a given card, or even cards made by different vendors to work with applications, without requiring a revision of the application each time a card changes (or a new one is introduced).

Both these goals are accomplished via the Slot Manager software and the firmware data structures contained in the configuration ROMs of each card. If a valid configuration ROM is present, system software as well as applications and drivers can take advantage of the Slot

Manager library of routines to perform the two functions stated above.

Let’s examine the architecture and concepts further.

The specific functions of the Slot Manager are as follows:

• Locate and list the cards with configuration ROMs (also known as declaration ROMs);
• Impose a uniform structure on the information in the declaration ROM and provide a set of library routines to access the information. One major advantage of this is that once certain sets of data structures are defined and made public, host applications may use information contained in the structures on any card regardless of vendor;
Provide routines to allow host applications to transparently access information in the ROMs without regard to NuBus or byte lanes;
• Provide a mechanism for cards to have initialization code run on the host CPU during the host's startup sequence;
• Allow cards to have drivers in their declaration ROMs loaded into the host CPU;
Initialize and manipulate the parameter RAM on the host CPU for the card in the slot during startup.

Without a configuration ROM, most of this will be impossible. Locating the card will be difficult. You can't ask the customer where it is, without violating the user-friendly interface of the Macintosh, and you can't find it with software without the possibility of stepping on other cards' registers, memory, and so on, not to mention having to deal with bus errors for empty slots. Configuration ROMs are well defined in terms of location in the slot address space and structure. Getting at stored nonvolatile data on the board will probably be different for each card. Having code and or drivers loaded and executed at startup time will not be possible.

In summary, you should make a configuration ROM, and use the Slot Manager. It will provide services to applications (and drivers) accessing the card, and will mean less maintenance for applications and drivers.

X-Refs:
"Device Manager," *Inside Macintosh* Volumes II and IV
"Slot Manager," *Inside Macintosh* Volume V