

The RadiSys logo is a dark blue rounded rectangle with the word "RadiSys." in white serif font. A thin white line extends from the right side of the logo, connecting to a small white circle on the left side of the title "PCI Server Reference".

RadiSys.

# PCI Server Reference

RadiSys Corporation  
5445 NE Dawson Creek Drive  
Hillsboro, OR 97124  
(503) 615-1100  
FAX: (503) 615-1150  
[www.radisys.com](http://www.radisys.com)  
07-0609-01  
December 1999

EPC, iRMX, INtime, Inside Advantage, and RadiSys are registered trademarks of RadiSys Corporation. Spirit, DAI, DAQ, ASM, Brahma, and SAIB are trademarks of RadiSys Corporation.

Microsoft and MS-DOS are registered trademarks of Microsoft Corporation and Windows 95 is a trademark of Microsoft Corporation.

IBM and PC/AT are registered trademarks of International Business Machines Corporation.

Microsoft Windows and MS-DOS are registered trademarks of Microsoft Corporation.

Intel is a registered trademark of Intel Corporation.

All other trademarks, registered trademarks, service marks, and trade names are property of their respective owners.

December 1999

Copyright © 1999 by RadiSys Corporation

All rights reserved.

# Quick Contents

---

<b>Chapter 1.</b>	<b>Description of the PCI Server</b>
<b>Chapter 2.</b>	<b>Communicating with the PCI Server</b>
<b>Chapter 3.</b>	<b>PCI Commands</b>
<b>Chapter 4.</b>	<b>Using the PCI Server</b>
<b>Appendix A.</b>	<b>Numerical List of Commands</b>
<b>Appendix B.</b>	<b>PSM Status Codes</b>
<b>Appendix C.</b>	<b>Related Publications</b>
	<b>Glossary</b>
	<b>Index</b>

# Notational Conventions

This manual uses the following abbreviations and terms in a specific way. Refer to the glossary for definitions of other technical terms used in the manual.

11	Hexadecimal base (for example, 0F0). Numbers without an 11 or Binary suffix are decimal numbers.
MB 11	Multibus 11 interface
MB 1	Multibus 1 interface
tape drive	Streaming cartridge tape drive
<b>Command Name</b>	Commands are shown in bold.



## **Note**

A note calls attention to an important fact.



## **CAUTION**

A caution points out something that could damage your hardware or data.

---

<b>1</b>	<b>Description of the PCI Server</b>	
	Introduction to the PCI Server .....	1
	Features .....	3
	Definition of Roles .....	4
	Products that Run the PCI Server .....	6
	Board Products .....	6
	Packaging of the PCI Server.....	7
	Firmware .....	7
	Pre-configured and Bootable.....	7
	Configurable.....	7
	PCI Server Hosts .....	8
	Standards .....	9
	Performance Features .....	9
	Cache and Read-ahead Configuration .....	9
	Reasons for Read-ahead Caching .....	9
	Read Operation with Caching.....	10
	Write Operation with Caching.....	10
	Guidelines for Setting the Cache Parameters .....	11
	Seek Optimization and Command Ordering.....	13
	Seek Optimization .....	13
	Command Ordering .....	14
	Statistics .....	14
	Direct Access.....	14
	SCSI Reset.....	15
	Initializing the PCI Server .....	17
	Loading the PCI Server .....	17
	Scanning the SCSI Bus.....	17
	Sending Commands.....	17
	Two Kinds of Commands.....	18
	Standard.....	18
	Implementation-specific .....	18
	Uses for SCSI Pass Thru .....	19
	SCSI Bus Error Handling .....	19
	Target Mode .....	19

---

<b>2</b>	<b>Communicating with the PCI Server</b>	
	Message Passing and Transport.....	23
	PCI for MB II Boards.....	24
	PCI for MB I Boards.....	24
	PCI for PCs.....	24
	Format of the PCI Data Structures.....	26
	Peripheral Command Message (PCM).....	27
	Peripheral Status Message (PSM).....	31

---

<b>3</b>	<b>PCI Commands</b>	
	List of PCI Server Commands.....	35
	Command Types.....	39
	Command Descriptions.....	40
	Clear Reset Blocking Condition.....	41
	Erase Unit.....	42
	Flush Get Send Data.....	43
	Format Unit.....	44
	Get Device List.....	50
	Get SCSI ID.....	53
	Get Send Data.....	54
	Get Server Information.....	57
	Get Statistics.....	59
	Get Target Mode Options.....	63
	Immediate Reset.....	65
	I/O Sleep.....	66
	Load.....	67
	Locate PCI Server.....	68
	Query Unit Characteristics.....	69
	Query Unit Options.....	78
	Read Data.....	81
	Read Data Direct.....	83
	Read Data and Verify.....	85
	Read Defect List.....	86
	Read Device Capacity.....	90
	Reassign Blocks.....	91
	Recalibrate.....	93
	Release.....	94
	Request Reset Notification.....	95
	Request Unit Attention Notification.....	96
	Reserve.....	97
	Reset Device.....	99

Reset Notification .....	100
Retension Tape .....	101
SCSI Pass Thru.....	102
Seek Beginning of Tape.....	111
Seek End of Data .....	112
Seek Filemark.....	113
Set Reset Blocking Time .....	115
Set Statistics .....	116
Set Target Mode Options.....	117
Set Unit Characteristics .....	119
Set Unit Options .....	126
Start Unit .....	130
Stop Unit.....	131
Synchronous Negotiation Control.....	132
Test I/O.....	136
Unit Attention Notification.....	137
Unload .....	138
Write Data .....	139
Write Data Direct .....	141
Write Filemark .....	143

---

## 4 Using the PCI Server

Finding the PCI Servers in the System .....	145
Example.....	146
Preparing to Send Commands.....	147
Changing the Parameters for a Unit.....	149
Reading a Block of Data from a Unit .....	150
Writing a Block of Data to a Unit.....	151
Subscribing to an Announcement Service .....	152
Using Target Mode.....	154
Role of the Sender .....	155
Role of the Receiver .....	157
Buffers for Target Mode.....	158
Data Buffers .....	159
Control Buffers.....	159
Buffer Size Compared to Data Size.....	160
Target Mode Operation .....	162
Commands and Target Mode .....	162
Doing Immediate Synchronous Negotiation.....	163
Sending Messages in the iRMX III OS.....	163

---

<b>A</b>	<b>Numerical List of Commands</b>	
<b>B</b>	<b>PSM Status Codes</b>	
	Error Handling.....	169
	PSM Status Code Definitions .....	170
<b>C</b>	<b>Related Publications</b>	
	Intel Publications .....	173
	IEEE Publications.....	174
	ANSI Publications .....	174
	<b>Glossary</b>	173
	<b>Index</b>	177

---

---

## Tables

Table 3-1. Alphabetical List of PCI Commands.....	36
Table 4-1. Buffer Size Compared to Data Size.....	161
Table A-1. Numerical List of PCI Commands.....	166
Table B-1. Status Codes.....	170

---

## Figures

Figure 1-1. The Server Handles Both the PCI and SCSI Interfaces.....	2
Figure 1-2. The Server Handles the PCI Interface and the SCSI Interface on a Single Board	2
Figure 1-3. PCI Terminology.....	5
Figure 1-4. Dividing the Cache.....	11
Figure 1-5. Clients Can Transfer Data Between Systems.....	20
Figure 1-6. A PCI Client Can Talk With A Non-PCI Program.....	20
Figure 1-7. The Target Server Offers Eight Logical Units.....	21
Figure 2-1. Regular and Short-circuit Message Passing.....	25
Figure 2-2. Byte Ordering.....	26
Figure 2-3. The PCM in the General Interrupt Message.....	28
Figure 2-4. The PCM in the Buffer Request Message.....	29
Figure 2-5. The PSM in the General Interrupt Message.....	31
Figure 2-6. The PSM in the Buffer Request Message.....	32
Figure 4-1. Example System.....	146
Figure 4-2. Set Unit Characteristics.....	147
Figure 4-3. Set Unit Options.....	149
Figure 4-4. Read Data.....	150
Figure 4-5. Write Data.....	151
Figure 4-6. Unit Attention Notification.....	153
Figure 4-7. To Transfer Data, Both Sending and Receiving Clients Must Use Commands	154
Figure 4-8. Client C2 Nests the SCSI Command in a Buffer Request.....	155
Figure 4-9. Sender's SCSI Pass Thru Message.....	156
Figure 4-10. Receiver's Get Send Data Command.....	157



# Description of the PCI Server

---

# 1

This user's guide tells how to use the Peripheral Controller Interface (PCI) server with any one of several different RadiSys board products. This manual is not a complete reference document; it refers to other documents about the related products, Multibus II architecture, and message passing.

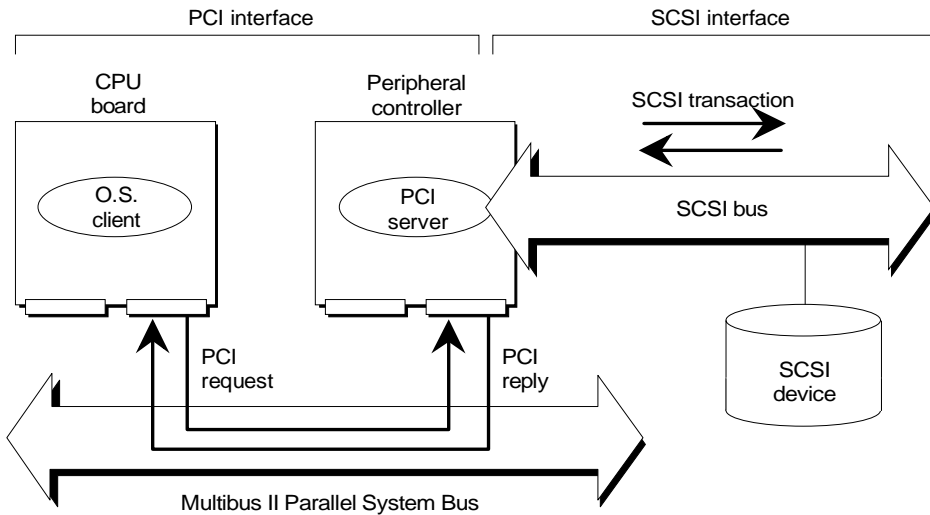
This manual is intended for people who are writing their own device driver or bootstrap loader. It also provides information for people doing one or more of the following things:

- Using a board with the PCI server in the firmware.
- Using an RadiSys-supplied iRMX<sup>®</sup> or UNIX System V/386 device driver for PCI.
- Configuring an iRMX system to include the PCI server. The PCI server is part of the iRMX software and you can configure it into your version of the operating system.
- Using a board with the PCI server configured as part of the operating system on the board. The PCI server is already configured in the UNIX System V/386 supplied by Intel.

## Introduction to the PCI Server

The Peripheral Controller Interface (PCI) server is system software or firmware that enables access to SCSI peripherals in Multibus systems. The server can transparently handle multiple, concurrent I/O commands from several boards running different operating systems. The server insulates the boards from the SCSI bus, so that each board does not need specific information about the SCSI bus and devices.

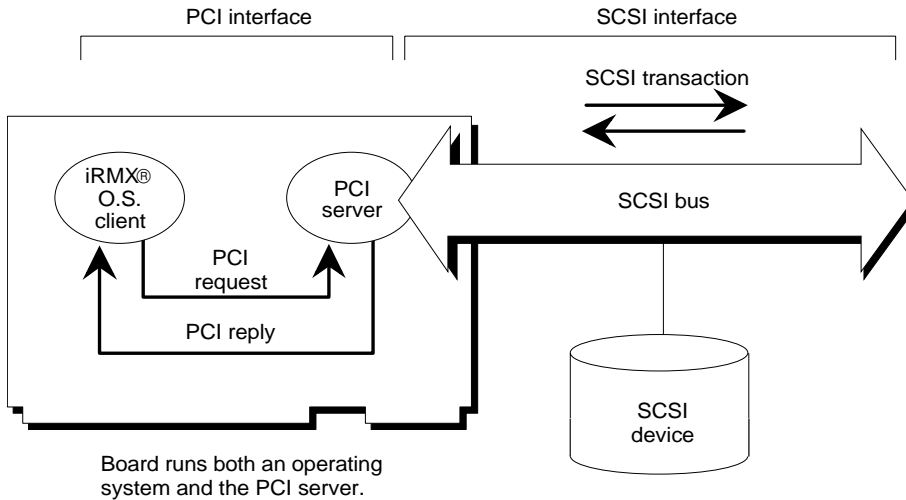
Figure 1-1 shows a typical access in a system: the server accepts an I/O command from the requesting board. The server then accesses SCSI peripheral devices by using the SCSI bus protocol. After the device completes the I/O operation, the server replies to the requesting board.



OM02080

**Figure 1-1. The Server Handles Both the PCI and SCSI Interfaces**

Figure 1-2 shows how the PCI server operates on a single board.



OM02081

**Figure 1-2. The Server Handles the PCI Interface and the SCSI Interface on a Single Board**

# Features

The PCI server offers these features:

- Provides an interface to SCSI peripherals that is hardware-independent. It provides this kind of access to hard disk drives, flexible disk drives, and streaming cartridge tapes.
- Allows the passing of SCSI commands directly to the devices, without interpreting the commands.

With this SCSI pass-thru method, the requesting board can use commands that are unique to a specific device. This method extends PCI's support of peripherals to other SCSI devices, such as image scanners, optical disks, and processor boards.

- Supports full SCSI arbitration, disconnect, and reselect.

The server can handle multi-threaded I/O operations with up to 56 SCSI peripheral units. It also allows multiple peripheral controllers on the same SCSI bus.

- Supports synchronous and asynchronous SCSI transfers.
- Processes up to 250 PCI commands concurrently.
- Allows logical addressing as provided by the SCSI protocol.
- Improves performance by offering adjustable block caching, read ahead, and I/O command ordering.
- Collects statistics, such as the number of disk errors, cache hits, and other recurring events. This information can help you tune the performance of the board.
- Supports a multiple board system. The server can concurrently handle I/O operations from multiple boards that are running different operating systems.
- Supports data accesses directly to and from a client's buffer if the client and server are on the same board. This avoids unnecessary copies of data and may improve performance.
- Supports target mode. Two chassis that are connected by a SCSI bus can communicate by using target mode.

See also: Using target mode, in Chapter 4

## Definition of Roles

This manual uses these names for the different parts of the system, as shown in Figure 1-3:

**host** A board that communicates across the Parallel System Bus (PSB) by using message passing.

**client** The requester at the PCI interface. It requests services from the PCI server. The client is typically an operating system's device driver on a CPU board.

**peripheral controller**

The board that has the PCI server software or firmware running on it. It attaches directly to the SCSI bus. (In some MB II and in all MB I configurations, the peripheral controller has both the client and server functions on it.)

**initiator**

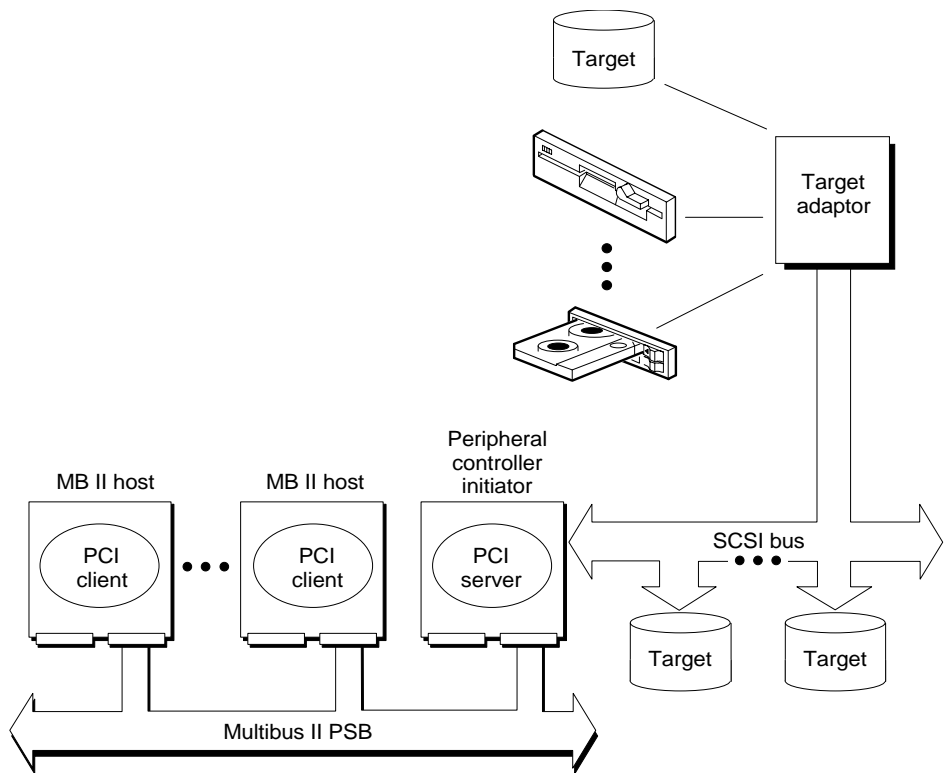
A device that starts a SCSI transaction (the requester on the SCSI bus). In this manual, it usually is the peripheral controller; however, for target mode, the initiator may be a second peripheral controller on the SCSI bus.

**server** The software or firmware that handles PCI requests from the clients and returns the results to them.

**target** The responder on the SCSI bus, responding to a SCSI transaction begun by an initiator. This is a SCSI device attached directly to the SCSI bus or SCSI device attached to a target adapter board, which is attached to the SCSI bus. In target mode, the PCI server may be a target.

**target adapter**

A board that controls from one to eight SCSI devices. Some devices have embedded controllers; others rely on separate controlling hardware, called the target adapter.



OM02082

**Figure 1-3. PCI Terminology**

# Products that Run the PCI Server

## Board Products

At the time of publication, the PCI server supports the SCSI channels on these boards:

- iSBC 386/258S and 386/258D

The PCI server is in the firmware. You can find out the firmware version by using the **Get Statistics** command.

- iSBC 386/258 with MSA SIT Kit

The 386/258 System Integration Tool (SIT) kit adds a full set of MSA (Multibus II System Architecture) firmware to the 386/258 board. However, the SIT kit firmware does not include the PCI server. Instead, the 386/258 can run the PCI server provided with the iRMX OS or the UNIX System V/386 operating system packages.

- iSBC 386/12S and iSBC 486/12S

These Multibus I CPU boards can run the PCI server from the iRMX OS. The PCI server on these boards uses short-circuit message passing.

- iSBC 486/133SE and iSBC 486/166E

These Multibus II CPU boards run the PCI server from the iRMX OS.

- iSBC p5090, iSBC P5120 class, iSBC P5200

These Multibus II CPU boards run the PCI server from the iRMX OS.

### ⇒ **Note**

The PCI server is not compatible with the iSBC 186/224A (Multibus II) or iSBC 221 (Multibus I) peripheral controller boards.

## Packaging of the PCI Server

The PCI server can be embedded in EPROM or boot-loaded onto a controller.

### Firmware

The iSBC 386/258S and 386/258D peripheral controller boards have the PCI server in their firmware.

### Pre-configured and Bootable

The Intel UNIX System V/386 and iRMX Operating Systems have the PCI server already configured and bound with the iRMX OS to form a boot image for the 386/258S and 386/258D boards that have MSA firmware.

### Configurable

The iRMX products include a binary file of the PCI server. If you want to use the server, you must configure it as a system job on ICU-configurable systems or load it as a loadable job for DOSRMX or iRMX for PCs systems. Many RadiSys boards can use the configurable version of the PCI server: 386/258 with MSA firmware, 486/133SE, 486/166, 486/12S, 386/12S, iSBC P5090, iSBC P5120, and iSBC P5200.

See also: *System Configuration and Administration manual*

The iRMX systems offer two modes of operation:

- The PCI server host runs a configuration of the operating system that includes the Human Interface. In this case, the server requires its own dedicated peripheral device.
- The PCI server host runs an I/O controller configuration of the operating system, and all peripherals are available to the other hosts.

# PCI Server Hosts

The PCI server must be used with the iRMX Operating System, and operates in several environments:

- With iRMX III on MBI and MBII, it runs as a first-level job.
- With MBI and MBII, the PCI server uses short-circuit message passing to communicate with PCI Clients on the same board.
- In the DOSRMX/iRMX for PCs environment, the PCI server can be loaded with **sysload**. It uses short-circuit message passing to communicate with the local PCI device driver, which is also loaded with **sysload**.

For the PCI server to operate, it must have:

- A SCSI interface chip on the board hosting the PCI server
- A SCSI host adapter present that the PCI server supports when the server is loaded with **sysload** into DOSRMX/iRMX for PCs

The PCI server has software that controls the SCSI interface chips on Intel's MBI & MBII boards.

In the PC environment, multiple SCSI host adapters are supported by multiple PCI servers. The servers must be loaded with **sysload** using appropriate configuration information for each SCSI host adapter. When you load a PCI server, you must also load a new PCI device driver to communicate with that server.

⇒ **Note**

The PCI server is not compatible with SCSI ASPI managers. If an ASPI manager is used, and the PCI server is loaded with **sysload**, the PCI server will not operate correctly.

The PCI server also runs on other host adapters.

See also: Products that Run the PCI Server, in this chapter

## Standards

The PCI server complies with these standards:

- SCSI (ANSI X3.131-1986)
- Common Command Set (CCS, ANSI X3T9.2, 85-52 Rev 4B)
- QIC 104 class of sequential storage device implementation for SCSI (ANSI X3.146-1987)

Contact your sales representative for current information on the makes and models of drives that you can use with the PCI server.

## Performance Features

The PCI server provides methods to tune and monitor its performance:

- Caching and read ahead
- Seek optimization
- Command ordering
- Statistics
- Direct accesses to a client's buffer

You can set parameters for all of these features for each unit at runtime. This means you can use the same peripheral controller for a wide range of applications.

## Cache and Read-ahead Configuration

### Reasons for Read-ahead Caching

The read-ahead cache feature improves performance in the two following situations:

- After the server finishes reading the block of data that it is currently reading, the client asks for the next sequential block.
- The client does not keep a cache of recently read data and it accesses the same data again.

To cover both of these cases, the server saves the requested data block and a number of surrounding blocks in its local memory. This local memory serves as a disk cache. On subsequent reads, the server checks if the requested block is in the cache. If it is in the cache, the operation is called a *cache hit* and the server retrieves the data from the cache. If not, it's called a *cache miss* and the server retrieves the data from the disk.

This process is called read-ahead data-block caching. Read-ahead data-block caching reduces the number of disk accesses and, therefore, the average I/O latency. In particular, it can improve the performance of multi-user and multi-tasking applications.

The cache is organized as a set of buffers called cache lines. Each line has the same size as the programmed read-ahead size.

If the cache is not full, the server stores the data to lines that are empty. If the cache is full, the server chooses which lines to replace by using the least-recently-used algorithm. This algorithm fills the line that the server has not read or written for the longest period of time.

## Read Operation with Caching

When a client sends a read command, the server first tries to retrieve the data from cache. If the data is not cached, the server reads it from the device and updates the cache according to the read-ahead parameters.

For example, if the client asks to read one block with a block size of 1K at block 0, and the read ahead size is 16 blocks, the server puts blocks 0 through 15 in the cache. In fact, if the client asks to read any single block from 0 through 15, the server puts blocks 0 through 15 in the cache.

## Write Operation with Caching

When a client sends a write command, the server manages the cache and then sends the data to the device. It uses the following write-through policy:

- If the data is not in the cache (cache miss), the server handles the cache in one of these ways:
  - If the amount of data is less than a full line, the server does not put the data in the cache because the server would need to make a disk read to get the full line.
  - If the amount of data is at least the size of the cache line, the server puts all of the data in the cache. However, if a few blocks at either end do not cover whole cache lines, the server does not put these few blocks in the cache.
- If data from the requested block address is already in the cache, the write operation is a cache hit. As soon as a write command comes from the client, the server updates the block(s) of data in the cache with the newly written data and then issues a write command to the device.

In summary, the data is retained in the cache in all cases in which it is possible to do so without requiring any data to be read from the disk. In all cases, the data is immediately written to the disk.

## Guidelines for Setting the Cache Parameters

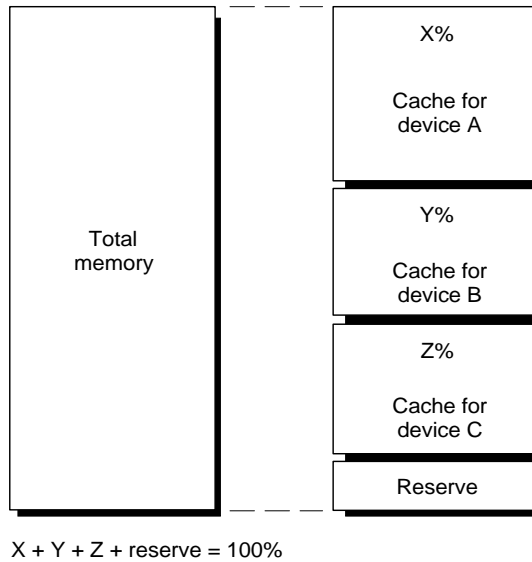
The **Set Unit Options** command includes these parameters for read-ahead caching:

- Cache enable/disable
- Buffer size
- Read-ahead size, which can be up to 63K bytes long

Figure 1-4 shows an example of a cache divided among three devices.

### Default Division of Cache

During initialization, the server looks for devices on the SCSI bus. Keeping a small part of the cache in reserve, the server divides the rest of the cache equally among the attached hard disk drives. The PCI server sets the default number of read-ahead blocks to the size of a single track.



OM02083

**Figure 1-4. Dividing the Cache**

## Changing the Total Size of the Cache

In Figure 1-4, 100% means the total cache available to the PCI server. The method for changing the size depends on the product:

- For the firmware version of PCI (embedded controller), the server uses almost all the on-board RAM. You can add more RAM to the board to get a larger cache.
- For the ICU-configurable software version of PCI, you can use the iRMX interactive configuration utility (ICU) to change the amount of cache.

See also: *ICU User's Guide and Quick Reference*

- For DOSRMX and iRMX for PCs, use the **sysload** command.

See also: *System Configuration and Administration*

## Changing the Default Settings

The default cache settings generally produce good results. Under two circumstances, however, you should consider changing them:

- If you can't guarantee that the drives will be powered up when the controller initializes them, you should use the **Set Unit Options** command to set the cache buffer size for all units.

Assume, for example, that the controller is attached to four drives and only two are fully powered when the controller initializes them. The controller assigns almost 50% of the cache buffer memory to each of the two drives. The remaining two drives have to share the small reserved space. To change this, you would use the **Set Unit Options** command to reduce the memory for the first two and then use the command to increase the memory for the remaining two.

- If your application does not do sequential accesses, then the read-ahead feature is actually a penalty and will make your accesses slower. In this case, you should use the **Set Unit Options** command to set the read-ahead option to your most common access size or disable the cache completely.

## Seek Optimization and Command Ordering

Two parameters, seek optimization and command ordering, govern the order used by the PCI server to satisfy outstanding (queued) requests for a single device. Together, these parameters provide 12 different ways to handle the order of requests.

### Seek Optimization

Excessive seeking can occur when a client makes concurrent accesses to multiple files in physically discontinuous areas on a single disk. By choosing from these algorithms, you can balance the system's needs for quick response and fair access.

<b>Method</b>	<b>Description</b>
First Come, First Served	The server satisfies requests based on the order of their arrival. This gives equal priority to all requests, but is least efficient for the movement of the drive's head.
Elevator	The server first satisfies requests for data that lies downstream of the direction a drive's head is moving. When all requests lying in that direction have been satisfied, the server reverses the seek direction. This algorithm is like an elevator that can pick up people when it's going up and, after it's gone all the way up, it can pick up people when it's going down.
CSCAN	The CSCAN algorithm is similar to the elevator algorithm. However, the server satisfies requests only when the head moves in one direction. As the head sweeps from the outside edge to the center of the disk, the PCI server satisfies requests. When the server has no more requests for data toward the center of the disk, the head moves back to satisfy the request closest to the edge and sweeps toward the center again. This algorithm is like an elevator that can pick up people only when it's going up. This algorithm is fairer than the elevator algorithm.
Shortest Seek Time First	The server satisfies the request for data blocks that lie closest to a drive's current head position, regardless of the direction. This method is most efficient for the movement of the drive's head, but it can starve the access to files at the edges of the disk.

## Command Ordering

Command ordering assigns one of these priorities to read and write operations:

- Writes before reads
- Reads before writes
- Equal priority of reads and writes

In the typical operating system, read calls are synchronous. Operating systems that maintain their own buffer caches often have asynchronous write calls. If an operating system has application read calls that are synchronous and application write calls that are asynchronous, it may benefit from the reads-before-writes option.

Command ordering takes precedence over seek optimization. As an example, consider the configuration that uses read-before-write ordering and shortest-seek-time-first optimization: when the server has a request for a write that is close to the head and a request for a read farther away, it will do the read first.

## Statistics

The PCI server keeps track of events such as the number of cache hits and misses, the total number of reads and writes, and the number of peripheral errors. By using these statistics, you can gauge the effectiveness of your configurations.

See also: **Get Statistics** and **Set Statistics** commands in Chapter 3

## Direct Access

If the client and server are on the same board, the client may give **Read Data Direct** and **Write Data Direct** commands to the server. The commands include a physical pointer to the client's buffer. The server uses the pointer to read or write directly to the buffer and does not cache the data. Because the server does not make extra copies of the data, performance is improved.

# SCSI Reset

Clients can cause the PCI server to reset the SCSI bus in one of three ways:

## Reset Unit command

To reset the SCSI bus, a client sends the **Reset Unit** command. Because this command is queued like any other command, it does not have any effect if the PCI server is hung before it has a chance to read the command.

## Immediate SCSI Reset command

A client can also send the **Immediate SCSI Reset** command, which causes the PCI server to immediately issue a SCSI bus reset. Again, if the server is hung, this command has no effect.



### CAUTION

If the server does act on the **Immediate Reset** command, it violates all of its internal mutual exclusion and synchronization, so it might crash. Use the **Immediate Reset** command only under extreme circumstances.

## Reset through interconnect space (for Multibus II)

In addition to resetting the SCSI bus, this option resets the peripheral controller. When the PCI server initializes itself on a Multibus II board, it examines byte 16 of the Firmware Communications Record:

Name	Bit	Value	Meaning
RST0	0	0	Not reset.
		1	First instance of PCI server resets its SCSI bus.
RST1	1	0	Not reset.
		1	Second instance of PCI server resets its SCSI bus.
RFU	7-2	0	Reserved.

If the client detects that the server and the SCSI bus are hung, it should set this Firmware Communications byte in interconnect space and use a Multibus II local reset to reset the peripheral controller board.

As implied by the RST0 and RST1 bits, a single peripheral controller board can have more than one PCI server.

See also: Chapter 2 shows an example of a board with two servers. For a general description of the Firmware Communications Record, see the *Multibus II Interconnect Interface Specification*. For more information about local resets, see the *Multibus II Initialization and Diagnostics Specification*.

Either the PCI server or a peripheral device on the SCSI bus may reset the SCSI bus. When the server detects a reset on the SCSI bus, it does the following:

1. Returns all pending commands that require SCSI bus access and sends the status code, I/O Interface Reset.
2. If target mode is enabled, the PCI server discards all state information about pending target mode SCSI transactions and returns all pending **Get Send Data** commands to the host with the status code, I/O Interface Reset.
3. Denies access to the SCSI bus for a period of time.

The **Set Reset Blocking Time** command sets the time period. The default time is 2 seconds; the maximum is infinity.

4. Notifies each client that the SCSI bus is unavailable, if the client previously asked for this service using the **Request Reset Notification** command.
5. Returns all new commands with the status code, I/O Interface Reset.
6. Resumes operation after the specified time period.

The client can use the **Clear Reset Blocking Time** command to resume operation before the time period is finished, which is especially useful when the time period is set to infinity.

# Initializing the PCI Server

## Loading the PCI Server

Firmware versions of the PCI server copy themselves from EPROM to RAM as soon as they are invoked. Software versions of the server are bootstrap loaded.

## Scanning the SCSI Bus

During initialization, the PCI server scans the SCSI bus by issuing the **SCSI Test Unit Ready** and **SCSI Inquiry** commands to find out how many units are attached. It sends each of these commands to all logical unit numbers (LUN) at every SCSI ID that it finds (except, of course, to itself).

After the server finds out how many units are attached, it calculates the default cache size and read-ahead size for each unit.

## Sending Commands

On Multibus II boards, after the PCI server initializes itself, it sets the first byte of the Firmware Communications Record in interconnect space to a value of 81H. Clients may use this as a synchronization point to start issuing commands to the server. However, the recommended method is to use the **Locate PCI Server** command to detect when the server is ready, because this method is independent of specific implementations.

If commands are issued before the server is ready, they may be lost.

# Two Kinds of Commands

The server has two kinds of commands: standard and implementation-specific.

## Standard

The PCI server offers a basic set of commands, called the standard commands. The standard commands provide generic I/O operations and can potentially be used to access I/O interfaces other than SCSI. The PCI standard commands will be compatible with future releases of the PCI server.

## Implementation-specific

The PCI server also offers some implementation-specific commands to do tasks that the standard PCI commands do not support:

### **SCSI Pass Thru**

This command lets the client send a SCSI command directly to a device. Many devices offer unique features, so the **SCSI Pass Thru** command lets the client provide the actual SCSI command. The PCI server manages the transaction but does not interpret the command.

### **Get Statistics** command

The PCI server keeps a record of several types of actions, including cache hits and disk errors. When the client issues the **Get Statistics** command, the server returns this information. The client can also use this command to find out the version number of the PCI server.



### **Note**

To avoid having to modify a device driver for future versions of the PCI server, use only the standard commands for the device driver. Implementation-specific commands are not guaranteed to be compatible in the future.

## Uses for SCSI Pass Thru

The **SCSI Pass Thru** command makes three different tasks possible:

- Sending vendor-unique commands to devices that are supported by the PCI server. The vendors of some disk and tape drives offer features that can only be accessed by writing vendor-specific information to the drives.
- Communicating with devices that cannot be accessed by using the standard PCI commands (for example, scanners, optical disks, and printers).
- Using the device controller as a traditional SCSI host adapter.

⇒ **Note**

Data transferred with the **SCSI Pass Thru** command is not cached. An I/O operation with this command does not provide access to the cache, command reordering, and other features of the PCI server.

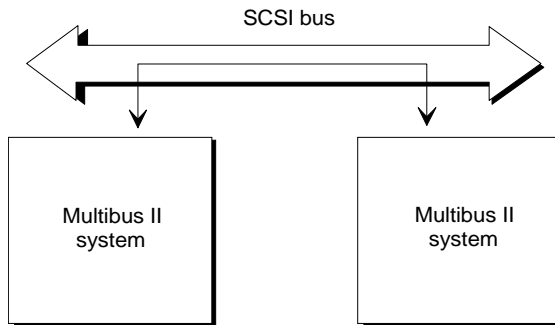
## SCSI Bus Error Handling

When the PCI server detects an error on the SCSI bus, it issues a **SCSI Request Sense** command, and it returns the Sense Key and Extended Sense Key values in its reply to the client. (The **SCSI Pass Thru** command is a possible exception to this; it has a bit that can be used to turn off this feature.)

See also: Device Specific Error in the Peripheral Status Message in Chapter 2 for specific information about this reply message

## Target Mode

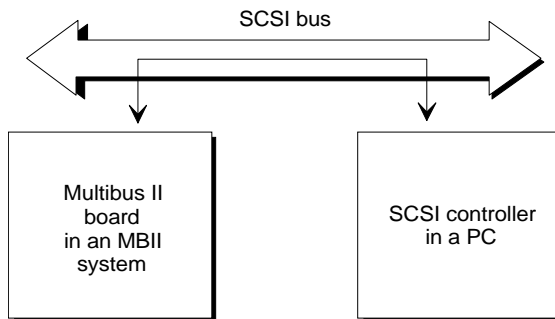
Target mode lets PCI clients communicate with other PCI clients over the SCSI bus. As shown in Figure 1-5, the clients do not have to be in the same chassis, as long as they are connected by the SCSI bus. The PCI server in one chassis acts as the initiator and starts the communication; the PCI server in the second chassis acts as the target and responds. These roles may change with each transaction, but they do not affect the normal access by the PCI server to the SCSI peripherals.



OM02084

**Figure 1-5. Clients Can Transfer Data Between Systems**

Target mode also lets a PCI client communicate over the SCSI bus with programs that do not follow PCI protocol, as shown in Figure 1-6.

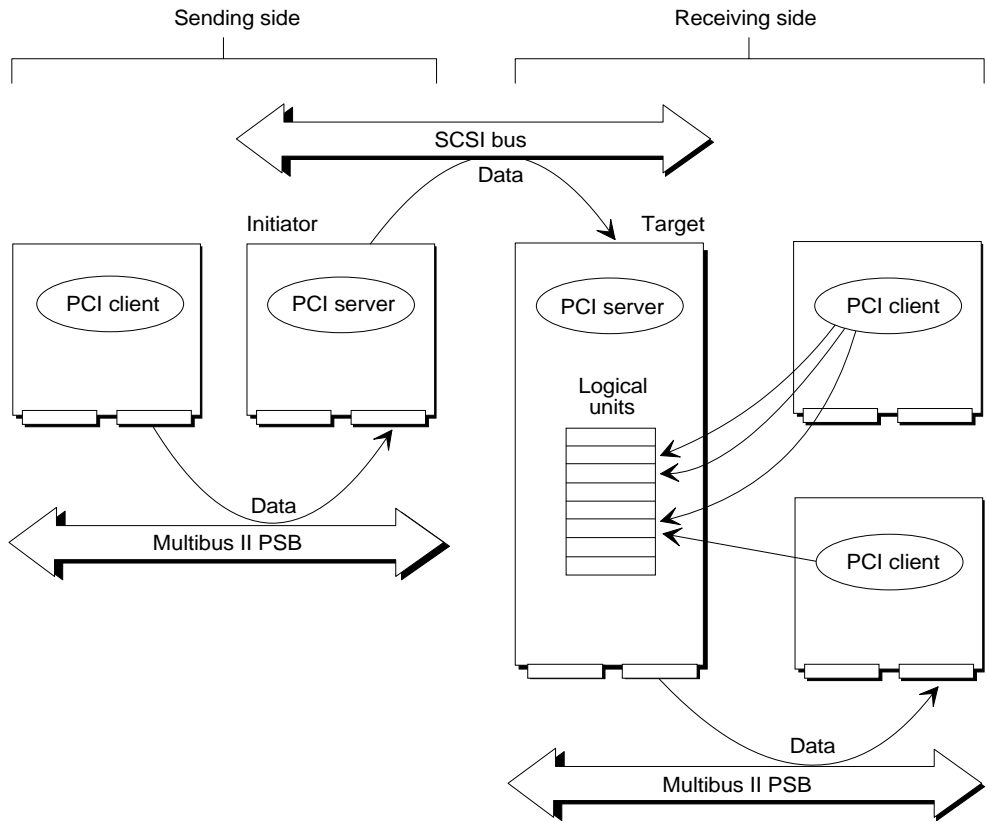


OM02085

**Figure 1-6. A PCI Client Can Talk With A Non-PCI Program**

To be a target, the server must let its board behave as a peripheral. The SCSI protocol uses the processor device type for this purpose. The PCI server supports the processor device type and these SCSI commands: **Test Unit Ready**, **Inquiry**, **Request Sense**, and **Send**.

Figure **Error! Reference source not found.-7** shows how the PCI server can handle eight SCSI logical units in target mode. It has independent buffers and status information for each one. A client may listen at one or all eight of the logical units on the server. Clients control the size and number of buffers by using the **Set Target Mode Options** command. To find out the settings, they can use the **Get Target Mode Options** command.



OM02086

**Figure 1-7. The Target Server Offers Eight Logical Units**

The sending client embeds the **SCSI Send** command in the PCI command called **SCSI Pass Thru**. The server puts the **Send** command on the SCSI bus with the data.

See also: The receiving client must issue a command called **Get Send Data** to retrieve the data from the PCI server. See Chapter 4 for more information.





This chapter describes:

- Message passing and transport for Multibus II boards
- PCI for Multibus I boards
- PCI server message passing
- PCI message formats

## Message Passing and Transport

Most Multibus II boards communicate by passing messages. Each board equipped with the message passing coprocessor (MPC) can send and receive solicited and unsolicited messages. Solicited messages transmit large amounts of data (up to 16 megabytes minus 1 byte) between Multibus II boards. Unsolicited messages are intelligent interrupts; they also prepare for the passing of solicited messages.

The *Multibus II Transport Protocol Specification and Designer's Guide* sets a standard for communication between Multibus II boards in a system. The transport protocol defines structured packets that the PCI server and clients use to pass commands, status, and data. The server and clients embed their commands and status in the last 20 bytes of a general interrupt message or in the last 16 bytes of a buffer request message.

The commands, status, and data do not contain process ID fields that bind replies with requests. Instead, the server and client use the transaction facilities of the Multibus II transport protocol to bind client requests with server replies.

### ⇒ **Note**

The server may use fragmentation for both request and reply. Clients must be able to support fragmentation.

See also: For a detailed discussion of message passing and fragmentation, see the *MPC User's Manual* and the *Multibus II Transport Protocol Specification and Designer's Guide*.

## PCI for MB II Boards

In a Multibus II system, clients use message passing to communicate with a PCI server across the parallel system bus. Any host may be a client of a server.

The client and server see each other as a socket, which is a combination of host ID and port ID. A host ID is a name for a piece of hardware, and a port ID is a name for a piece of software.

If a host is using the iRMX operating system, the client can be on the same board as the server. Because PCI servers and clients (device drivers) view each other as sockets, it doesn't matter to the client or the server whether they are on the same board or not. If they are on the same board, as in Figure 2-1, they use short-circuit message passing to communicate through the equivalent of MB II message space.

## PCI for MB I Boards

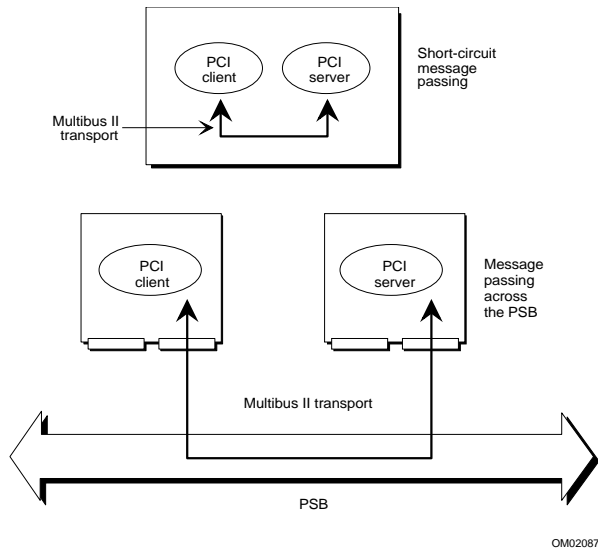
In a Multibus I system, the PCI server must use short-circuit message passing, as shown at the top of Figure 2-1. In short-circuit message passing, the device driver (client) and the PCI server are on the same board, but they communicate through the equivalent of MB II message space.

The iRMX III operating system is the only operating system that use short-circuit message passing. The UNIX operating system has a PCI client, but not a PCI server.

## PCI for PCs

As is the case for Multibus I systems, the PCI server must use short-circuit message passing. The device driver (client) and the PCI server are on the same board, but they communicate through the equivalent of MB II message space.

To summarize, in an MB I or PC system the PCI client and server are always on the same board. In an MB II system, they may be on the same board or on different boards. The iRMX operating system is the only RadiSys operating system that supports the short-circuit (same board) configuration.



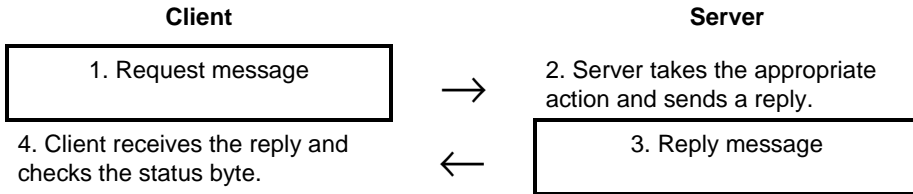
**Figure 2-1. Regular and Short-circuit Message Passing**

# Format of the PCI Data Structures

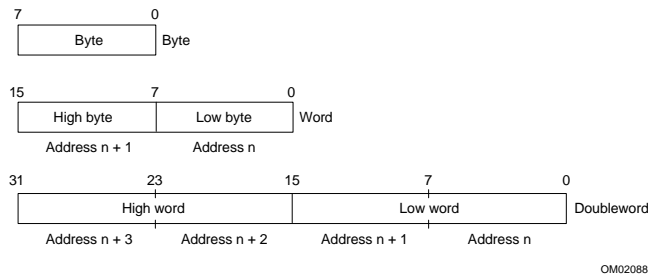
The client and server use two data structures to communicate with each other:

- Peripheral command message (PCM)
- Peripheral status message (PSM)

Clients initiate requests. A client sends a PCM to the server and the server returns a PSM. The PCM is called the request, and the PSM is called the reply.



In the PCM and PSM, all fields that are marked “Reserved” are reserved for future use and should be set to zero. All the fields that have more than one byte use the byte-ordering scheme shown in Figure 2-2.



**Figure 2-2. Byte Ordering**

## Peripheral Command Message (PCM)

The client sends a PCM to request a service from the PCI server. The PCM is a data structure embedded in the user data portion of a general interrupt, buffer request, or broadcast message.

- A general interrupt message is a form of unsolicited message.
- A buffer request message is classed in the transport protocol as an unsolicited message. Because it is used to set up a solicited message, this manual uses the term solicited message to mean a PCI request that is embedded in a buffer request.

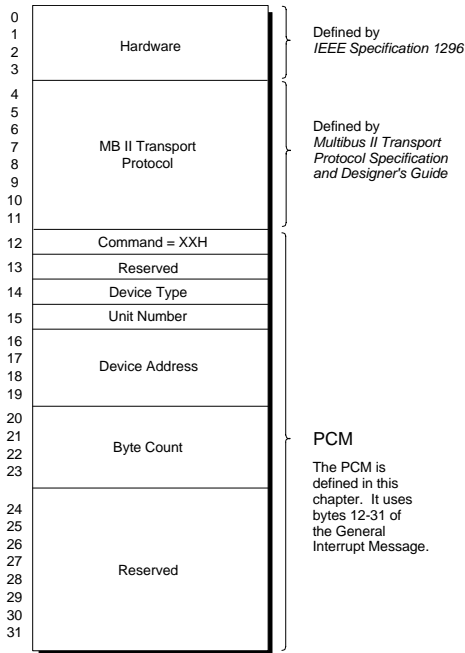
A solicited message includes data. The data portion depends on the kind of command, as shown in these examples:

<b>Command</b>	<b>Type of Data</b>
Format	Defect list
Reassign Blocks	Block list
Set Unit Characteristics	Parameters
Write Data	Data

- A broadcast message is an unsolicited message sent to all boards in a Multibus II system. The only PCI command that uses a broadcast message is the **Locate PCI Server** command.

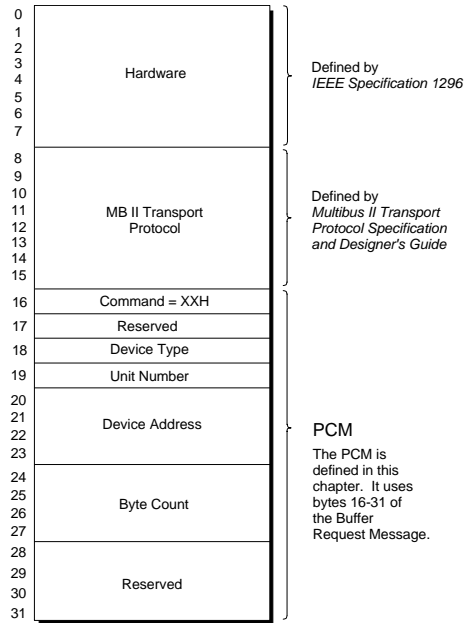
Figure 2-3 shows that a general interrupt message contains the PCM in bytes 12-31; Figure 2-4 shows that a buffer request message contains the PCM in bytes 16-31.

This chapter defines the fields of the PCM. The OS automatically handles the other message fields, which are defined by the IEEE 1296 and the transport protocol specifications.



OM02089

**Figure 2-3. The PCM in the General Interrupt Message**



OM02090

**Figure 2-4. The PCM in the Buffer Request Message**

The following list defines the generic PCM fields:

- Command**            The operation that the client requests.  
See also: Chapter 3 for command definitions
- Device Type**        The logical code for the type of device being accessed. These device types are currently defined as:

<b>Code</b>	<b>Device Type</b>
0	Reserved
1	Fixed media drive (hard disk)
2	Removable media drive (flexible disk)
3	Streaming cartridge tape
4	CD-ROM drive
5-7FH	Reserved
80H	Pass thru device type
81-FFH	Reserved

When you choose the pass thru device type, you have access to only those commands marked with a "PT" in Tables 3-1 and A-1.

- Unit Number**        The client chooses a specific device by placing the device's SCSI ID and LUN (logical unit number) in this field:

<b>Bits</b>	<b>Unit Number</b>
0-2	SCSI target ID (bit 0 is LSB)
3-5	SCSI LUN (bit 3 is LSB)
6-7	Reserved

The SCSI target ID is set by jumpers on the controller board and devices. Clients can calculate the Unit Number:

$$\text{Unit Number} = [(\text{LUN} \times 8) + \text{Target ID}]$$

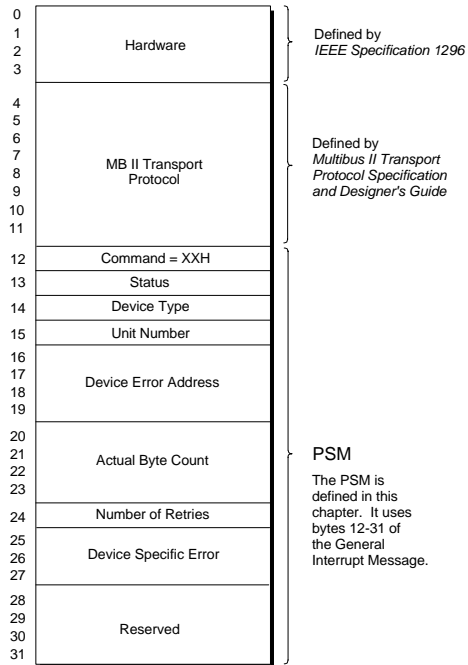
Except for the four target mode commands, the Unit Number is the SCSI target ID and the LUN of the remote device. For target mode commands (**Get Send Data, Flush Get Send Data, Get Target Mode Options, Set Target Mode Options**), the SCSI target ID is that of the PCI server's peripheral controller. The LUN is the logical unit that the client wants to contact. The sending and receiving clients both use the same unit number.

- Device Address**    Logical block number being addressed. Tape drives do not use this field.
- Byte Count**        Number of bytes to transfer.

## Peripheral Status Message (PSM)

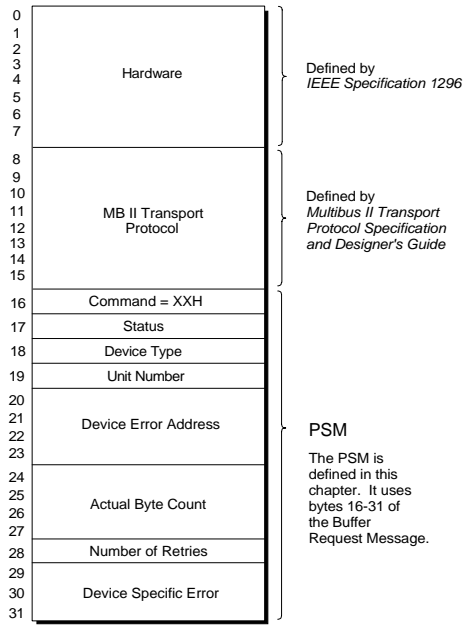
To give status about one of the client's previous requests, the server sends a peripheral status message (PSM) to a client. Figures 2-5 and 2-6 show the PSM embedded in the user data field of a general interrupt and a buffer request message.

This chapter defines the fields of the PSM. The operating system automatically handles the other message fields, which are defined by the IEEE 1296 and the transport protocol specifications.



OM02091

**Figure 2-5. The PSM in the General Interrupt Message**



OM02092

**Figure 2-6. The PSM in the Buffer Request Message**

The following list defines the generic PSM fields:

**Command**            The operation that the client requested.

**Status**              Completion status of the command:

<b>Bit</b>	<b>Value</b>
0	Command completed successfully
Not Zero	An error occurred

See also:      Appendix B lists the status codes; the Device Specific Error field can supply additional status information

If a command asked for a data transfer and the status code indicates an error, check the Actual Byte Count in the PSM against the requested Byte Count. If the actual count equals the requested count, the controller was able to recover from the error. If the actual count is not equal to the requested count, the controller was unable to recover from the error.

**Device Type**        Logical code for the type of device being accessed; the same as for the PCM.

**Unit Number**      Unit number of the peripheral that was accessed or where the error occurred; the same as for the PCM.

**Device Error Address**

If an error occurred, this field contains the logical block address of the error.

**Actual Byte Count**

Number of bytes that were transferred between the client and the server.

**Number of Retries**

Number of retries that the server did while trying to execute the command.

## Device Specific Error

Additional status information, if available. If the command completed successfully, all three bytes are set to zero.

In the event of a SCSI I/O error (including soft errors), the server returns these error bytes:

<b>Byte</b>	<b>Description</b>
DSE0 Byte 0	SCSI status returned by the device during the SCSI Status Phase
DSE1 Byte 1	4-bit sense key in byte 2 of the sense data information, which is obtained from the device using the SCSI <b>Request Sense</b> command
DSE2 Byte 2	Extended sense key from the sense data



This chapter describes the PCI server commands. It starts by listing all the commands in alphabetical order in Table 3-1. Then it describes each command in alphabetical order.

See also: Table A-1 in Appendix A lists all of the commands by number.

## List of PCI Server Commands

Table 3-1 uses these symbols:

Symbol	Meaning	Explanation
H	Hard disk drive	The command may be used with a hard disk drive.
F	Flexible disk drive	The command may be used with a flexible disk drive.
T	Tape drive	The command may be used with a cartridge tape.
PT	Pass Thru	The command must use the pass-thru device type.
I	Ignored	The command ignores the device type and unit number. They should be set to 0.
R	Required	The command must be preceded by two preliminary commands: <b>Set Unit Characteristics</b> and <b>Set Unit Options</b> .
Prohib	Prohibited	These target mode commands must not be preceded by the two commands, <b>Set Unit Characteristics</b> and <b>Set Unit Options</b> .
A	Allowed	The command may be used while a unit is reserved.

Blank spaces in the Table 3-1 columns have these meanings:

<b>Column</b>	<b>Meaning</b>
Hard Disk	Command cannot be used with the device.
Flex Disk	Command cannot be used with the device.
Tape	Command cannot be used with the device.
Pass Thru	Command cannot be used with the device.
Ignore Dev Type and Unit Number	Command uses the device type and unit number.
Requires Preliminary Commands	Command does not require preliminary commands.
Allow While Reserved	Command may not be used while a unit is reserved.

**Table 3-1. Alphabetical List of PCI Commands**

<b>Command</b>	<b>Hex Value</b>	<b>Hard Disk</b>	<b>Flex Disk</b>	<b>Tape</b>	<b>CD-ROM</b>	<b>Pass Thru</b>	<b>Ignore Dev Type &amp; Unit #</b>	<b>Req. Prelim Ccmds.</b>	<b>Allow While Rsvd</b>
Clear Reset Blocking Condition	93H						I		A
Erase Unit	29H			T				R	
Flush Get Send Data	9CH					PT		Prohib	
Format Unit	08H	H	F					R	
Get Device List	05H	H	F	T	C	PT	I*		A
Get SCSI ID	9BH						I		
Get Send Data	98H					PT		Prohib	
Get Server Information	18H						I		A
Get Statistics	81H	H	F	T	C	PT			
Get Target Mode Options	9AH					PT		Prohib	

continued

\* For the Get Device List command, the server ignores the unit number, but the device type must be valid.

**Table 3-1. Alphabetical List of PCI Commands (continued)**

<b>Command</b>	<b>Hex Value</b>	<b>Hard Disk</b>	<b>Flex Disk</b>	<b>Tape</b>	<b>CD-ROM</b>	<b>Pass Thru</b>	<b>Ignore Dev Type &amp; Unit #</b>	<b>Req. Prelim Cmds.</b>	<b>Allow While Rsvd</b>
Immediate Reset	94H					PT	I		A
I/O Sleep	83H	H	F	T	C	PT			A
Load	2CH			T	C			R	
Locate PCI Server	01H						I		
Query Unit Characteristics	17H	H	F	T	C				A
Query Unit Options	07H	H	F	T	C	PT			A
Read Data	11H	H	F	T	C			R	
Read Data Direct	12H	H						R	
Read Data and Verify	22H	H	F	T	C			R	
Read Defect List	0AH	H						R	
Read Device Capacity	0BH	H			C			R	
Reassign Blocks	09H	H						R	
Recalibrate	23H	H	F					R	
Release	04H	H	F	T	C	PT			A
Request Reset Notification	90H					PT	I		A
Request Unit Attention Notification	30H	H	F	T	C			R	
Reserve	03H	H	F	T	C	PT			
Reset Device	02H	H	F	T	C	PT			
Reset Notification **	91H						I	R	
Retension Tape	26H			T				R	
SCSI Pass Thru	80H	H	F	T	C	PT			

continued

**Table 3-1. Alphabetical List of PCI Commands (continued)**

<b>Command</b>	<b>Hex Value</b>	<b>Hard Disk</b>	<b>Flex Disk</b>	<b>Tape</b>	<b>CD-ROM</b>	<b>Pass Thru</b>	<b>Ignore Dev Type &amp; Unit #</b>	<b>Req. Prelim Cmds.</b>	<b>Allow While Rsvd</b>
Seek Beginning of Tape (BOT)	2AH			T				R	
Seek End of Data (EOD)	2BH			T				R	
Seek Filemark (Space Fwd FM)	27H			T				R	
Set Reset Blocking Time	92H						I		A
Set Statistics	82H	H	F	T	C				
Set Target Mode Options	99H					PT		Prohib	
Set Unit Characteristics	10H	H	F	T					
Set Unit Options	06H	H	F	T	C	PT			
Start Unit	0FH	H	F		C			R	
Stop Unit	0EH	H	F		C			R	
Synchronous Negotiation Control	84H	H	F	T	C	PT			
Test I/O	24H	H	F	T	C				
Unit Attention Notification **	2FH	H	F	T	C			R	
Unload	25H			T	C			R	
Write Data	14H	H	F	T				R	
Write Data Direct	15H	H						R	
Write Filemark	28H			T				R	

\*\* The server issues these commands: Reset Notification and Unit Attention Notification. The client issues all of the other commands.

## Command Types

As described in Chapter 1, the PCI commands are divided into two types:

<b>Command Type</b>	<b>Description</b>
Standard	Guaranteed to be compatible with future versions of PCI, and numbered less than 80H. PCI uses the SCSI Common Command Set to send these commands to the SCSI bus.
Implementation-specific	Added for unique implementations, and are numbered 80H and above. PCI uses most of these commands only on the peripheral controller board; it does not send them to the SCSI bus. However, PCI uses one of these commands, <b>SCSI Pass Thru</b> , to send vendor-unique commands directly to the SCSI device.

## Command Descriptions

The command descriptions are in alphabetical order, as in Table 3-1. Following a brief description of each command, each message format is described.

- Unsolicited means the PCM or PSM is embedded in a general interrupt message.
- Solicited means the PCM or PSM is embedded in a buffer request message.

If only one format is listed, Unsolicited PCM, for example, that means the PCM must be unsolicited. If both unsolicited and solicited messages are listed, that means the message can be either unsolicited or solicited, depending on the results of the operation.

⇒ **Note**

Each section describes only the extra or special information about the message fields. See Chapter 2 for the generic descriptions of the fields.

Bit-level subfields are listed from bit 0 (least significant bit) to bit 7 (most significant bit). The bit order is as follows:

Bit 7    Bit 6    Bit 5    Bit 4    Bit 3    Bit 2    Bit 1    Bit 0

For example, if bits 4-5 have the binary value 10, the byte order would be:

x x 1 0 x x x x

### Clear Reset Blocking Condition

Clears the blocking condition created by the server after a SCSI reset.

The **Set Reset Blocking Time** command sets the time period that the server waits before it accepts commands after a reset. This command, **Clear Reset Blocking Condition**, aborts the waiting period so the server can accept commands sooner.

### Unsolicited Peripheral Command Message

Byte	Description
0	93H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	93H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### Erase Unit

**Erase Unit** erases the tape from the beginning of the tape (BOT) to the end of the tape (EOT), and then rewinds the tape to the beginning.

This command cannot be used for hard or flexible disks.

### Unsolicited Peripheral Command Message

Byte	Description
0	29H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	29H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number Retries
13-15	Device-specific Error
16-19	Reserved

## Flush Get Send Data

Cancels all previous **Get Send Data** commands.

When the client sends this command to the server, the server returns a PSM for each **Get Send Data** command that it had received and not used. Each PSM contains an OK status and an actual length of zero. As the last step, the server returns the **Flush Get Send Data** PSM.

## Unsolicited Peripheral Command Message

Byte	Description
0	9CH
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

Where:

Device Type      Must be 80H.

## Unsolicited Peripheral Status Message

Byte	Description
0	9CH
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

Where:

Device Type      Must be 80H.

### Format Unit

Formats a hard or flexible disk.

This command prepares a hard or flexible disk for data recording. It writes the gaps, sector headers, and data fields on the device.

Based on information from a previous **Set Unit Characteristics** command for the unit, the **Format Unit** command reserves recording space and allocates alternate spaces.

This command offers a choice of options for mapping media defects. To assign alternate blocks, it can use the primary defect list, the grown defect list, the defect list sent with the command, or all three. If you enable the certification process, the drive updates the grown defect list with defects it finds during the certification process.

This command cannot be used for tape drives.

#### ⇒ **Notes**

The drive must support the defect list format that you choose.

Hard disk drives must support Pages 3 and 4 of the **SCSI Mode Sense** and **Mode Select** commands.

Flexible disk drives must support Page 5 of the **SCSI Mode Sense** and **Mode Select** commands.

### Unsolicited Peripheral Command Message

The client uses the unsolicited form of the PCM when the client does not provide a defect list to the server.

Byte	Description
0	08H
1	Reserved
2	Device Type
3	Unit Number
4	Flags
5-19	Reserved

Where:

Device Type	Value	Meaning
	1	Hard disk drive
	2	Flexible disk drive

Flags This table describes the flag bits:

Name	Bit	Value	Meaning
Map Bad Blocks	0-1	00	Use primary defect list to assign alternate blocks.
		01	Use grown defect list to assign alternate blocks.
		10	Use both primary and grown defect lists to assign alternate blocks.
		11	Don't use either list.
Certification Process	2	0	Disabled
		1	Enabled
Reserved	3	0	Reserved
Format of Defects	4-5	00	Bytes from index pulse
		01	Block format
		10	Physical sector
		11	Reserved
Stop Format	6	0	Finish format if defect list is unavailable.
		1	Stop format if defect list is unavailable.
Use Drive Defaults	7	0	Disabled
		1	Enabled

⇒ **Note**

If the certification process is enabled, defects found during the certification process are always added to the grown defect list.

### Solicited Peripheral Command Message

The client uses the solicited form of the **Format Unit** PCM when it gives a defect list to the controller board.

#### PCM Control Portion

Byte	Description
0	08H
1	Reserved
2	Device Type
3	Unit Number
4	Flags
5-7	Reserved
8-11	Defect Count
12-15	Reserved

Where:

Device Type	Value	Meaning
	1	Hard disk drive
	2	Flexible disk drive

Flags                    The flags are identical to the flags of the unsolicited PCM.

Defect Count          The number of defects in the data buffer, which contains the changes for the grown defect list.

#### ⇒ **Notes**

If the certification process is enabled, defects found during the certification process are always added to the grown defect list.

If the client sends a solicited message with a new defect list, the server uses the new defect list and adds it to the grown defect list. The server also uses other defect lists, as selected by the Map Bad Blocks flag.

**Solicited PCM Data Portion**

Bits 4 and 5 in the flags field show the format of the defect list, which is passed in the data portion of the response. The next three sections give the three formats for the data:

- Bytes from the index pulse
- Block format
- Physical sector

**PCM Data Portion (Bytes From the Index Pulse)**

<b>Byte</b>	<b>Description</b>
0-2	Defect 0 Cylinder
3	Defect 0 Head
4-7	Defect 0 Bytes from Index
...	...
...	Defect <i>n</i> Cylinder
...	Defect <i>n</i> Head
...	Defect <i>n</i> Bytes from Index

Where:

Cylinder            Cylinder number of the defect.

Head                Head number of the defect.

Bytes from Index

Number of bytes from the index pulse to the location of the defect.  
 A value of 0FFFF FFFFH means that the entire track should be mapped as defective.

### PCM Data Portion (Block Format)

Byte	Description
0-3	Defect 0 Logical Block Number
...	...
...	Defect $n$ Logical Block Number

Where:

Defect 0 Logical Block Number

Address of the block that contains defect number 0 (byte 0 is least significant).

Defect  $n$  Logical Block Number

Address of the block that contains defect number  $n$ .

### PCM Data Portion (Physical Sector Format)

Byte	Description
0-2	Defect 0 Cylinder
3	Defect 0 Head
4-7	Defect 0 Sector
...	...
...	Defect $n$ Cylinder
...	Defect $n$ Head
...	Defect $n$ Sector

Where:

Cylinder            Cylinder number of the defect.

Head                Head number of the defect.

Sector              Sector number of the defect.

## Unsolicited Peripheral Status Message

Byte	Description
0	08H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Get Device List

Finds out if a specific device type is connected to the controller.

This command asks the server to find out how many units of a certain device type are available to the server. The command checks for only one device type at a time. The server refers to the results of the most recent bus scan and sends the information to the client.

In the normal case, the server sends a solicited reply, so the client should be prepared to receive 256 bytes. However, if the server finds an error, it sends an unsolicited reply.

### Unsolicited Peripheral Command Message

Byte	Description
0	05H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

Where:

Device Type      Device type that the client wants to know about.

Unit Number      Always set to 0 to indicate the client is asking for information about all the units of this device type.

### Unsolicited Peripheral Status Message

Byte	Description
0	05H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

## Solicited Peripheral Status Message

### PSM Control Portion

Byte	Description
0	05H
1	Status
2	Device Type
3	Unit Number
4-15	Reserved

Where:

Unit number      Should be 0, which shows that the information in the data buffers is for all units of the specified device type.

### PSM Data Portion

The format of the **Get Device List** buffer is a 256-byte array of encoded bytes:

Byte	Description
0	PCI Unit 0 Encoded Byte
1	PCI Unit 1 Encoded Byte
2	PCI Unit 2 Encoded Byte
...	...
255	PCI Unit 255 Encoded Byte

Where:

Encoded Byte      Each byte has this format:

Bit	Value	Meaning
0	0	Unit not present
	1	Unit present
1-7	0	Reserved

Unit Number      A number computed from the SCSI LUN and the SCSI Target ID as follows:  
 $(\text{LUN} \times 8) + \text{SCSI target ID}$

## Get Device List

---

This list shows an example of a **Get Device List** buffer:

<b>Byte</b>	<b>Value</b>	<b>Description</b>
0 ...		PCI Unit 0 Encoded Byte
... ..		...
11 01H		PCI Unit 11 present
... ..		...
255 ...		PCI Unit 255 Encoded Byte

The byte at offset 11 shows that PCI Unit 11 is present.

## Get SCSI ID

Finds out the SCSI ID of the board running the PCI server.

If the client does not know the ID of its server, it can send this message. For the **Set Target Mode Options**, **Get Target Mode Options**, **Get Send Data**, and **Flush Get Send Data** commands, the client needs to use the server's SCSI target ID.

### Unsolicited Peripheral Command Message

Byte	Description
0	9BH
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

Where:

Device Type      Must be 0.

Unit Number      Must be 0.

### Unsolicited Peripheral Status Message

Byte	Description
0	9BH
1	Status
2	Device Type
3	Unit Number
4	SCSI ID
5-19	Reserved

Where:

Device Type      Must be 0.

Unit Number      Must be 0.

SCSI ID            SCSI target ID of the server's board.

### Get Send Data

Retrieves data sent with a **SCSI Send** command.

In target mode, a client uses this command to receive data sent from another SCSI initiator across the SCSI bus. The receiving client sends this command to its own server, generally before the server receives the data.

See also: For more information, see the section on target mode in Chapter 4.

### Unsolicited Peripheral Command Message

Byte	Description
0	98H
1	Reserved
2	Device Type
3	Unit Number
4-7	Length
8-19	Reserved

Where:

Device Type      Must be 80H.

Unit Number      The client can calculate the unit number as:  
(LUN × 8) + SCSI target ID

Length            Maximum amount of data the client is willing to receive.

To find out if the server has any data and, if so, how much, the client may send this command with the Length field set to 0. In this case, the server returns an unsolicited message.

## Unsolicited Peripheral Status Message

Byte	Description
0	98H
1	Status
2	Device Type
3	Unit Number
4-7	Remaining Length
8-11	Remaining SCSI Length
12	Flags
13	Remote Target ID
14-19	Reserved

Where:

Device Type      Must be 80H.

Remaining Length

Length of the data that is buffered on the PCI server and has not been sent to the client yet.

Remaining SCSI Length

Length of the data from the **SCSI Send** command which remains to be sent to the PCI server over the SCSI bus.

Flags

This field is used only if the EOM flag is set.

Bit	Value	Description
0		End of Message (EOM).
	0	Server has more data; client must send another <b>Get Send Data</b> command.
	1	All data has been sent.
1-7	0	Reserved

Remote Target ID

Initiator's SCSI ID.

### Solicited Peripheral Status Message

#### PSM Control Portion

Byte	Description
0	98H
1	Status
2	Device Type
3	Unit Number
4-7	Remaining Length
8-11	Remaining SCSI Length
12	Flags
13	Remote Target ID
14-15	Reserved

Where:

Device Type      Must be 80H.

Remaining Length

Length of the data that is buffered on the PCI server and has not been sent to the client yet.

Remaining SCSI Length

Length of the data from the **SCSI Send** command which remains to be sent to the PCI server over the SCSI bus.

Flags

This field is used only if the EOM flag is set.

Bit	Value	Description
0		End of Message (EOM).
	0	Server has more data; client must send another <b>Get Send Data</b> command.
	1	All data has been sent.
1-7	0	Reserved.

Remote Target ID

Initiator's SCSI ID.

#### PSM Data Portion

The buffer contains the data from the sending client.

## Get Server Information

Finds out the number of PCI servers and their port IDs on one peripheral controller.

See also: Chapter 4 for information on finding the PCI servers in the system.

### Unsolicited Peripheral Command Message

Byte	Description
0	18H
1	Reserved
2	Device Type
3	Unit Number
4	Command
5	Reserved
6-7	Argument 1
8-19	Reserved

Where:

Device Type      Must be 0.

Unit Number      Must be 0.

Command          **Value**    **Meaning**

00H            Asks the PCI server to return the total number of PCI servers running on the board.

01H            Asks the PCI server to return the port ID of the number (the instance) of server specified in the Argument 1 field, described next.

02-FFH      Reserved.

Argument 1      If the Command field was 01H, this field gives the number of the PCI server. The servers are numbered 0 to  $(N-1)$ , where  $N$  is the total number of servers running on the board. For example, if there are two servers on the board, you can get the port ID of the second one by putting the number 1 in this field.

### Unsolicited Peripheral Status Message

Byte	Description
0	18H
1	Status
2	Device Type
3	Unit Number
4	Command
5	Reserved
6-7	Result 1
8-19	Reserved

Where:

Device Type      Set to 0.

Unit Number      Set to 0.

Command          Same value used in the PCM.

Result 1          If the Command field is 00H, this field gives the total number of the PCI servers running on the board. If the command field is 01H, this field gives the port ID of the specified PCI server.

## Get Statistics

Retrieves tallies kept by the PCI server.

The client sends this message to read the statistical counts maintained by the PCI server. The peripheral controller keeps some of these statistics for each peripheral unit and some globally for all units. The statistics counters do not overflow when the count exceeds the maximum. Thus, if the maximum value is returned, it implies a count value of at least the maximum.

Normally the server sends a solicited reply, but if an error occurs, it sends an unsolicited reply.

### Unsolicited Peripheral Command Message

Byte	Description
0	81H
1	Reserved
2	Device Type
3	Unit Number
4	Flags
5-19	Reserved

Where:

Device Type      Must refer to a valid peripheral unit.

Unit Number      Must refer to a valid peripheral unit.

Flags	Bit	Value	Meaning
	0	0	The statistics counters must not be changed by this command.
		1	All statistics counters are reset to 0.
	1-15	0	Reserved (value must be 0).

### Unsolicited Solicited Peripheral Status Message

Byte	Description
0	81H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### Solicited Peripheral Status Message

#### PSM Control Portion

Byte	Description
0	81H
1	Status
2	Device Type
3	Unit Number
4-15	Reserved

## PSM Data Portion

The data portion of the response includes global statistics, followed by unit statistics.



### Note

The client should reserve a buffer of at least 256 bytes for this data.

Byte	Description
0-3	Re-transmit Count
4-7	Transport Errors
8-11	Reserved
12-15	Reserved
16-19	Server Version Number
20-21	Number of Buffers
22-23	Buffer Size
24-27	Quick Read Hits
28-31	Reserved
32-35	Read Hits
36-39	Read Misses
40-43	Writes
44-47	Disk Errors
48-51	Physical Blocks Read
52-55	Read I/Os
56-59	Physical Blocks Written
60-63	Write I/Os

Where:

### Re-transmit Count

Number of retransmissions made by the PCI server due to NACKs. The counter is global across all attached peripheral units.

### Transport Errors

Number of message passing errors found. These are errors that were not recoverable; the server could not recover from transmission errors after 16 consecutive re-transmissions. The counter is global across all attached peripheral units.

Server Version Number	A four-byte ASCII string containing the version number of PCI server.
Number of Buffers	Number of data buffers allocated for the specified unit.
Buffer Size	Size in bytes of each data buffer for the specified unit. It is the same as the cache line size for the unit.
Quick Read Hits	Number of blocks that the server read from cache and returned to the client by an optimized control path.
Read Hits	Number of blocks that the server read from cache and returned to the client by a normal cache access path. (A read hit requires one more iRMX task switch than a quick read hit.) The sum of read hits and quick read hits gives the total number of blocks that were read from cache.
Read Misses	Number of disk blocks read that required access to a SCSI peripheral because they were not in the cache: $\text{cache hit ratio} = (\text{read hits} + \text{quick read hits}) / (\text{read hits} + \text{quick read hits} + \text{read misses})$ .
Writes	Number of blocks that the client wrote.
Disk Errors	Total number of SCSI I/O errors, including soft errors from which the server recovered .
Physical Blocks Read	Total number of blocks read from the unit.
Read I/Os	Total number of physical read I/Os issued to the unit: $\text{average read I/O size} = (\text{physical blocks read}) / (\text{read I/Os})$ .
Physical Blocks Written	Total number of blocks written to the unit.
Write I/Os	Total number of physical write I/Os issued to the unit: $\text{average write I/O size} = (\text{physical blocks written}) / (\text{write I/Os})$ .

## Get Target Mode Options

Finds out about control and data buffer settings.

With this command, the client reads the server's current target mode settings.

### Unsolicited Peripheral Command Message

Byte	Description
0	9AH
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

Where:

Device Type      Must be 80H.

Unit Number      Contains the SCSI target ID of the PCI server's peripheral controller and the LUN that the client wants to use.

The client can calculate the Unit Number as:

$$[(\text{LUN} \times 8) + \text{SCSI target ID}]$$

### Unsolicited Peripheral Status Message

Byte	Description
0	Function = 9AH
1	Status
2	Device Type
3	Unit Number
4-5	Number of Control Buffers
6-7	Number of Data Buffers
8-9	Data Buffer Size
10-11	Max Ctrl Bufs Per Init
12-13	Max Pending Get Send Data Cmds
14-19	Reserved

## Get Target Mode Options

---

Where:

Device Type      Must be 80H.

Number of Control Buffers  
Number of control buffers allocated by the PCI server for this target mode logical unit.

Number of Data Buffers  
Number of data buffers allocated by the PCI server for this target mode logical unit.

Data Buffer Size  
Size of each data buffer allocated by the PCI server for this target mode logical unit.

Max Ctrl Bufs Per Init  
Maximum number of control buffers the PCI Server allocates for any single initiator for this logical unit.

Max Pending Get Send Data Cmds  
Maximum number of **Get Send Data** commands pending for any logical unit.

## Immediate Reset

Requests an immediate reset on the SCSI bus.

With this command, the client asks the server to reset the SCSI bus immediately. The server does the reset and takes the same actions as when it detects a reset initiated by another device on the SCSI bus.

See also: Chapter 1 describes the SCSI reset actions.



### CAUTION

If the server does act on this command, it violates all of its internal mutual exclusion and synchronization, so it might crash. Use this command only under extreme circumstances or when the PCI server is quiescent.

## Unsolicited Peripheral Command Message

Byte	Description
0	94H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

## Unsolicited Peripheral Status Message

Byte	Description
0	94H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### I/O Sleep

Suspends operations on a unit.

This command causes the PCI server to suspend operations on a specified unit for a specified period of time. During the suspension, test programs can check if commands are queued and pending at the controller.

### Unsolicited Peripheral Command Message

Byte	Description
0	83H
1	Reserved
2	Device Type
3	Unit Number
4-5	Sleep Time
6-19	Reserved

Where:

Sleep Time      Time, in units of 10 ms, for the suspension of commands to the unit; a value of 0FFFFH is illegal. For example, 0064H = 1 second.

### Unsolicited Peripheral Status Message

Byte	Description
0	83H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

## Load

Rewinds the tape to the beginning.

### Unsolicited Peripheral Command Message

Byte	Description
0	2CH
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	2CH
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number Retries
13-15	Device-specific Error
16-19	Reserved

### Locate PCI Server

Finds PCI servers on the parallel system bus.

This command finds out which slots in the Multibus II backplane contain PCI servers. It can be either a broadcast message or a general interrupt non-RSVP message:

- When sending a broadcast message, the client must be prepared to receive multiple responses. It must have enough control buffers to receive responses from all the servers. The command is embedded in bytes 12 through 31 of the broadcast message.
- The general interrupt non-RSVP message can be used to verify that the PCI server is at the location where the client expects it.

See also: Chapter 4 for information on how to find a PCI server.

⇒ **Note**

This is the only command that can be used in a broadcast message; it must be non-RSVP if used in a general interrupt message.

### Unsolicited Peripheral Command Message

Byte	Description
0	01H
1-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	01H
1	Status
2-19	Reserved

⇒ **Note**

The PSM is a non-RSVP message.

## Query Unit Characteristics

Reads the physical characteristics of the specified drive.

This command asks the server for information about the physical characteristics of the specified device.

If the server finds an error, it returns an unsolicited PSM.

### Unsolicited Peripheral Command Message

Byte	Description
0	17H
1	Reserved
2	Device Type
3	Unit Number
4	Flags
5-19	Reserved

Where:

Flags                      This table describes the flag bits:

Name	Bit	Value	Meaning
Source of Info	0	0	Return info set by client.
		1	Return info from device (hard disk device only).
Which Info	1-2	00	Return current device values.
		01	Return default device value.
		10	Return changeable device values.
		11	Reserved.
Reserved	3-7	0	Reserved.

### Unsolicited Peripheral Status Message

Byte	Description
0	17H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### Solicited Peripheral Status Message

#### PSM Control Portion

Byte	Description
0	17H
1	Status
2	Device Type
3	Unit Number
4-15	Reserved

#### PSM Data Portion

The information received in the data portion of the message depends on the device type. The next two sections describe two different data buffers:

- Direct access devices (hard and flexible disk drives)
- Tapes

### PSM Data Portion (Direct Access Devices)

The server uses the following data buffer for direct access devices (Device Type =1 or 2):

<b>Byte</b>	<b>Description</b>
0-3	Number of Cylinders
4-5	Number of Surfaces
6-7	Sectors/Track
8-9	Bytes/Sector
10-11	Flexible Disk Format
12-15	RWC Cylinder
16-19	WPC Cylinder
20-21	Step Rate
22-23	Alternate Sectors/Zone
24-25	Alternate Tracks/Zone
26-27	Alternate Tracks/Volume
28-31	Tracks/Zone
32-33	Track Skew
34-35	Cylinder Skew
36-37	Interleave
38	Pin 2
39	Pin 34
40-255	Reserved

Where:

Number of Cylinders

The total number of cylinders on the device, including alternates.

Number of Surfaces

The number of data recording surfaces on the device.

Sectors/Track

The total number of sectors per track, including alternates.

Bytes/Sector

The number of data bytes per sector.

## Query Unit Characteristics

---

### Flexible Disk Format

This field contains two bytes (bits 15-0). This table defines the low byte (bits 7-0).

<b>Name</b>	<b>Bit</b>	<b>Value</b>	<b>Meaning</b>
Transfer Rate <sup>1</sup>	0-1	00	125 Kbits/second
		01	250 Kbits/second
		10	300 Kbits/second
		11	500 Kbits/second
Step Pulses <sup>2</sup>	2-3	00	1 step pulse per cylinder (normal usage)
		01	2 step pulses per cylinder
		10	3 step pulses per cylinder
		11	4 step pulses per cylinder
True Ready	4	0	Server must not rely on Ready signal from drive to assume drive readiness.
		1	Drive provides a true Ready signal which the server uses to sense drive readiness.
Reserved	5-7	0	Reserved.

#### Notes:

- 1 The transfer rate depends on the recording density and encoding method.
- 2 Non-zero values for step pulses can be used to access a flexible disk formatted on a drive with a lower tpi (tracks per inch) value. For example, a value of 01 allows a 96 tpi drive to access a flexible disk formatted on a 48 tpi drive.

This table defines the high byte (bits 8-15) of the Flexible Disk Format field, which represent the media type. (Bit 8 is the least significant.)

Value	Disk Diameter	Bit Density (bits/radian)	Track Density	Number of Sides
00H	Reserved			
01H	8"	6631	48 tpi	1
02H	8"	6631	48 tpi	2
03H	8"	13262	48 tpi	1
04H	8"	13262	48 tpi	2
05H	5.25"	3979	48 tpi	1
06H	5.25"	7958	48 tpi	2
07H	5.25"	7958	96 tpi	2
08H	5.25"	13262	96 tpi	2
09H	3.5"	7958	135 tpi	2
0AH	Uses Currently Mounted Media			
0BH–0FFH	Reserved			

RWC Cylinder Starting cylinder number to use for reduced write current

WPC Cylinder Starting cylinder number to use for write precompensation.

Step Rate Drive step rate expressed in units of 0.1 microseconds.

Tracks/Zone Size of defect handling zones created on the device when it is formatted. All zones on the device are the same size, except that the last zone may be different. If this field is zero, a single zone takes up the whole device.

Alternate Sector/Zone

Number of sectors per zone reserved during formatting for replacing defective sectors in that zone.

Alternate Tracks/Zone

Number of tracks per zone reserved during formatting for replacing defective tracks in that zone.

Alternate Tracks/Volume

Number of tracks reserved during formatting for replacing defective tracks on the device. This parameter is only significant if zone-based defect handling is not supported.

## Query Unit Characteristics

---

Track Skew	Number of sectors between the last data sector of a track and the first data sector on the next sequential track of the same cylinder.						
Cylinder Skew	Number of sectors between the last data sector of a cylinder and the first data sector on the next sequential cylinder.						
Interleave	Interleave factor for formatting the tracks of the device: <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>0</td><td>The default interleave factor.</td></tr><tr><td>1</td><td>Consecutive data sectors are placed in consecutive physical order.</td></tr></tbody></table>	Value	Meaning	0	The default interleave factor.	1	Consecutive data sectors are placed in consecutive physical order.
Value	Meaning						
0	The default interleave factor.						
1	Consecutive data sectors are placed in consecutive physical order.						

Pin 2 Defines the use of pin 2 of the flexible disk drive interface. Different models of drives use this pin differently. The controller reads this byte to find out how to use pin 2.

Bit	Value	Meaning
0-2		Function
	000	Open
	001	Ready
	010	Disk changed
	011	Speed
	100	(Not used)
	101	Density
	110	(Not used)
3		Polarity
	0	Active-low
	1	Active-high
4-6	0	Reserved
7		Indicates if the field is valid or not.
	0	Use the default (compatible with previous versions of the PCI server).
	1	Use the other bits.

This table summarizes the value and meaning for all the valid bit combinations:

Value	Meaning
00H	Use the default (speed/active low)
80H	Open/active low
88H	Open/active high
81H	Ready/active low
89H	Ready/active high
82H	Disk changed/active low
8AH	Disk changed/active high
83H	Speed/active low
8BH	Speed/active high
85H	Density/active low
8CH	Density/active high

## Query Unit Characteristics

---

Pin 34

Defines the use of pin 34 of the flexible disk drive interface. Different models of drives use this pin differently.

Bit 7 indicates if the field is valid or not.

Bit	Value	Meaning
0-2		Function
	000	Open
	001	Ready
	010	Disk changed
	011	(not used)
	111	
3		Polarity
	0	Active-low
	1	Active-high
4-6	0	Reserved
7		Indicates if the field is valid or not.
	0	Use the default (compatible with previous versions of the PCI server).
	1	Use the other bits.

This table summarizes the value and meaning for all the valid bit combinations:

Value	Meaning
00H	Use the default (speed/active low)
80H	Open/active low
88H	Open/active high
81H	Ready/active low
89H	Ready/active high
82H	Disk changed/active low
8AH	Disk changed/active high

### PSM Data (Cartridge Tape Devices)

The server uses the following data buffer for tapes (Device Type = 3):

Byte	Description
0-1	Bytes/block
2-255	Reserved

Where:

Bytes/block      Number of data bytes per tape block.

### Query Unit Options

Returns information about the mode of a specific unit.

This command returns information about the mode, the requested number of retries, the command ordering, the buffer size, and the amount of read-ahead caching for a unit.

### Unsolicited Peripheral Command Message

Byte	Description
0	07H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	07H
1	Status
2	Device Type
3	Unit Number
4	Reserved
5	Mode
6	Reserved
7	Number of Retries
8	Command Ordering
9	Buffer Size
10	Read Ahead
11-19	Reserved

Where:

Mode

This table describes the mode bits:

Mode	Bit	Value	Meaning
Address Mode	0	0	Logical addressing
Cache Mode	1	0	Disabled
		1	Enabled
Command Ordering Mode	2	0	Re-ordering disabled
		1	Re-ordering enabled
Reserved	3-7	0	Reserved

Number of Retries

Maximum number of times the server should retry a failed operation before reporting a hard error.

Command Ordering

This table describes the Command Ordering bits:

Name	Bit	Value	Meaning
Read/write Relative Priority <sup>1</sup>	0-1	00	Equal priority
		01	Reads before writes
		10	Writes before reads
Seek Ordering Algorithm <sup>1</sup>	2-5	11	Reserved
		00H	First-come-first-served
		01H	Shortest-seek-time-first
		02H	Elevator
		03H	CSCAN
Reserved	6-7	04H-0FH	Reserved
		0	Reserved

Note:

<sup>1</sup> Only valid if command re-ordering is enabled in the mode field.

Buffer Size

Value	Meaning
00H	Default minimum.
1-64H	Percentage of total cache memory allocated to this device.
65-FEH	Reserved.

## Query Unit Options

---

Read Ahead      This field is only valid if cache is enabled.

<b>Value</b>	<b>Meaning</b>
--------------	----------------

00H	Invalid value.
-----	----------------

1-FEH	Minimum number of sectors read by server in any read operation.
-------	---

0FFH	Controller selects the default size.
------	--------------------------------------

## Read Data

Read Data from mass storage.

This command transfers a block of data from a device to the client's memory. If the server did not read any data successfully, it sends an unsolicited PSM. If the server read at least some data successfully, it sends a solicited message.

If the reply is fragmented and an error is found after the server has sent some data to the client, the server will stop the reply phase of the transactions prematurely, as allowed by the Multibus II transport protocol.

### Unsolicited Peripheral Command Message

Byte	Description
0	11H
1	Reserved
2	Device Type
3	Unit Number
4-7	Device Address
8-11	Byte Count
12-19	Reserved

Where:

Device Address    This field is not specified for a tape.

Byte Count        Number of bytes to transfer from the requested device to the client.  
This must be a multiple of the block size of the device.

## Read Data

---

### Unsolicited Peripheral Status Message

Byte	Description
0	11H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Byte Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Solicited Peripheral Status Message

#### PSM Control Portion

Byte	Description
0	11H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Byte Count
12	Number of Retries
13-15	Device-specific Error

#### PSM Data Portion

This buffer contains the data read from the device.

## Read Data Direct

Retrieves data from mass storage into a buffer.

Used when the client and server are on the same board, this command causes the server to read directly from the device into a buffer defined by the client on the board.

The PCM is embedded in an RSVP unsolicited message.

### Unsolicited Peripheral Command Message

Byte	Description
0	12H
1	Reserved
2	Device Type
3	Unit Number
4-7	Device Address
8-11	Byte Count
12-15	Physical Address
16-19	Reserved

Where:

Device Type	Must be 01H (hard disk drive).
Device Address	This address is either the starting cylinder, head, and sector or the logical block number.
Byte Count	Number of bytes to be transferred from the requested device to the host. This value must be a multiple of device block size.
Physical Address	Physical address of the buffer that will receive the data (the address must be even).

### Unsolicited Peripheral Status Message

Byte	Description
0	12H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Byte Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Read Data and Verify

Determines whether data can be read error-free.

This command causes the server to try to read directly from the device, rather than from the cache. It does not transfer data to the client or do any compare operations on the data fields.

### Unsolicited Peripheral Command Message

Byte	Description
0	22H
1	Reserved
2	Device Type
3	Unit Number
4-7	Device Address
8-11	Byte Count
12-19	Reserved

Where:

Byte Count      This value must be a multiple of device block size.

### Unsolicited Peripheral Status Message

Byte	Description
0	22H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Byte Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Read Defect List

Reads the defect lists from a hard disk into a buffer.

The command is used in two ways:

- To get the length of the defect list, the client issues a **Read Defect List** command in the form that expects an unsolicited response.
- To get the actual defect list, the client then issues a **Read Defect List** command in the form that expects a solicited response. The server returns either the entire defect list or the maximum that can fit in the reply buffer posted by the client.



#### Notes

This command cannot be used for flexible disk drives or cartridge tapes.

The drive must support the format of the defect list.

### Unsolicited Peripheral Command Message

Byte	Description
0	0AH
1	Reserved
2	Device Type
3	Unit Number
4	Flags
5-7	Reserved
8-11	Defect Count
12-19	Reserved

Where:

Flags                      This table describes the flag bits:

Name	Bit	Value	Meaning
Value Read	0-1	00	Primary defect list read. <sup>1</sup>
		01	Grown defect list read. <sup>1</sup>
		10	Length of primary defect list read. <sup>2</sup>
		11	Length of grown defect list read. <sup>2</sup>
Format of Defects In Data Buffer		00	Bytes from index pulse format
		01	Block format
		10	Physical sector format
		11	Reserved
Reserved	4-7	0	Reserved

Notes:

<sup>1</sup> The reply will be solicited if the request message uses 00 or 01.

<sup>2</sup> The reply will be unsolicited if the request message uses 10 or 11.

**Defect Count**            Number of defects the receiving client can fit in the buffer for the data response. If this field is greater than zero, the value read field in the flags parameter must be 00 or 01. If the value read field is not 00 or 01, this field is ignored.

### Unsolicited Peripheral Status Message

Byte	Description
0	0AH
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Defect Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Solicited Peripheral Status Message

#### PSM Command Portion

Byte	Description
0	0AH
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Defect Count
12	Number of Retries
13-15	Device-specific Error

#### PSM Data Portion

This data buffer contains either the primary defect list or the grown defect list that was read from the disk. The PCM chooses one of these formats:

- Bytes from index pulse
- Block format
- Physical sector

#### PSM Data Portion (Bytes from Index Pulse Format)

Byte	Description
0-2	Defect 0 Cylinder
3	Defect 0 Head
4-7	Defect 0 Bytes from Index
8-10	Defect 1 Cylinder
11	Defect 1 Head
12-15	Defect 1 Bytes from Index

Where:

Cylinder            Cylinder number of the defect.

Head                Head number of the defect.

Bytes from index

Bytes from index pulse where the defect resides. A value of 0FFFF FFFFH indicates that the entire track should be reassigned.

**PSM Data Portion (Block Format)**

Byte	Description
0-3	Defect 0 Logical Block Number
...	...
...	Defect $n$ Logical Block Number

Where:

Logical Block Number

Address of the block that contains the defect. The defect descriptors should in ascending order (0, 1, 2, ...,  $n$ ).

**PSM Data Portion (Physical Sector Format)**

Byte	Description
0-2	Defect 0 Cylinder
3	Defect 0 Head
4-7	Defect 0 Sector
8-10	Defect 1 Cylinder
11	Defect 1 Head
12-15	Defect 1 Sector
...	...

Where:

Cylinder            Cylinder number of the defect.

Head                Head number of the defect.

Sector              Sector number on the cylinder of the defect.

### Read Device Capacity

Asks for the total block count of a drive.

This command asks the server for the total block count of the device.

### Unsolicited Peripheral Command Message

Byte	Description
0	OBH
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	OBH
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Total Block Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

Where:

Total Block Count

Capacity of the device, expressed in number of blocks;  
0 = the server could not find out the size.

## Reassign Blocks

Reassigns defective logical blocks to a reserved area on the logical hard disk.

This command asks the server to do two things:

- Reassign defective logical blocks to an area on the logical device reserved for this purpose.
- Add these blocks to the grown defect list on the disk.

The format of the defect list is the logical block format.

This command does not apply to a flexible disk or cartridge tape.

## Solicited Peripheral Command Message

### PCM Control Portion

Byte	Description
0	09H
1	Reserved
2	Device Type
3	Unit Number
4-7	Reserved
8-11	Defect Count
12-15	Reserved

Where:

Defect Count      Number of defects in the buffer that need to be added to grown defect list.

### PCM Data Portion

The data portion lists the defects to add to the grown defect list. The format is that of the logical block numbers.

Byte	Description
0-3	Defect 0 Logical Block Number
...	Defect 1 Logical Block Number
...	...
...	Defect <i>n</i> Logical Block Number

Where:

Logical Block Number

Address of the block that contains the defect. The defect descriptors should be in ascending order (0, 1, 2, ..., *n*).

### Unsolicited Peripheral Status Message

Byte	Description
0	09H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Defect Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

Where:

Actual Defect Count

Number of defects actually mapped out.

## Recalibrate

Positions the read/write heads of the hard or flexible disk at track 0.

This command does not apply to tapes.

### Unsolicited Peripheral Command Message

Byte	Description
0	23H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	23H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Release

Returns a reserved device to general use.

This command releases the device from a previous reserve operation (see the **Reserve** command). If a non-reserved device receives this command, it does not return an error.

The client that reserved the device is normally the only one that can release it. However, any client may release a device by setting the force bit in the flags field of the PCM. Once the device is released, all clients are able to access the device until it is reserved again.

### Unsolicited Peripheral Command Message

Byte	Description
0	04H
1	Reserved
2	Device Type
3	Unit Number
4	Flags
5-19	Reserved

Where:

Flags                      This table describes the flag bits:

Name	Bit	Value	Meaning
Force Mode	0	0	Disabled
		1	Enabled
Reserved	1-7	0	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	04H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### Request Reset Notification

Asks to be notified of a SCSI bus reset.

This command asks the server to send a message to the client when the server detects a reset on the SCSI bus. The server immediately replies that it received the request. Later, when the server detects a reset, it sends a non-RSVP message to the client. Multiple clients may issue this request for the same device.

Clients can also use this command to cancel a previous **Request Reset Notification** command.

### Unsolicited Peripheral Command Message

Byte	Description
0	90H
1	Reserved
2	Device Type
3	Unit Number
4-5	Port ID
6-19	Reserved

Where:

**Port ID** Port ID of the client. When the server detects a reset on the SCSI bus, it should send a **Reset Notification** to this port ID.

To cancel a previous request, the client sets this field to zero.

### Unsolicited Peripheral Status Message

Byte	Description
0	90H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### Request Unit Attention Notification

Asks to be notified of a SCSI bus reset.

This command asks the server to send a message to the client when a device needs attention, such as for a media change or a device ready state change. The server immediately replies that it received a request. Later, when a device requires attention, the server sends a non-RSVP message to the client. See the description of **Unit Attention Notification** for more information.

Multiple clients can issue this request for the same device. Clients can also use this command to cancel a previous **Request Unit Attention Notification** command.

### Unsolicited Peripheral Command Message

Byte	Description
0	30H
1	Reserved
2	Device Type
3	Unit Number
4-5	Port ID
6-19	Reserved

Where:

Port ID                      To request the service, the client sends its port ID in this field. To cancel a previous request, the client sets this field to zero.

Value	Meaning
Client's port ID	Request to be notified.
0	Cancel previous request.

### Unsolicited Peripheral Status Message

Byte	Description
0	30H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

## Reserve

Gains exclusive use of a unit.

A client uses this command to reserve a unit for the client's exclusive use. After the unit is reserved, the server accepts commands for the unit only from the reserving client. It is not possible to reserve a portion of a unit; the command reserves an entire unit.

The reservation can apply to one or all of the client's Multibus II transport ports.

To cancel a reservation, the client that reserved the unit must issue a **Release** command. Other clients can release the unit by sending the **Release** command with the flag field's force bit set.

The server executes all commands that precede the **Reserve** command in the queue. The **Reserve** command affects only those commands issued after it.

Any client that did not reserve the unit is limited to the commands that are marked "A" in Tables 3-1 and A-1.

## Unsolicited Peripheral Command Message

Byte	Description
0	03H
1	Reserved
2	Device Type
3	Unit Number
4-5	Port ID
19	Reserved

Where:

Unit Number      Note that the entire unit will be reserved.

Port ID	Value	Meaning
	0	All of the client's ports have access to the unit.
	1-0FFFFH	Only this port may have access.

### Unsolicited Peripheral Status Message

Byte	Description
0	03H
1	Status
2	Device Type
3	Unit Number
4-5	Port ID
6-19	Reserved

## Reset Device

Resets the unit.



### CAUTION

When the server receives this command for a SCSI device, it resets the whole SCSI bus.

## Unsolicited Peripheral Command Message

Byte	Description
0	02H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

## Unsolicited Peripheral Status Message

Byte	Description
0	02H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### Reset Notification

Tells clients about SCSI reset.

To all clients that requested this service, the server sends a notice when it detects that the SCSI bus has been reset.

Clients can request this service by sending a **Request Reset Notification** to the server. Once a client receives a **Reset Notification**, it must send another **Request Reset Notification** to the server to continue the service.

This message is actually the second reply to a client that requested this service. The first reply simply acknowledged that the server received the request. This reply is the actual notice.

### Unsolicited Peripheral Status Message

Byte	Description
0	91H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### Retension Tape

Rewinds the tape to the beginning, fast forwards to the end, and rewinds to the beginning.

This command prepares the tape for subsequent operations. It rewinds the tape to the beginning-of-tape marker, then fast forwards to the end-of-tape marker (EOT), and finally rewinds back to the BOT marker. This command re-stacks the tape in the cartridge, thus ensuring unobstructed tape movement.

This command does not apply to hard or flexible disk drives.

### Unsolicited Peripheral Command Message

Byte	Description
0	26H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

Where:

Device Type      The only valid device type is 3 (tape drive).

### Unsolicited Peripheral Status Message

Byte	Description
0	26H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### SCSI Pass Thru

Sends a SCSI command directly to the device.

Using this implementation-specific command, a client can send SCSI commands directly to a SCSI device. A client can issue a **SCSI Pass Thru** command at any time, except when the unit is reserved by a different client.

After receiving a **SCSI Pass Thru** command, the PCI server does the following:

1. Selects the logical unit number (LUN) of the target. (The LUN is part of the PCI unit number.)
2. During the SCSI Command phase, transfers the command bytes specified by the client.
3. If the SCSI command has a SCSI Data Out phase, sends out the data bytes to the target. (The client specifies the data bytes in the data portion of the solicited request message.) The data sent must have an even byte count.

Otherwise, if the SCSI command has a SCSI Data In phase, reads in the data bytes from the target during the SCSI Data In phase.

4. Waits until the SCSI command finishes and then fills in the status and device-specific status bytes in the PCI reply message.
5. Sends the PCI reply message to the client.



#### CAUTIONS

The PCI server does not interpret the **SCSI Pass Thru** command and data bytes. The client is responsible for sending the correct information.

Before using the **PCI Read Data** command to read data that was written with the **SCSI Pass Thru** command, the client must disable the cache.

If the **SCSI Pass Thru** command changes the disk geometry parameters, the client must inform the PCI server of these changes before accessing the disk with standard PCI commands. To inform the server, the client sends the **PCI Set Unit Characteristics** command with the Force Mode enabled.

The data portion must have an even number of bytes.

## ⇒ Notes

The SCSI commands which have Data In and Data Out phases at the same time are not supported by the **SCSI Pass Thru** command.

The device type in the **SCSI Pass Thru** command must match the type of device being accessed (for example, 1 for hard disk, 2 for flexible disk, 3 for tape). Or, it can be 80H, which represents any device type.

The **Set Unit Options** command may be used for all device types (even **SCSI Pass Thru**) to adjust the amount of buffer space available. This is important because the transfer length cannot exceed the buffering space.

There are two methods of sending a **SCSI Pass Thru**, determined by the setting of bit 5 (the CDB\_IN\_DATA flag) of the Peripheral Command Message (PCM). Method 1 supports SCSI commands up to 12 bytes in length. Method 2 is limited to commands which are 10 bytes or less in length, and is available for compatibility with existing code.

## Method 1

Set bit 5 of the PCM to 1. The PCM must always be solicited, since the SCSI Command Block (CDB) is contained in the data part of the message.

## Solicited Peripheral Control Message for Method 1

### PCM Control Portion

Byte	Description
0	80H
1	Flags
2	Device Type
3	Unit Number
4-15	Reserved

Where:

Flags

This table describes the flag bits:

Name	Bit	Value	Meaning
Reserved	0	0	Must be set to 0.
Disconnect <sup>1</sup>	1	0	PCI server disables the SCSI disconnect feature during the command.
		1	PCI server lets the SCSI target disconnect during the command.
Extended Cmnd	2	0	Ignored.
Request Sense	3	0	PCI server issues a SCSI <b>Request Sense Disable</b> command if the passed-thru SCSI command returns a SCSI Check Condition status.
		1	PCI server does not issue the <b>Request Sense</b> command, and the client has to issue the <b>Request Sense</b> command with a subsequent <b>SCSI Pass Thru</b> request.
Extended Data	4		Ignored.
CDB In Data	5	0	Must be set to 1.
Reserved	6-7	0	Must be set to 0.

Notes:

- <sup>1</sup> Generally, the Disconnect Flag (bit 1) is set to 1. When this flag is cleared (0), data transfer is slower and data transfer length is restricted.

## PCM Data Portion

The data part of the PCM message has a header with the following structure:

Byte	Description
0-1	Reserved
2-3	SCSI Command Length (6, 10, or 12 bytes)
4-7	Data In Length
8-15	Reserved
16-127	SCSI COMMAND (maximum of 12 bytes)
128-n	Data Out (optional)

Where:

### SCSI Command Length

The length of the SCSI COMMAND that you want to send out onto the SCSI bus — should be either 6, 10, or 12 bytes.

**Data In Length** A 32-bit binary number indicating the number of bytes to be received during the Data In phase.

**SCSI Command** The SCSI Command itself — limited to 12 bytes.

**Data Out** An optional field for any data that PCI should send out during the Data Out phase.

When using Method 1, your application must always send the entire 128-byte header. Along with it you can send:

- A Data Out phase (a write). The value in the Data In field is not used.
- A Data In phase (a read), with the Data In Length field indicating the number of bytes you expect PCI to receive during the SCSI Data In phase.
- Neither a Data Out nor a Data In phase. Set the Data In Length field to zero.

You cannot send a command that has both a Data Out and a Data In phase.

If you send more than 128 bytes in the Message Data phase, the PCI server assumes that the data following the header is to be sent out to the SCSI bus during the SCSI Data Out phase. If you send more than 128 bytes of data and you have a non-zero value in the Data In Length field, the SCSI Data Out phase will take precedence because PCI assumes that this SCSI command has a Data Out phase but no SCSI Data In phase.

### Unsolicited Peripheral Status Message

Byte	Description
0	80H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Solicited Peripheral Status Message

Byte	Description
0	80H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Count
12	Number of Retries
13-15	Device-specific Error

## Method 2

This method is compatible with existing applications. Set bit 5 of the PCM to 0, then:

For Data Out phase (a write)

Send a solicited PCM with the Data Length field set to 0. The server assumes the data length to be the Data Length shown in the Buffer Request message. The PSM will be unsolicited.

For Data In phase (a read)

Send an unsolicited PCM, specifying the Data Length fields. The PSM will be solicited.

## Unsolicited Peripheral Command Message for Method 2

### PCM Control Portion

Byte	Description
0	80H
1	Flags
2	Device Type
3	Unit Number
4-5	Data Length (low)
6	SCSI Command (0)
7	SCSI Command (1)
8	SCSI Command (2)
9	SCSI Command (3)
10	SCSI Command (4)
11	SCSI Command (5)
12	SCSI Command (6)
13	SCSI Command (7)
14	SCSI Command (8)
15	SCSI Command (9)
16-17	Data Length (high)
18-19	Reserved

Definitions are the same as for the Solicited PCM on the next page.

### Solicited Peripheral Command Message for Method 2

Byte	Description
0	80H
1	Flags
2	Device Type
3	Unit Number
4-5	Data Length (low)
6	SCSI Command (0)
7	SCSI Command (1)
8	SCSI Command (2)
9	SCSI Command (3)
10	SCSI Command (4)
11	SCSI Command (5)
12	SCSI Command (6)
13	SCSI Command (7)
14	SCSI Command (8)
15	SCSI Command (9)

Where:

Flags

This table describes the flag bits:

Name	Bit	Value	Meaning
Reserved	0	0	Must be set to 0.
Disconnect <sup>1</sup>	1	0	PCI server disables the SCSI disconnect feature during the command.
		1	PCI server lets the SCSI target disconnect during the command.
Extended Cmnd	2	0	The SCSI command is a six-byte command, and only the first 6 bytes of the SCSI command are used.
		1	The SCSI command is an extended command; that is, all 10 bytes of the SCSI Command field are valid.

Request Sense	3	0	PCI server issues a SCSI <b>Request Sense Disable</b> command if the passed-thru SCSI command returns a SCSI Check Condition status.										
		1	PCI server does not issue the <b>Request Sense</b> command, and the client has to issue the <b>Request Sense</b> command with a subsequent <b>SCSI Pass Thru</b> request.										
Extended Data	4		Relevant only if a Data In phase is expected.										
		0	PCI server expects the Data In phase to have a length of Data Length (low) bytes (see the Data Length parameter).										
		1	PCI server expects the Data In phase to have a length of $([\text{Data Length (high)} \times 64\text{K}] + \text{Data Length (low)})$ bytes (see the Data Length parameter). The byte order is: Data Length (high) followed by Data Length (low):										
			<table border="0" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: left;">Byte</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>17</td> <td>Most significant byte<sup>2</sup></td> </tr> <tr> <td>16</td> <td>2nd MSB<sup>2</sup></td> </tr> <tr> <td>5</td> <td>3rd MSB</td> </tr> <tr> <td>4</td> <td>Least significant byte</td> </tr> </tbody> </table>	Byte	Description	17	Most significant byte <sup>2</sup>	16	2nd MSB <sup>2</sup>	5	3rd MSB	4	Least significant byte
Byte	Description												
17	Most significant byte <sup>2</sup>												
16	2nd MSB <sup>2</sup>												
5	3rd MSB												
4	Least significant byte												
Reserved	5-7	0	Must be set to 0.										

Notes:

- <sup>1</sup> Generally, the Disconnect Flag (bit 1) is set to 1. When this flag is cleared (0), data transfer is slower and data transfer length is restricted.
- <sup>2</sup> Bits 17 and 16 are for the unsolicited message only.

## SCSI Pass Thru

---

**Data Length** A 32-bit binary number indicating the number of bytes to be transferred during the Data In or Data Out phase.

This field is relevant only if the **SCSI Pass Thru** command is in an unsolicited message. To indicate a Data Out phase, send a solicited PCM. The Data Length must be even.

This table shows the meanings of the Data Length byte and the extended data flag:

<b>Data Length</b>	<b>Extended Data Flag</b>	<b>Meaning</b>
0	Don't care	No data phase is expected (use this for a solicited PCM).
Not zero	0	A data phase of this length is expected (less than 64K).
	1	A data phase of more than 64K-1 is expected. (This means the client should use the information in the Data Length (high) field.)

**SCSI Command** The actual SCSI command bytes that are sent on the SCSI bus during the Command phase. The server uses only the first 6 bytes if the Extended Command flag is 0.

**Reserved** These 4 bytes are present only in the case of an unsolicited message. They must be set to 0.

### Peripheral Status Messages

Both the solicited and unsolicited Peripheral Status Messages are the same as for Method 1.

## Seek Beginning of Tape

Rewinds tape.

This command rewinds the tape to the beginning of tape marker. This command does not apply to hard or flexible disk drives.

### Unsolicited Peripheral Command Message

Byte	Description
0	2AH
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	2AH
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Seek End of Data

Moves tape to end of data (EOD) filemark.

This command searches for the end of data and positions the tape at the end of data filemark. This allows the server to append data to data that already exists; the server leaves the EOD filemark at the end of the new data. This command does not apply to hard or flexible disk drives.

### Unsolicited Peripheral Command Message

Byte	Description
0	2BH
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	2BH
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

## Seek Filemark

Moves the tape forward.

This command advances the tape a specified number of filemarks or until end-of-media is reached, whichever occurs first. It positions the tape just after the specified filemark.

This command does not apply to hard or flexible disk drives.

### Unsolicited Peripheral Command Message

Byte	Description
0	27H
1	Reserved
2	Device Type
3	Unit Number
4-7	Reserved
8-11	Count
12-19	Reserved

Where:

**Count**                      Number of filemarks the tape device will seek. A count of 0 is valid, but performs no operation.

### Unsolicited Peripheral Status Message

Byte	Description
0	27H
1	Status
2	Device Type
3	Unit Number
4-7	Reserved
8-11	Actual Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

Where:

Actual Count      Number of filemarks the tape was moved.

### Set Reset Blocking Time

Tells the server how long to wait before accepting commands after a reset.

For a given period of time after a SCSI reset, the PCI server returns all commands that require access to the SCSI bus. The server puts the I/O Interface Reset status code in the reply message. By using this command, the client sets this time period, which is called the reset blocking time.

### Unsolicited Peripheral Command Message

Byte	Description
0	92H
1	Reserved
2	Device Type
3	Unit Number
4-5	Delay Time
6-19	Reserved

Where:

Delay Time      Each 01H increment is 10 ms. 0FFFFH is an infinite delay.

### Unsolicited Peripheral Status Message

Byte	Description
0	91H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### Set Statistics

Modifies the statistics counters.

The client can send this implementation-specific command to the server to change the statistics counters kept by the controller board. This message is useful if the client needs to reset some (but not all) of the statistics counters.

To reset certain counters to zero, first use the **Get Statistics** command to get a buffer containing the current values. Then, use the **Set Statistics** command to put zeros in the chosen value fields in the buffer and send the buffer back to the server.

A client can issue a **Set Statistics** command to the server at any time, except when the unit is reserved by another client.

### Solicited Peripheral Command Message

#### PCM Control Portion

Byte	Description
0	82H
1	Reserved
2	Device Type
3	Unit Number
4-15	Reserved

#### PCM Data Portion

The data portion of the request includes statistics for all the devices on the SCSI bus, followed by statistics for the specified unit. See the **Get Statistics** command for the format of the data.

### Unsolicited Peripheral Status Message

Byte	Description
0	82H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

## Set Target Mode Options

Allocate control and data buffers.

For target mode, the client that will receive data sends this command to its server. If the client does not know its server's target ID, it can use the **Get SCSI ID** command. The client can read the current options with the **Get Target Mode Options** command.

### ⇒ Note

For clients that use target mode, the applications programs that communicate with each other must use the same SCSI target ID for the target server. Because there are eight LUNs available, the target server can handle up to eight different applications programs.

## Unsolicited Peripheral Command Message

Byte	Description
0	99H
1	Reserved
2	Device Type
3	Unit Number
4-5	Reserved
6-7	Number of Control Buffers
8-9	Number of Data Buffers
10-11	Data Buffer Size
12-13	Maximum Control Buffers Per Init
14-19	Reserved

Where:

Device Type      Must be 80H.

Unit Number      The client can calculate the unit number as:  
[(LUN × 8) + SCSI target ID].

Number of Control Buffers  
Number of control buffers that the PCI server allocates for this target mode logical unit.

Number of Data Buffers  
Number of data buffers that the PCI server allocates for this target mode logical unit.

## Set Target Mode Options

---

### Data Buffer Size

Size of each data buffer that the PCI server allocates for this target mode logical unit.

### Max Ctrl Bufs Per Init

Maximum number of control buffers the PCI Server allocates for any single initiator. This number must be at least 2.

## Unsolicited Peripheral Status Message

Byte	Description
0	99H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

Where:

Device Type      Must contain 80H.

## Set Unit Characteristics

Describes the physical characteristics of a unit to the server.

This command tells the server about the physical characteristics of a unit.

Before the client may send any PCI data transfer commands to a unit, it must first send a **Set Unit Characteristics** command followed by a **Set Unit Options** command. After any client issues the first **Set Unit Characteristics** command to a unit, a subsequent **Set Unit Characteristics** command has no effect unless the Force Mode flag is set. The force Mode flag causes the command to take effect unconditionally. If the Force Mode flag is not set, the reply includes a Device Already Initialized status code.

### ⇒ **Note**

For hard disk drives, the server does not issue a **SCSI Mode Select** command.

For flexible disk drives, the server issues **SCSI Mode Sense** and **Mode Select** commands.

For tape drives, the server issues a **SCSI Mode Sense** command. If the block size set on the drive does not match the block size requested by the client, the server also issues a **SCSI Mode Select** command to set the tape drive to the new block size.

## Solicited Peripheral Command Message

### PCM Control Portion

Byte	Description
0	10H
1	Reserved
2	Device Type
3	Unit Number
4	Flags
5-15	Reserved

## Set Unit Characteristics

---

Where:

Flags                      This table describes the flag bits:

<b>Name</b>	<b>Bit</b>	<b>Value</b>	<b>Meaning</b>
Force Mode	0	0	Disabled
		1	Enabled
Reserved	1-7	0	Reserved

### PCM Data Portion (Hard and Flexible Disks)

This section shows the data for direct access devices, which are hard and flexible disk drives. (Device Types: 1 = hard disk drive; 2 = flexible disk drive.)

<b>Byte</b>	<b>Description</b>
0-3	Number of Cylinders
4-5	Number of Surfaces
6-7	Sector/Track
8-9	Bytes/Sector
10-11	Flexible Disk Format <sup>1</sup>
12-15	RWC Cylinder
16-19	WPC Cylinder
20-21	Step Rate <sup>1</sup>
22-23	Alternate Sector/Zone <sup>2</sup>
24-25	Alternate Tracks/Zone <sup>2</sup>
26-27	Alternate Track/Volume <sup>2</sup>
28-31	Tracks/Zone <sup>2</sup>
32-33	Track Skew <sup>2</sup>
34-35	Cylinder Skew <sup>2</sup>
36-37	Interleave
38	Pin 2 <sup>1</sup>
39	Pin 34 <sup>1</sup>
40-255	Reserved

Notes:

<sup>1</sup> Fields that apply only to flexible disks:

Flexible Disk Format  
Step Rate

<sup>2</sup> Fields that apply only to hard disks:

Alternate Sector/Zone                      Tracks/Zone  
Alternate Tracks/Zone                      Track Skew  
Alternate Tracks/Volume                      Cylinder Skew

Where:

Number of Cylinders

Total number of cylinders on the unit including alternatives.

Number of Surfaces

Number of data recording surfaces on the unit.

Sectors/Track

Total number of sectors per track including alternates.

Bytes/Sector

Number of data bytes per sector.

Flexible Disk Format

This table describes bits 0-7 of this 16-bit field:

<b>Name</b>	<b>Bit</b>	<b>Value</b>	<b>Meaning</b>
Transfer Rate <sup>1</sup>	0-1	00	125 Kbits/second
		01	250 Kbits/second
		10	300 Kbits/second
		11	500 Kbits/second
Step Pulses <sup>2</sup>	2-3	00	1 step pulse per cylinder (normal usage)
		01	2 step pulses per cylinder
		10	3 step pulses per cylinder
		11	4 step pulses per cylinder
True Ready	4	0	Server must not rely on Ready signal from drive to assume drive readiness.
		1	Drive provides a true Ready signal which the server uses to sense drive readiness.
Reserved	5-7	0	Reserved.

Notes:

<sup>1</sup> The transfer rate depends on the recording density and encoding method.

<sup>2</sup> Non-zero values for step pulses can be used to access a flexible disk formatted on a drive with a lower tpi (tracks per inch) value; for example, a value of 01 allows a 96 tpi drive to access a flexible disk formatted on 48 tpi drive.

## Set Unit Characteristics

---

This table describes bits 8-15 of the flexible disk format. These bits define the media type:

<b>Value (Bits 8-15)</b>	<b>Disk Diameter</b>	<b>Bit Density (bits/ radian)</b>	<b>Track Density</b>	<b>Number of Sides</b>
00H	Reserved			
01H	8"	6631	48 tpi	1
02H	8"	6631	48 tpi	2
03H	8"	13262	48 tpi	1
04H	8"	13262	48 tpi	2
05H	5.25"	3979	48 tpi	1
06H	5.25"	7958	48 tpi	2
07H	5.25"	7958	96 tpi	2
08H	5.25"	13262	96 tpi	2
09H	3.5"	7958	135 tpi	2
0A-FFH	Reserved			

RWC Cylinder	Starting cylinder number for reduce write current
WPC Cylinder	Starting cylinder number for write precompensation
Step Rate	Drive step rate expressed in units of 0.1 microseconds
Tracks/Zone	Size of defect handling zones created on the unit when it is formatted. Except possibly for the last zone, all zones on the device are the same size. If this field is set to 0., a single zone spans the whole unit.
Alternate Sector/Zone	Number of sectors per zone that will be reserved, when the unit is formatted, to replace defective sectors in that zone.
Alternate Tracks/Zone	Number of tracks per zone that will be reserved, when the unit is formatted, to replace defective tracks in that zone.
Alternate Tracks/Volume	Number of tracks reserved to replace defective tracks on the unit. This parameter is only significant if zone-based defect handling is not supported.
Track Skew	Number of sectors between the last data sector of one track and the first data sector on the next sequential track of the same cylinder.

**Cylinder Skew**    Number of sectors between the last data sector of one cylinder and the first data sector on the next sequential cylinder.

**Interleave**        Interleave factor used by the server to format the tracks of the unit.

<b>Value</b>	<b>Meaning</b>
0	Default interleave.
1	Consecutive data sectors are placed in consecutive physical order.

**Pin 2**                Defines the use of pin 2 of the flexible disk drive interface. Different models of drives use this pin differently. The controller reads this byte to find out how to use pin 2.

<b>Bit</b>	<b>Value</b>	<b>Meaning</b>
0-2		Function
	000	Open
	001	Ready
	010	Disk changed
	011	Speed
	100	Density
	101–	Reserved
	111	
3	0	Active-low polarity
	1	Active-high polarity
4-6	0	Reserved
7		Indicates if the field is valid or not.
	0	Use the default (compatible with previous versions of the PCI server).
	1	Use the other bits.

This table shows the valid values of the Pin 2 field:

Value	Meaning
00H	Use the default (speed/active low)
80H	Open/active low
88H	Open/active high
81H	Ready/active low
89H	Ready/active high
82H	Disk changed/active low
8AH	Disk changed/active high
83H	Speed/active low
8BH	Speed/active high
85H	Density/active low
8CH	Density/active high

Pin 34

Defines the use of pin 34 of the flexible disk drive interface. Different models of drives use this pin differently.

Bit	Value	Meaning
0-2		Function
	000	Open
	001	Ready
	010	Disk changed
	011– 111	Reserved
3	0	Active-low polarity
	1	Active-high polarity
4-6	0	Reserved
7		Indicates if the field is valid or not.
	0	Use the default, which is ready/active low (compatible with previous versions of the PCI server).
	1	Use the other bits.

This table shows the valid values for the Pin 34 field:

Value	Meaning
00H	Use the default (ready/active low)
80H	Open/active low
88H	Open/active high
81H	Ready/active low
89H	Ready/active high
82H	Disk changed/active low
8AH	Disk changed/active high

### PSM Data Portion (Tape Drives)

This section shows the data for tape drives (Device Type = 3):

Byte	Description
0-1	Bytes/Block
2-255	Reserved

Where:

Bytes/Block  
Number of data bytes per tape block.

### Unsolicited Peripheral Status Message

Byte	Description
0	10H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Set Unit Options

Chooses operational modes for a unit.

With this command, the client sets a variety of operational modes for a unit, including number of retries, cache buffer size, command ordering, and read-ahead.

The client must send a **Set Unit Characteristics** before it sends a **Set Unit Options** command. The client must send both these commands to a unit, and they must complete successfully, before the client can send any data transfer commands to the unit.

If the client issues a subsequent **Set Unit Characteristics** command to the same unit, it should also issue another **Set Unit Options** command.

After the first **Set Unit Options** command to a unit, subsequent **Set Unit Options** commands have no effect unless the Force Mode flag is set. The Force Mode flag causes the command to take effect unconditionally. If the Force Mode flag is not set, the server replies with a Device Already Initialized status code.

⇒ **Note**

For hard disk drives, the default buffers size is the total available buffer split equally among the hard disk drives that are present. The default read-ahead is the track size minus any alternates assigned per track.

For flexible disk drives, the server always allocates one buffer the same size as a track. This size cannot be changed.

For tape drives, the server always allocates one 63K buffer unless the client requests the default minimum. For the default minimum, the server allocates one 1K buffer.

The maximum read-ahead is 63K.

## Unsolicited Peripheral Command Message

Byte	Description
0	06H
1	Reserved
2	Device Type
3	Unit Number
4	Flags
5	Mode
6	Reserved
7	Number of Retries
8	Command Ordering
9	Cache Buffer Size
10	Read Ahead
11-19	Reserved

Where:

Flags This table describes the flag bits:

Name	Bit	Value	Meaning
Force Mode		0	Disabled
		1	Enabled
Reserved	1-7	0	Reserved

If the Force Mode flag is set (1), the client can overwrite the settings of a previous **Set Unit Options**.

Mode This table describes the mode bits:

Name	Bit	Value	Meaning
Address Mode		0	Logical addressing
		1	Address Mode not used
Cache Mode	1	0	Disabled
		1	Enabled
Command Ordering Mode <sup>1</sup>	2	0	Disabled <sup>2</sup>
		1	Enabled <sup>3</sup>
Reserved	3-7	0	Reserved

Notes:

<sup>1</sup> Command Ordering Mode does the scheduling for requests that aren't satisfied by the cache and need to go to the disk.

<sup>2</sup> Strictly first-come-first-served.

<sup>3</sup> Queuing order not guaranteed.

## Set Unit Options

---

### Number of Retries

The server ignores this field.

### Command Ordering

This table describes the command ordering bits:

<b>Name</b>	<b>Bit</b>	<b>Value</b>	<b>Meaning</b>
Read/Write	0-1	00	Equal priority
Relative Priority		01	Reads before writes (default)
		10	Writes before reads
		11	Use the default <sup>2</sup>
Seek Ordering Algorithm <sup>1</sup>	2-5	00H	First come first served
		01H	Shortest seek time first
		02H	Elevator (default)
		03H	CSCAN
		04-0EH	Reserved
		0FH	Use the default <sup>2</sup>
Reserved	6-7	0	Reserved

Notes:

<sup>1</sup> Only valid if Command Ordering Mode is enabled in the Mode byte.

<sup>2</sup> This value allows the default to change dynamically.

The Read/Write Relative Priority takes precedence over the Seek Ordering Algorithm for all queued commands. All reads are in one queue, writes are in another queue, and each queue is ordered by the Seek Ordering Algorithm.

### Cache Buffer Size

This table describes the Cache Buffer Size field:

<b>Value</b>	<b>Meaning</b>
00H	Default minimum.
01H-64H	Percentage of total cache memory to allocate to this unit (64H = 100%).
65H-FEH	Reserved.
FFH	Default size.

### Read Ahead

This table describes the Read Ahead field:

<b>Value</b>	<b>Meaning</b>
00H	Invalid value.
01H-FEH	Minimum number of blocks read by server in any read operation.
FFH	Server selects default size.

## **Unsolicited Peripheral Status Message**

<b>Byte</b>	<b>Description</b>
0	06H
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

### Start Unit

Asks the server to spin up the unit.

This command does not apply to tape drives.

### Unsolicited Peripheral Command Message

Byte	Description
0	0FH
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	0FH
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

## Stop Unit

Asks the server to stop (spin down) the unit.

This command does not apply to tape devices.

### Unsolicited Peripheral Command Message

Byte	Description
0	0EH
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	0EH
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

### Synchronous Negotiation Control

Sets up and asks about synchronized transfers.

This command allows the server to control some synchronous negotiation parameters and to ask for the state of synchronous negotiation on a device. For more information about these parameters, see the manual for your drive.



#### **CAUTION**

Some devices require the Disconnect bit to be set at all times, so the PCI server provides this. However, the server assumes that the device will not actually disconnect because there is no need for it to do so. If the device disconnects, the server will probably hang.

See also: To do an immediate synchronous negotiation, see the procedure in Chapter 4.

### Unsolicited Peripheral Command Message

Byte	Descriptions
0	84H
1	Reserved
2	Device Type
3	Unit Number
4	Subfunction Code
5-15	Reserved

Where:

Subfunction Code

Value	Meaning
0	Reserved.
1	Set Disconnect bit in the Identify message. (This sets the Disconnect bit only in the Identify message used during synchronous negotiation.)
2	Don't set Disconnect bit in the Identify message (default).
3	Clear the synchronous state information. (This sets the Need to Negotiate flag and sets all the synchronous state values to their defaults. Synchronous negotiation will occur during the next initialization or the next access to the device.)
4	Return the synchronous state information. (The PSM returns the state information in fields 4 through 8. The information shows whether the current accesses are synchronous or asynchronous and whether PCI will do synchronous negotiation on its next initialization, read, or write.)
5	Do synchronous negotiation. (Synchronous negotiation will occur only if the Need to Negotiate field is 01H.)
6-255	Reserved.

## Unsolicited Peripheral Status Message

Byte	Description
0	84H
1	Status
2	Device Type
3	Unit Number
4	Need to Negotiate Flag
5	Synchronous Flag
6	Request/Acknowledge Offset
7	Transfer Period
8	Do Disconnect
9-19	Reserved



### Note

Fields 4 through 8 are valid only if the PCM set the subfunction code to 4.

Where:

Need to Negotiate Flag

Value	Meaning
00H	Server will not try synchronous negotiation.
01H	Server will try synchronous negotiation: <ul style="list-style-type: none"><li>• The next time the device is initialized</li><li>• On the next read or write to the device</li><li>• When the server receives a <b>Synchronous Negotiation Control</b> command with function code 5, "do synchronous negotiation"</li></ul>

Synchronous Flag

Value	Meaning
00H	Data is transferred asynchronously.
01H	Data is transferred synchronously.

### Request/Acknowledge Offset

Maximum number of REQ pulses that may be outstanding before the driver receives the corresponding ACK pulse.

If the number of REQ pulses exceeds the number of ACK pulses by the Request/Acknowledge Offset, the drive does not assert REQ until the next ACK pulse is received. Successful completion of the data phase requires that the REQ and ACK pulses be equal.

### Transfer Period

Minimum time between the leading edge of a REQ pulse and the corresponding ACK pulse.

This value is: (Transfer Period returned)  $\times$  4 ns.

### Do Disconnect Flag

<b>Value</b>	<b>Meaning</b>
00H	Disconnect bit will not be set in the Identify message during synchronous negotiation.
01H	Disconnect bit will be set in the Identify message during synchronous negotiation.

### Test I/O

Tests a unit for device ready status.

#### Unsolicited Peripheral Command Message

Byte	Description
0	24H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

#### Unsolicited Peripheral Status Message

Byte	Description
0	24H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

## Unit Attention Notification

Notifies the client that a device needs attention.

Clients can request this service by sending a **Request Unit Attention Notification** command to the server. Once a client receives a **Unit Attention Notification** message, it must send another **Request Unit Attention Notification** command to the server to continue the service.

The message is the second reply sent to any client that sent the **Request Unit Attention Notification** command. The first reply acknowledges that the server received the request. This reply is the actual notice of a unit attention condition.

### ⇒ Notes

The server sends this notice when it detects a transition of the Ready signal to Not Ready. To be able to detect the transition, the server polls only those flexible disk and tape drives that were Ready the last time it checked; it polls them every 250 ms. If a drive moves from the Not Ready to the Ready state, the server finds out when it receives the next command for that unit.

Up to 21 clients can **Request Unit Attention** on a single drive.

## Unsolicited Peripheral Status Message

Byte	Description
0	2FH
1	Status
2	Device Type
3	Unit Number
4-19	Reserved

Where:

Status	Set to 00H.
Device Type	Logical device code for the unit that needs attention.
Unit Number	Number of the unit that needs attention.

### Unload

Rewinds a tape.

This command does not apply to hard or flexible disk drives.

### Unsolicited Peripheral Command Message

Byte	Description
0	25H
1	Reserved
2	Device Type
3	Unit Number
4-19	Reserved

### Unsolicited Peripheral Status Message

Byte	Description
0	25H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Reserved
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

## Write Data

Transfers data from client to device.

This command transfers a block of data from the client's memory to the specified device. If the request is fragmented and an error is found after the server receives some data fragments, the server may stop the request phase of the transaction prematurely, as allowed by the Multibus II transport protocol.

## Solicited Peripheral Command Message

### PCM Control Portion

Byte	Description
0	14H
1	Reserved
2	Device Type
3	Unit Number
4-7	Device Address
8-11	Byte Count
12-15	Reserved

Where:

Byte Count      Number of bytes to be transferred from the client to the requested device. This count must be a multiple of the device block size. The length field in the buffer request should match this number.

### PCM Data Portion

The data portion is the data that the client wants to write to the device.

### Unsolicited Peripheral Status Message

Byte	Description
0	14H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Byte Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

## Write Data Direct

Transfers a block of data from host memory to a device.

Used when the client and server are on the same board, this command causes the server to write directly to the device from a buffer defined by the client on the board.

### Unsolicited Peripheral Command Message

Byte	Description
0	15H
1	Reserved
2	Device Type
3	Unit Number
4-7	Device Address
8-11	Byte Count
12-15	Physical Address
16-19	Reserved

Where:

Device Type	Must be 01H (hard disk drive).
Device Address	This address is either the starting cylinder, head, and sector or the logical block number. This field is not used for tapes.
Byte Count	Number of bytes to be transferred from system memory to the requested device. This value must be a multiple of device block size.
Physical Address	Physical address of the buffer that contains the data (the address must be even.)

### Unsolicited Peripheral Status Message

Byte	Description
0	15H
1	Status
2	Device Type
3	Unit Number
4-7	Device Error Address
8-11	Actual Byte Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved

## Write Filemark

Writes a filemark on the page.

This command writes one or more filemarks on the tape at the current tape position. It does not rewind the tape, but positions the tape just after the filemark so that additional data can be written to the tape. This command does not apply to hard or flexible disk drives.

## Unsolicited Peripheral Command Message

Byte	Description
0	28H
1	Reserved
2	Device Type
3	Unit Number
4-7	Reserved
8-11	Count
12-19	Reserved

Where:

Count                      Number of filemarks to be written.

## Unsolicited Peripheral Status Message

Byte	Description
0	28H
1	Status
2	Device Type
3	Unit Number
4-7	Reserved
8-11	Actual Count
12	Number of Retries
13-15	Device-specific Error
16-19	Reserved





This chapter gives the procedures for using the PCI commands. This chapter includes:

- Finding the PCI servers in the system
- Preparing to send commands
- Changing the parameters for a unit
- Reading a block of data from a unit
- Writing a block of data to a unit
- Subscribing to an announcement service
- Using Target Mode
- Doing immediate synchronous negotiation
- Sending messages in the iRMX III OS

## Finding the PCI Servers in the System

A peripheral controller board can contain more than one PCI server. The first server listens at the Multibus II transport port ID 520H (which is different from an I/O port address).

To find out how many PCI servers are in the system and what their port IDs are, you can use this procedure:

1. Use one of two methods to find all potential sites for PCI servers:
  - Scan interconnect space for the appropriate board IDs, and send the **Locate PCI Server** command to port ID 520H at each peripheral controller (such as the 386/258 board) in the system.
  - Broadcast the **Locate PCI Server** command.

Broadcasting is the more efficient method. If you get a reply from either method, it means that at least one PCI server is present at that site.

2. Send the **Get Server Information** command to port ID 520H with the Command field set to 00H.

The server at 520H returns the number of the PCI servers on the board.

3. Send the **Get Server Information** command to port ID 520H. Set the Command field to 01H, and set the Argument 1 field to the number of the server you want to know about. The first instance of the server is 0, and the second instance is 1.

The server returns the port ID of the server that you asked about. For example, if the board has two servers, the first instance uses port ID 520H, and the second instance uses a different port ID.

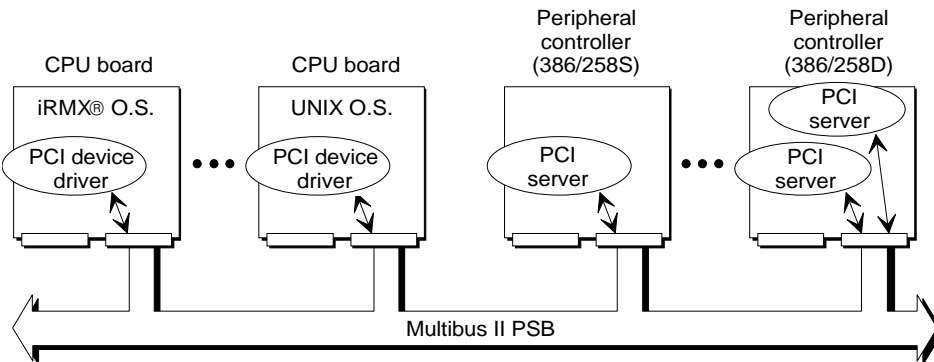
See also: Chapter 3 for a description of the **Get Server Information** command bits

## Example

The single-ended SCSI 386/258S peripheral controller supports one SCSI bus and runs one PCI server. The dual version of the board, the 386/258D, supports both a single-ended SCSI bus and a differential SCSI bus and runs two servers. Both servers have the same host ID but listen at different port IDs. These two servers operate independently and have different SCSI target IDs.

Figure 4-1 shows an example system that includes at least two CPU boards and two 386/258 boards. The CPU boards running the operating systems contain only the PCI driver, not the PCI server. The PCI server on the 386/258S listens at port ID 520H.

On the dual board, the PCI server for the single-ended interface listens at port ID 520H. By sending a **Get Server Information** command to this port, you can get the port ID of the PCI server that controls the differential interface. Even if the single-ended interface is not connected to any devices and only the differential interface is used, the single-ended PCI server always runs.



OM02094

Figure 4-1. Example System

## Preparing to Send Commands

Before the client may read or write to a unit, it must initialize the unit:

1. Use a **send\_sol\_RSVP** call with a pointer to the **Set Unit Characteristics** PCM and a pointer to the data.

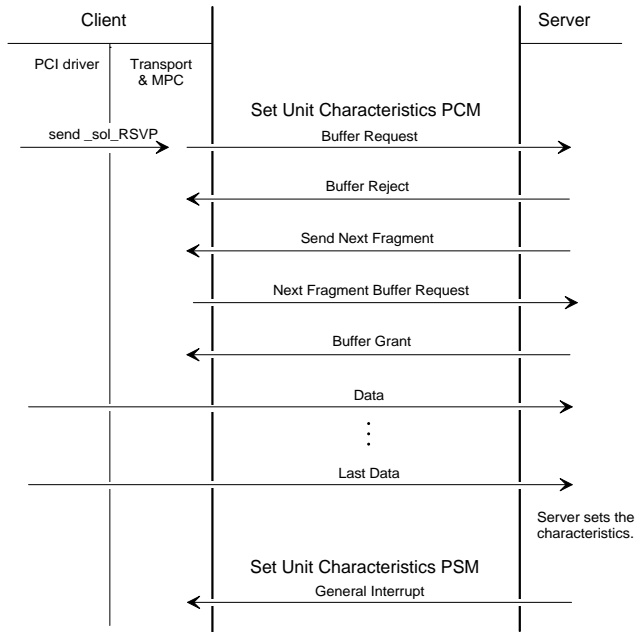
As shown in Figure 4-2, this solicited message starts with a buffer request that is first rejected because the PCI server uses fragmentation mode.

When the server grants the buffer, the MPC driver sends the data. After the data transfer, the server sets the characteristics and returns the status in a **Set Unit Characteristics** PSM.

2. Check the Status field in the PSM. If it is 00H, Command Completed, continue with the next step. If the status is not 00H, a problem exists that needs to be corrected.

See also: Appendix B lists the PSM status codes

3. Use a **send\_unsol\_RSVP** with a pointer to the **Set Unit Options** PCM (see Figure 4-3). The server returns the status in a **Set Unit Options** PSM.
4. Check the status field in the PSM. If it is 00H, Command Completed, the client may now send any combination of commands. If the status is not 00H, there is a problem that needs to be corrected.



OM02095

**Figure 4-2. Set Unit Characteristics**

## Changing the Parameters for a Unit

To change the parameters for a unit after the first **Set Unit Options** PCM for a drive:

1. Use a **send\_unsol\_RSVP** call with a pointer to the **Set Unit Options** PCM. The Force Mode bit in the PCM must be set to cause the parameters to change.
2. Check the Status field in the PSM. If the status is not 00H, a problem exists that needs to be corrected.

See also: Appendix B lists the status codes



### Note

This change affects the operation of the unit for all clients that access it. However, the client does not need to guarantee that all clients are quiescent before sending the message.

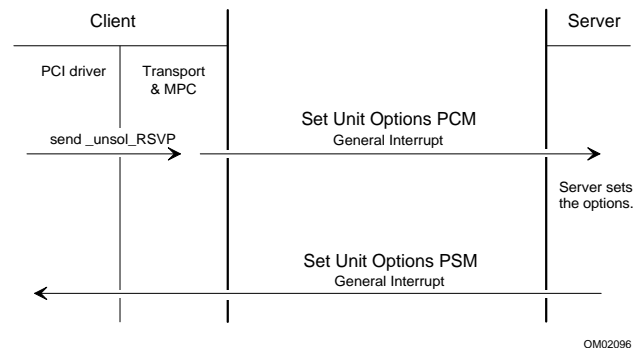


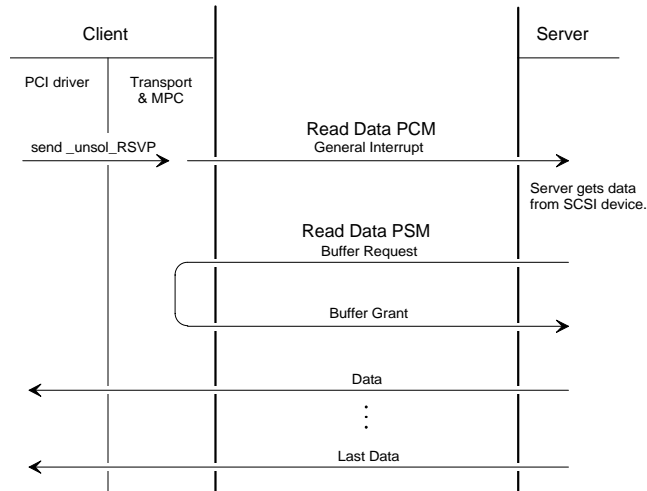
Figure 4-3. Set Unit Options

# Reading a Block of Data from a Unit

To read a block of data from a unit:

1. Use a `send_unsol_RSVP` call with a pointer to the **Read Data** PCM.

The server reads the data from the device and then requests a buffer. After the client grants the buffer, the server sends the block of data. See Figure 4-4.



OM02097

**Figure 4-4. Read Data**

The server may use a fragmented reply.

If the server runs into a read error, it may send an unsolicited (general interrupt) message as the reply.

## Writing a Block of Data to a Unit

This list and Figure 4-5 show how to write a block of data:

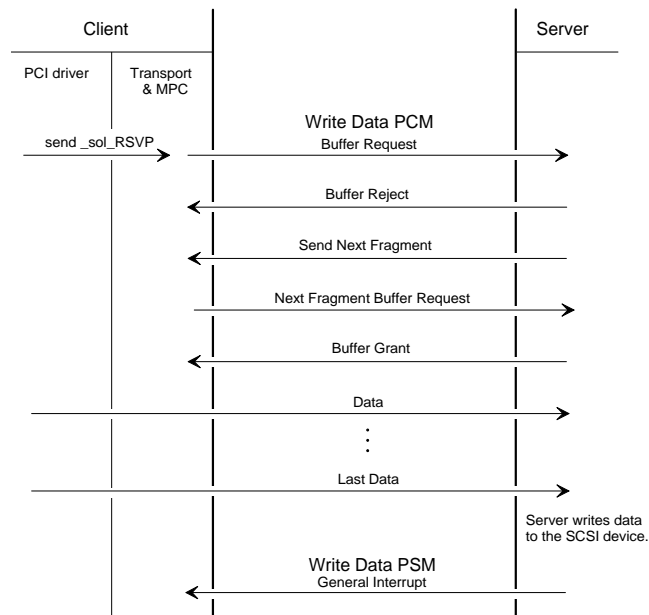
1. Use a `send_sol_RSVP` call with a pointer to the **Write Data** PCM and a pointer to the data.

Multibus II transport code automatically handles the buffer request, buffer reject, and next fragment messages. It also automatically sends the data after the buffer is set up.

The server sends the status in a **Write Data** PSM.

2. Check the Status field in the PSM. If the status is not 00H, a problem exists that needs to be corrected.

See also: Appendix B lists the status codes



OM02098

Figure 4-5. Write Data

## Subscribing to an Announcement Service

The PCI server can send a notice to the clients when one of these events occurs:

- When the SCSI bus is reset.
- When a unit needs attention, such as when the flexible disk or cartridge tape door is open.

The server sends the notice to the client only if the client asked for the service by using one of these commands: **Request Reset Notification**, or **Request Unit Attention Notification**.

Although this section describes how to use **Request Unit Attention Notification**, the client uses the same procedure for the **Request Reset Notification** command.

To ask the PCI server to send notice when a unit needs attention:

1. Use a **send\_unsol\_RSVP** call with a pointer to the **Request Unit Attention Notification** PCM. (See Figure 4-6.)

The OS and the MPC driver use the information from the **send\_unsol\_RSVP** command to prepare and send the **Request Unit Attention Notification** command to the server.

Under normal circumstances, the server replies that it received the request.

2. Check the Status field in the PSM.
3. If the status is not 00H, a problem exists that needs to be corrected.

See also: Appendix B lists the status codes

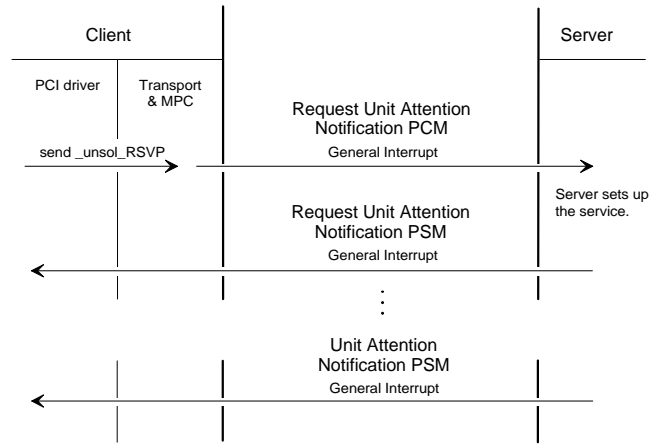
4. The client should be prepared to receive a **Unit Attention Notification** from the server. The server will send it when a device needs attention.

### ⇒ **Notes**

Multiple hosts may issue this request for the same unit.

After receiving unit attention notice, the client must subscribe again.

To cancel the subscription, the client sends the same command with 0 for the Port ID.



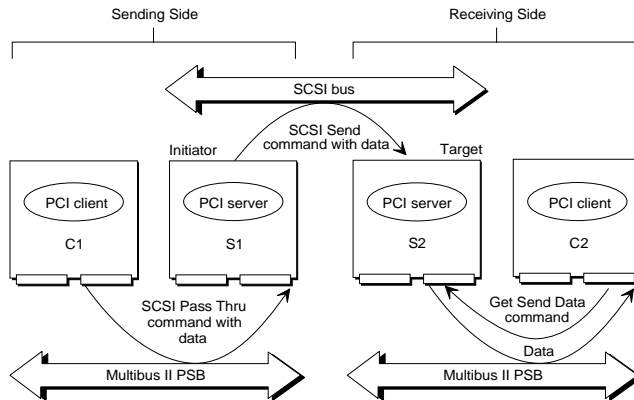
OM02099

**Figure 4-6. Unit Attention Notification**

## Using Target Mode

To use target mode, both the sender and the receiver are clients and they must send commands to their servers (see Figure 4-7). For this example, the sender is a PCI client, C1, that is sending data to another PCI client, C2.

Client C1 sends commands and data to its server, S1. S1 is the initiator and sends the data to the target server, S2. S2 automatically picks up the data, and client C2 retrieves the data from S2.



OM02100

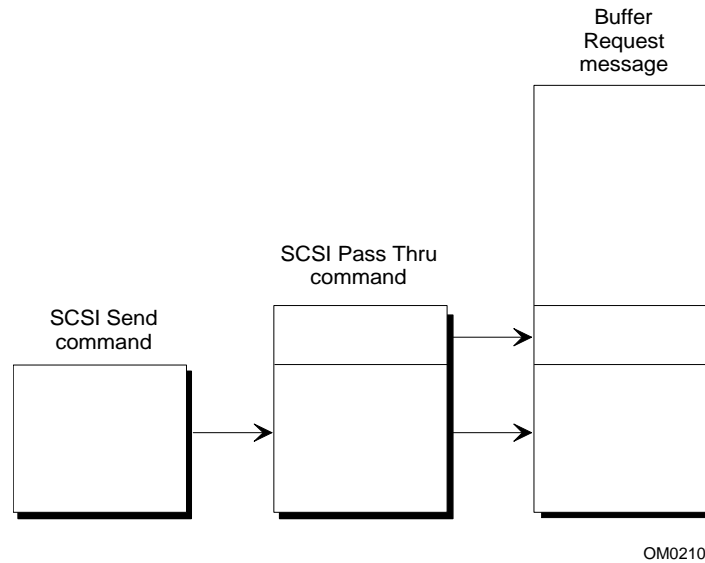
**Figure 4-7. To Transfer Data, Both the Sending and Receiving Clients Must Use Commands**

## Role of the Sender

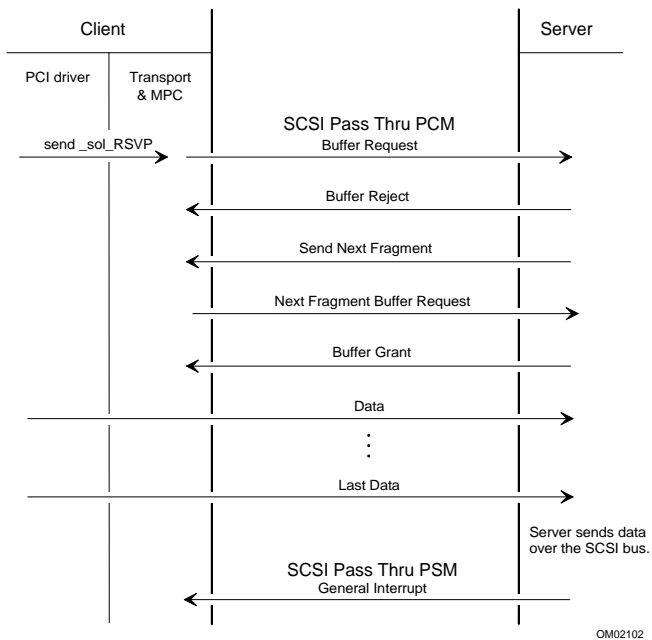
Client C1 must do the following (see Figures 4-8 and 4-9):

1. If the PCI server's buffer is not large enough, use a **Set Unit Options** command to increase it.
2. Embed a **SCSI Send** command in the PCI command called **SCSI Pass Thru**.
  - a. Set the Disconnect bit in the Flags field of the **SCSI Pass Thru** message.
  - b. Set the Device Type to the pass-thru type, 80H.
  - c. Set the Unit Number for the SCSI target ID and the LUN of the target.
3. Send the **SCSI Pass Thru** PCM to the server S1. The PCM must be in a buffer request, which is a solicited message.

S1 then puts the SCSI **Send** command and the data on the SCSI bus, and S2 picks up the data.



**Figure 4-8. Client C2 Nests the SCSI Command in a Buffer Request**



OM02102

**Figure 4-9. Sender's SCSI Pass Thru Message**

## Role of the Receiver

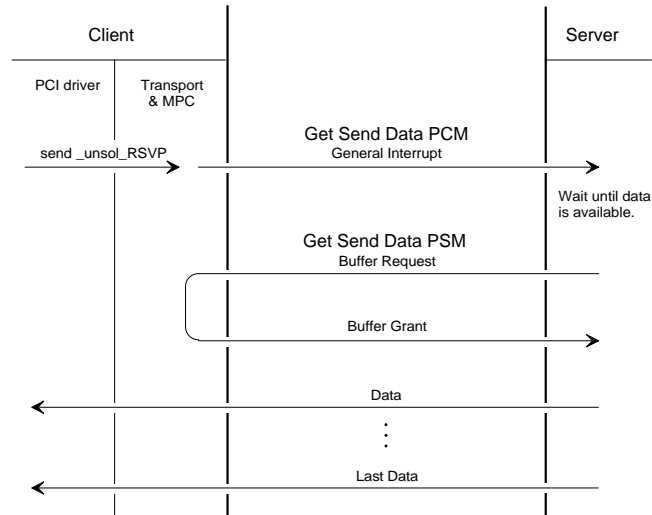
Client C2 must do the following to retrieve the data from server S2. For target mode, C2 does not initialize the driver with the **Set Unit Options** and **Set Unit Characteristics** commands. See Figure 4-10.

1. If the buffers on the server are not large enough, enlarge them by using the **Set Target Mode Options** command.
2. If the client does not know the ID of its server, use the **Get SCSI ID** command.

The SCSI ID for target mode is the same ID the PCI server uses on the SCSI bus when it is an initiator.

The sending and receiving clients must be programmed to use the same LUN. Because there are eight logical unit numbers, the server can handle eight pairs of client programs.

3. Prepare a **Get Send Data** command.
  - a. Set the Device Type to 80H (pass thru).
  - b. Use a Unit Number constructed from the target ID of the peripheral controller that the Server S2 controls and the LUN that the client wants to contact:  $[(LUN \times 8) + \text{SCSI target ID}]$ .



OM02103

**Figure 4-10. Receiver's Get Send Data Command**

## Buffers for Target Mode

The PCI server maintains a pool of data and control buffers for each logical unit. When the server receives a SCSI **Send** command, it uses data buffers to receive the data and a control buffer to maintain state information about the transaction.

The receiving client should set up the number and size of control and data buffers on its server with the **Set Target Mode Options** command. The client can find out what is available for a logical unit by first sending the **Get Target Mode Options** command.

## Data Buffers

The client should make the data buffer space as large as is practical. The data space may be one or more large buffers, or a larger number of smaller buffers.

## Control Buffers

The PCI server takes different actions depending on the control buffers and the permission to disconnect from the SCSI bus:

No control buffer or not enough control buffers

If the server does not have enough control buffers, it stops the SCSI transaction with a Busy status.

Disconnect not allowed

In some cases the server does not have enough data buffer space to take all the data at once. Ideally, the server can disconnect from the SCSI bus until the client takes some of the data. Then the server reselects the initiator to receive the rest of the SCSI data.

However, the initiator must give permission for a disconnect in the SCSI identify message. If it does not give this permission, and if the server cannot buffer all of the data from the SCSI bus, the Version 2.0 server does not accept any of the data. Instead it returns a Busy SCSI status.

It is recommended that SCSI **Send** commands give permission for disconnecting.

Maximum buffers and disconnect allowed

In this case the initiator has given permission for a disconnect and the client has allocated the maximum number of control buffers for a particular initiator. The PCI server disconnects from the SCSI bus before it uses the last control buffer and records the transaction in the last control buffer. Thus, it avoids having to send a Busy SCSI status to the initiator.

## Buffer Size Compared to Data Size

Table 4-1 shows the relation of the length of data to the EOM flag.

Key to the Table 4-1:

Is send complete?

If send is complete, the PCI server received all of the data from the SCSI bus and completed the **SCSI Send** operation.

Data Length

“=” means that the length field in the **Get Send Data** command is equal to the amount of data buffered in the server.

“>” means that the length field in the **Get Send Data** command is greater than the amount of data buffered in the server.

“<” means that the length field in the **Get Send Data** command is less than the amount of data buffered in the server.

GSD

Abbreviation for the **Get Send Data** command.

EOM Flag

The server sets the EOM flag to 1 only when there is no data pending and no SCSI transaction in progress.

Assumptions made for table:

- Data transfers on SCSI bus and Multibus II are as large as possible, limited only by the size of the buffers.
- The server may disconnect for other reasons than those mentioned in the table. If the client has chosen the maximum number of control buffers, the server keeps track of the number and uses the last one to store the information about the transaction and then disconnects until the control buffers are free again.

**Table 4-1. Buffer Size Compared to Data Size**

<b>Is Send Complete?</b>	<b>Data Length</b>	<b>Server Actions</b>	<b>EOM Flag</b>	<b>Client Follow-up</b>	<b>Server Follow-up</b>
Yes	=	The server gives all data to client.	1	none	none
Yes	>	The server gives all data to client.	1	none	none
Yes	<	The server fills up the GSD reply. One GSD is not enough.	0	Issue more GSD commands until one arrives with the EOM flag set.	Reply to subsequent GSD commands until all the data has been sent.
No	=	The server received as much data as possible from the SCSI bus and then disconnected. Server gives client the data received so far and does not set the EOM flag.	0	Issue more GSD commands until one arrives with the EOM flag set.	Reselect SCSI initiator to get the rest of the data.
No	>	The server received as much data as possible from the SCSI bus and then disconnected. Server gives client the data received so far in a Multibus II reply fragment. Server does not send control portion of message.	GSD message is not complete.	none	Reselect SCSI initiator to get the rest of the data.
No	<	The server received as much data as possible from the SCSI bus and then disconnected. Server gives client the data received so far and does not set the EOM flag.	0	Issue more GSD commands until one arrives with the EOM flag set.	Reselect SCSI initiator to get the rest of the data.

## Target Mode Operation

Version 2.0 handles target mode in this way:

- If a **SCSI Send** command does not allow the server to disconnect, the PCI server accepts the data only if it has enough buffers for all of the data. Otherwise, as mentioned earlier, it simply returns a Busy SCSI status.
- If a **SCSI Send** command has a length of zero, the server accepts it, but does not record the transaction or tell the client about it.
- If the server receives an unsupported SCSI command, it completes the Command Phase, returns a Check Condition SCSI status, and records the sense key as Illegal Request.
- The server accepts synchronous negotiation after being selected and receiving the Identify message.

## Commands and Target Mode

If the data is too long for the first **Get Send Data** command, the server clears the EOM bit in the PSM and tells the client how much data remains. The client must send another **Get Send Data** command. The server sends as much as it can without exceeding the client's buffer.

The client may keep multiple **Get Send Data** commands posted at the server, and may find out the limit by sending the **Get Target Mode Options** command.

To cancel **Get Send Data** commands, the client can use the **Flush Get Send Data** command. The server returns a PSM with a length of 0 and an OK status for each **Get Send Data** command. It finishes by sending the PSM for the **Flush** command.

The receiving server does not combine data from different SCSI **Send** commands. If it receives one block of 1000 bytes in one SCSI **Send** and another block of 500 bytes in another SCSI **Send** from the sender and a **Get Send Data** PCM from the receiving client for 2000 bytes, the server sends only the first block (that is, 1000 bytes) in the **Get Send Data** PSM. The client must send another command to get the other block.

In target mode, the client must not initialize logical units of the server with the **Set Unit Characteristics** and **Set Unit Options** commands.

## Doing Immediate Synchronous Negotiation

To do an immediate synchronous negotiation, send three **Synchronous Negotiation** commands in this order:

Step	Subfunction	
	Code Set To	Meaning
1	03	Clear the synchronous state.
2	01 or 02	Set (01) or don't set (02) the Disconnect bit.
3	05	Do synchronous negotiation.

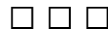
See also: **Synchronous Negotiation** command, in Chapter 3

## Sending Messages in the iRMX III OS

The actual method of sending **send\_unsol\_RSVP** and **send\_sol\_RSVP** is different for each operating system.

For the iRMX III OS, the device driver uses a Nucleus system call, **send\_rsvp**, to send either an unsolicited or solicited RSVP message.

See also: **rq\_send\_rsvp** description in the *System Call Reference*





# Numerical List of Commands

# A

Table A-1 uses these symbols:

<b>Symbol</b>	<b>Meaning</b>	<b>Explanation</b>
H	Hard disk drive	The command may be used with a hard disk.
F	Flexible disk drive	The command may be used a flexible disk.
T	Tape drive	The command may be used with a cartridge tape.
PT	Pass thru	The command must use the pass-thru device type.
I	Ignored	The command ignores the device type and unit number. They should be set to zero.
R	Required	The command must be preceded by two preliminary commands: <b>Set Unit Characteristics</b> and <b>Set Unit Options</b> .
Prohib	Prohibited	These target mode commands must not be preceded by the two commands, <b>Set Unit Characteristics</b> and <b>Set Unit Options</b> .
A	Allowed	The command may be used while a unit is reserved.

Blank spaces in the Table A-1 columns mean that: the command cannot be used with the device; the command uses the device type and unit number; the command does not require preliminary commands; or the command may not be used while a unit is reserved.

**Table A-1. Numerical List of PCI Commands**

Hex Value	Command	Hard Disk	Flex Disk	Tape	CD-ROM	Pass Thru	Ignore Dev Type & Unit #	Req. Prelim Cmds.	Allow While Rsvd
00H	Reserved								
01H	Locate PCI Server						I		
02H	Reset Device	H	F	T	C	PT			
03H	Reserve	H	F	T	C	PT			
04H	Release	H	F	T	C	PT			
05H	Get Device List	H	F	T	C	PT	I*		
06H	Set Unit Options	H	F	T	C	PT			
07H	Query Unit Options	H	F	T	C	PT			A
08H	Format Unit		H	F				R	
09H	Reassign Blocks	H						R	
0AH	Read Defect List	H						R	
0BH	Read Device Capacity	H			C			R	
0C- 0DH	(Reserved)								
0EH	Stop Unit	H	F		C			R	
0FH	Start Unit	H	F		C			R	
10H	Set Unit Characteristics	H	F	T					
11H	Read Data	H	F	T	C			R	
12H	Read Data Direct	H						R	
13H	(Reserved)								
14H	Write Data	H	F	T				R	
15H	Write Data Direct	H						R	
16H	(Reserved)								
17H	Query Unit Characteristics	H	F	T	C				A

continued

\* For the Get Device List command, the server ignores the unit number, but the device type must be valid.

**Table A-1. Numerical List of PCI Commands (continued)**

Hex Value	Command	Hard Disk	Flex Disk	Tape	CD-ROM	Pass Thru	Ignore Dev Type & Unit #	Req. Prelim Cmds.	Allow While Rsvd
18H	Get Server Information						I		A
19-21H	(Reserved)								
22H	Read Data and Verify	H	F	T	C			R	
23H	Recalibrate	H	F					R	
24H	Test I/O	H	F	T	C				
25H	Unload			T	C			R	
26H	Retention Tape			T				R	
27H	Seek Filemark (Space Fwd FM)			T				R	
28H	Write Filemark			T				R	
29H	Erase Unit			T				R	
2AH	Seek Beginning of Tape (BOT)			T				R	
2BH	Seek End of Data (EOD)			T				R	
2CH	Load			T	C			R	
2D-2EH	(Reserved)								
2FH	Unit Attention Notification**	H	F	T	C			R	
30H	Request Unit Attention Notification	H	F	T	C			R	
31-7FH	Reserved								
80H	SCSI Pass Thru	H	F	T	C	PT			
81H	Get Statistics	H	F	T	C	PT			
82H	Set Statistics	H	F	T	C				
83H	I/O Sleep	H	F	T	C	PT			

continued

**Table A-1. Numerical List of PCI Commands (continued)**

Hex Value	Command	Hard Disk	Flex Disk	Tape	CD-ROM	Pass Thru	Ignore Dev Type & Unit #	Req. Prelim Cmds.	Allow While Rsvd
84H	Synchronous Negotiation Control	H	F	T	C	PT			
85-8FH	(Reserved)								
90H	Request Reset Notification					PT	I		A
91H	Reset Notification**						I	R	
92H	Set Reset Blocking Time						I		A
93H	Clear Reset Blocking Condition						I		A
94H	Immediate Reset					PT	I		A
95-97H	Reserved								
98H	Get Send Data					PT		Prohib	
99H	Set Target Mode Options					PT		Prohib	
9AH	Get Target Mode Options					PT		Prohib	
9BH	Get SCSI ID							I	
9C	Flush Get Send Data					PT		Prohib	
9D-FFH	Reserved								

\*\* The server issues these commands: Reset Notification and Unit Attention Notification. The client issues all of the other commands.

□ □ □

## Error Handling

To handle any message and data transfer errors, the PCI server supports the error actions specified by the Multibus II transport protocol. To handle any errors that are related to a peripheral, the server sends a PSM to the host to indicate the end of the transaction. If the controller recovers from an error that did not involve data transfer, it does not report the error.

The PSM can contain any of the status codes listed in Table B-1. It can also contain three bytes called the Device Specific Error (DSE).

# PSM Status Code Definitions

These status codes may be returned in the status field of a PSM:

**Table B-1. Status Codes**

<b>Code</b>	<b>Name</b>	<b>Description</b>
00H	Command Completed	The command completed successfully and no status is pending.
01H	End of Media	While trying to transfer several sectors (hard or flexible disk), the server found the end of media.
02H	End of Recorded Medium	No more data blocks or filemarks are recorded on the tape from the current tape position to the end of the tape.
03H	Filemark Detected	The tape drive found a filemark during a <b>Read Data</b> or a <b>Read Data and Verify</b> command.
04H	Command Timeout	The server issued a command to the SCSI device and did not receive a reply.
05H	Device Reserved	The requested device is currently reserved by another host.
06H	Device Not Initialized	The client issued a command to a device that was not initialized. The device requires that the <b>Set Unit Characteristics</b> or <b>Set Unit Options</b> commands be performed before it responds to a data transfer command.
07H	Device Already Initialized	The device previously received a <b>Set Unit Characteristics</b> or <b>Set Unit Options</b> command, and the Force Mode flag bit was not set on the command that was recently sent.
08H	Invalid PCM Field	One of the fields in the PCM contains an incorrectly specified field.
09H	Invalid Parameter Buffer	The parameter buffer associated with the PCM contains an incorrectly specified field.
0AH	Device Not Ready	The peripheral device specified in the PCM is not ready.

continued

**Table B-1. Status Codes (continued)**

<b>Code</b>	<b>Name</b>	<b>Description</b>
0BH	Write Protected	The client tried to write to a device that is write-protected.
0CH	Media Error	The server stopped the command because of a flaw in the medium or an error in the recorded data. The Device Specific Error field in the PSM contains information about the error. Some reasons for this error are:  For hard disk drives: ECC error found on the Data field or ID field.  For flexible disks: CRC error found on the Data field or ID field.  For hard disk drives or flexible disks: sector not found, drive positioning error, or cylinder miscompare.
0DH	I/O Interface Reset	The I/O interface associated with the unit has been reset.

□ □ □



# Glossary

---

If a term is not defined here, use the index to find the term in the text of the manual.

<b>alternate track</b>	A track used in place of a defective track.
<b>BOT</b>	Beginning Of Tape
<b>buffer memory size</b>	Maximum size of a data transfer on the SCSI bus. The transfer between the client and the peripheral controller can be larger.
<b>cache</b>	Part of controller's RAM used by the server to store copies of recently-used data from peripheral devices. The cache discussed in this manual is a disk cache, not a CPU cache.
<b>cache hit</b>	For a read or write operation, the PCI server checks the cache before going to the peripheral device. If data from the requested block address is already in the cache, the operation is a cache hit.
<b>cache miss</b>	For a read or write operation, the PCI server checks the cache before going to the peripheral device. If data from the requested block address is not already in the cache, the operation is a cache miss.
<b>cache size</b>	The amount of cache for a unit. Data transferred with the SCSI Pass Thru command is not cached.
<b>client</b>	In this manual, the client is an application that requests services from the PCI server. The client is typically an operating system's device driver on a CPU board.
<b>CRC (Cyclic Redundancy Check)</b>	A calculation done on data transfers to determine their validity.
<b>ECC (Error-Correcting Code)</b>	A method that allows the system to identify and try to correct a data error.
<b>EOM (End Of Media)</b>	The highest cylinder, head, and sector numbers possible for a given volume on a disk.
<b>EOT</b>	End Of Tape

<b>filemark</b>	A magnetic code that separates physical files on a tape.
<b>global</b>	Pertaining to the PCI server as a whole, rather than a specific device.
<b>host</b>	A board that communicates across the parallel system bus (PSB) by using message passing.
<b>initiator</b>	The requestor at the SCSI bus. Usually, the initiator is the peripheral controller, but for target mode, the initiator can be a client.
<b>message passing</b>	The process of exchanging packets of data over the parallel system bus. Message passing uses the message address space defined by the IEEE 1296 specification.
<b>Multibus II transport protocol</b>	A communications protocol that uses Multibus II message passing and supports real-time distributed processing.
<b>non-RSVP</b>	A message sent without expectation of reply. In a non-RSVP message for PCI, the transaction ID of the PCM byte 10 is 0.
<b>PCI</b>	Peripheral Controller Interface (PCI in this book does not refer to the PCI hardware interface for the personal computer).
<b>PCM</b>	Peripheral Control Message
<b>PCI server</b>	The software or firmware that handles PCI requests from the clients and returns the results to them.
<b>peripheral controller</b>	The board that has the PCI server software or firmware running on it. It attaches directly to the SCSI bus. In some MB II and in all MB I configurations, the peripheral controller has both the client and server functions on it.
<b>PSM</b>	Peripheral Status Message
<b>server</b>	An application that provides services to clients. In this manual, server refers only to the PCI server.
<b>socket</b>	A combination of host ID and port ID. A socket uniquely identifies a piece of software.
<b>target</b>	The responder at the SCSI bus: a SCSI device attached to a target adaptor board, which is attached to the SCSI bus, a SCSI device attached directly to the SCSI bus, or, in target mode, the peripheral controller.

<b>target mode</b>	A mode that treats the peripheral controller board as a target, allowing one client to communicate with another client program via the SCSI bus.
<b>QIC</b>	Quarter Inch Cartridge (tape)
<b>SCSI command</b>	In this manual, the names of all commands used on the SCSI bus begin with the word SCSI. In addition, the name of one PCI command begins with the word SCSI (SCSI Pass Thru).
<b>target adaptor</b>	A board that controls from one to eight SCSI devices. Some devices have built-in controllers and are called embedded devices; others are controlled by a separate piece of hardware called the target adaptor.
<b>unit</b>	One specific peripheral device. For example, two identical hard disk drives on the same SCSI bus have different unit numbers.





## A

Actual Count field, 114  
Alternate Sector/Zone field, 73, 122  
Alternate Tracks/Volume field, 73, 122  
Alternate Tracks/Zone field, 73, 122  
Argument 1 field, 57

## B

buffer request message, 29  
Buffer Size field, 62, 79  
Byte Count field, 30, 81, 83, 85, 139, 141  
Bytes from Index field, 47  
Bytes/Block field, 77, 125  
Bytes/Sector field, 71, 121

## C

Cache Buffer Size field, 128  
caching, 9  
CD-ROM drive  
    device type for, 30  
certification process, 44  
Clear Reset Blocking Condition command, 41  
client, defined, 4  
Command field, 30, 33, 57, 58  
command ordering, 13, 14  
Command Ordering field, 79, 128  
Count field, 113, 143  
Cylinder field, 47, 48  
Cylinder Skew field, 74, 123

## D

Data Buffer Size field, 118  
Data Length field, 110  
Defect Count field, 46, 87, 91

defect list, 44, 46  
defect list, command to read, 86  
Defect n Logical Block Number field, 48  
Delay Time field, 115  
Device Address field, 30, 81, 83, 141  
Device Error Address field, 33  
Device Type field, 30, 33, 45  
Disk Errors field, 62  
Do Disconnect Flag field, 135

## E

Encoded Byte field, 51  
End of Message (EOM), 56

## F

Flags field, 45, 46, 55, 56, 59, 69, 87, 94, 104, 108, 120, 127  
flexible disk  
    device type for, 30  
    PCI commands for  
        Format Unit, 44  
        Get Device List, 50  
        Get Statistics, 59  
        I/O Sleep, 66  
        Query Unit Characteristics, 69  
        Query Unit Options, 78  
        Read Data, 81  
        Read Data and Verify, 85  
        Recalibrate, 93  
        Release, 94  
        Request Unit Attention Notification, 96  
        Reserve, 97  
        Reset Device, 99  
        SCSI Pass Thru, 102  
        Set Statistics, 116  
        Set Unit Characteristics, 119

- Set Unit Options, 126
- Start Unit, 130
- Stop Unit, 131
- Synchronous Negotiation Control, 132
- Test I/O, 136
- Unit Attention Notification, 137
- Write Data, 139
- Write Data Direct, 141
- Flexible Disk Format field, 72, 121
- Flush Get Send Data command, 43
- Format Unit command, 44

## G

- Get Device List command, 50
- Get SCSI ID command, 53
- Get Send Data command, 54. See also Flush Get Send Data command
- Get Server Information command, 57
- Get Statistics command, 59
- Get Target Mode Options command, 63
- grown defect list, 44, 91

## H

- hard disk
  - device type for, 30
  - PCI commands for
    - Format Unit, 44
    - Get Device List, 50
    - Get Statistics, 59
    - I/O Sleep, 66
    - Query Unit Characteristics, 69
    - Query Unit Options, 78
    - Read Data, 81
    - Read Data and Verify, 85
    - Read Data Direct, 83
    - Read Defect List, 86
    - Read Device Capacity, 90
    - Reassign Blocks, 91
    - Recalibrate, 93
    - Release, 94
    - Request Unit Attention Notification, 96
    - Reserve, 97
    - Reset Device, 99
    - SCSI Pass Thru, 102

- Set Statistics, 116
- Set Unit Characteristics, 119
- Set Unit Options, 126
- Start Unit, 130
- Stop Unit, 131
- Synchronous Negotiation Control, 132
- Test I/O, 136
- Unit Attention Notification command, 137
- Write Data, 139
- Write Data Direct, 141

- Head field, 47, 48
- host, defined, 4

## I

- I/O Sleep command, 66
- Immediate Reset command, 65
- immediate synchronous negotiation, doing, 163
- implementation-specific commands, 18, 39
- initiator, defined, 4
- Interleave field, 74, 123

## L

- Length field, 54
- Locate PCI Server command, 68

## M

- map bad blocks, 46
- Max Ctrl Bufs Per Init field, 118
- memory allocation, 12
- Mode field, 79, 127

## N

- Need to Negotiate Flag field, 134
- Number of Buffers field, 62
- Number of Control Buffers field, 117
- Number of Cylinders field, 121
- Number of Data Buffers field, 117
- Number of Retries field, 79, 128
- Number of Surfaces field, 121

## P

- parameters, changing for a unit, 149
- pass thru device type
  - device type for, 30
  - PCI commands for
    - Flush Get Send Data, 43
    - Get Device List, 50
    - Get Send Data, 54
    - Get Statistics, 59
    - Get Target Mode Options, 63
    - I/O Sleep, 66
    - Immediate Reset, 65
    - Query Unit Characteristics, 69
    - Release, 94
    - Request Reset Notification, 95
    - Set Target Mode Options, 117
    - Set Unit Options, 126
    - Synchronous Negotiation Control, 132
- PCI server
  - error handling, 169
  - features, 3
  - hosts, 8
  - introduction, 1
  - list of commands, 36, 166
  - loading, 17
  - locating, 68, 145
  - packaging, 7
  - performance features, 9
  - preparing to send commands to, 147
  - sending notices to clients, 152
  - standards, 9
  - supported products, 6
- PCM buffer request message, 29
- PCM general interrupt message, 28
- peripheral command message (PCM), 26, 27
- peripheral controller, defined, 4
- peripheral status message (PSM), 26, 31
- Physical Blocks Written field, 62
- Pin 2 field, 75, 123
- Pin 34 field, 76, 124
- Port ID field, 95, 96, 97
- primary defect list, 44
- PSM, See peripheral status message (PSM)

## Q

- Query Unit Characteristics command, 69
- Query Unit Options command, 78

## R

- Read Ahead field, 80, 128
- Read Data and Verify command, 85
- Read Data command, 81
- Read Data Direct command, 83
- Read Defect List command, 86
- Read Device Capacity command, 90
- Read Hits field, 62
- Read Misses field, 62
- Reassign Blocks command, 91
- Recalibrate command, 93
- Release command, 94
- Remaining Length field, 56
- Remaining SCSI Length field, 55, 56
- Remote Target ID field, 55, 56
- Request Reset Notification command, 95
- Request Unit Attention Notification command, 96
- Request/Acknowledge Offset field, 135
- Reserve command, 97
- Reset Device command, 99
- Reset Notification command, 100
- Result 1 field, 58
- Retention Tape command, 101
- Re-transmit Count field, 61
- RWC Cylinder field, 73, 122

## S

- SCSI
  - bus reset, 95
  - error handling, 19
  - immediate reset, 15
- SCSI Command field, 110
- SCSI Pass Thru command, 19, 102
  - Method 1, 103
  - Method 2, 107
- Sector field, 48
- Sectors/Track field, 71, 121
- Seek Beginning of Tape command, 111
- Seek End of Data command, 112

- Seek Filemark command, 113
- seek optimization, 13
- Server Version Number field, 62
- server, defined, 4
- Set Reset Blocking Time command, 115
- Set Statistics command, 116
- Set Target Mode Options command, 117
- Set Unit Characteristics command, 119
- Set Unit Options command, 126
- soft errors, 34
- standard commands, 18, 39
- Start Unit command, 130
- status codes, list, 170
- Status field, 33
- Step Rate field, 73, 122
- Stop Unit command, 131
- Subfunction Code field, 133
- suspend operations, command for, 66
- Synchronous Flag field, 134
- Synchronous Negotiation Control command, 132

## T

- tape drive
  - device type for, 30
  - PCI commands for
    - Get Device List, 50
    - Get Statistics, 59
    - I/O Sleep, 66
    - Load, 67
    - Query Unit Characteristics, 69
    - Query Unit Options, 78
    - Read Data, 81
    - Read Data and Verify, 85
    - Release, 94
    - Request Unit Attention Notification, 96
    - Reserve, 97
    - Reset Device, 99
    - Retension Tape, 101

- SCSI Pass Thru, 102
- Seek Beginning of Tape, 111
- Seek End of Data, 112
- Seek Filemark, 113
- Set Statistics, 116
- Set Unit Characteristics, 119
- Set Unit Options, 126
- Synchronous Negotiation Control, 132
- Test I/O, 136
- Unit Attention Notification, 137
- Unload, 138
- Write Data, 139
- Write Filemark, 143

- target adaptor, defined, 4
- target mode, 19, 154
  - Get Target Mode Options command, 63
  - Set Target Mode Options command, 117
- target, defined, 4
- Test I/O command, 136
- Total Block Count field, 90
- Track Skew field, 74, 122
- Tracks/Zone field, 73, 122
- Transfer Period field, 135
- Transport Errors field, 61

## U

- Unit Number field, 30, 33, 50, 53
- Unload command, 138

## W

- WPC Cylinder field, 73, 122
- Write Data command, 139
- Write Data Direct command, 141
- Write Filemark command, 143
- Write I/Os field, 62
- Writes field, 62
- writing a block of data to a unit, 151