CHAPTER 12

PC Card

This chapter presents requirements and recommendations for PC Card 16 (previously referred to as PCMCIA cards) and CardBus under the Microsoft Windows family of operating systems.

Version 1.1
Includes changes to items 1a, 16, 17, 18, 19, 22, References for PC Card, and Checklist for PC Card, as previously published in the PC 97 FAQ on http://www.microsoft.com/hwdev/pc97.htm and the PC 97 OnNow Requirements on http://www.microsoft.com/hwdev/desguid/onnowpc97.htm

Contents

Overview for PC Card

Windows supports PC Card I/O cards. Memory PC Cards are only supported as legacy devices. For any PC Card device to work effectively with Windows, the manufacturer must implement a minimum set of tuples documented in the PC Card Standard. Windows uses these tuples to identify and configure any PC Card.

For PC 97, the new design concerns for PC Card are the following:

- New design guidelines for CardBus host controllers and cards.
- New power management guidelines, including requirements for the PC Card controller to meet new PCMCIA class power-management specifications being developed as part of the OnNow design initiative.

These new design details are discussed in this chapter.

Note Because CardBus implementations are just beginning, additional requirements might arise. As implementation details become available, Microsoft will provide information about revisions to the hardware requirements and recommendations at http://www.microsoft.com/hwdev/.

PC Card Basic Requirements

This section summarizes the basic standards for PC.

1. All devices compliant with the PC Card Standard released February 1995
Required

Designs for PC Card socket controllers, host controllers, and cards must all be based on the PCMCIA standards.

The February 1995 PC Card Standard adds requirements for minimum card information structure (CIS), 3.3v cards, multifunction cards, and cards that use DMA, and it introduces requirements for 32-bit PC Cards (CardBus). All PC Card devices must comply with these standards for PC 97.

Version 1.1 Addition:

1a. Surprise removal of a PC Card device must not cause a system failure
Required

Such a failure can include any spontaneous reboot, system stall, or blue screen. At a minimum, the PC Card driver must ensure that the PC system does not fail if the user accidentally pulls the PC Card out of its socket. The best solution is to lock the PC Card in the socket.

CardBus cards share the same PCI interrupt as their socket controller. For example, interrupt handlers for both the CardBus Media Access Control (MAC) driver and socket driver are chained to the same PCI interrupt request (IRQ), usually with the MAC driver chained ahead of the socket driver. When there is a surprise removal of a CardBus network card, a card removal interrupt is fired for the socket controller, but the network card's driver is at the head of the chain, so its interrupt handler gets called first.

For some PC Card drivers, the MAC driver erroneously reports that it owns the interrupt (even though its hardware is removed), probably because it reads the 0xff in its interrupt status register and interprets this as meaning that all its interrupts are pending.

Because the socket controller doesn't get called, it can't clear the interrupt, so the interrupt keeps firing until the nested interrupts cause a fault. To prevent this problem, the MAC driver should check whether its device is still present whenever it reads a value such as 0xff in its status register, and then recover gracefully when this occurs.

WHQL is testing and reporting on support for surprise removal now. This is required as of **October 1, 1997.** (Change date: April 30, 1997)

PC Card Socket Controllers

This section summarizes PC Card requirements and standards for socket controllers.

2. Support the Industry-standard ExCA base-register set

Required

The built-in PC Card 16 software in Windows includes drivers for the Industry-standard ExCA–compatible socket controllers. To be compatible with these drivers, socket controller implementations must support the Industry-standard ExCA base-register set. Notice that some controllers do not fully implement the base-register set and are incompatible. Also, some controllers implement extended registers or enhancements. The built-in Windows drivers do not exploit these features, even though the controller might be compatible.

3. Maintain mapping of IRQ Routing Register bits to system interrupt vectors

Required

The system design must maintain the mapping of the PC Card controller’s IRQ Routing Register bits to system interrupt vectors. This means that when an interrupt is programmed in the controller to occur on the IRQ*x* pin, the system’s IRQ routing causes the interrupt controller to generate the interrupt vector for IRQ*x*, and no other IRQ.

4. Support the industry-standard definition for CardBus bridges

Required

If the system supports CardBus, it must support the definition in “PCI to PCMCIA CardBus Bridge Register Description” (“Yenta” specification) for CardBus controllers (PCI-to-CardBus bridges). This definition includes a common PCI configuration space header assigned the Header Type field value of 82H. Although this is not yet incorporated into the PCI standard, Windows supports it. Any controller features that are not part of the “Yenta” specification will not be used in standard drivers. Any hardware-initialization or setup required to make the controller comply with “Yenta” or the other requirements listed here are the responsibility of the BIOS.

Because CardBus host controllers are PCI bus bridges, they will be supported (enumerated and configured) by the PCI software in Windows in the same manner as other PCI bus bridges. For information, see the “PCI” chapter in Part 3 of this guide.

5. Support both ISA and PCI interrupts on CardBus controllers

Required

The PC Card software will dynamically configure the bridge to utilize ISA interrupts for PC Card 16 cards, and use PCI interrupts for CardBus cards. Notice the requirement to "maintain mapping of IRQ routing" earlier in this section also applies to CardBus controllers. Also notice that systems implementing CardBus controllers must fully support PCI v. 2.1 and the additional PCI requirements for IRQ routing described in the "PCI" chapter in Part 3 of this guide.

6. BIOS initializes CardBus controller in 82365-compatible mode and reports it as PNP0E03 for backward compatibility

Recommended

CardBus host controllers are enumerated and configured in Windows 95, just as other PCI bus bridges. The PCI bus bridge support in Windows 95 is based on new requirements for PCI interrupt routing and bridge-window configuration. Because of this, full compliance with the latest PCI requirements is required for CardBus support.

For backward compatibility with Windows 95, there are steps the BIOS can take. Specifically, the BIOS must initialize the CardBus controller in Intel 82365-compatible mode and report it as device "PNP0E03, Intel 82365-compatible CardBus controller." Specifically, the requirements are as follows for BIOS POST time (CardBus controller ConfigSpace initialization):

- Command register (that is, offset 0x04) set to 0x07 (that is, IOSpaceEnable, MemSpaceEnable, BusMasterEnable).
- RegisterBaseAddress (that is, offset 0x10) set to 0.
- All memory and I/O windows set to 0 (that is, offset 0x1c−0x38).
- Interrupt Line register (that is, offset 0x3c) set to 0xff (no IRQ is assigned).
- LegacyBaseAddress (that is, offset 0x44) set to legacy mode I/O base address (such as 0x3e0).
- Other controller-specific initialization required to put the controller in legacy mode.

This puts the CardBus controller into Legacy mode where Socketsv.vxd (the Windows 95 Socket Services driver) can access it as an Intel PCIC-compatible controller at an I/O address (for example, 0x3e0).

Notice that the BIOS must be at least PCI 2.1 compliant, meaning that it must at least support BIOS function (AX=0xb10e) GetIRQRouting and return the necessary PCI IRQ routing information, including the routing information for the CardBus controller. In general, if the CardBus controller is on the system board, there must be a slot routing entry for it in the table. If the CardBus controller is a PCI add-on card, there must be routing information entries for each PCI slot in the system.

During Plug and Play BIOS enumeration, the BIOS should report the CardBus controller as *pnp0e03 with a compatible ID of *pnp0e00 and the I/O resource of two ports (for example, 0x3e0−0x3e1). Windows 95 does not know about *pnp0e03, but it does know about *pnp0e00, so it will load Socketsv.vxd (for generic Intel PCIC-compatible controllers) and everything will work with the above BIOS initialization. The Windows 95 PCI enumerator (Pci.vxd) does not know about the CardBus controller's PCI header (Type 2), so it will not report the CardBus controller device at all.

In Windows 95, when the BIOS enumerator sees *pnp0e03, it will hide the device and call the BIOS to disable it. When the BIOS receives the disable call for *pnp0e03, it should do the following:

- Command register (that is, offset 0x04) set to 0.
- RegisterBaseAddress (that is, offset 0x10) set to 0.
- Interrupt Line register (that is, offset 0x3c) set to 0xff.
- LegacyBaseAddress (that is, offset 0x44) set to 0.
- Other controller-specific initialization required to put the controller in CardBus mode.

7. Writable PCI Configuration Space bits not shared by CardBus controllers

Required

CardBus controllers are multifunction PCI devices, and Windows treats each function of a multifunction PCI device as an independent device. As such, there can be no sharing between functions of writable PCI Configuration Space bits (such as the Command register).

Notice that the PC Card 16-bit Interface Legacy Mode Base Address Register (offset 44h in the Type 2 PCI header) is the only exception to this requirement. This register must be shared between the two functions, as they must share the same registers for compatibility with the ExCA programming model.

8. Each R2 memory window in CardBus controller has it own page register

Required

For complete flexibility and support for typical configurations, CardBus controllers must support the independent location of R2 memory windows anywhere in the full system address space as recommended in the Industry-standard "Yenta" specification. Controllers that share a single page register among all R2 memory windows place the constraint on the system that all R2 memory windows must be located within the same 16-MB block. This is often not possible with typical DRAM (16 MB) and bridge (positive decode) configurations, and consequently will result in the disabling of cards in these cases.

Plug and Play Design for PC Card 16

This section summarizes the Plug and Play requirements for PC Card 16 cards.

The Windows operating system determines what type of card is plugged into the PC Card socket by examining the tuples on the card. For Plug and Play functionality, PC I/O cards must support a set of required information and configuration tuples. The PCMCIA bus enumerator uses these tuples to identify the card, load the correct device driver, and indicate all the possible configurations to the Plug and Play configuration manager. The configuration manager then dynamically assigns a valid configuration based on this information.

9. Required I/O card tuples supported

Required

You must implement the following items for any PCMCIA I/O card that connects to a PC 97 system:

- The PC Card must contain the device information tuple (CISTPL_DEVICE, 01h), the level 1 version/product information tuple (CISTPL_VERS_1, 15h), the configuration tuple (CISTPL_CONFIG, 1Ah), and the configuration table entry tuple (CISTPL_CFTABLE_ENTRY, 1Bh).
- The level 1 version/product information tuple must contain the name of the manufacturer and the name of the product in the product information string (TPLLV1_INFO, byte 4).
- The name of the manufacturer and the name of the product in the level-1 version/product information tuple must be composed only of ASCII characters greater than ASCII 20h and less than ASCII 7Fh.

Windows uses the information contained in the required and recommended tuples to create a unique device identifier for the card and to assimilate configuration information for the device. Windows uses the device configuration tuples to determine the general characteristics of the card.

Required I/O Card Tuples

Tuple identifier	Tuple code	Description and comments
01h	CISTPL_DEVICE	Device information (common memory). For non-memory cards, this tuple must be present, but the device type will be NULL.
15h	CISTPL_VERS_1	Level 1 version/product information: Product information string Product name Product number Other manufacturer information
1Ah	CISTPL_CONF	Configuration. Indicates the location of configuration registers and the registers present.
1Bh	CISTPL_CE	Configuration table entry. Appropriate configuration requirements for I/O space, interrupts, memory, and so on should be specified.
20h	CISTPL_MANFID	Manufacturer identifier. Card manufacturer identifier code. Defines manufacturer for this card.
21h	CISTPL_FUNCID	Function identifier. Provides function information about the card. Also includes system initialization information.

The device information tuple provides information about the memory devices used in the card's common memory space. The device type, size, and speed are used to configure the socket for efficient access to the card. This tuple must be present on PCMCIA I/O cards, but the device type must be NULL.

The Level 1 version/product information tuple contains human-readable information about the product and its manufacturer. This information is intended to be displayed to the user, where necessary. Windows uses the information contained in the product information and product name strings to construct the device identifier for that card. It also scans through the tuple, starting at the very beginning and going to the end of the product name string. The information gathered from the scan is one source for creating a 16-bit cyclic redundancy check (CRC) that Windows uses to construct the device identifier. Because the optional third and fourth strings in the tuple are not used in the CRC scan, devices that require unique numbers on each card can use these strings to store that information.

The configuration tuple tells the software where to locate the configuration registers that program the card's configuration, as well as which registers are present on the card.

Each configuration table entry tuple completely describes one valid configuration in which the card can operate. Each entry describes power, timing, I/O space, interrupt, and memory space requirements for the given configuration. Configuration software selects one of these configurations for the card based on the resources currently available in the system.

The manufacturer identifier tuple (CISTPL_MANFID, 20h) and the function identification tuple (CISTPL_FUNCID, 21h) add extra flexibility to a PC Card that connects to the PC:

- The manufacturer identifier tuple provides unique information about the manufacturer of this card. This code is registered with PCMCIA. Windows uses the manufacturer identifier tuple as one source for creating a 16-bit CRC used in the construction of the device identifier.
- The function identification tuple provides information about the class of device or what function the card provides (for example, memory, modem, disk, and so on). This information helps the software perform necessary installation tasks and locate compatible drivers. Although it is not required to make this determination, Windows uses the function identification tuple internally to determine what type of device is on the PC Card.

10. Configuration entry tuples listed in priority order

Required

Place the configuration entry tuples in the preferred order for configuring the device. Windows processes the tuples in the order they are placed in the card’s information structure (CIS). From these tuples, Windows creates logical configuration in this order and prioritizes them in the same order. Notice that for multiple voltage cards the voltage policy is to prioritize 3.3V configurations (if supported by the system) over 5V configurations, regardless of the order of the configuration table entry tuples (CISTPL_CFTABLE_ENTRY).

11. Maximum configuration options specified

Required

Many older PC Cards specified “fixed” configurations to address compatibility with existing software. However, this is not the intended use for tuples; the configuration software should be responsible for compatibility. The tuples should be used only to rule out configurations not supported by the hardware. If you must provide fixed configurations for an operating system other than Windows, you must still provide one or more entries that specify the maximum configurability the hardware can handle.

Plug and Play Design for CardBus

This section summarizes the Plug and Play requirements for CardBus cards. CardBus was designed as a combination of PC Card 16 and PCI. The goal is to gain the benefits of PCI in the PC Card format. Consistent with this goal, Windows support for CardBus places specific requirements on CardBus cards.

12. Configuration space meets the Common Silicon Guidelines

Required

The standard for CardBus defines a PCI-like configuration space that is not fully compliant with the PCI specification. Specifically, under the CardBus Standard, card vendors do not have to implement certain critical fields in the configuration space (described in the PC Card Standard as “allocated”). In the PC Card Standard Guidelines for silicon common to both PCI and CardBus products, the implementation of these fields is recommended.

However, to maintain compatibility with existing PCI system software and drivers for PC 97, Windows will support only CardBus cards whose configuration space is designed to meet the Common Silicon Guidelines.

This is required because CardBus configuration is performed by the PCI software, which knows how to deal with all aspects of PCI topology configuration, including bridging. Without the “allocated” fields, the cards cannot be treated fully as PCI devices and therefore cannot be supported under Windows.

The required “allocated” fields are listed in the following table.

Required “Allocated” Fields

Field	Description and comments
Vendor ID	This read-only field contains a Unique ID (in PCI space) for the manufacturer of the card. It is allocated by the PCI SIG.
Device ID Revision ID	These read-only fields are vendor-assigned values that uniquely identify the device (among all vendors of PCI or CardBus products).
Class Code	These read-only fields are defined in the PCI 2.1 specification. They describe what type of device this card is.
Max_Lat Min_Gnt	These read-only fields specify the desired settings for Latency Timer values, according to the PCI 2.1 specification. Values of 0 indicate the device has no major requirements for the settings of Latency Timers.
Interrupt Line	This register must be read-write and must not be connected to anything, just as on PCI cards. This register is used to store the current IRQ routing for the device.

13. RESERVED fields compliant with PCI v. 2.1

Required

The CardBus specification also lists two fields as RESERVED (offset 2C in the configuration space), which have since been defined in the PCI v. 2.1 specification. These are also required on CardBus cards for Windows compatibility.

Required RESERVED Fields

Field	Description
SubSystem ID	If different from Device ID
SubSystem Vendor ID	If different from Vendor ID

14. CardBus required and recommended tuples implemented

Required

For CardBus, Windows also requires the same set of card tuples as recommended in the PC Card Guidelines, as summarized in the following table.

Required Tuples for CardBus

Tuple identifier	Tuple code	Comments
04h	CISTPL_CONFIG_CB	
05h	CISTPL_CFTABLE_ENTRY_CB	
07h	CISTPL_BAR	
13h	CISTPL_LINKTARGET	Required as first tuple by PC Card Standard
15h	CISTPL_VERS_1	
20h	CISTPL_MANFID	
FFh	CISTPL_END	Required as end-of-chain tuple by PC Card Standard
21h	CISTPL_FUNCID	Recommended in PC Card Standard; required for PC 97

15. Writable PCI Configuration Space bits not shared by CardBus controllers

Required

CardBus controllers are multifunction PCI devices, and Windows treats each function of a multifunction PCI device as an independent device. As such, there can be no sharing between functions of writable PCI Configuration Space bits (such as the Command register).

For more information requirements related to the PCI Configuration Space, see the "PCI" chapter in Part 3 of this guide.

PC 97 Requirements for PC Card

This section summarizes additional PC 97 Hardware Design Guidelines.

Power Management for PC Card

This section summarizes the specific power management requirements for PC Card. Power management requirements for specific device classes are defined in the related chapters in Part 4 of this guide.

16. Compliance with "Device Class Power Management Reference Specification" for PC Card controller

Required

The "Device Class Power Management Reference Specification" for the PC Card Controller Device Class provides class-specific definitions of the OnNow device power states (D0–D3) for these devices. The specification also covers device functionality expected in each power state and the possible Wakeup event definitions for the class (for example, whether a card insertion should wake the system).

Version 1.1 Clarification:

Logo compliance testing for PC Card controllers began **July 1, 1997,** for support of the *PC Card Device Class Power Management Reference Specification* and the *Default Device Class Power Management Specification,* as described in the clarifications to item 5 of the "Basic PC 97" chapter.

17. PC Card 16 cards implement power-related events using the ReqAttn bit and the #STSCHG mechanism

Required

Any PCMCIA 5.0 card capable of signaling a Wakeup event to the system (as defined in the Device Class Power Management Reference Specification for its class) must implement the ReqAttn bit in the Extended Status register, its associated enable bit, and signaling on the #STSCHG line. PCMCIA 2.x cards must conform to the ExCA wakeup specification using but 4 of the Card Configuration and Status register to enable wakeup. Also, any card with a battery must support the Battery Voltage Detection (xBVDn) bits in the Pin Replacement register as currently defined in the PC Card Standard, and must support signal changes using the #STSCHG mechanism.

Version 1.1 Change:
Comment does not apply to support for the Battery Voltage Detection (xBVDn) bits. (Change date: August 6, 1997)

Power management requirements for PC Card 16 cards must be met as of **July 1, 1997**.

18. CardBus controllers and cards implement PCI power management specifications

Required

PCI-to-CardBus bridges and CardBus cards must conform to the requirements for power management on the PCI bus.

For information about PCI power management specifications, see the “PCI” chapter in Part 3 of this guide.

Version 1.1 Changes:
PCI-to-CardBus bridges must comply with the requirements defined in *PCI Bus Power Management Interface Specification, Revision 1.0* or higher. CardBus cards must also comply with the requirements defined in *PCI Bus Power Management Interface Specification, Revision 1.0* or higher. This requires that the CardBus card must use the CSTSCHG pin to signal wake-up events (since there is no PME# pin on the CardBus interface) and use PME_EN in the card's Configuration Space to enable wake-up events. Specifically, setting the PME_EN bit in the card's Configuration Space must provide the same behavior as setting both the GWAK and WKUP bits in the card's Function Event Mask Register. (Change date: August 6, 1997)

Logo compliance testing for CardBus controllers and cards will begin **April 1, 1998.** This is because this support requires PCI power management, which is not required until April 1, 1998.

19. ZV-compatible PC Cards compliant with the PC Card standard definitions for Zoomed Video

Required

The PC Card Standard defines the requirements for ZV cards.

Version 1.1 Change:

Power management requirements for Zoomed Video (ZV) must be met as of **July 1, 1997**.

Device Drivers and Installation for PC Card

This section summarizes requirements for device drivers for PC Card.

20. No user intervention required for correctly installing devices

Required

The user must not be required to perform any action other than to insert disks that contain drivers and other files.

21. Device functional immediately without restarting the system

Required

The user must not have to restart the system to be able to begin using the device, either after installation is complete or whenever the device is inserted in the system.

22. ZV-compatible PC Card driver uses DirectDraw Live Video Extensions

Required

ZV-compatible PC Card drivers must use software interfaces based on 32-bit DirectDraw Live Video Extensions (LVE) to configure the graphics controller to receive video input using the ZV Port. This includes programming the graphics controller to configure the format of the video data, its location on screen, and so on. LVE is part of DirectX 3.0.

ZV card device drivers must handle dynamic graphics state changes, such as resolution changes, color depth changes, and switching to and from full-screen MS-DOS–based applications.

Version 1.1 Change:
The driver should use Video Port Extensions (VPE), which is part of DirectX 5.0. (Change date: September 12, 1997)

References for PC Card

This section lists some of the publications, services, and tools available to help build hardware that works with Windows operating systems.

PC Card support in Windows 95
http://www.microsoft.com/hwdev/busbios/

Windows 95 DDK
MSDN Professional membership

Microsoft diagnostic utility for PC Card
Microsoft provides a diagnostic utility that supports multiple voltage cards, MFC cards and the new Device ID mechanism. This utility—Dtpl.exe—is posted on the Microsoft FTP and web servers.

"Yenta" specification: PCI to PCMCIA CardBus Bridge Register Description
Published by Intel Corporation, available to all "Yenta" members.
Phone: (800) 879-4683

PCMCIA Standards
Personal Computer Memory Card International Association
2635 North First Street, Suite 209
San Jose, CA 95134 USA

Device Class Power Management Reference Specification
http://www.microsoft.com/hwdev/onnow.htm

Version 1.1 References Update:
CardBus host controllers and Windows compatibility white paper
http://www.microsoft.com/hwdev/busbios/

PC Card Controller Device Class Power Management Reference Specification, Version 1.0
http://www.microsoft.com/hwdev/onnow.htm

PC Card diagnostic utility and white papers
Microsoft provides a diagnostic utility that supports multiple voltage cards, multifunction child cards, and the new device ID mechanism, and that can be used to help design or verify CIS for use with PC Card support under Windows. This utility (Dtpl.exe), plus white papers related to PC Card implementations under Windows operating systems, is available on the web site at http://www.microsoft.com/hwdev/busbios/.

PCI Bus Power Management Interface Specification, Revision 1.0
PCI Local Bus Specification, Revision 2.1 (PCI 2.1)
PCI SIG
Phone: (800) 433-5177
http://www.pcisig.com

PCMCIA standards, including *PC Card Standard Guidelines* and *PCI to PCMCIA CardBus Bridge Register Description* (Yenta specification)
PCMCIA
2635 North First Street, Suite 209
San Jose, CA 95134 USA
Phone: (408) 433-2273
Fax: (408) 433-9558
E-mail: office@pcmcia.org
http://www.pc-card.com/

Windows and Windows NT DDKs
MSDN Professional membership

Checklist for PC Card

PC Card Basic Requirements

1. All devices compliant with the PC Card Standard released February 1995
Required

1a.Surprise removal of a PC Card device must not cause a system failure
Required

PC Card Socket Controllers

2. Support the Industry-standard ExCA base-register set
Required

3. Maintain mapping of IRQ Routing Register bits to system interrupt vectors
Required

4. Support the industry-standard definition for CardBus bridges
Required

5. Support both ISA and PCI interrupts on CardBus controllers
Required

6. BIOS initializes CardBus controller in 82365-compatible mode and reports it as PNP0E03 for backward compatibility
Recommended

7. Writable PCI Configuration Space bits not shared by CardBus controllers
Required

8. Each R2 memory window in CardBus controller has it own page register
Required

Plug and Play Design for PC Card 16

9. Required I/O card tuples supported
Required

10. Configuration entry tuples listed in priority order
Required

11. Maximum configuration options specified
Required

Plug and Play Design for CardBus

12. Configuration space meets the Common Silicon Guidelines
Required

13. RESERVED fields compliant with PCI v. 2.1
Required

14. CardBus required and recommended tuples implemented
Required

15. Writable PCI Configuration Space bits not shared by CardBus controllers
Required

PC 97 Requirements for PC Card

Power Management for PC Card

16. Compliance with “Device Class Power Management Reference Specification” for PC Card controller
Required

17. PC Card 16 cards implement power-related events using the ReqAttn bit and the #STSCHG mechanism

Required

18. CardBus controllers and cards implement PCI power management specifications
Required

19. ZV-compatible PC Cards compliant with the PC Card standard definitions for Zoomed Video
Required

Device Drivers and Installation for PC Card

20. No user intervention required for correctly installing devices
Required

21. Device functional immediately without restarting the system
Required

22. ZV-compatible PC Card driver uses DirectDraw Live Video Extensions
Required