PCI Hot Plug Software Usage Model

August 2000

THIS DOCUMENT IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

1.0 Introduction

PCI Hot Plug (PHP) is the concept of removing or inserting a standard PCI adapter card from a system without interrupting normal system operation or powering-down the system as a whole. PHP technology improves Reliability, Availability, and Serviceability (RAS) by allowing PCI adapters to be replaced, upgraded or added to a system while other adapters in the system provide uninterrupted service.

This document is intended to provide a high level overview of the software components required to support PCI Hot Plug on Intel Server Platforms. It also explains, at a user level, various cases that could occur in a Hot Plug environment and their expected results.

In order to support PCI Hot Plug, systems require Hot Plug hardware, a Hot Plug operating system, and Hot Plug capable adapter drivers. To insure backward compatibility, any combination of Hot Plug and conventional versions of each of these components is permitted, including mixing both Hot Plug and conventional adapter drivers. If a conventional driver is loaded under a Hot Plug operating system, or a Hot Plug driver is loaded under a conventional operating system, the driver will continue to have the same capability it always had in the conventional environment.

2.0 PCI Hot Plug Operations

2.1 PCI Hot Plug Operations

The PCI Hot Plug Specification discusses the sequence of steps involved in a “Hot Removal” and a “Hot Insertion” of a PCI adapter card. The next two sections explain, at a user level, the sequence of steps that may occur during a Hot Removal and a Hot Insertion.

2.1.1 Sequence of Steps for a Hot Removal

The following is an example of the sequence of steps that could take place during a Hot Removal. The actual sequence of steps may vary from one operating system to another but the following is typical:

1. The user informs the software via the UI or button of the desire to Hot Remove an adapter card.
2. The Hot Plug software quiesces the appropriate adapter activity.
    a. Adapter driver completes or terminates any incomplete operations.
    b. Adapter driver puts itself and the adapter card into a state in which it will not initiate any PCI bus activity.
3. The Hot Plug software disconnects the slot from the rest of the PCI bus.
4. The Hot Plug software powers off the appropriate slot and turns off the green power LED.
5. The Hot Plug software updates the UI to indicate the slot status.
6. The user removes the adapter card.

2.1.2 Sequence of Steps for a Hot Insertion

The following is the sequence of steps that could takes place during a Hot Insertion. The actual sequence of steps may vary from one operating system to another but the following is typical:

1. The Hot Plug slot is powered off by the method established by the system designer (i.e. the UI).
2. The user inserts the new adapter card.
3. The user notifies the Hot Plug software to power on the slot via the UI.
4. The Hot Plug software powers on the appropriate slot and turns on the green power LED.
5. The Hot Plug software connects the appropriate slot to the PCI bus.
6. The Hot Plug software notifies the OS that the adapter card is installed and the OS:
   a. Initializes the adapter card's Configuration Space Header and assigns system resources.
   b. Readies the appropriate adapter driver for use.
   c. Completes the initialization of the adapter card.
7. The Hot Plug software updates the UI to indicate the slot status and notifies the user that the card is ready.

The specification does not define higher level operations that are based on the Hot Removal or Hot Insertion of an adapter card. The specification leaves the implementation of these operations up to the operating system vendor.

Three terms that are commonly used to describe PCI Hot Plug operations are Hot Replace, Hot Add, and Hot Upgrade.

- **Hot Replace** – To Hot Remove an adapter card and then Hot Insert an identical adapter into the same slot. The replacement adapter card will use the same PCI resources that were assigned to the previous card and its driver will not be updated. Hot Replace is also commonly referred to as “Like-for-Like Replacement”.
- **Hot Add** – To Hot Insert an adapter card to a previously unoccupied slot. This operation requires that a driver also be loaded for the hot added adapter. This operation requires that PCI resources have been reserved by the system BIOS for the Hot Added adapter card. Hot Add is also sometimes referred to as “Hot Expansion”.
- **Hot Upgrade** – To Hot Remove an adapter card and then Hot Insert an upgraded adapter (i.e. new revision) that requires different PCI resources than the original card. The adapter's driver may or may not use the same driver as the previous adapter.

Not all of these operations are supported by every Hot Plug capable operating system. Furthermore, each operating system may implement these operations in a different manner. Refer to the Operating System PHP Support section for details.

3.0 BIOS PHP Support

To support the addition of a new adapter card into a PCI slot that was empty when the server was booted the system BIOS needs to reserve PCI resource space for the new adapter card.

One implementation is to use a PCI Hot Plug Resource Table (HRT). The system BIOS will build the HRT during POST. The HRT will contain the available PCI resources for each slot in the system whether or not it is occupied. The OS or PHP controller driver will use the HRT during a Hot Plug operation to determine if a Hot Plugged card can fit and where to configure it.

An HRT is currently required to support Hot Add and Hot Upgrade in NetWare* 4.11/5.0, in the Windows* 2000 filter driver implementation and all Hot Plug operations in the Windows 2000 filter driver implementation and UnixWare* 7.x.

The Windows 2000 implementation that is based on Advanced Configuration and Power Interface (ACPI) does not require an HRT but will require resource padding and the necessary ASL code for the PHP controller.

Note: The HRT is valid for the system configuration as it existed at system POST. If the system configuration is changed as the result of a Hot Plug Add or Hot Plug Remove, then the HRT no longer represents the system state. If the PHP-C driver, or whatever system software that interprets the HRT, is unloaded from the system and reloaded, it cannot rely on the HRT to describe the current state of the system. The PHP-C driver must either rely on a method of saving the current state of the PCI resource pool or prevent it from being reloaded if the HRT does not reflect the current system state.

4.0 Operating System PHP Support

The following operating systems have, or will have, support for PCI Hot Plug:

- Microsoft NT* 4.0
- Microsoft Windows* 2000
- Novell NetWare* 4.11 and 5.0
- SCO UnixWare* 7.x

Table 1 shows the adapter driver and BIOS requirements for supporting PHP for each OS.

Table 1. Operating System PHP Support

Operating System	Adapter Driver Requirements	HRT Required	Hot Plug Operations Supported		
			Hot Replace	Hot Add	Hot Upgrade
NT 4.0	Modified to support PHP	No	Yes	No	No
NetWare 4.11/5.0	NWPA 2.32 or ODI 3.31	Yes, for Hot Add and Upgrade	Yes	Yes	Yes
UnixWare 7.x	DDI-8 w/Hot Plug support	Yes, for all PHP operations	Yes	Yes	Yes
Windows 2000 Filter Driver	WDM	Yes	Yes	Yes	Yes
ACPI	WDM	No, but BIOS must support ACPI and contain the necessary ASL code	Yes	Yes	Yes

4.1 Windows* NT 4.0

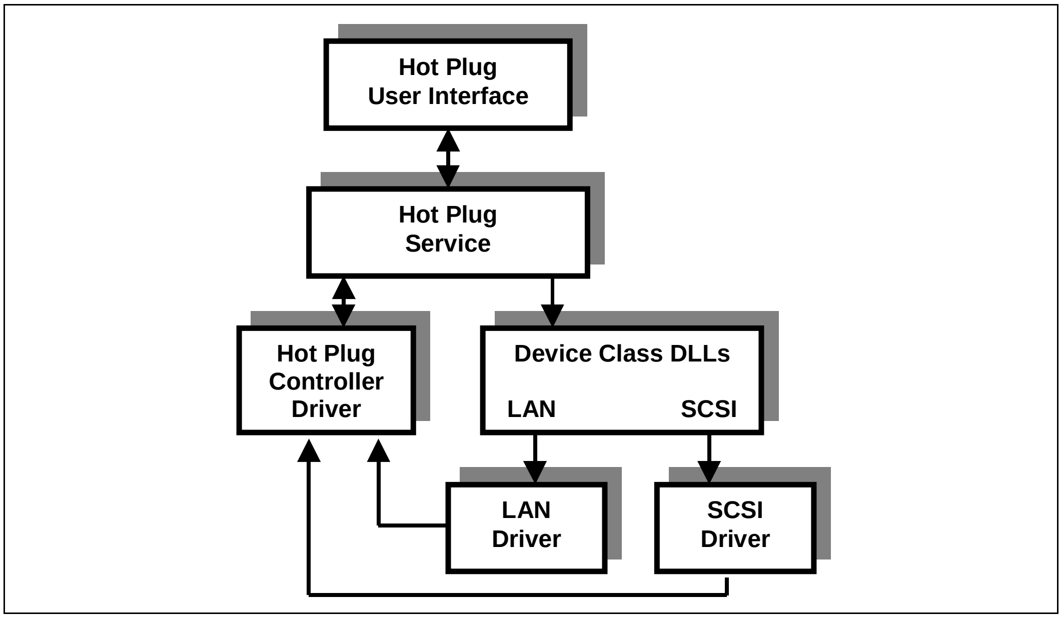
Native NT 4.0 does not support PCI Hot Plug. Hot Plug support was developed by adding new modules and modifying device drivers without modifying the NT kernel. NT 4.0 treats resource allocation and PCI device configuration as OS functions therefore making it very difficult to allocate resources to a new device outside the OS. Because of this NT 4.0 only supports Hot Replace (i.e. like-for-like replacement). There is no additional BIOS support required.

The add-in PHP software components of NT 4.0 are:

- **Hot Plug User Interface** – The graphical user interface providing access to the Hot Plug functionality.
- **Hot Plug Service** – An NT software construct used to encapsulate control and communication about PCI Hot Plug.
- **Hot Plug Controller Driver** – The device driver for the Hot Plug Controller that provides an interface to the Hot Pug Service to control the Hot Plug hardware and to read and set PCI device configuration space.
- **Hot Plug Device Class DLLs (storage and LAN)** – Provides a standard interface between the Hot Plug service and Hot Plug aware storage and LAN drivers.
- **Hot Plug Aware Adapter Drivers** – Standard miniport drivers with added IOCTLs (or OIDs) to provide support for PCI Hot Plug. They are modified to accept quiesce and resume requests from the Hot Plug Service.

The Hot Plug User Interface, Hot Plug Service, Hot Plug Controller Driver and Hot Plug Device Class DLLs are installed by running the software stack installation program. Hot Plug aware adapter drivers can be installed before or after the install program is run. Refer to the installation document for more details. Figure 1 shows the key components of the NT 4.0 PCI Hot Plug architecture.

Figure 1. NT 4.0 PCI Hot Plug Architecture



4.2 Windows* 2000

Microsoft will not support PCI Hot Plug in the initial release of Windows 2000. Development of PCI Hot Plug will begin following the release of Windows 2000, and support will ship in a later release. Because of this, all information regarding PCI Hot Plug support under Windows 2000 in this document is speculation and subject to change. Refer to http://www.microsoft.com/hwdev/pci/hotplugpci.htm for more details.

The software interface that Microsoft will ultimately support for PCI Hot Plug in Windows 2000 is the Advanced Configuration and Power Interface (ACPI) specification. There will no be PCI Hot Plug Controller driver necessary in the ACPI implementation. The system BIOS must contain ASL code to control the Hot Plug controller.

Although the basic infrastructure is in place in Windows 2000, the feature will not have been fully tested. Vendors can continue development and testing to be ready for support in a future release.

Drivers for adapters that can be Hot Plugged have no additional requirements beyond the standard Plug and Play support that every Windows 2000 driver must have. As long as they conform to the Windows Driver Model (WDM) then they will support PCI Hot Plug.

The following information regarding Windows 2000 WHQL testing of a PCI Hot Plug system with was obtained from http://www.microsoft.com/hwdev/pci/hotplugtest.htm:

“Some current systems are being designed to use an alternative, non-ACPI based mechanism for hot-plugging PCI devices, developed by Compaq and licensed by other vendors. Systems can be submitted to WHQL for testing with this method of supporting hot-pluggable PCI if they meet the following test requirements:

1. A system running Windows NT 4.0 SPx or Windows 2000 that uses this non-ACPI solution should be capable of supporting the following upgrade scenarios:
   - Upgrading successfully to Windows 2000 with the OEM’s non-ACPI PCI hot-plugging solution for Windows 2000.
   - Upgrading successfully to the future release of Windows 2000 that supports Hot-Plug PCI, with the end-user experience being the same as a clean retail installation of the future Hot-Plug PCI release on an ACPI-based Hot-Plug PCI platform.
2. A system running Windows NT 4.0 SPx or Windows 2000 that uses this non-ACPI solution should support successfully installing the retail version of Windows 2000 and the future Hot-Plug PCI release.

   Notice that unless the OEM provides the drivers for the non-ACPI Windows 2000 PCI hot-plug solution, hot-plugging functionality will not be guaranteed to be present in this case; however, all other system functionality must remain after installing a retail version of Windows NT.

OEMs will have to indicate on the system’s testing paperwork that these installation tests have all been performed and that the system works as required. OEMs must also provide the non-ACPI PCI hot-plugging drivers for systems submitted to WHQL for testing.

Important: Only two variants of PCI hot-plugging support will be allowed under this framework: those that are based on the Compaq technology, and those that meet requirements for ACPI-based support as defined in the design guides for PC 98, PC 99, and servers. Also, no new systems will be accepted for testing with non-ACPI based solutions as of 60 days after the release of the Windows 2000 version that provides Hot-Plug PCI support using ACPI methods.”

4.2.1 Windows* 2000 PCI Hot Plug Filter Driver

To support non-ACPI compatible versions of Hot Plug Controllers a filter driver was developed that layers into the standard Windows 2000 PCI bus driver to provide PCI Hot Plug capability. The filter driver also acts as the device driver for the Hot Plug Controller. This filter driver will be used until ACPI support is developed.

The Filter Driver implementation supports all three Hot Plug operations (Hot Replace, Add, and Upgrade). With the Windows 2000 Filter driver, Hot Replace doesn't exist as a single operation, but is implemented as a Hot Remove followed by a Hot Add. BIOS (HRT) support is required for all Hot Plug operations.

4.3 Novell NetWare* 4.11/5.0

Both NetWare 4.11 SP5b (or greater) and NetWare 5.0 support all three PCI Hot Plug operations (Hot Replace, Add, and Upgrade). For Hot Add and Hot Upgrade BIOS support (HRT) is required.

The PHP software components of NetWare 4.11 and 5.0 are:

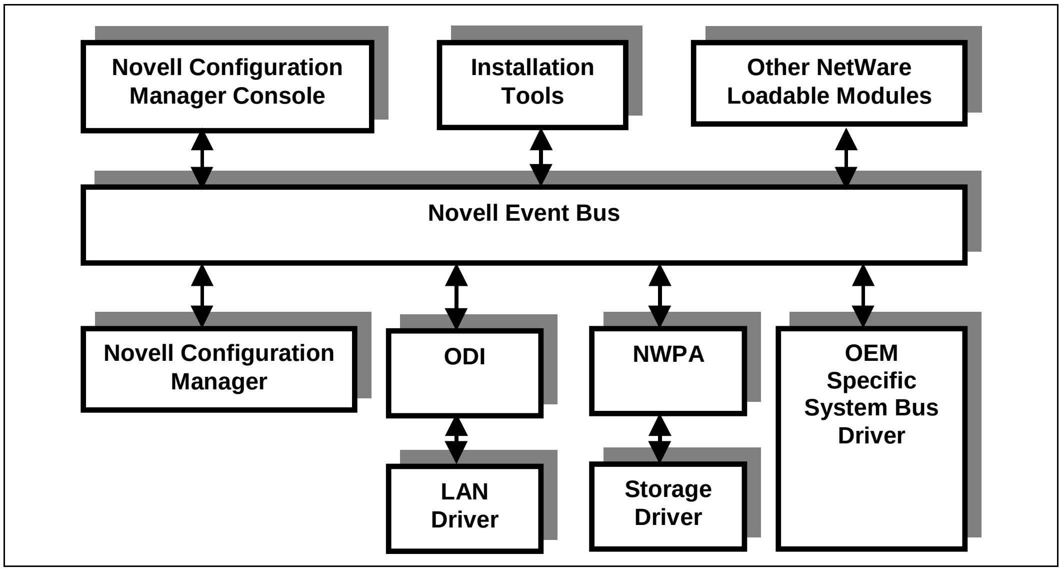
- **Novell Event Bus –** A communication channel for signaling between modules.
- **Novell Configuration Manager Console** – The user interface.
- **Novell Configuration Manager** – Manages Hot Plug operations.
- **OEM-Specific System Bus Driver** – The device driver for the Hot Plug controller.
- **Installation Tool** – Locates the appropriate device driver for a newly installed device.
- **Hot Plug Aware Adapter Drivers –** Storage adapter drivers must comply with NWPA 2.32 and LAN adapter drivers must comply with ODI 3.31 to support PCI Hot Plug. No additional driver modifications are required.

NetWare 4.11 requires the Novell Event Bus, Novell Configuration Manager Console, Novell Configuration Manager, and the Installation Tool to be obtained and loaded from Service Patch 5b (SP5b). In addition, the OEM-Specific Bus Driver and an additional NLM (ioconfig.nlm) are obtained from Intel. All of these components should be loaded in the AUTOEXEC.NCF file. Refer to the installation document for more details.

NetWare 5.0 incorporates the Hot Plug components into the OS distribution media. The end user is only required to obtain the appropriate OEM-Specific System Bus Driver from Intel. The driver is written following Novell's System Bus Driver Spec.

Note: The NetWare PHP-C driver relies on a ROM BIOS function that allows it to store HRT information in the system's NVRAM (see BIOS section 4). The current Intel BIOS does not implement this ROM BIOS function. Therefore, current Intel PHP implementation does not support unloading and re-loading the PHP-C driver (i.e. downing a NetWare to DOS and starting it back up without rebooting the system.) Figure 2 shows the key components of the NetWare PCI Hot Plug architecture.

Figure 2. Novell NetWare PCI Hot Plug Architecture


Novell Configuration Manager Console
Installation Tools
Other NetWare Loadable Modules
Novell Event Bus
Novell Configuration Manager
ODI
NWPA
OEM Specific System Bus Driver
LAN Driver
Storage Driver

4.4 SCO UnixWare* 7.x

UnixWare 7.x supports all three PCI Hot Plug operations (Hot Replace, Add, and Upgrade). BIOS (HRT) support is required for all Hot Plug operations.

The PHP software components of UnixWare 7.x are:

- **Hot Plug Manager User Interface (HPMUI)** – The administrative interface through which the user requests Hot Plug operations.
- **Hot Plug Service Library (HPSL)** – The user-level library used to implement a new HPMUI. These routines access information in the resource manager database, send operation requests to the kernel-level components and receive status information back.
- **Hot Plug Controller Interface (HPCI)** – The primary component to manage PHP controller drivers and other device drivers. The HPCI manages system component information and the Hot Plug support features. The HPCI also manages all interactions between the Hot Plug subsystem and the resource manager database and the specific kernel subsystem that implements special device files.
- **Hot Plug Controller Driver (HPCD)** – The device driver for the Hot Plug Controller. It passes Hot Plug operation requests from the HPCI to the bus and passes events to the HPCI.
- **Hot Plug Aware Adapter Driver** – Storage and LAN device drivers are required to comply with the DDI-8 specification and support its Hot Plug requirements. See the UnixWare 7 HDK for more details.

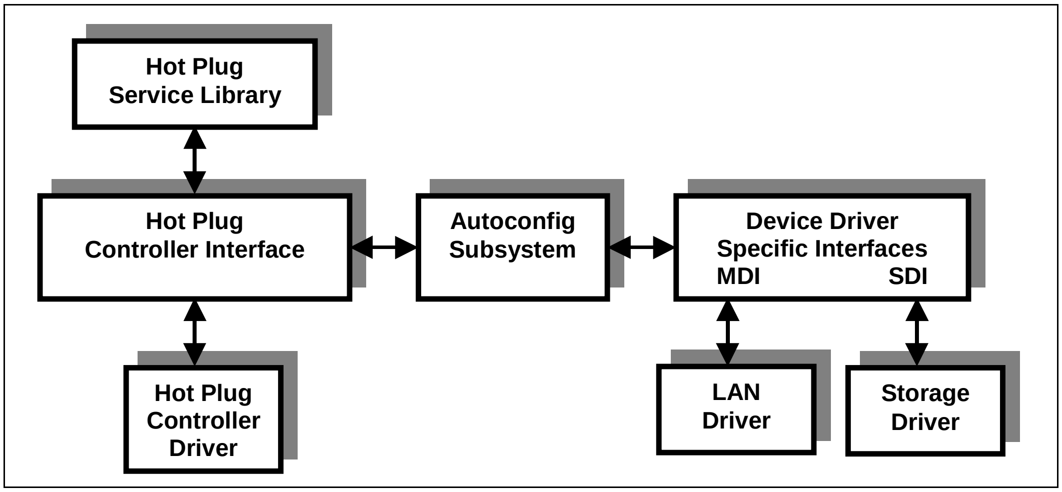
UnixWare 7.x incorporates the Hot Plug components into the OS distribution media. The end user is only required to obtain the appropriate Hot Plug Controller Driver from Intel. This driver is written to the SCO’s hot plug driver spec.

The following information was taken from http://chac.sco.com/LateNews/latenews.html#late.5:

> “The hot remove operation for an HBA driver works only if none of the devices connected to this HBA are in use. The user must close all applications and unmount all file systems which are using any devices connected to this controller. This step is not necessary for the hot replace operation, but it is always safe to perform it.
>
> At present, only hot add operation is supported for NIC controllers. The user must use the SCOadmin Hot Plug Manager to power up the controller and then run the SCOadmin Network Configuration Manager to configure the devices connected to this controller.
>
> The hot add operation is not supported for the first NIC controller in the system. The first NIC controller must be added to the system when the system is powered off.”

These limitations will be addressed in future releases of the UnixWare operating system. Figure 3 shows the key components of the UnixWare 7.x PCI Hot Plug architecture.

Figure 3. SCO UnixWare 7.x PCI Hot Plug Architecture


Hot Plug Service Library
Hot Plug Controller Interface
Autoconfig Subsystem
Device Driver Specific Interfaces
MDI
SDI
Hot Plug Controller Driver
LAN Driver
Storage Driver

5.0 Usage Model

This section is intended to provide a model for users of PCI Hot Plug technology on an Intel Server Platform.

5.1 Intel Server Platform-specific Information

Intel servers include two LEDs located next to each PCI Hot Plug slot. A green LED acts as a power indicator (on = power, off = no power) and an amber LED acts as an attention indicator (on = error, off = no error) for the slot.

Some Intel servers include a mechanical slot switch for each PCI Hot Plug Slot. When the switch is opened the slot is isolated from the PCI bus. The switch can be enabled/disabled through a PHP override jumper and/or the system BIOS. When the override jumper is set or the switch is disabled through the BIOS, the slot remains connected to the PCI bus regardless of the state of the switch.

Some Intel servers include a push button for each PHP slot. When the button is pushed it sends an interrupt to the PHP software to power on/off a slot. The intent of this button is to provide PHP support w/o having to use the GUI. In most cases, when powering on a slot via the button, the user will still have to configure networking or storage devices. Not all OSs have support for this push button, in which case pushing them will have no effect.

Intel servers contain may contain different PCI Hot Plug controllers and therefore have different driver requirements. The goal is to combine driver all requirements into one PHP controller driver for each operating system.

Troubleshooting tips can be found at http://www-osd.dp.intel.com/php/PHPTroubleshoot.htm.

5.2 Usage Cases

The following are cases that could occur with an Intel Server Platform in a PCI Hot Plug environment and their expected results as seen by the end user.

Case 1. User fails an adapter through the User Interface (UI)

NT 4.0: The Hot Plug software must quiesce the adapter driver activity and turn on the amber fault LED for the failed slot. The UI must be updated to show the state of the amber fault LED to notify the user of the failed adapter. If the adapter is configured as part of a “team” then adapter activity should be transferred to the other team member. No other device should be affected.

NW, UW, W2K: Failing an adapter through the UI is not implemented.

Case 2. User powers down a slot through the UI

NT 4.0, UW if no adapter is present: The Hot Plug software must turn off the power to the slot and its green power LED. The UI must be updated to show the state of the green power LED to notify the user of the slot status. No other slot or device should be affected.

NT 4.0, UW if a non-Hot Plug adapter is present: This action is not supported. The Hot Plug software must not allow a slot to be powered off through the UI if a non-Hot Plug adapter is present. No other slot or device should be affected.

NT 4.0, UW, W2K if a Hot Plug adapter is present: The Hot Plug software must quiesce the adapter driver activity and power down the specific PCI slot. The software must also turn off the green power LED for that slot. The UI must be updated to show the state of green power LED to notify the user of the slot power change. If the adapter is configured as part of a “team” then adapter activity should be transferred to the other team member. No other slots or device should be affected.

NW: NetWare doesn’t have a specific option to power down a slot through the UI.

Case 3. User powers up a slot through the UI

NT 4.0, UW if no adapter is present: The Hot Plug software must turn on the power to the slot and its green power LED. The UI must be updated to show the state of the green power LED to notify the user of the slot status. No other slot or device should be affected.

NT 4.0, UW if a non-Hot Plug adapter is present: This action is not supported. The Hot Plug software must not allow a slot to be powered on through the UI if a non-Hot Plug adapter driver is loaded for an adapter. No other slot or device should be affected.

NT 4.0, UW if Hot Plug adapter is present: After first considering passing the requirements of cases four and five the Hot Plug software must connect the appropriate slot to the PCI bus, power on the appropriate slot and notify the OS that the adapter card is installed. The OS then:

1. Initializes the adapter card's Configuration Space Header and assigns system resources.
2. Readies the appropriate adapter driver for use.
3. Completes the initialization of the adapter card.

The Hot Plug software must update the status of the green power LED to notify the user that the card is ready. If this operation was part of a Hot Replace of an adapter that was part of a "team" then the adapter should rejoin the team. No other slot or device should be affected.

NW, W2K: NetWare and Windows 2000 don't have a specific option to power up a slot through the UI.

Case 4. User powers down a slot via the push button

NT 4.0, NW, W2K if no adapter is present: The software will blink the green power LED for ~5 seconds to allow the user to abort the operation. If the button is pushed during the time when the green power LED is blinking, the operation will be aborted and the state of the machine will be returned. If the button is not pushed a second time, the Hot Plug software must turn off the power to the slot and its green power LED. The UI must be updated to show the state of the green power LED to notify the user of the slot status. No other slot or device should be affected.

NT 4.0, NW, W2K if a non-Hot Plug adapter is present: This action is not supported. The Hot Plug software must not allow a slot to be powered off via the push button if a non-Hot Plug adapter is present. The green power LED will blink for ~5 seconds and then the system will return to its original state. No other slot or device should be affected.

NT 4.0, NW, W2K if a Hot Plug adapter is present: The software will blink the green power LED for ~5 seconds to allow the user to abort the operation. If the button is pushed during the time when the green power LED is blinking, the operation will be aborted and the state of the machine will be returned. If the button is not pushed a second time, the Hot Plug software must quiesce the adapter driver activity and power down the specific PCI slot. The software must also turn off the green power LED for that slot. The UI must be updated to show the state of green power LED to notify the user of the slot power change. If the adapter is configured as part of a "team" then adapter activity should be transferred to the other team member. No other slots or device should be affected.

UW: UnixWare doesn't currently have a specific option to power down a slot via the push button. Pushing the button will have no affect in UnixWare. This support is expected in UnixWare 7.2.

Case 5. User powers up a slot via the push button

NT 4.0, NW, W2K if no adapter is present: The software will blink the green power LED for ~5 seconds to allow the user to abort the operation. If the button is pushed during the time when the green power LED is blinking, the operation will be aborted and the state of the machine will be returned. If the button is not pushed a second time, the Hot Plug software must turn on the power to the slot and its green power LED. The UI must be updated to show the state of the green power LED to notify the user of the slot status. No other slot or device should be affected.

NT 4.0, NW, W2K if a non-Hot Plug adapter is present: This action is not supported. The Hot Plug software must not allow a slot to be powered on via the button if a non-Hot Plug adapter driver is loaded for an adapter. No other slot or device should be affected.

NT 4.0, NW, W2K if Hot Plug adapter is present: The software will blink the green power LED for ~5 seconds to allow the user to abort the operation. If the button is pushed during the time when the green power LED is blinking, the operation will be aborted and the state of the machine will be returned. If the button is not pushed a second time, then after first considering passing the requirements of cases six and seven the Hot Plug software must connect the appropriate slot to the PCI bus, power on the appropriate slot and notify the OS that the adapter card is installed. The OS then:

1. Initializes the adapter card's Configuration Space Header and assigns system resources.
2. Readies the appropriate adapter driver for use.
3. Completes the initialization of the adapter card.

The Hot Plug software must update the status of the green power LED to notify the user that the card is ready. If this operation was part of a Hot Replace of an adapter that was part of a "team" then the adapter should rejoin the team. No other slot or device should be affected.

UW: UnixWare doesn't currently have a specific option to power up a slot via the push button. Pushing the button will have no affect in UnixWare. This support is expected in UnixWare 7.2.

Case 6. User tries to Hot Replace a non-identical adapter

NT 4.0, NW (w/o HRT): This action is not supported. A valid Hot Removal will occur but when the non-identical adapter is inserted the Hot Plug software should detect that it is a non-identical and adapter turn on the amber fault LED and update its status in the UI to notify the user of the invalid operation. The power will not be turned back on to the slot. No other device should be affected.

NW (w/HRT), UW, W2K: A valid Hot Removal will occur followed by a Hot Add or Hot Upgrade, depending on if the adapter is a new or upgraded adapter. No other device should be affected.

Case 7. User tries to insert a 33 MHz adapter in a slot operating at 66 MHz

NT 4.0, UW, NW, W2K: This is not a supported action. The Hot Plug software must prevent the PCI bus from being connected to the slot containing the 33 MHz adapter. The Hot Plug software must:

1. Disconnect the slot from the bus.
2. Power on the slot.
3. Detect that the adapter is not capable of running at 66 MHz.
4. Power off the slot without ever connecting it to the PCI bus.
5. Turn on the amber fault LED and update its status in the UI to notify the user of the invalid operation.

No other device should be affected.

Case 8. User inserts a 66 MHz capable adapter in a slot running at 33 MHz

NT 4.0, NW, UW, W2K: Since a 66 MHz capable adapter must also be capable of running at 33 MHz this action is supported. The Hot Plug software must treat this scenario as if the adapter was a 33 MHz adapter. The 66 MHz adapter will end up operating at 33 MHz after being inserted into the 33 MHz slot.

Case 9. PHP capable adapter driver detects an error in the adapter

NT 4.0, NW, UW, W2K: The adapter driver detects the error in the adapter and notifies the hot plug software which turns on the amber fault LED. The UI must be updated to show the state of the amber fault LED to notify the user of the adapter error. If the adapter is configured as part of a "team" then adapter activity should be transferred to the other team member. No other device or slot should be affected.

Case 10. Hot Plug Controller Driver detects a slot power fault

NT 4.0, NW, UW, W2K: The slot power controller will immediately disconnect the power to the slot and inform the Hot Plug Controller when it detects a power fault. The Hot Plug software must turn on the amber fault LED and turn off the green power LED for the slot with the power fault. The UI must be updated with the LEDs status to notify the user of the fault. There is no restriction on how the adapter driver must handle this situation so the end result is undetermined. No other devices should be affected.

Case 11. User opens the mechanical slot switch without using the software interface

NT 4.0, NW, UW, W2K if PHP override jumper is not set: This is not a supported action. The slot will immediately be disconnected from the PCI bus and the result of this action is unpredictable. An NMI may occur.

NT 4.0, NW, UW, W2K if PHP override jumper is set: If the PHP override jumper is set then the mechanical slot switches will not function. Therefore, when opened there will be no affect on the system.

Case 12. Slot power status when system is booted

NT 4.0, NW, UW, W2K: The OS drivers turn power off to empty slots when they load.

6.0 Conventions and Terminology

This document uses the following terms and abbreviations:

ACPI	Advanced Configuration and Power Interface
ASL	ACPI Source Language
ASIC	Application Specific Integrated Circuit
BIOS	Basic Input/Output System
DDI	Device Driver Interface (SCO UnixWare)
DLL	Dynamic Linked Library (Microsoft NT)
GUI	Graphical User Interface
HDK	Hardware Development Kit (SCO UnixWare)
HPCD	PCI Hot Plug Controller Driver
HRT	PCI Hot Plug Resource Table
IOCTL	Input/Output Control (Microsoft NT)

LAN	Large Area Network
MDI	LAN (or MAC) Driver Interface (SCO UnixWare)
NIC	Network Interface Card
NW	Novell IntraNetWare Operating System
NWPA	NetWare Peripheral Architecture
NT	Microsoft Windows NT Operating System
ODI	Open Data-Link Interface (Novell NetWare)
OEM	Original Equipment Manufacturer
OID	Object Identifier (Microsoft NT)
OS	Operating System
PCI	Peripheral Component Interconnect
PHP	PCI Hot Plug
POST	Power On Self Test
RAS	Reliability, Availability, and Serviceability
SDI	Storage Driver Interface (SCO UnixWare)
SW	Software
UI	User Interface
UW	SCO UnixWare Operating System
WDM	Windows Driver Model (Microsoft NT)
WHQL	Microsoft Windows Hardware Quality Labs