Updating ARCNET NIMS from ISA Bus to PCI Bus

Many computers are now being made without an ISA bus[1]. This absence of an ISA bus is forcing many users to use network interface modules (*NIMs*)[2] that have been built for PCI slots. Contemporary Controls provides ARCNET devices for this purpose in its ***PCI20 series*** of NIMs—but the evolution to PCI NIMS raises several driver issues.

The PCI20 NIM is shipped with an ***enabler*** for Windows 95/98. The enabler is not truly a driver[3]—it simply "enables" Windows to assign system resources to the "jumper-less"[4] NIM during Plug-and-Play (PNP) resource allocation—after that, it has no further effect.

Once installed, an enabler or a driver can be difficult to remove. Often the choice of driver depends on the software supplied by the Original Equipment Manufacturer (OEM). Therefore, before deciding whether to use the enabler and/or some other driver with the PCI20, it is very important to contact the OEM for guidance.

NOTE: *Only one driver at a time can be installed for a specific PNP device. If a* ***second*** *driver is installed, the first should be removed in the process. If this does not work, the system may fail to recognize the intended driver and this could cause system control to seriously malfunction. A common mistake is to install the PCI20 enabler, discover that it was not needed, then install another driver without the enabler being fully removed. In such a case, the system may still recognize the first association between the enabler and the PCI20, and ignore attempts to associate the PCI20 with any other driver as long as the first association is maintained.*

What System Resources Are Involved

The PCI20 needs one 16-byte block of input/output (***I/O***) space and one interrupt (***IRQ***). *In addition* to these I/O and IRQ settings, a 128-byte block of memory and an extra 128-byte block of I/O space will be reserved by the system for the purpose of controlling the PCI20. In some systems Device Manager will report these two 128-byte spaces, but they **must not** be altered nor accessed by the user.

Typical resource problems occur when the system operating in Plug-and-Play mode allocates resources **automatically**, but the application requires a **specific** I/O address and/or IRQ. Windows NT and 2000, do not provide the means for changing the resource assignments. If the application is inflexible and demands a specific resource, consult with the OEM for guidance in dealing with this problem.

The PCI20 enabler disk includes a DOS program named *findpci.exe* that reports the assigned IRQ and I/O range. This program is useful in understanding how the operating system has allocated resources.

CONTEMPORARY CONTROLS®

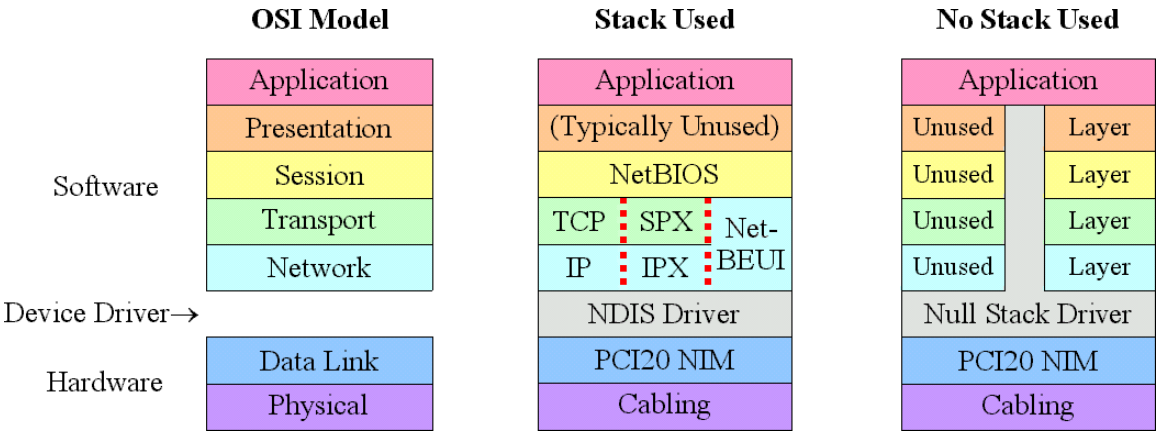
2431 Curtiss Street • Downers Grove, Illinois 60515 • USA
TEL 630.963.7070 **FAX** 630.963.0109

When the Enabler SHOULD NOT Be Used

The PCI20 enabler does not currently function with Windows NT/2000. In these systems a driver *must* be used to operate the PCI20 NIM, but it must be flexible enough to use any I/O address or IRQ assigned by the system. The assigned I/O address can vary from 100–FFFF and the IRQ can vary from 3–15.

In the 7-layer OSI Model depicted in the illustration below, a gap has been inserted to distinguish the upper five layers from the lower two layers. Connectivity between the upper (software) layers and the lower (hardware) layers is provided by a device driver although the precise nature of the connectivity varies with the type of driver.

A ***protocol driver*** (or ***stack***) implements a specific network protocol[5] such as TCP/IP, IPX/SPX, or NetBEUI. Upon installation, Windows will typically "bind" the PCI20 to one or more protocols by default. Use of a different protocol will require the user to specify the one desired. For stack operation without an OEM-supplied driver, an *NDIS driver*[6] must be used so that the PCI20 can communicate with the Application through whichever stack has been selected for use. This situation is illustrated below in the middle chart entitled "Stack Used" where communication between the NIM and NetBIOS is provided by either the TCP/IP stack or the SPX/IPX stack or the NetBEUI stack.



NDIS drivers *require* a standard protocol. For users who use non-standard protocols, this is a problem that can be solved by using ***Null Stack drivers***[7] designed for Windows 95/98/ME/NT or 2000. Although not written to any standard, these drivers allow the PCI20 NIM to send and receive raw packets. In such a situation, illustrated above in the chart entitled "No Stack Used," the Null Stack driver provides operational connectivity between the Application and the PCI20. In most ARCNET LANs, the Presentation, Session, Transport and Network Layer functionality is unused, but in some cases the Application software may include Presentation and Session functionality.

Both NDIS and Null Stack drivers are available for each NIM at www.arccontrol.com.

When the Enabler SHOULD Be Used

The illustrations on the previous page do not depict enabler usage. The enabler— h ch ***works only with Windows 95/98/ME***—merely sets the resources for the PCI20 NIM. The enabler *does not* provide any connectivity between the application and the PCI20.

Sometimes a Null Stack driver lacks the ability to establish resources for the PCI20 NIM. This is the case with the Null Stack drivers for Windows 95/98. If running the PCI20 ***Windows 95/98/ME*** with a Null Stack driver, the enabler ***will still be needed*** to set resources.

Occasionally, an OEM may supply driver software that lacks the ability to allocate resources. In this case, the OEM should specify that an enabler is needed. But, if the OEM does not specifically state that the PCI20 enabler is to be used, **do not** use it.

Drivers originally written for ISA NIMs typically do not provide resource allocation because resources for those cards were traditionally set by jumpers on the NIM. When implementing a ***Legacy*** or ***DOS*** driver for the PCI20, resources need to be allocated with the enabler.

Summary

When upgrading NIMs from ISA to PCI …

1. Consult with the OEM about driver support and NIM resource requirements.
2. Determine whether a Stack or a Null Stack driver is to be used.
3. If an enabler is also needed, install it before the driver.

More specific information about drivers is available in Technical Note 2.

---

[1] Machines that do not have an ISA bus are sometimes known as “Legacy-free” computers.

[2] NIMs are also known as network interface cards (NICs) or network adapters.

[3] Drivers are discussed in detail in Technical Note TN-2.

[4] In a Legacy machine, NIM resources are typically set by jumpers on the card. In “jumper-less” mode, the user assigns resources via software. In Plug-and-Play mode, the system *assigns* resources to the NIM.

[5] A protocol is a network “language.” For computers to communicate, they must use the *same* protocol.

[6] The NDIS drivers supplied by Contemporary Controls follow an existing Microsoft standard and are suitable for the many applications written to work with NDIS drivers. However, it is common to find a user that has written a custom driver which follows no standard or the user’s application writes directly to the ARCNET controller chip on the PCI20.

[7] Someone creating an ARCNET application could use Null Stack drivers to talk to embedded systems with little trouble. But existing applications would need to be modified to work with Null Stack drivers. Many customers have taken both routes with Null Stack drivers from Contemporary Controls.